ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN

PHẠM DOÃN CẢNH

KHÓA LUẬN TỐT NGHIỆP

ỨNG DỤNG CÔNG NGHỆ TẠO SINH TĂNG CƯỜNG TRONG CẢI THIỆN KHẢ NĂNG TỰ HỌC VÀ RA QUYẾT ĐỊNH CHO TRỢ LÝ ẢO

RETRIEVAL-AUGMENTED GENERATION TECHNOLOGY IN IMPROVING SELF-LEARNING CAPABILITIES AND DECISION MAKING FOR VIRTUAL ASSISTANTS

CỬ NHÂN NGÀNH CÔNG NGHỆ THÔNG TIN

TP. HỒ CHÍ MINH, 2025

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN

PHẠM DOÃN CẢNH – 21520642

KHÓA LUẬN TỐT NGHIỆP

ỨNG DỤNG CÔNG NGHỆ TẠO SINH TĂNG CƯỜNG TRONG CẢI THIỆN KHẢ NĂNG TỰ HỌC VÀ RA QUYẾT ĐỊNH CHO TRỢ LÝ ẢO

RETRIEVAL-AUGMENTED GENERATION TECHNOLOGY IN IMPROVING SELF-LEARNING CAPABILITIES AND DECISION MAKING FOR VIRTUAL ASSISTANTS

CỬ NHÂN NGÀNH CÔNG NGHỆ THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN
TS. NGUYỄN VĂN KIỆT

TP. HỒ CHÍ MINH, 2025

THÔNG TIN HỘI ĐỒNG CHẤM KHÓA LUẬN TỐT NGHIỆP

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …….. ngày …….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

LỜI CẢM ƠN

Xuyên suốt khoảng thời gian được trải nghiệm và thực hành trên trường cũng như là làm khoá luận tốt nghiệp tại trường Đại học Công nghệ Thông tin, Đại học Quốc gia TP. Hồ Chí Minh là một chặng đường vô cùng quá giá và quan trọng, một khoản thời gian vô cùng đáng nhớ đối với tôi. Quá đó tôi cũng muốn gửi một lời cảm ơn chân thành và sâu sắc nhất đến những người thầy người cô đã tận tâm hỗ trợ và hướng dẫn tôi nhiệt tình, cũng như là những người đã đồng hành cùng tôi qua bốn năm học tập tại trường.

Trước hết, tôi xin được gửi lời cảm ơn đến Khoa Khoa học và Kỹ thuật Thông tin, trường Đại học Công nghệ Thông tin cũng như các thầy cô trong khoa, nơi đã đem đến cho tôi rất nhiều trải nghiệm và những bài học mà chắc chắn tôi sẽ không thể nào kiếm được ở những nơi khác, một môi trường đã mãi giũa lên một người làm ra đề tài khoá luận này.

Đặc biệt, tôi muốn gửi lời cảm ơn sâu đậm nhất đến sự nhiệt tình và những lời khuyên đến từ thầy TS. Nguyễn Văn Kiệt, người đã trực tiếp đưa ra những hướng dẫn và các hướng đi trong xuyên suốt quá trình thực hiện đề tài này. Sự nhiệt huyết và những lời khuyên quý giá đến từ thầy không chỉ giúp tôi vượt qua khó khăn mà còn định hướng rõ ràng hơn, chỉ điểm cho tôi đích đến đúng đắn trong quá trình thực hiện khoá luận.

Đề tài khóa luận tốt nghiệp này là thành quả đúc kết được sau quá trình nỗ lực của bản thân cùng sự hỗ trợ to lớn từ quý thầy cô và những người đồng hành của tôi trong khoản thời gian ở giảng đường. Tuy nhiên, không thể thiếu những thiếu xót trong quá trình làm, tôi rất mong nhận được sự góp ý quý báu từ Thầy Cô để đề tài được có thể vượt vũ môn và hoàn thành một cách trọn vẹn.

TP. Hồ Chí Minh, tháng 12 năm 2024

Sinh viên thực hiện

Phạm Doãn Cảnh

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

TÓM TẮT KHÓA LUẬN

Đề tài đã đề xuất giải pháp ứng dụng kết hợp công nghệ RAG phát triển một trợ lý ảo có thể tự phân tích và đưa ra các quyết định dựa trên bộ nhớ dài hạn được triển khai dựa trên công nghệ "Prompt Engineering" nhằm thực hiện việc đưa ra các hành động hợp lý hơn với các nhóm người dùng khác nhau. Đồng thời đề tài cũng nhấn mạnh việc xây dựng hệ thống tích hợp các công nghệ tiên tiến xung quanh các mô hình large language model để trợ lý ảo có thể tổng hợp thông tin một cách đầy đủ bao gồm khả năng truy cập và phân tích file, trích xuất thông tin phân tích thời gian thực hay thậm chí là yêu cầu thông tin từ cơ sở dữ liệu để sắp xếp các hoạt động của người sử dụng kết hợp với các khả năng đầu cuối trong module nhận diện giọng nói, video call, share screen để mở rộng tương tác giúp người muốn sử dụng trợ lý ảo có thêm những phương thức gần gũi hơn.

Công nghệ sử dụng bao gồm:

- LangChain Framework: Hỗ trợ xử lý ngôn ngữ và tích hợp dữ liệu.
- Node.js, FastAPI: Xây dựng server và API.
- ReactJS: Phát triển giao diện người dùng.
- MySQL, ChromaDB: Lưu trữ dữ liệu quan hệ và dữ liệu vector.
- Google Calendar API: Quản lý lịch và nhắc nhở.

Khóa luận cũng nghiên cứu các kỹ thuật tiên tiến về việc phân đoạn và phân tích các nội dung văn bản thông qua hai kĩ thuật Agentic Chunking và Semantic Chunking, nhằm tối ưu hóa việc xử lý, lưu trữ, và truy xuất dữ liệu ngữ nghĩa, giúp cải thiện hiệu quả hoạt động của trợ lý ảo trong các nhiệm vụ tìm kiếm và tổng hợp thông tin.

Nội dung luận văn được tổ chức thành 6 chương

Chương 1: Tổng quan chung về chủ đề

Giải thích lý do chọn chủ đề, mục tiêu của nghiên cứu bao gồm việc tạo ra một hệ thống trợ lý ảo thông minh cho việc quản lý lịch và tìm kiếm thông tin. Phương pháp triển khai sử dụng các công nghệ tiên tiến nhằm nâng cao trải nghiệm của người dùng.

Chương 2: Các công trình nghiên cứu và dự án khác trong lĩnh vực này

Tổng quan về các công trình nghiên cứu hiện có và các hệ thống hiện tại của trợ lý ảo, phân tích các kỹ thuật như nhận diện giọng nói, phát triển chatbot và tích hợp nền tảng đa dụng.

Chương 3: Cơ sở lý thuyết và công nghệ áp dụng trong công trình này

Liệt kê các công nghệ và kĩ thuật đã được triển khai trong dự án luận văn được giải thích bao gồm LangChain, ReactJS, Node.js, FastAPI, MySQL và ChromaDB. Các công nghệ tiên tiến như Kỹ thuật câu lệnh (Prompt Engineering), RAG và các framework hỗ trợ phát triển AI cũng được phân tích chi tiết.

Chương 4: Cách tiếp cận thực hiện

Trên thực tế, người dùng được cung cấp thông tin như mô tả kiến trúc hệ thống, các sơ đồ xử lý dữ liệu và các khả năng chức năng như lưu trữ dài hạn, tìm kiếm thông tin, cũng như tương tác đồng bộ thời gian thực qua video call hoặc chia sẻ màn hình.

Chương 5: Cài đặt hệ thống và kết quả

Chi tiết về quy trình cài đặt hệ thống, giao diện ứng dụng và các tính năng đã triển khai như lập lịch, tìm kiếm thông tin, giao tiếp với trợ lý ảo, v.v.

Chương 6: Kết luận và Triển vọng phát

Tóm tắt các ưu điểm, hạn chế của hệ thống và đề xuất hướng phát triển trong tương lai, như cải tiến khả năng học tập của trợ lý ảo hoặc mở rộng ứng dụng vào các lĩnh vực khác.

Chương 1. **TỔNG QUÁT ĐỀ TÀI**

1.1. Lý do chọn đề tài

Ngày nay, trí tuệ nhân tạo (AI) đã trở thành một thuật ngữ quen thuộc thu hút sự quan tâm lớn và được chú ý rộng rãi trên toàn cầu nhờ vào khả năng mang lại những giải pháp rất là tiện lợi cho người dùng, cũng như là giúp hỗ trợ cải thiện từ những điều nhỏ nhất trong cuộc sống của loài người. Và trong số các ứng dụng của trí tuệ nhân tạo hiện nay, trợ lý ảo AI là một trong những ứng dụng nổi bật và xu hướng nhất hiện nay, nhờ tính tiện lợi và khả năng tương tác tự nhiên mà người dùng nhận được từ trợ lý ảo. Lấy cảm hứng từ J.A.R.V.I.S, một nhân vật giả tưởng đến từ Vũ trụ Điện ảnh Marvel, các hệ thống trợ lý ảo đang dần hiện thực hóa các ý tưởng từng được cho là không thể, giúp đem lại những trải nghiệm tiện ích vượt trội hơn cho đời sống hàng ngày của con người.

1.2. Mục tiêu đề tài

Đề tài mong muốn xây dựng một hệ thống trợ lý ảo nhằm hỗ trợ người dùng tự động hóa các công việc liên quan đến quản lý lịch trình cá nhân, tìm kiếm, tổng hợp và tóm tắt thông tin từ nhiều nguồn khác nhau. Hệ thống này không chỉ dừng lại ở việc thực hiện các tác vụ thông thường mà còn Tập trung vào việc cải thiện trải nghiệm người dùng. thông qua các kỹ thuật xoay quanh công nghệ đang làm mưa làm gió trong thời đại hiện nay như xử lý ngôn ngữ tự nhiên (NLP), nhận diện giọng nói, prompt engineering.

Trợ lý ảo được thiết kế nhằm giúp người dùng tương tác một cách tự nhiên nhất thông qua giọng nói, đồng thời tích hợp khả năng nhận diện và hiểu bối cảnh thông qua thị giác. Điều này góp phần kết nối chặt chẽ hơn giữa con người và công nghệ, mang lại cảm giác giao tiếp gần gũi và hiệu quả hơn.

Một trong những mục tiêu hàng đầu của hệ thống là Khả năng tự động xử lý và lưu trữ thông tin và sử dụng các thông tin quan trọng liên quan đến từng

người dùng. Thông qua việc lưu trữ dữ liệu trong bộ nhớ dài hạn, hệ thống có thể học hỏi từ quá trình tương tác để đưa ra các câu trả lời và gợi ý được cá nhân hóa cao, đáp ứng nhu cầu và sở thích riêng của từng người dùng. Điều này không chỉ tăng cường tính linh hoạt mà còn tối ưu hóa hiệu quả trong việc hỗ trợ người dùng xử lý các vấn đề thường gặp trong cuộc sống và công việc hàng ngày.

Ngoài ra, quá trình quản lý lịch trình cũng được hệ thống tập trung tối ưu hóa, hỗ trợ người dùng theo dõi và sắp xếp công việc, thời gian một cách khoa học hơn. Kết hợp với khả năng tìm kiếm và tổng hợp thông tin nhanh chóng từ nhiều nguồn, trợ lý ảo sẽ trở thành một công cụ hữu ích, giúp người dùng tiết kiệm thời gian và nâng cao năng suất.

1.3. Phương pháp thực hiện

Việc chọn lựa đúng công nghệ đóng vai trò cốt lõi để tạo nên một ứng dụng hoàn chỉnh, bởi lẽ người dùng không chỉ đánh giá dựa trên các tính năng mà còn đặt sự chú trọng vào chất lượng tổng thể mà ứng dụng đem lại. Do đó tôi đã quyết định sử dụng những công nghệ phổ biến và đáng tin cậy, Không chỉ đảm bảo hoạt động tốt trên nhiều trình duyệt, mà còn đáp ứng yêu cầu về hiệu suất ổn định và tốc độ xử lý nhanh trong quá trình vận hành:

- Sử dụng Framework React và các thư viện liên quan để xây dựng giao diện cho trang web
- Nghiên cứu sử dụng một số opensource để triển khai cho chức năng mô phỏng lại quá trình video call và screen share giữa người dùng và trợ lý ảo như react-media-recorder, silence-aware-recorder và merge-images
- Xây dựng Web API server cho hệ thống bằng Node.js Express để xử lý việc truyền nhận dữ liệu cho ứng dụng. Đồng thời, sử dụng FastAPI để triển khai thêm các service nhằm tận dụng ưu thế của Python trong việc hỗ trợ các thư viện mạnh mẽ liên quan đến xử lý dữ liệu và tích hợp với các mô hình ngôn ngữ.

- Áp dụng hệ quản trị cơ sở dữ liệu MySQL để quản lý và lưu trữ thông tin của toàn bộ hệ thống, bên cạnh đó sử dụng thêm ChromaDB để hỗ trợ lưu trữ các dữ liệu dưới dạng vector phục vụ cho các dữ liệu làm việc trực tiếp với trợ lý ảo như bộ nhớ dài hạn và dữ liệu từ tệp tin người dùng đăng tải
- Sử dụng **Socket.io** và **Socket.io-Client** để triển khai một hệ thống giao tiếp thời gian thực giữa client và server, đem đến các chức năng như cập nhật trạng thái video call, chia sẻ màn hình, và nhận thông báo các dữ liệu hay thông tin, tin nhắn từ trợ lý ảo.
- Sử dụng Framework Langchain nhằm hỗ trợ các quy trình xử lý prompt, dữ liệu liên quan đến mô hình ngôn ngữ và trợ lý ảo

1.4. Giới hạn phạm vi

Đề tài hướng đến việc xây dựng một hệ thống tận dụng AI tạo sinh kết hợp công nghệ "Retrieval-augmented generation" để xây dựng nên một trợ lý ảo có khả năng ghi nhớ và học hỏi từ đó đem đến tính linh hoạt hơn trong việc hỗ trợ người dùng xây dựng, quản lý lịch trình học tập và làm việc, tự động nhắc nhở các nhiệm vụ cần làm. Đồng thời cung cấp khả năng tìm kiếm và tổng hợp thông tin từ internet hay từ một hoặc nhiều tập dữ liệu, tài liệu một cách nhanh chóng và chính xác, với trải nghiệm gần gũi và dễ tương tác giữa người và máy tính thông qua khả năng lắng nghe và quan sát bối cảnh xung quanh môi trường tương tác.

Chương 2. **CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN**

Hiện nay, đã có nhiều nghiên cứu tập trung vào việc phát triển các hệ thống trợ lý ảo nhằm giải quyết các vấn đề cụ thể của con người. Chẳng hạn, nghiên cứu của S. Siddesh và cộng sự (2020) cùng Dũng và cộng sự (2022) đã chú trọng vào khả năng nhận diện giọng nói tự động, giúp người dùng giảm bớt thao tác thủ công khi tương tác với thiết bị. Bên cạnh đó, Rohit Tamrakar và cộng sự (2021) đã nghiên cứu tích hợp hệ thống chatbot qua các nền tảng phổ biến như Slack, WhatsApp, Telegram, giúp mở rộng khả năng ứng dụng của trợ lý ảo trong nhiều môi trường khác nhau. Những công trình này không chỉ mở ra nhiều tiềm năng trong việc cải tiến công nghệ mà còn đặt nền tảng cho việc ứng dụng rộng rãi hơn các trợ lý ảo vào thực tế.

Với xu hướng phát triển đó, đề tài nghiên cứu của tôi tập trung vào việc đề xuất một giải pháp xây dựng hệ thống trợ lý ảo có khả năng hoạt động trên nền tảng web. Hệ thống này không chỉ dừng lại ở các chức năng cơ bản như giao tiếp qua văn bản hay giọng nói, mà còn mở rộng thêm các khả năng tiên tiến hơn. Một số tính năng nổi bật được nghiên cứu và phát triển trong hệ thống bao gồm:

- **Tương tác qua video call và chia sẻ màn hình:** cho phép trợ lý ảo nhận diện và xử lý thông tin bối cảnh tốt hơn trong các tình huống thực tế.
- **Tìm kiếm và tổng hợp thông tin đa nguồn:** giúp người dùng nhanh chóng tiếp cận thông tin từ nhiều nguồn khác nhau một cách chính xác và hiệu quả.
- **Phân tích và học tập dữ liệu từ các cuộc trò chuyện:** từ đó cải thiện khả năng đáp ứng nhu cầu cá nhân hóa của từng người dùng.
- **Quản lý lịch trình và thời gian:** hỗ trợ người dùng sắp xếp công việc và sinh hoạt hợp lý hơn.

Chương 3. CƠ SỞ LÝ THUYẾT VÀ CÁC CÔNG NGHỆ SỬ DỤNG

3.1. Langchain

Hình 3.1 LangChain

LangChain là một framework mã nguồn mở, được thiết kế nhằm tối ưu hóa khả năng ứng dụng của các mô hình ngôn ngữ lớn (LLMs) như ChatGPT, LlaMA,... trong các bài toán thực tế. Framework này cho phép xây dựng các ứng dụng AI phức tạp bằng cách tích hợp khả năng xử lý ngôn ngữ tự nhiên với các nguồn dữ liệu bên ngoài, workflows, và logic tùy chỉnh.

Ra mắt lần đầu vào tháng 10 năm 2022, LangChain đã nhanh chóng đạt được sự quan tâm lớn từ cộng đồng phát triển phần mềm. Theo thống kê từ GitHub, tính đến tháng 12 năm 2024, LangChain đã đạt 95.5K stars, vượt qua cả PyTorch (84.6K stars), một framework nổi tiếng lâu đời trong lĩnh vực học sâu. Điều này minh chứng cho sự phổ biến và tiềm năng phát triển vượt bậc của LangChain, đặc biệt trong các ứng dụng tích hợp mô hình ngôn ngữ lớn. [Nguồn: GitHub LangChain Repository]

Tuy nhiên, mặc dù nhận được nhiều sự quan tâm, việc sử dụng LangChain trong các dự án thực tế vẫn đối mặt với một số thách thức, như yêu cầu kỹ thuật cao và khả năng mở rộng của framework. Những yếu tố này cần được xem xét kỹ lưỡng trong quá trình triển khai.

Mục tiêu chính của framework này là giúp các nhà phát triển dễ dàng tích hợp AI vào các hệ thống thực tế, chẳng hạn như chatbot, công cụ tự động hóa quy trình, và hệ thống hỗ trợ ra quyết định.

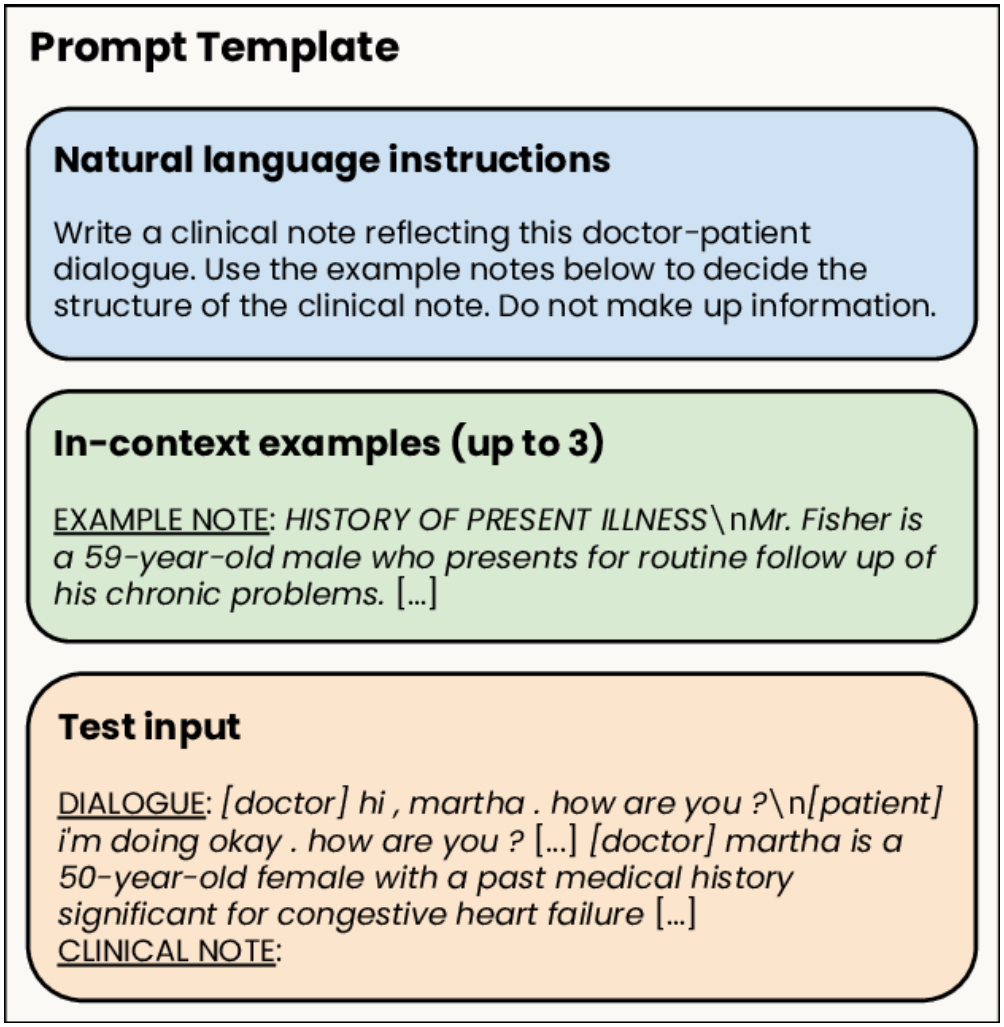
LangChain hoạt động dựa trên các **khối cơ bản (modules)**, cho phép tùy chỉnh theo nhu cầu của từng ứng dụng. Một số thành phần chính:

3.1.1. Prompts:

Cho phép tùy chỉnh cách giao tiếp với các mô hình ngôn ngữ bằng cách thiết kế các lời nhắc (prompts) phù hợp với mục tiêu ứng dụng. LangChain cung cấp các công cụ mạnh mẽ để xây dựng và quản lý các prompt thông qua lớp PromptTemplate.

Một prompt thường bao gồm các thành phần sau:

- Hướng dẫn (Instructions): Chỉ định cho mô hình biết cần làm gì, cách sử dụng thông tin bên ngoài nếu có, xử lý truy vấn như thế nào và cấu trúc đầu ra ra sao.
- Thông tin bên ngoài (External Information): Cung cấp thêm nguồn kiến thức cho mô hình, có thể được chèn thủ công vào prompt, truy xuất qua cơ sở dữ liệu vector (retrieval augmentation) hoặc thông qua các phương thức khác như API.
- Đầu vào từ người dùng (User Input): Thường là truy vấn được nhập vào hệ thống bởi người dùng.
- Chỉ báo đầu ra (Output Indicator): Đánh dấu sự bắt đầu của văn bản sẽ được mô hình tạo ra.

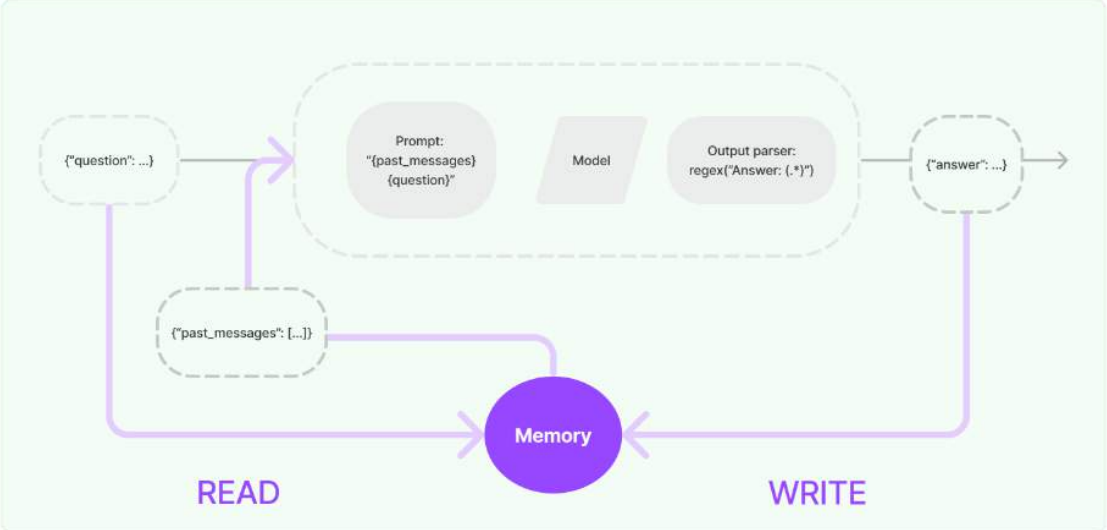


Hình 3.2 Ví dụ về prompt template

Các thành phần này thường được sắp xếp theo thứ tự: hướng dẫn, thông tin bên ngoài (nếu có), đầu vào từ người dùng và cuối cùng là chỉ báo đầu ra. Việc thiết kế prompt hiệu quả là một kỹ năng quan trọng khi làm việc với LLMs. Prompt tốt có thể cải thiện đáng kể chất lượng đầu ra của mô hình, trong khi prompt kém có thể dẫn đến kết quả không mong muốn. Sử dụng các công cụ như PromptTemplate và FewShotPromptTemplate trong LangChain giúp chuẩn hóa và tối ưu hóa quá trình này, cho phép xây dựng các ứng dụng AI mạnh mẽ và linh hoạt hơn.

3.1.2. Memory:

Trong các ứng dụng sử dụng Mô hình Ngôn ngữ Lớn (LLM), khả năng ghi nhớ và truy xuất thông tin từ các tương tác trước đó—được gọi là "memory"—là một yếu tố quan trọng để duy trì ngữ cảnh và cung cấp phản hồi phù hợp. LangChain cung cấp một loạt các tiện ích để tích hợp chức năng này vào hệ thống của bạn.



Hình 3.3 Cơ chế lưu trữ bộ nhớ LangChain

3.1.2.1. Chức năng của Memory trong LangChain

Một hệ thống memory cần hỗ trợ hai hành động cơ bản: đọc và ghi. Trong quá trình thực thi của một chuỗi (chain), hệ thống sẽ:

- **Đọc (READ):** Sau khi nhận đầu vào từ người dùng nhưng trước khi thực thi logic chính, chuỗi sẽ đọc từ hệ thống memory để bổ sung thông tin cần thiết.
- **Ghi (WRITE):** Sau khi thực thi logic chính nhưng trước khi trả về kết quả, chuỗi sẽ ghi lại các đầu vào và đầu ra hiện tại vào memory để sử dụng trong tương lai.

3.1.2.2. Lưu trữ và Truy vấn Trạng thái

- **Lưu trữ (Storing):** Dưới lớp nền của bất kỳ hệ thống memory nào là lịch sử của tất cả các tương tác trò chuyện. LangChain cung cấp các tích hợp để lưu trữ các tin nhắn này, từ danh sách trong bộ nhớ đến cơ sở dữ liệu lưu trữ lâu dài.
- **Truy vấn (Querying):** Việc chỉ lưu trữ danh sách các tin nhắn là chưa đủ; cần có các cấu trúc dữ liệu và thuật toán để truy xuất thông tin một cách hiệu quả. Các hệ thống memory đơn giản có thể chỉ trả về các tin nhắn gần đây nhất, trong khi các hệ thống phức tạp hơn có thể tạo ra tóm tắt hoặc trích xuất thông tin về các thực thể được đề cập trong cuộc trò chuyện.

3.1.2.3. Tích hợp Memory vào Hệ thống

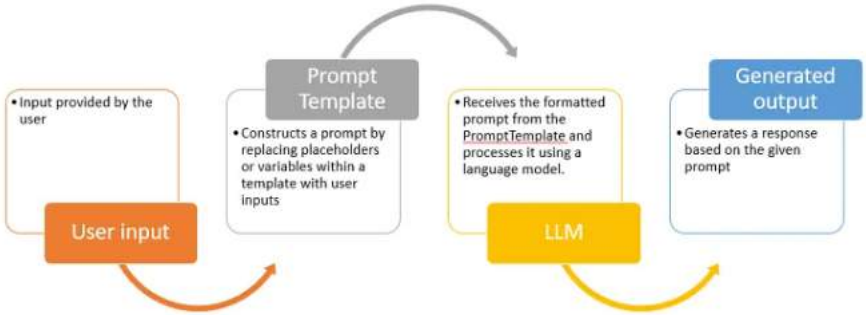
LangChain cung cấp các lớp memory khác nhau để phù hợp với nhu cầu cụ thể của ứng dụng, bao gồm:

- **ConversationBufferMemory:** Lưu trữ toàn bộ lịch sử cuộc trò chuyện trong một bộ đệm và truyền nó vào mẫu prompt.
- **ConversationBufferWindowMemory:** Tương tự như trên nhưng chỉ lưu trữ một số lượng tin nhắn gần đây nhất.
- **ConversationSummaryMemory:** Tạo ra tóm tắt của cuộc trò chuyện để duy trì ngữ cảnh mà không cần lưu trữ toàn bộ lịch sử.
- **VectorStoreRetrieverMemory:** Sử dụng các vector nhúng để lưu trữ và truy xuất thông tin dựa trên ngữ nghĩa.

3.1.3. Chains (Workflow):

Là tập hợp các thành phần được kết nối tuần tự để thực hiện một nhiệm vụ cụ thể. Mỗi chuỗi có thể bao gồm các mô hình ngôn ngữ, bộ truy xuất tài liệu, hoặc thậm chí các chuỗi khác, tạo nên một quy trình xử lý linh hoạt và mạnh mẽ. Cho phép kết hợp nhiều tác vụ nhỏ (như gọi API, xử lý dữ liệu) thành một luồng công việc thống nhất, từ đó có thể dễ dàng tạo ra được các các ứng dụng

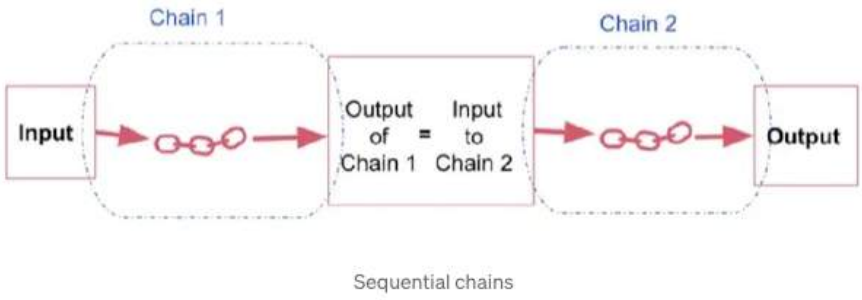
đơn giản, tự nhiên và nhanh chóng nhất. Ví dụ, một ứng dụng có thể nhận input từ người dùng, xử lý dữ liệu, gọi API LLM và trả về kết quả.



Hình 3.4 Tổng quan về một mô hình chuỗi căng bản

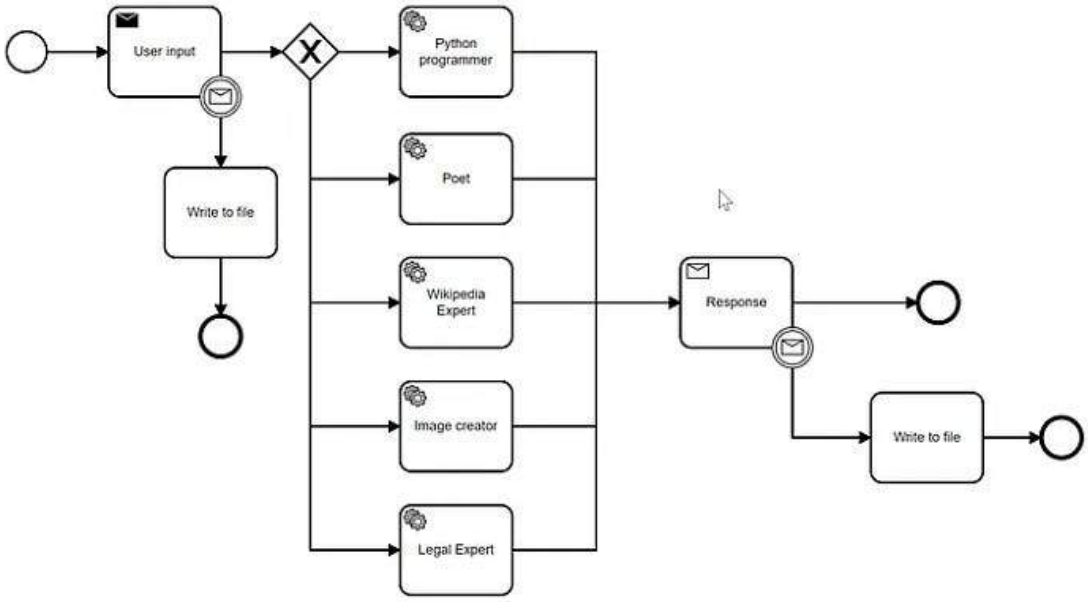
Có nhiều loại chuỗi trong LangChain, bao gồm:

- **Sequential Chain (Chuỗi tuần tự)**: Chuỗi này thực hiện các bước theo thứ tự, trong đó đầu ra của bước trước trở thành đầu vào của bước sau. Điều này cho phép xử lý thông tin một cách tuần tự và logic.



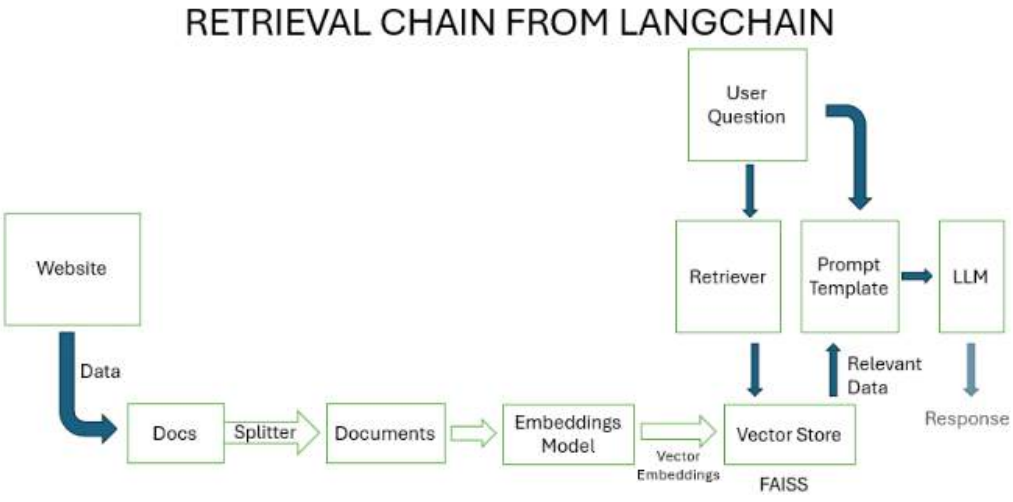
Hình 3.5 Mô hình chuỗi tuần tự

- **Router Chain (Chuỗi định tuyến)**: Sử dụng mô hình ngôn ngữ để quyết định hướng đi phù hợp cho truy vấn, chọn lựa giữa các chuỗi khác nhau dựa trên ngữ cảnh hoặc loại câu hỏi.



Hình 3.6 Mô hình chuỗi định tuyến

- **Retrieval Chain (Chuỗi truy xuất)**: Kết hợp với các công cụ truy xuất thông tin để tìm kiếm và lấy dữ liệu liên quan, sau đó sử dụng mô hình ngôn ngữ để xử lý và trả lời dựa trên dữ liệu đó.



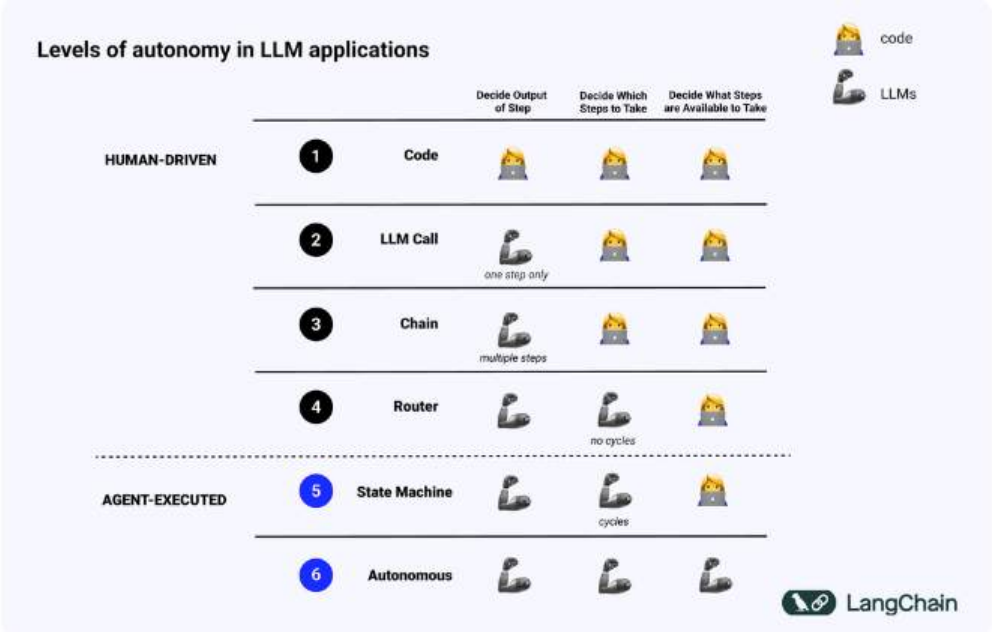
Hình 3.7 Mô hình chuỗi truy xuất

Data Connection: LangChain hỗ trợ tích hợp với các nguồn dữ liệu bên ngoài (như cơ sở dữ liệu, tài liệu PDF, web) để truy vấn và xử lý dữ liệu trước khi gửi đến mô hình AI.

3.1.4. Agent:

Agent là một thành phần đặc biệt trong các ứng dụng trí tuệ nhân tạo, sử dụng Mô hình Ngôn ngữ Lớn (LLM) để tự động ra quyết định và thực hiện các hành động cụ thể dựa trên thông tin đầu vào từ phía của người dùng. Không giống như các chuỗi xử lý cố định (Chains), nơi các bước được xác định trước, Agents có khả năng linh hoạt điều chỉnh luồng thực hiện dựa trên trạng thái hiện tại và thông tin thu thập được trong quá trình hoạt động. Điều này cho phép chúng kết hợp việc xử lý ngôn ngữ tự nhiên, tìm kiếm thông tin từ nhiều nguồn dữ liệu khác nhau, và thực hiện các tác vụ thông qua API hoặc công cụ bên ngoài. Trong LangChain, một Agent hoạt động như một hệ thống động, sử dụng LLM để phân tích ngữ cảnh đầu vào, xác định hành động một cách phù hợp, và gọi các công cụ bên ngoài để thực hiện và hoàn thành nhiệm vụ. Song mức độ "tác nhân" (agentic) của một hệ thống phụ thuộc vào mức độ mà LLM quyết định hành vi của hệ thống đó. Ví dụ như là một hệ thống sử dụng LLM để định tuyến đầu vào vào một quy trình làm việc cụ thể có một mức độ "tác nhân" nhất định, trong khi một hệ thống phức tạp hơn, nơi LLM quyết định tiếp tục

hoặc kết thúc một quy trình, sẽ có mức độ "tác nhân" cao hơn.



Hình 3.8 Các cấp độ tác nhân

Agent hoạt động thông qua một quy trình lặp, bao gồm các bước sau:

- Nhận đầu vào: Agent nhận đầu vào từ người dùng hoặc một hệ thống khác.
- Phân tích ngữ cảnh: Agent sử dụng LLM để phân tích ngữ cảnh và quyết định hành động tiếp theo.
- Lựa chọn công cụ: Agent quyết định công cụ nào cần được sử dụng (như tìm kiếm thông tin, truy xuất cơ sở dữ liệu, hoặc gọi API).
- Thực thi và phản hồi:
    - Agent thực hiện tác vụ bằng công cụ đã chọn.
    - Sau khi có kết quả, Agent lặp lại quy trình để xử lý thêm hoặc trả kết quả cho người dùng.

Ưu điểm

- Tích hợp dễ dàng với LLMs: Hỗ trợ nhiều nền tảng mô hình ngôn ngữ lớn như OpenAI, Hugging Face, Anthropic, Google PaLM.

- Khả năng lưu trữ trạng thái: Memory cho phép các ứng dụng duy trì ngữ cảnh hội thoại hoặc chuỗi hành động, tạo trải nghiệm người dùng liền mạch.
- Tùy chỉnh linh hoạt: Hỗ trợ xây dựng workflows và logic phức tạp thông qua các chains.
- Kết nối đa dạng: Dễ dàng tích hợp với các nguồn dữ liệu, API, và tài liệu, giúp AI đưa ra câu trả lời dựa trên dữ liệu thực tế.
- Mở rộng và tái sử dụng: Kiến trúc module hóa của LangChain giúp việc mở rộng và tái sử dụng trong các dự án khác nhau trở nên dễ dàng.

Nhược điểm

- Phụ thuộc vào mô hình bên ngoài: LangChain cần sử dụng API từ các nhà cung cấp mô hình ngôn ngữ lớn, dẫn đến chi phí cao khi xử lý lượng dữ liệu lớn.
- Độ phức tạp ban đầu: Người dùng mới có thể gặp khó khăn khi làm quen với các thành phần như Chains, Prompts, hoặc Memory.
- Hiệu suất hạn chế: Khi tích hợp với mô hình lớn hoặc dữ liệu lớn, thời gian phản hồi của hệ thống có thể bị chậm.
- Rủi ro bảo mật: Khi làm việc với dữ liệu nhạy cảm, cần chú ý về bảo mật vì các API LLM thường xử lý dữ liệu qua máy chủ bên ngoài.
- Cần kỹ năng lập trình: Để khai thác tối đa LangChain, đòi hỏi người dùng phải hiểu rõ Python, API, và các khái niệm về mô hình AI.

3.2. Reactjs

Hình 3.9 ReactJS

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng để xây dựng giao diện người dùng (UI) trong các ứng dụng web. React cho phép tạo các thành phần UI (components) có thể tái sử dụng, giúp quản lý giao diện hiệu quả hơn trong các ứng dụng phức tạp.

Kiến trúc:

React hoạt động dựa trên mô hình **Component-Based Architecture**. Mỗi thành phần (component) đại diện cho một phần giao diện riêng lẻ, và các thành phần có thể lồng ghép với nhau để tạo nên giao diện tổng thể.

- **Kiến trúc Component-Based:** React hoạt động dựa trên kiến trúc Component-Based Architecture, trong đó giao diện được chia thành các thành phần nhỏ. Mỗi thành phần đảm nhiệm một phần công việc cụ thể (ví dụ: nút bấm, danh sách, menu).

- **Component:** Mỗi component là một khối giao diện độc lập, có thể nhận dữ liệu đầu vào (props) và tự quản lý trạng thái của chính nó (state).

Hình 3.10 Ví dụ về component

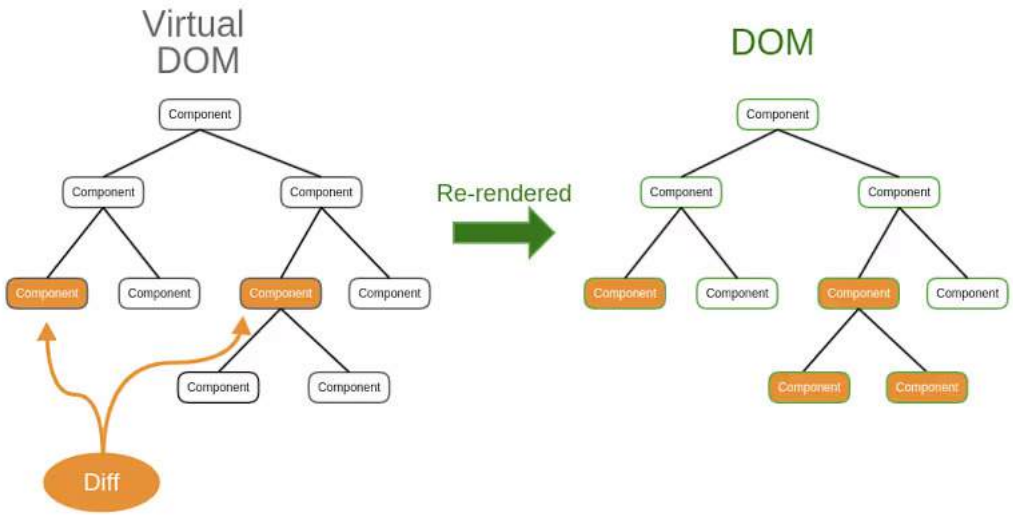
- **Hierarchy:** Các component có thể lồng ghép với nhau để tạo thành giao diện phức tạp.

Hình 3.11 Ví dụ về Hierarchy với các component

- **Virtual DOM (Document Object Model Ảo):** Virtual DOM là một trong những tính năng cốt lõi giúp React vượt trội so với các thư viện hoặc framework khác. Thay vì thao tác trực tiếp trên DOM thật (vốn chậm), React tạo ra một DOM ảo - một bản sao nhẹ hơn của DOM thật. Khi dữ liệu hoặc trạng thái thay đổi:

  - React so sánh DOM ảo mới với DOM ảo cũ (quá trình diffing).

- Chỉ những thay đổi cần thiết sẽ được cập nhật vào DOM thật, giúp cải thiện hiệu suất và tăng tốc độ phản hồi giao diện.



Hình 3.12 Virtual DOM (Document Object Model Ảo)[1]

- **One-Way Data Flow (Luồng dữ liệu một chiều):** React quản lý luồng dữ liệu theo hướng từ trên xuống (parent → child). Điều này giúp dữ liệu được kiểm soát tốt hơn và tránh xung đột khi truyền dữ liệu giữa các component.

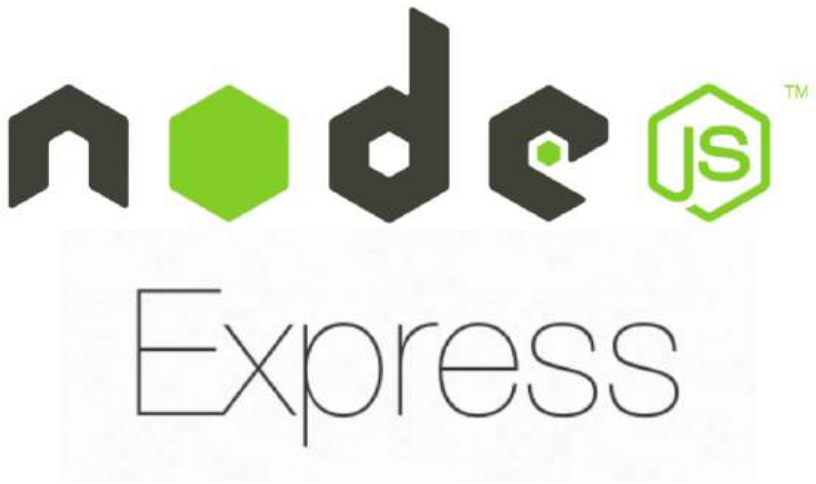
- **Hooks và Lifecycle:** React cung cấp Hooks (như useState, useEffect, useContext) để quản lý trạng thái và thực hiện các tác vụ liên quan đến vòng đời component (lifecycle). Hooks giúp viết code đơn giản hơn so với các phương pháp truyền thống như class component.

3.3. Nodejs (Express)

Node.js là một môi trường chạy JavaScript trên máy chủ, xây dựng dựa trên công cụ JavaScript V8 của Google nổi bật với khả năng xử lý bất đồng bộ. và kiến trúc dựa trên sự kiện (event-driven). Một trong những đặc điểm đáng

[1]Nguồn : Holistic SEO » TechSEO » What is Document Object Model?

chú ý của Node.js là cơ chế **non-blocking**, cho phép chương trình tiếp tục thực thi các tác vụ khác mà không cần chờ đợi một tác vụ cụ thể (như đọc/ghi tệp hoặc truy vấn cơ sở dữ liệu) hoàn tất. Thay vì tạm dừng để chờ kết quả, Node.js sử dụng callback, promise, hoặc async/await để xử lý kết quả khi nó sẵn sàng. Cơ chế này không chỉ tối ưu hóa hiệu suất mà còn giúp Node.js xử lý đồng thời hàng ngàn kết nối mà không làm nghẽn tài nguyên hệ thống, là giải pháp lý tưởng cho các ứng dụng hoạt động theo thời gian thực hoặc các hệ thống đòi hỏi hiệu suất cao về hiệu năng. Ra mắt vào năm 2009, Node.js đã nhanh chóng vươn lên thành một nền tảng được ưa chuộng trong việc triển khai các ứng dụng web. Trong khi đó, Express.js là một framework đơn giản và linh hoạt phát triển dựa trên Node.js hỗ trợ mạnh mẽ trong việc phát triển các ứng dụng web và API. Express cung cấp các công cụ và thư viện để quản lý định tuyến, xử lý yêu cầu HTTP, và sử dụng middleware giúp đơn giản hóa quá trình phát triển backend.



Hình 3.13 Nodejs (Express)

Ưu điểm của Nodejs:

Một trong những lý do khiến Node.js trở thành nền tảng phát triển phổ biến do có sự hỗ trợ vượt trội từ **npm (Node Package Manager)**, hệ thống quản lý thư viện và gói mở rộng lớn nhất dành cho JavaScript. Với hơn 2 triệu

gói thư viện có sẵn, npm trang bị cho nhà phát triển các công cụ để xử lý mọi vấn đề từ cơ bản đến nâng cao, giúp tiết kiệm thời gian và nâng cao hiệu quả trong quá trình phát triển phần mềm.

Trong đề tài này, Node.js và Express.js giữ vai trò then chốt trong việc xây dựng hệ thống backend. Cụ thể, chúng được sử dụng để tạo ra các API xử lý dữ liệu và giao tiếp với cơ sở dữ liệu MySQL, cung cấp dữ liệu theo định dạng JSON cho frontend (React.js). Hệ thống còn tích hợp các API bên ngoài như Google Calendar API thông qua Express, đồng thời sử dụng thư viện Socket.io trên Node.js nhằm quản lý dữ liệu theo thời gian thực giữa máy khách và máy chủ. Kiến trúc này giúp tối ưu hóa tốc độ phản hồi cũng như đảm bảo khả năng mở rộng của hệ thống.

Nhược điểm của Nodejs:

Mặc dù có nhiều ưu điểm, Node.js và Express.js cũng vẫn còn một số điểm hạn chế. Do sử dụng một luồng duy nhất, Node.js không tối ưu cho các tác vụ yêu cầu xử lý tính toán phức tạp (CPU-bound tasks). Ngoài ra, việc lập trình không đồng bộ với Node.js đòi hỏi người phát triển phải quản lý lỗi một cách kỹ lưỡng, Nếu không, hệ thống có thể dễ dàng rơi vào tình trạng mất ổn định. Tuy nhiên, nhờ vào tính đơn giản và hiệu quả trong phát triển API, cùng với khả năng tích hợp linh hoạt với các công nghệ khác, Node.js và Express.js vẫn được xem là giải pháp lý tưởng cho hệ thống backend trong đề tài này.

3.4. Socket.io

Hình 3.14 Socket.io

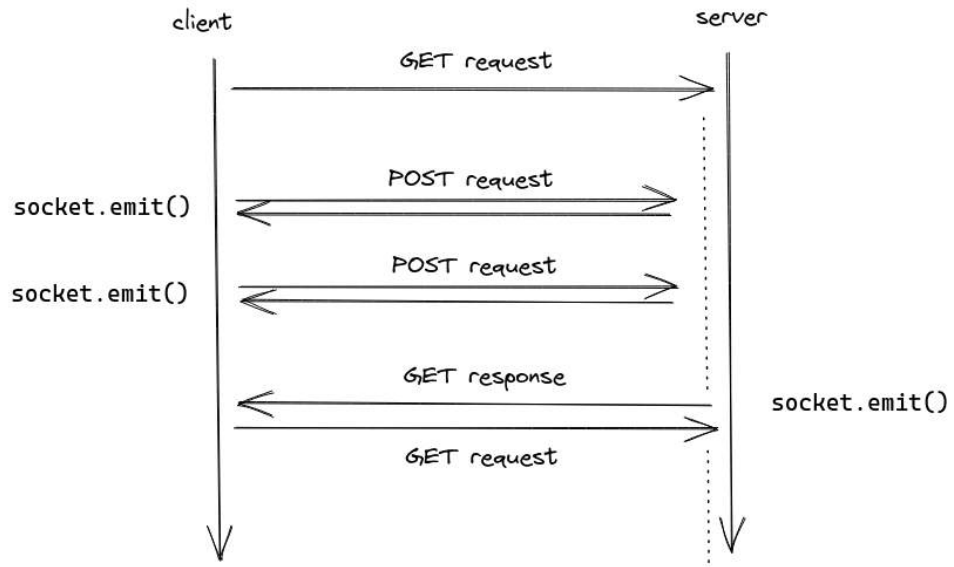
Được xem là một thư viện mạnh mẽ dành cho Node.js, Socket.io hỗ trợ xây dựng các ứng dụng thời gian thực (real-time) bằng cách sử dụng giao thức WebSocket. Mục tiêu để giúp các ứng dụng có thể truyền dữ liệu qua lại giữa hai chiều (bi-directional) từ client sang server và ngược lại một cách dễ dàng. Socket.IO cung cấp cho nhà phát triển các tính năng như truyền dữ liệu thời gian thực, sự kiện tùy chỉnh, và tự động xử lý các vấn đề về kết nối như khôi phục hoặc thay thế giao thức (fallback) khi cần.

Đặc điểm chính của Socket.IO

- **Truyền dữ liệu thời gian thực:** Cho phép gửi và nhận dữ liệu tức thì giữa server và client.
- **Hỗ trợ giao thức WebSocket:** Tự động chuyển sang giao thức khác (polling) khi WebSocket không được hỗ trợ.
- **Quản lý kết nối tự động:** Socket.IO tự động xử lý khi kết nối bị mất hoặc cần thiết lập lại.

- **Sự kiện tùy chỉnh:** Cho phép tạo và sử dụng các sự kiện tùy chỉnh (custom events) để quản lý giao tiếp.
- **Khả năng mở rộng:** Hỗ trợ phân cụm (clustering) và cân bằng tải (load balancing) để quản lý nhiều kết nối đồng thời.
- **Hỗ trợ đa nền tảng:** Socket.IO có thể hoạt động trên trình duyệt, ứng dụng di động, hoặc ứng dụng desktop.

Cách hoạt động của Socket.IO



Hình 3.15 Nguyên lý hoạt động của Socket.io[2]

Socket.IO hoạt động trên mô hình client-server, trong đó:

- **Client:** Một ứng dụng (thường là trình duyệt) kết nối tới server qua Socket.IO client library.
- **Server:** Một ứng dụng Node.js sử dụng thư viện Socket.IO để lắng nghe kết nối từ các client.

[2] Nguồn: Socket.IO Documentation

- **Kết nối:** Khi client yêu cầu kết nối đến server, một socket (kết nối hai chiều) được tạo để trao đổi dữ liệu.

3.5. FastAPI

Hình 3.16 FastAPI[3]

FastAPI là một framework hiện đại và mạnh mẽ để phát triển ứng dụng web và API bằng Python. Ra đời vào năm 2018, FastAPI đã nhanh chóng trở thành một lựa chọn phổ biến nhờ hiệu năng cao, khả năng xử lý không đồng bộ và khả năng tạo tài liệu API tự động (Swagger UI và ReDoc). Framework này được xây dựng trên hai nền tảng mạnh mẽ là **Starlette** (hỗ trợ xử lý HTTP và WebSocket không đồng bộ) và **Pydantic** (xác thực và quản lý dữ liệu), giúp đảm bảo độ chính xác và hiệu quả trong việc xử lý các yêu cầu API.

Trong đề tài này, FastAPI được sử dụng với mục đích xử lý dữ liệu liên quan đến **agentic chunking** – một phương pháp xử lý dữ liệu theo từng phần (chunk) để cải thiện hiệu quả truy xuất và tối ưu hóa phản hồi từ mô hình ngôn ngữ lớn (LLM). Tính năng này đòi hỏi một số thư viện Python có hiệu năng và

[3] Nguồn https://viblo.asia/p/huong-dan-co-ban-framework-fastapi-tu-a-z-phan-1-V3m5W0oyKO7

độ linh hoạt tốt hơn so với Node.js. Bên cạnh đó, **LangChain Python** cung cấp nhiều chức năng hơn so với phiên bản **LangChain JavaScript**, đặc biệt trong các bài toán phức tạp như quản lý bộ nhớ, xử lý dữ liệu truy xuất, và tích hợp các công cụ bên ngoài.

FastAPI đóng vai trò như một công cụ trung gian mạnh mẽ để giao tiếp giữa các hệ thống và triển khai các chức năng này. Cụ thể, với sự hỗ trợ của các thư viện Python như **numpy** và **pandas** (xử lý dữ liệu), **ChromaDB** (quản lý cơ sở dữ liệu ngữ nghĩa), cùng các tính năng của LangChain Python, hệ thống có thể thực hiện các tác vụ agentic chunking một cách chính xác và hiệu quả. Ngoài ra, tính năng xử lý không đồng bộ của FastAPI đảm bảo hiệu suất cao, giúp tối ưu hóa thời gian phản hồi khi giao tiếp với mô hình ngôn ngữ lớn và người dùng.

Mặc dù FastAPI mang lại nhiều ưu điểm vượt trội, việc triển khai trên môi trường thực tế có thể đòi hỏi sự tối ưu hóa về mặt cấu hình để đáp ứng khối lượng truy vấn lớn một cách ổn định. Tuy nhiên, với sự hỗ trợ toàn diện từ hệ sinh thái Python và sự linh hoạt của FastAPI, framework này được chọn làm nền tảng xử lý các chức năng phức tạp và đặc thù của đề tài, đảm bảo sự hiệu quả và tính mở rộng cao cho hệ thống.

3.6. MySQL

Hình 3.17 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở, được phát triển bởi MySQL AB và hiện thuộc sở hữu của Oracle Corporation. Nó sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để quản lý, lưu trữ, và truy vấn dữ liệu. MySQL được thiết kế để xử lý khối lượng dữ liệu lớn và các giao dịch phức tạp với hiệu năng cao, đồng thời vẫn đảm bảo tính ổn định và dễ sử dụng.

3.7. ChromaDB

Hình 3.18 Logo ChromaDB

3.7.1. ChromaDB là gì?

Được xem là một cơ sở dữ liệu vector mã nguồn mở, chuyên biệt được thiết kế để phục vụ cho công tác quản lý và truy xuất các vector nhúng hoặc vector đặc trưng, ChromaDB cung cấp những vector biểu diễn toán học cho dữ liệu, thường được trích xuất từ các mô hình học sâu hoặc trí tuệ nhân tạo, chẳng hạn như mô hình xử lý ngôn ngữ tự nhiên (NLP) hay thị giác máy tính (computer vision). ChromaDB cho phép lưu trữ các vector cùng với siêu dữ liệu liên quan và hỗ trợ các ứng dụng tích hợp giành cho mô hình ngôn ngữ lớn (LLM), là một công cụ tìm kiếm dựa trên ngữ nghĩa cho các dữ liệu văn bản, và phân tích các dữ liệu đa phương tiện khác như hình ảnh hoặc âm thanh. Bên cạnh đó, ChromaDB cung cấp cũng mang đến cho các nhà phát triển những tính

năng vô cùng mạnh mẽ như tìm kiếm vector, tìm kiếm toàn văn bản, lưu trữ tài liệu, lọc siêu dữ liệu, và truy xuất đa phương tiện, giúp tối ưu hóa các tác vụ như tìm kiếm ngữ nghĩa, phân cụm, hoặc gợi ý thông minh.

3.7.2. Lịch sử phát triển của ChromaDB

ChromaDB có một lịch sử phát triển đi đổi với sự bùng nổ của học máy và trí tuệ nhân tạo, đặc biệt là nhu cầu xử lý dữ liệu phi cấu trúc trong các ứng dụng hiện đại, chẳng hạn như dữ liệu văn bản và hình ảnh. Khi các mô hình học sâu ngày càng phát triển, chúng tạo ra các vector đặc trưng có độ chính xác và khả năng khái quát hóa cao hơn, dẫn đến nhu cầu ngày càng tăng về một hệ thống cơ sở dữ liệu tối ưu để xử lý chúng.

Giai đoạn khởi đầu

ChromaDB được ra đời để phục vụ cho nhu cầu lưu trữ và truy xuất vector đặc trưng với hiệu suất cao và khả năng mở rộng. Những cơ sở dữ liệu vector ban đầu trên thị trường thường gặp phải hạn chế về tốc độ truy xuất và khả năng tích hợp với các mô hình AI. Các nhà phát triển ChromaDB đã tập trung vào việc giải quyết những vấn đề này thông qua việc tối ưu hóa kiến trúc lưu trữ và thuật toán tìm kiếm vector.

Sự phát triển gần đây

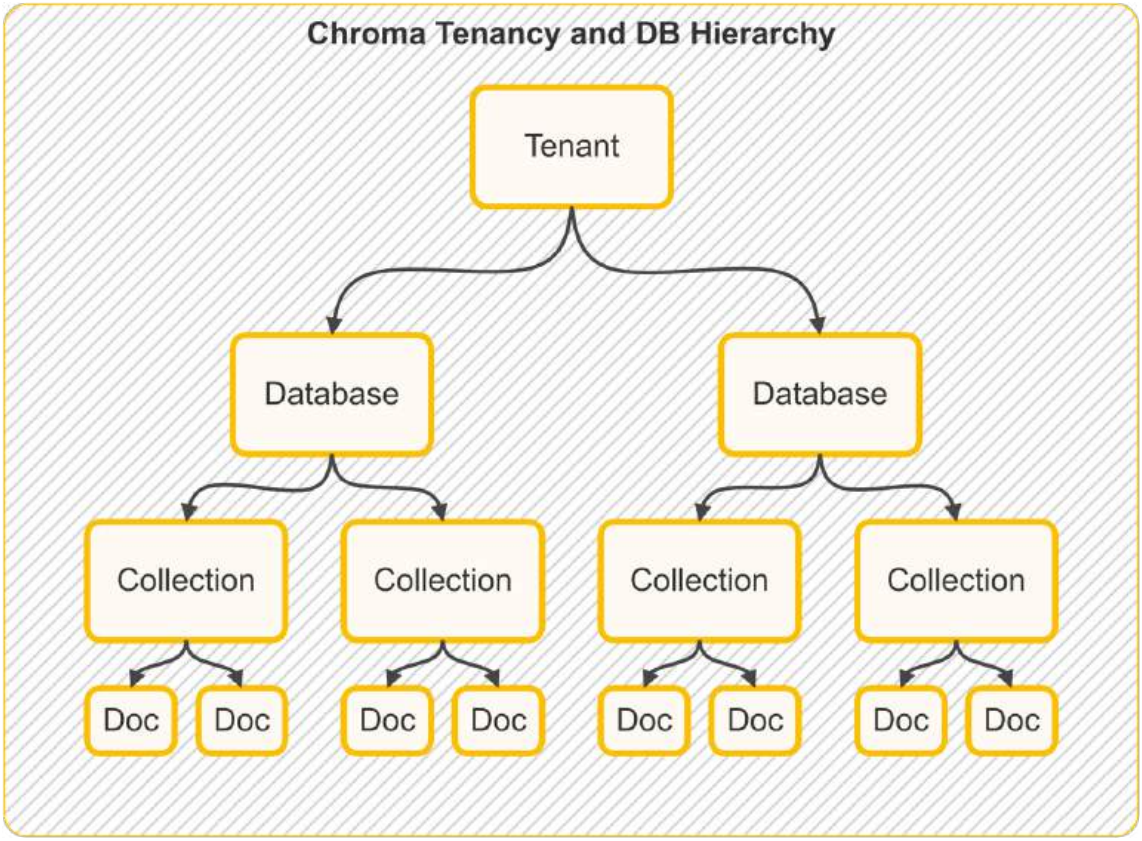
Trong những năm gần đây, ChromaDB đã trở thành một điểm đến phổ biến trong sự lựa chọn để phát triển một hệ sinh thái AI, nhờ khả năng tích hợp tốt với các mô hình AI hàng đầu như GPT của OpenAI hoặc các mô hình từ Hugging Face. Đồng thời, ChromaDB cũng hỗ trợ tìm kiếm tương tự dựa trên thuật toán gần đúng, như Approximate Nearest Neighbor (ANN), giúp tăng tốc độ truy xuất trên tập dữ liệu khổng lồ mà không làm ảnh hưởng đến độ chính xác quá nhiều.

ChromaDB ngoài việc dược thiết kế để xử lý tốt các tập dữ liệu lớn, cơ sở dữ liệu này còn được phát triển để chú trọng vào tính khả dụng và dễ tích hợp. Điều này giúp các nhà phát triển ứng dụng AI, đặc biệt là trong các lĩnh vực như

chatbot, tìm kiếm thông minh, và phân tích ngữ nghĩa, dễ dàng triển khai và ứng dụng.

ChromaDB đang dần phản ánh sự trưởng thành của lĩnh vực cơ sở dữ liệu vector, khi nhu cầu xử lý dữ liệu phức tạp ngày càng tăng và các công cụ như ChromaDB như là một trong những yếu tố quan trọng trong công cuộc phát triển và hỗ trợ các ứng dụng AI trong thời buổi ngày nay.

3.7.3. Cấu trúc của ChromaDB



Hình 3.19 Mô hình cấu trúc của ChromaDB[4]

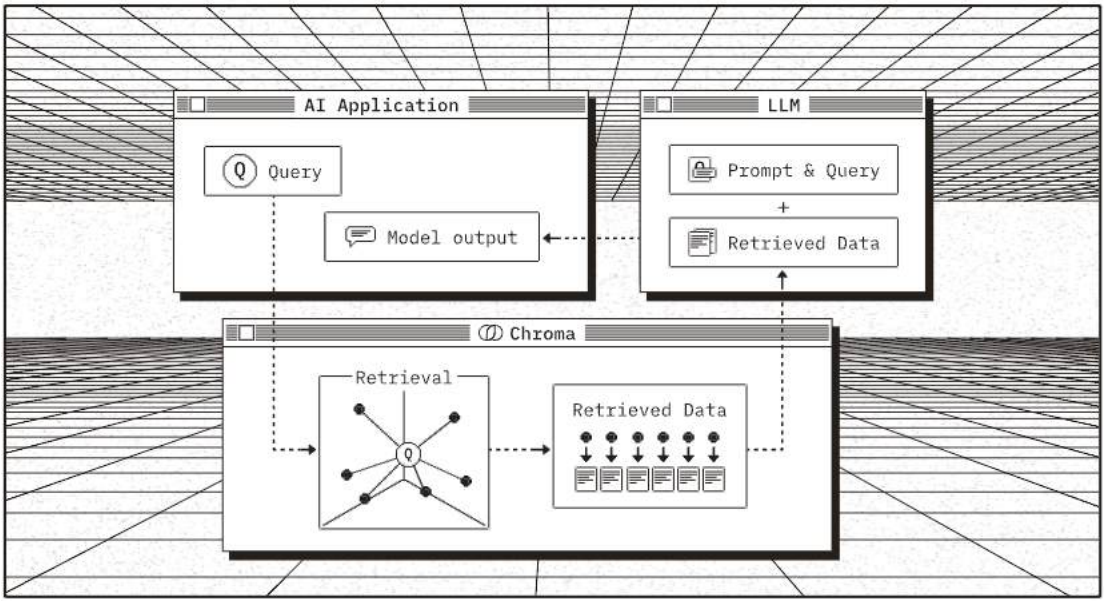
Trong ChromaDB, dữ liệu được tổ chức theo cấu trúc phân cấp:

- **Tenant:** Đại diện cho một tổ chức hoặc cá nhân sử dụng ChromaDB, chứa một tập hợp các cơ sở dữ liệu.

[4] Nguồn: ChromaDB Cookbook

- **Cơ sở dữ liệu**: Mô hình hóa một ứng dụng hoặc dự án cụ thể, chứa nhiều bộ sưu tập.
- **Bộ sưu tập (Collection):** Nhóm các nhúng, tài liệu và siêu dữ liệu liên quan.
- **Tài liệu (Document):** Các đoạn văn bản được liên kết với một nhúng và được lưu trữ trong cơ sở dữ liệu để truy vấn.
- **Siêu dữ liệu (Metadata):** Tập hợp các cặp khóa-giá trị liên quan đến nhúng, hỗ trợ việc lọc và truy vấn dữ liệu.

3.7.4. Cách hoạt động của ChromaDB



Hình 3.20 Nguyên lý hoạt động của ChromaDB[5]

1. **Tạo bộ sưu tập**: Tương tự như tạo bảng trong cơ sở dữ liệu quan hệ, chứa các nhúng, tài liệu và siêu dữ liệu.
2. **Thêm tài liệu**: ChromaDB tự động chuyển đổi văn bản thành nhúng sử dụng mô hình nhúng được chỉ định.
3. **Truy vấn**: Sử dụng nhúng để tìm các tài liệu tương tự, có thể áp dụng bộ lọc dựa trên siêu dữ liệu.

[5] Nguồn: ChromaDB Documentation

3.7.5. Tính năng của ChromaDB

- **Đơn giản và mạnh mẽ:** Dễ dàng cài đặt và tích hợp với các SDK Python và JavaScript/TypeScript.
- **Đầy đủ tính năng:** Hỗ trợ tìm kiếm vector, tìm kiếm toàn văn bản, lưu trữ tài liệu, lọc siêu dữ liệu và truy xuất đa phương tiện.
- **Khả năng mở rộng cao:** Hỗ trợ các backend lưu trữ như DuckDB cho sử dụng độc lập hoặc ClickHouse cho ứng dụng lớn.
- **Hỗ trợ đa ngôn ngữ:** Cung cấp SDK cho nhiều ngôn ngữ lập trình phổ biến.
- **Tích hợp:** Tích hợp với các mô hình nhúng từ HuggingFace, OpenAI, Google và các công cụ như LangChain và LlamaIndex.
- **Mã nguồn mở:** Được cấp phép dưới Apache 2.0.

Ứng dụng của ChromaDB

ChromaDB được áp dụng trong các ứng dụng cần xử lý dữ liệu vector có kích thước lớn, như mô hình ngôn ngữ lớn và công cụ tìm kiếm ngữ nghĩa trên dữ liệu văn bản. Nó cho phép lưu trữ các vector nhúng kèm theo siêu dữ liệu liên quan, đáp ứng các trường hợp sử dụng phức tạp hơn so với việc chỉ lưu trữ dữ liệu đơn thuần.

ChromaDB là một công cụ đắc lực dành cho các ứng dụng AI và ML, cung cấp giải pháp tối ưu để lưu trữ và truy xuất các vector nhúng. Với khả năng tích hợp linh hoạt và hỗ trợ đa ngôn ngữ, ChromaDB đang dần trở thành lựa chọn ưu tiên của các nhà phát triển trong việc phát triển các ứng dụng xử lý ngôn ngữ tự nhiên và tìm kiếm ngữ nghĩa.

3.8. Google Calendar API

Google Calendar API là một giao diện lập trình ứng dụng RESTful cho phép tích hợp và tương tác với dịch vụ Lịch Google. API này hỗ trợ và cho phép các ứng dụng có thể thực hiện các thao tác tuỳ chỉnh dữ liệu liên quan đến sự

kiện, quản lý lịch và thiết lập quyền truy cập, tương tự như các chức năng có trong giao diện web của Lịch Google.

3.8.1. Các khái niệm chính trong Google Calendar API:

- **Sự kiện (Event):** Đại diện cho một hoạt động trên lịch, chứa thông tin như tiêu đề, thời gian bắt đầu và kết thúc, và danh sách người tham gia. Sự kiện có thể là đơn lẻ hoặc lặp lại.
- **Lịch (Calendar):** Tập hợp các sự kiện, mỗi lịch có siêu dữ liệu như mô tả và múi giờ mặc định.
- **Danh sách lịch (Calendar List):** Danh sách tất cả các lịch mà người dùng có trong giao diện Lịch Google, bao gồm các thuộc tính như màu sắc và thông báo cho sự kiện mới.
- **Thiết lập (Setting):** Tùy chọn của người dùng từ giao diện Lịch, chẳng hạn như múi giờ.
- **Quy tắc kiểm soát truy cập (ACL):** Quy tắc cấp quyền cho người dùng hoặc nhóm người dùng với mức độ truy cập cụ thể vào một lịch.

3.8.2. Cách thức hoạt động của Google Calendar API:

Google Calendar API hoạt động dựa trên các yêu cầu HTTP, cho phép ứng dụng tương tác với dịch vụ Lịch Google. Các nhà phát triển có thể sử dụng các thư viện ứng dụng do Google cung cấp để đơn giản hóa quá trình này.

3.8.3. Các bước cơ bản để sử dụng Google Calendar API:

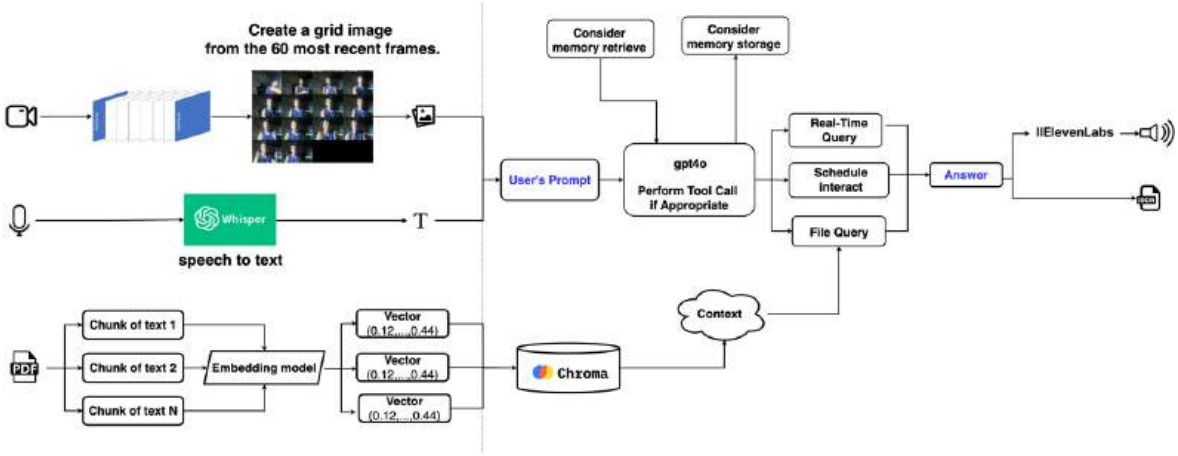
- **Kích hoạt API:** Truy cập Google Cloud Console, tạo một dự án mới hoặc sử dụng dự án hiện có, sau đó kích hoạt Google Calendar API cho dự án đó.
- **Thiết lập xác thực:** Cấu hình màn hình chấp thuận OAuth và tạo thông tin xác thực phù hợp (như OAuth 2.0 Client ID) để ứng dụng có thể truy cập API một cách an toàn.
- **Cài đặt thư viện ứng dụng:** Tùy thuộc vào ngôn ngữ lập trình sử dụng, cài đặt thư viện ứng dụng tương ứng để tương tác với API.

- **Thực hiện yêu cầu API:** Sử dụng thư viện ứng dụng để gửi yêu cầu đến API, chẳng hạn như tạo sự kiện mới, truy vấn danh sách sự kiện hoặc cập nhật thông tin sự kiện hiện có.

Chương 4. **PHƯƠNG PHÁP TRIỂN KHAI**

4.1. Tổng quan kiến trúc hệ thống

Luồng xử lý của hệ thống bao gồm 4 giai đoạn chính: (1) Xử lý và phân tích thông tin được tiếp nhận; (2) Phân tích và đồng bộ dữ liệu hội thoại; (3) Phân tích yêu cầu và xử lý kết quả; (4) Xuất dữ liệu.



Hình 4.1 Luồng xử lý hệ thống

4.1.1. Xử lý và phân tích thông tin được tiếp nhận:

Quá trình này tiếp nhận thông tin đầu vào qua các phương thức tương tác: bằng văn bản, file, hình ảnh, một url bất kỳ, giọng nói và đặc biệt là khả năng tương tác trực tiếp qua việc gọi video và chia sẻ màn hình. Sự linh hoạt này cho phép hệ thống tương tác với người dùng trong nhiều hoàn cảnh và lĩnh vực khác nhau, từ đó nâng cao khả năng tiếp cận và phản hồi trước yêu cầu đa dạng của người dùng:

4.1.1.1. Tương tác qua giọng nói:

Tận dụng chức năng từ opensource silence-aware-recorder[6] để giải quyết vấn đề về việc khi nào nên ngừng ghi âm và bắt đầu gọi API. Đây là một nguồn mở cho phép nhận diện khi người dùng ngừng nói trong một khoảng thời gian

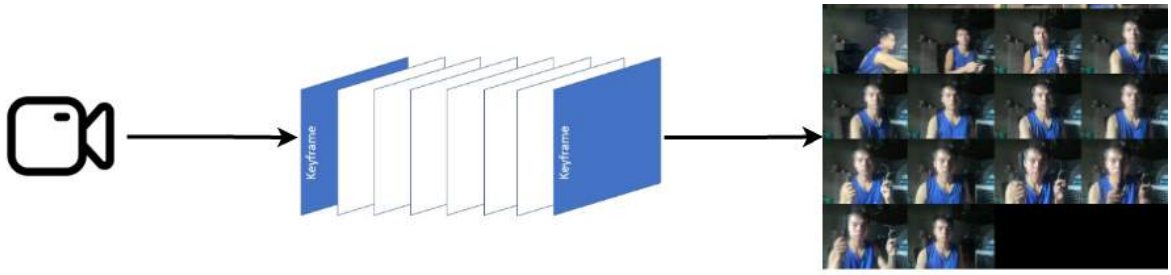
[6] Nguồn https://github.com/teunlao/silence-aware-recorder

nhất định, giúp người dùng không cần thao tác thủ công. Với khả năng nhận diện đa ngôn ngữ của mô hình Whisper được cung cấp bởi OpenAI, công nghệ này xử lý quá trình chuyển đổi từ giọng nói thành văn bản một cách chính xác giải quyết cho vấn đề nhận diện đa ngôn ngữ cho trợ lý ảo.

4.1.1.2. Tương tác qua gọi video và chia sẻ màn hình:

Đối với vấn đề về việc làm sao để một trợ lý ảo có thể nhận diện bối cảnh và tương tác với lượng dữ liệu trong quá trình gọi video và chia sẻ màn hình liên tục. Việc tích hợp khả năng cho phép xử lý dữ liệu đầu vào thông qua thị giác trực tiếp với mô hình ngôn ngữ lớn để nhận biết bối cảnh vẫn là một nhiệm vụ khó khăn và nhiều hạn chế chẳng hạn như giới hạn về lượng "context window"[7] của một mô hình ngôn ngữ làm cho việc xử lý liên tục khối lượng lớn dữ liệu như một video trong một lần rất là khó khăn.

Ở đề tài này đề xuất một giải pháp để xử lý cho vấn đề trên bằng cách chuyển đổi video thành từng frame và sử dụng khả năng nhận diện khi người dùng ngừng nói, như đã đề cập ở phần tương tác bằng giọng nói, để xây dựng và mô phỏng quá trình ghi nhận thông tin bối cảnh từ người dùng đang tương tác.



Create a grid image from the 60 most recent frames.

Hình 4.2 Chuyển đổi video thành một tấm ảnh từ các frames khoảnh khắc

[7] Context window: là giới hạn số lượng từ hoặc token mà mô hình ngôn ngữ có thể xử lý trong một lần, ảnh hưởng đến khả năng ghi nhớ ngữ cảnh của mô hình.

Dữ liệu video trong quá trình tương tác sẽ được ghi nhận và tổng hợp thành tối đa 60 khung hình gần nhất. Các khung hình này sau đó sẽ được gộp lại thành một tấm ảnh dạng lưới thể hiện các khoảnh khắc chuyển động trong một bức ảnh duy nhất từ đó, đưa vào mô hình xử ngôn ngữ lớn các khả năng xử lý hỉnh ảnh như gpt4o kết hợp với cấu trúc prompt đã được định nghĩa bối cảnh là một môi trường tương tác video giữa người dùng và trợ lý ảo.

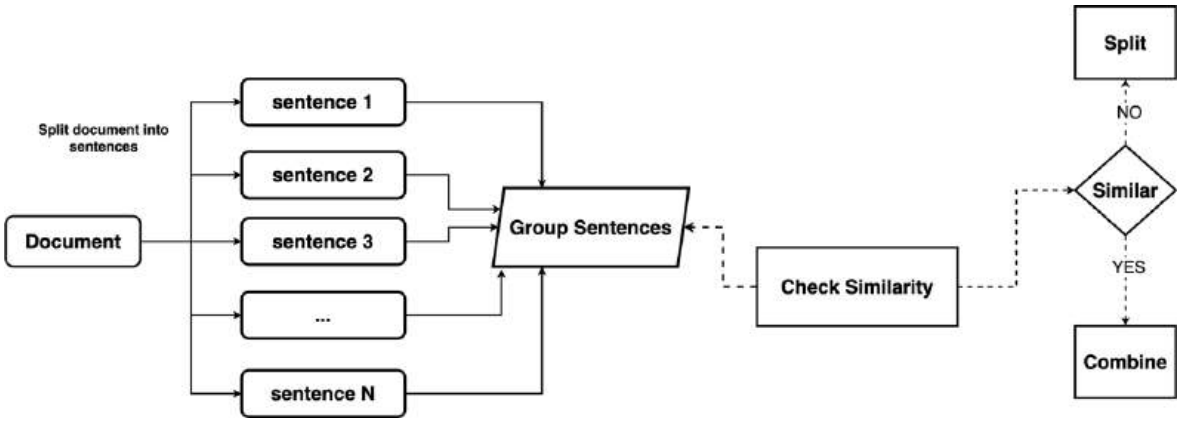
```
Context: The AI receives a tiled series of screenshots from a human's live
video feed. These screenshots represent sequential frames from the video,
capturing distinct moments. The AI is to analyze these frames as a
continuous video feed, answering human's questions while focusing on direct
and specific interpretations of the visual content.

1. When the human asks a question, use spatial and temporal information
from the video screenshots.
2. Respond with brief, precise answers to the human questions. Go straight
to the point, avoid superficial details. Be concise as much as possible.
3. Address the human directly, and assume that what is shown in the images
is what the human is doing.
4. Use "you" and "your" to refer to the human.
5. DO NOT mention a series of individual images, frames, a strip, a grid, a
pattern or a sequence. Do as if the AI is interacting with the human
through the video call screen.
6. DO NOT be over descriptive.
7. AI will not interact with what is shown in the images. It is the human
that is interacting with the objects in the images.
8. Keep in mind that the grid of images will show the same object in a
sequence of time. E.g. If an identical glass is shown in several
consecutive images, it is the same glass and NOT multiple glasses.
9. When asked about spatial questions, provide clear and specific
information regarding the location and arrangement of elements within the
frames. This includes understanding and describing the relative positions,
distances, and orientations of objects and people in the visual field, as
if observing a real-time 3D space.
10. If the human gives instructions, follow them precisely.
```

Hình 4.3 Prompt định nghĩa bối cảnh tương tác video cho mô hình ngôn ngữ

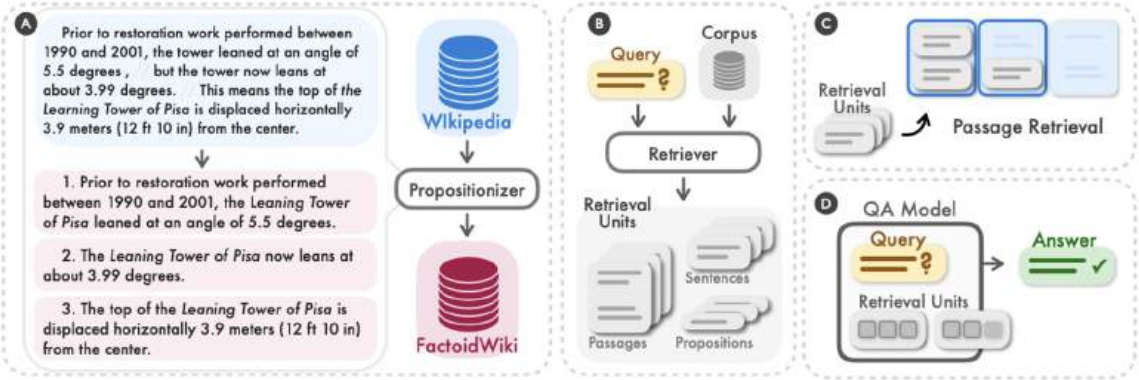
Từ đó giúp mô hình có thể dễ dàng nhận biết được bối cảnh để đưa ra loại câu trả lời phù hợp cho một hoàn cảnh tương tác gọi điện video.

4.1.1.3. Tiền xử lý file



Hình 4.4 Tổng quan phương pháp Agentic chunking

Các tập dữ liệu lớn sẽ được tiền xử lý bằng cách chia nhỏ toàn bộ nội dung của thành các chunk thông qua phương pháp phân đoạn nội dung (chunking) để chia văn bản thành các chunk có ý nghĩa, hoàn chỉnh về mặt ngữ nghĩa. Tận dụng phương pháp proposal-based chunking được nghiên cứu từ công trình của Tong Chen và cộng sự năm 2023 [7].



Hình 4.5 Phương pháp proposal-based chunking

Nghiên cứu này đề xuất một giải pháp xử lý thông tin văn bản lớn thành các mệnh đề nhỏ, tại đó mỗi mệnh đề đều có ý nghĩa riêng dựa trên bối cảnh của chỉnh nó, giúp cho các phân đoạn văn bản nhỏ có thể đứng độc lập và cung cấp

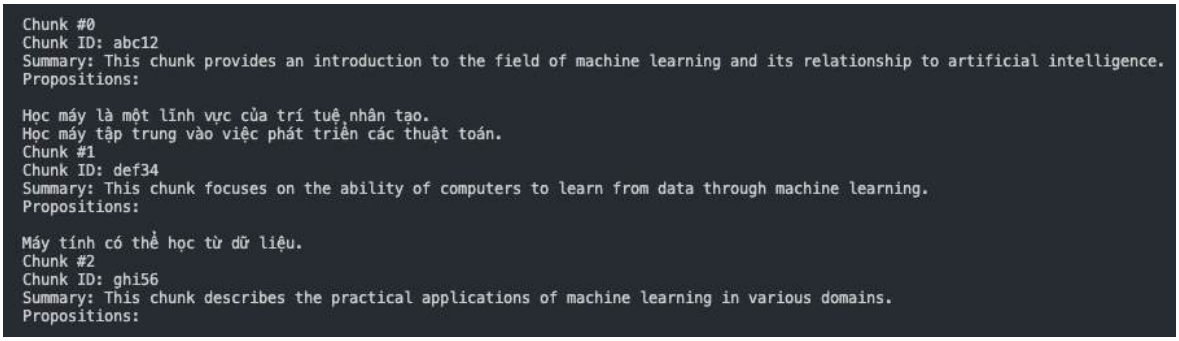
một ý nghĩa hoàn chỉnh mà không cần phụ thuộc vào các yếu tố diễn giải từ các đoạn văn bản trước đó.

Bảng 4.1 Bảng ví dụ của phương pháp proposition chunking

Văn bản gốc	**Phân đoạn theo "proposition":**
Học máy là một lĩnh vực của trí tuệ nhân tạo, tập trung vào việc phát triển các thuật toán để máy tính có thể học từ dữ liệu. Các ứng dụng bao gồm nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và dự đoán dữ liệu.	1. Học máy là một lĩnh vực của trí tuệ nhân tạo. 2. Học máy tập trung vào việc phát triển các thuật toán. 3. Máy tính có thể học từ dữ liệu. 4. Các ứng dụng bao gồm nhận dạng hình ảnh. 5. Các ứng dụng bao gồm xử lý ngôn ngữ tự nhiên. 6. Các ứng dụng bao gồm dự đoán dữ liệu.

Từ bảng 4.1 cho thấy được một cách xử lý giúp giữ vững được tính toàn vẹn của thông tin trong văn bản gốc trong quá trình truy xuất, giúp cho kết quả chính xác hơn và phù hợp với ngữ cảnh hơn, dùng mô hình ngôn ngữ để dánh giá bối cảnh toàn cục và viết lại, bổ sung thêm chủ ngữ cho các câu phân đoạn.

Đề tài đã tận dụng từ phương pháp trên, nghiên cứu và tận dụng thêm một phương pháp gọi là “Agentic Chunking” để giúp tối ưu hoá dữ liệu hơn nữa trước khi đưa vào cơ sở dữ liệu phụ vụ cho các câu trả lời của trợ lý ảo, bằng cách sử dụng mô hình ngôn ngữ để tóm tắt, từ đó phân tích và nhóm các đoạn văn bảng có chung ý nghĩ lại thành một đoạn văn bảng lớn



Hình 4.6 Ví dụ về phương pháp agentic chunking

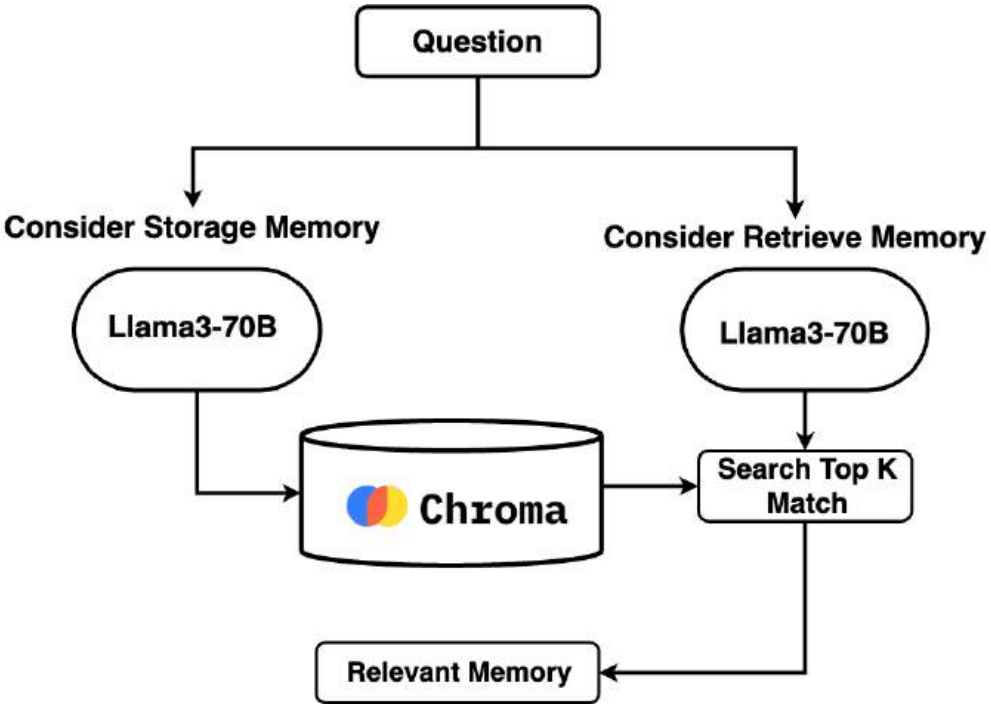
Các chunk này sau đó sẽ được chuyển đổi dưới dạng các vector bằng mô hình embedding text-embedding-3-large từ OpenAI và lưu trữ dưới vector database để phục vụ cho việc hỏi đáp sau này.

4.1.2. Phân tích và đồng bộ dữ liệu hội thoại

Ngoài khả năng ghi nhớ thông tin trong lịch sử của từng cuộc hội thoại như các trợ lý ảo hiện nay, để mô phỏng lại khả năng ghi nhớ của con người cho trợ lý ảo và cải thiện điều đó, đề tài triển khai hai phương thức để phân tích xử lý thông tin cho bộ nhớ dài hạn và bộ nhớ ngắn hạn cho trợ lý ảo

4.1.2.1. Phân tích và xử lý thông tin cho bộ nhớ dài hạn

Tận dụng từ mô hình kỹ thuật Tree of Thought Prompting (ToT) trong Prompt Engineering được nghiên cứu bởi Shunyu Yao và cộng sự năm 2023 [1] và nghiên cứu của Jieyi Long năm 2023 [2] , đây là kĩ thuật cho phép mô hình ngôn ngữ khám phá nhiều nhánh suy luận khác nhau, đánh giá các quyết định từ đó cải thiện khả năng giải quyết vấn đề của các mô hình ngôn ngữ lớn, từ đây đề tài xây dựng nên khả năng ghi nhớ dài hạn cho phép trợ lý ảo từ việc việc phân tích nội dung từ người dùng thông qua nhiều bước để đưa ra các quyết định lưu trữ, học tập và truy xuất lại các thông tin đã được học từ các cuộc hội thoại trước, giúp nâng cao tính linh hoạt và khả năng phản hồi chính xác, tăng trải nghiệm hơn cho người dùng.



Hình 4.7 Quy trình phân tích và tìm kiếm bộ thông tin cho bộ nhớ dài hạn

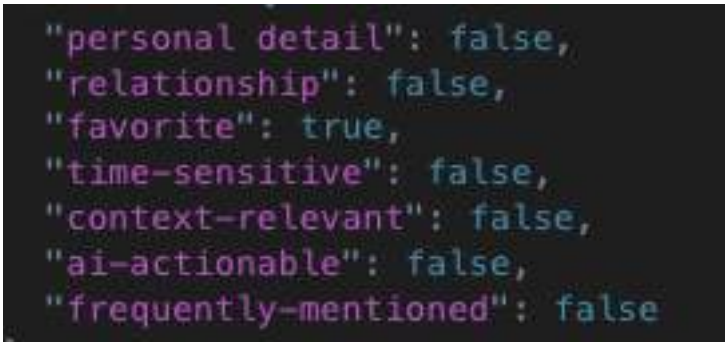
Đề tài xây dưng một tính năng mô phỏng lại quá trình tư duy theo chuỗi của mộ hình ngôn ngữ để giải quyết cho bài toán:

- Có nên **lưu thông tin** từ câu nói của người dùng hay không ?
- Có nên **truy xuất thông tin** từ bộ nhớ liên quan hay không ?

Từ đó xây dựng nên hai quy trình phân tích, đó là “phân tích để lấy dữ liệu liên quan từ vector database" và "phân tích để lưu trữ vào bộ nhớ", hai quy trình này cùng góp phần vào khả năng ghi nhớ và sử dụng thông tin của trợ lý ảo, mỗi quy trình tập trung vào một khía cạnh khác nhau của việc quản lý thông tin.

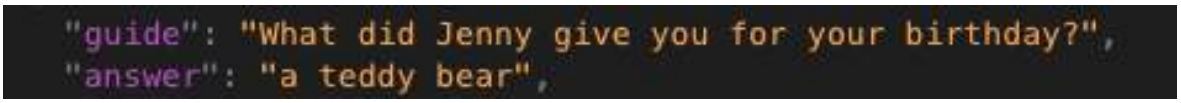
Phân tích để lưu trữ vào bộ nhớ (Memo Storage):

Thông qua mô hình gpt4o-mini prompt từ người dùng sẽ được chia ra và phân tích lần lượt dựa theo các tiêu chí nhỏ



Hình 4.8 Các tiêu chí đánh giá độ liên quan của thông tin

Từ đó giá nội dung có liên quan đến người dùng và có ích cho việc sử dụng sau này không, nếu các kết quả phân tích đều thoả mãn, mô hình ngôn ngữ sẽ dựa trên prompt của người để phân tích và trích xuất ra cặp câu trả lời và một lời khuyên liên quan đến câu trả lời đó để lưu trữ cặp dữ liệu đấy vào cơ sở dữ liệu vector.



Hình 4.9 Ví dụ về cặp câu trả lời gợi ý cho bộ nhớ

Phân tích để lấy dữ liệu liên quan từ vector database (Memo Retrieval):

Mô hình gpt4o-mini sẽ phân tích prompt đầu vào để xác định các khía cạnh quan trọng của tin nhắn. sau đó sử dụng tin nhắn hoặc các phần tổng quát hóa của nó để tìm kiếm các memo liên quan trong cơ sở dữ liệu vector. Các dữ liệu được truy xuất sẽ được so sánh với tin nhắn hiện tại để đảm bảo chúng đủ liên quan. Các dữ liệu có khoảng cách nhỏ hơn ngưỡng xác định trước từ trong vector database sẽ được chọn lọc và trở thành dữ liệu bộ nhớ có liên quan và sẽ được sử dụng để bổ sung vào ngữ cảnh cuộc trò chuyện hiện tại, giúp mô hình có thể trả lời câu hỏi của người dùng một cách chính xác và cụ thể hơn.

4.1.2.2. Phân tích và xử lý thông tin cho bộ nhớ ngắn hạn

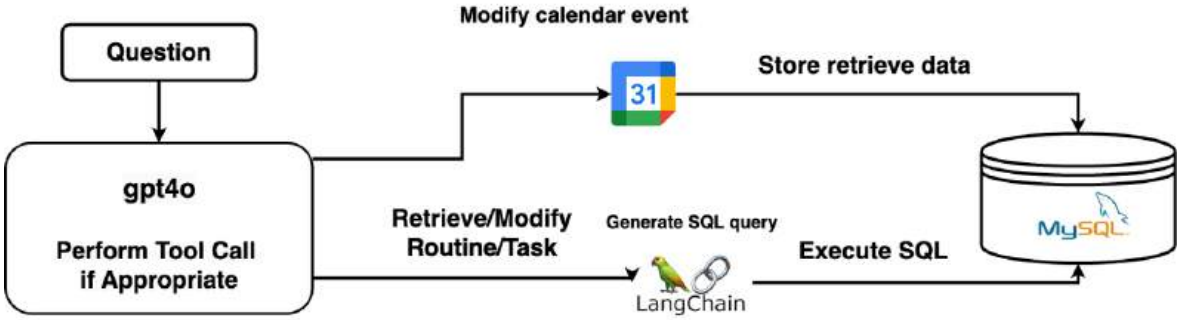
Bộ nhớ ngắn hạn bao gồm dữ liệu được tóm tắt lại từ toàn bộ lịch sử của các lần trò chuyện trước đó giữa người dùng và trợ lý ảo để giúp cho mô hình

có thể nhận biết được tổng quan về cuộc trò chuyện trước đó, bên cạnh đó bộ nhớ ngắn hạn cũng bao gồm các dữ liệu "instruction prompt"[8] liên quan đến ngày tháng hiện tại hay các thiết lập về tính cách và chỉ dẫn mà trợ lý ảo phải tuân theo.

4.1.3. Phân tích yêu cầu và xử lý kết quả

Tận dụng chức năng tool calling của mô hình gpt4-o với khả năng nhận diện và phân tích prompt từ yêu cầu mà người dùng đang hướng đến để xem xét việc sử dụng sự trợ giúp từ các công cụ đã được thiết lập sẵn cho trợ lý ảo

4.1.3.1. Đặt lịch nhắc nhở, lên kế hoạch:



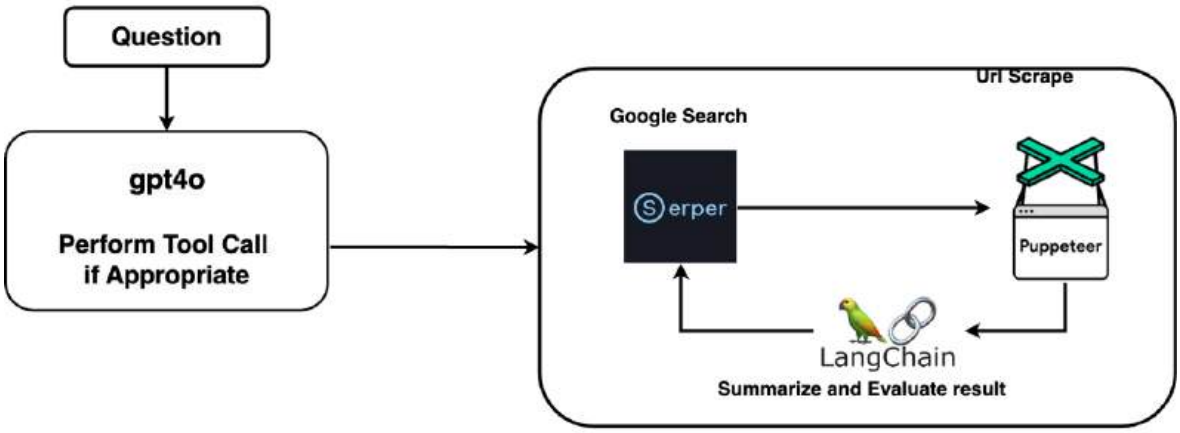
Hình 4.10 Quy trình lên kế hoạch/lịch hẹn của trợ lý ảo

Dựa trên đánh giá về độ chính xác từ khả năng chuyển đổi ngôn ngữ tự nhiên sang câu lệnh SQL của mộ hình ngôn ngữ được nghiên cứu từ công trình của Nitarshan Rajkumar và cộng sự năm 2022[3], đề tài tận dụng phương pháp này để xây dựng khả năng tương tác với cơ sở dữ liệu trong phạm vi các table lưu trữ dữ liệu nhắc nhở từ langchain cho phép mô hình có thể thực hiện các thao tác tìm kiếm thêm xóa sửa đối với các dữ liệu nhắc nhở dựa trên prompt của người dùng, đồng thời dữ liệu các lịch nhắc nhở sẽ luôn được đồng bộ lên google calendar tương ứng với tài khoản google đã được liên kết với tài khoản

[8] **Instruction prompt**: là một dạng hướng dẫn được đưa ra cho mô hình AI, mô tả rõ ràng nhiệm vụ cần thực hiện. Nó thường bao gồm các chỉ dẫn ngắn gọn, cụ thể, giúp AI hiểu cách trả lời hoặc xử lý một vấn đề nhất định.

hiện tại đang tương tác với trợ lý ảo, với tính năng parallel tool calling cho phép mô hình có thể thực hiện quá trình thêm bớt hay điều chỉnh dữ liệu nhắc nhở nhiều lần, điều này mang lại cho trợ lý ảo khả năng thiết lập hay lên một lịch trình học tập, làm việc một cách tự động và nhanh chóng

4.1.3.2. Phân tích tìm kiếm thông tin từ internet:



Hình 4.11 Quy trình tìm kiếm thông tin từ internet của trợ lý ảo

Với các yêu cầu đòi hỏi những thông tin mang tính chính xác thời gian thực, các dữ liệu mang tính biến động nhiều và nằm ngoài dữ liệu được huấn luyện của mô hình, trợ lý ảo sẽ trực tiếp tìm kiếm và phân tích các thông tin từ internet để trả lời thông qua quy trình sau:

- **Bước 1:** Thực hiện tìm kiếm các kết quả từ Google thông qua Serper API. Mô hình sẽ tự động phân tích các từ khóa liên quan đến câu hỏi từ prompt để thực hiện việc tìm kiếm.
- **Bước 2:** Sau khi có được các URL cần thiết, sử dụng công cụ Puppeteer để scrape toàn bộ nội dung từ các trang web.
- **Bước 3:** Tiến hành tóm tắt, lại số dữ liệu vừa scrape được, để đảm bảo token tối đa cho phép với kỹ thuật map reduce từ Langchain để thực hiện tóm tắt tất cả dữ liệu cần thiết

- **Bước 4:** Mô hình sẽ phân tích dữ liệu cuối cùng để đánh giá tính đáp ứng với câu hỏi. Nếu không đáp ứng, quy trình sẽ quay lại từ bước 1, tìm kiếm theo từ khóa khác và lặp lại cho đến khi có câu trả lời phù hợp. Sau đó, mô hình ngôn ngữ sẽ điều chỉnh lại câu từ để đưa ra thông tin tóm gọn mà người dùng mong muốn.

Với mỗi câu trả lời liên quan đến việc tìm kiếm từ internet, trợ lý ảo sẽ luôn đính kèm đường link nguồn từ các trang web có kết quả phù hợp.

4.1.3.3. Hỏi đáp về nội dung liên quan đến các file đã được phân tích

Ứng dụng công nghệ tạo sinh có tăng cường truy xuất (Retrieval Augmented Generation) để có thể giúp cho mô hình ngôn ngữ có thể đưa ra các câu trả lời dựa trên nguồn kiến thức từ các tập dữ liệu lớn đã được phân tích trước đó và phòng tránh các trường hợp "Hallucination". Do đó với các trường hợp prompt yêu cầu trả lời về các thông tin liên quan đến nội dung từ các file đã được tải lên, trợ lý ảo sẽ thực hiện cơ chế retrieval dữ liệu đã được xử lý trước đó bởi phương pháp "Agentic chunking" dưới dạng vector trong database để xác định các phân đoạn tri thức (paragraphs) nào có ngữ nghĩa tương đồng với câu truy vấn của người dùng. Các paragraphs này sau đó được đưa vào mô hình ngôn ngữ lớn để làm tăng context cho quá trình sinh ra câu trả lời, sau đó các paragraphs này được kết hợp với câu query ban đầu của user tạo thành 1 câu prompt. Câu prompt này được bổ sung thêm context sau đó được đưa qua LLM để sinh ra câu phản hồi cuối cùng theo context bổ sung.

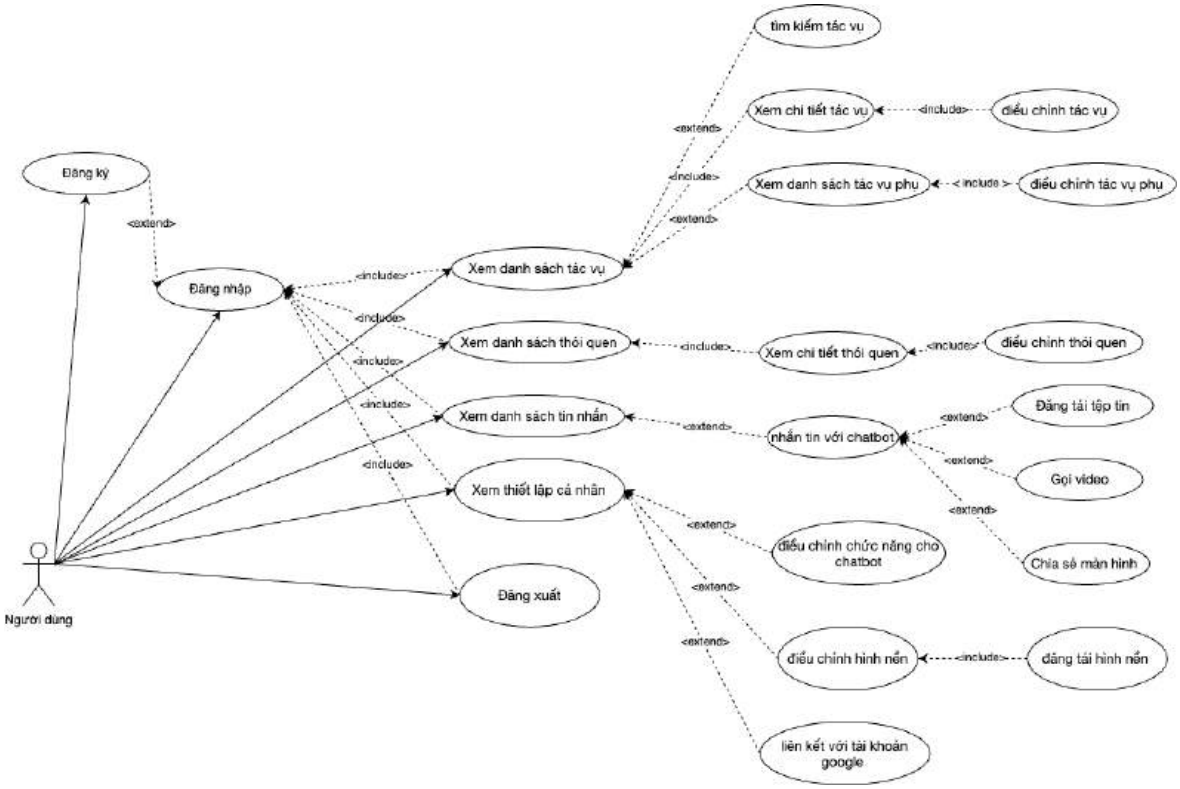
4.1.4. Xuất dữ liệu

Tận dụng khả năng hỗ trợ chunk stream từ mô hình ngôn ngữ, đề tài ứng dụng socket để hỗ trợ khả năng gửi liên tục các dữ liệu chunk được tạo ra về giao diện, đối với các trường hợp mô hình yêu cầu sử dụng tool, socket cũng sẽ thực hiện trả về dữ liệu và quá trình xử lý của các tool đó, điều này cho phép người dùng có thể nắm bắt được quá trình xử lý của trợ lý ảo và đặc biệt là giúp

tăng trải nghiệm sử dụng, người dùng không cần phải đợi toàn bộ quá trình xử lý hoàn thành mà vẫn có thể giám sát được quy trình xử lý khi đang tương tác trên giao diện ứng dụng.

Đối với tính năng gọi video, đề tài áp dụng phương pháp chia nhỏ câu trả lời ra thành nhiều chunk từ đó sử dụng mô hình chuyển đổi tts-1 của openAI để chuyển đổi lần lượt các dạng dữ liệu văn bản sang dữ liệu âm thanh, các dữ liệu âm thành này sau đó sẽ được phát lần lượt cho người dùng, quá trình này cho phép tối ưu được tổng thời gian xử lý của quá trình chuyển đổi bằng việc mô hình sẽ tốn ít thời gian để xử lý từng cụm dữ liệu nhỏ hơn là xử lý một lượng lớn dữ liệu trong một lần.

4.2. Sơ đồ usecase



Hình 4.12 Sơ đồ usecase

4.2.1. Danh sách các tác nhân

- User (Người dùng): Là tác nhân có khả năng try cập vào tất cả các chức năng được hiện thị trên màn hình giao diện trong hệ thống

4.2.2. Danh sách usecase

STT	Use case chính	Ý nghĩa/Ghi chú
1	Đăng ký	Người dùng sử dụng các thông tin như tên tài, email và mật khẩu để tạo một tài khoản mới nhằm truy cập vào hệ thống
2	Đăng nhập	Người dùng sử dụng các tài khoản đã được đăng ký trước đó để nhân đăng nhập vào hệ thống
3	Xem danh sách tác vụ	Hỗ trợ người dùng trong việc theo dõi, quản lý các tác vụ của mình theo ngày, tuần hoặc trạng thái (quá hạn, hôm nay, sắp tới).
4	Tìm kiếm tác vụ	Người dùng có thể chọn một khoảng thời gian nhất định dược hiển thị theo dạng lịch biểu từ một thời điểm nào đó đến một thời điểm bất kỳ khác nhằm lọc và hiển thị các tác vụ trong khoảng thời gian đó.
5	Xem chi tiết tác vụ	Cho phép người dùng xem các thông tin đầy đủ của một tác vụ cụ thể, bao gồm tiêu đề, màu tag, danh mục liên quan, thời hạn, ghi chú, và tệp đính kèm, từ đó hỗ trợ quản lý và chỉnh sửa dễ dàng.
6	Điều chỉnh tác vụ	Người dùng có thể thay đổi các thông tin liên quan đến chi tiết tác vụ, bao gồm: tiêu đề, màu tag, danh mục liên quan, thời hạn,

		nội dung ghi chú.
7	Xem danh sách tác vụ phụ	Hỗ trợ theo dõi và quản lý các công việc nhỏ (subtasks) liên quan đến một tác vụ chính, giúp việc kiểm soát chi tiết hơn các bước cần thực hiện để hoàn thành toàn bộ nhiệm vụ.
8	Điều chỉnh tác vụ phụ	Thay đổi thông tin hoặc trạng thái(hoàn thành hoặc chưa hoàn thành) của tác vụ phụ
9	Xem danh sách thói quen	Cung cấp các dữ liệu liên quan đến thói quen của người dùng theo hai mục chính, việc cần làm và việc không cần phải làm (việc không cần phải làm là những việc mà người dùng tạm thời)
10	Xem chi tiết thói quen	Cho phép người dùng truy cập và xem đầy đủ thông tin về một thói quen, bao gồm tiêu đề, màu tag, danh mục liên quan (như sức khỏe, tài chính, vui chơi, v.v.), trạng thái (hoạt động hoặc không hoạt động), thời gian lặp lại (routine time), ngày hoàn thành, và ghi chú chi tiết.
11	Điều chỉnh thói quen	Cho phép cập nhật và chỉnh sửa các thông tin chi tiết liên quan đến một thói quen, bao gồm tiêu đề, màu tag, danh mục liên quan, trạng thái hoạt động, thời gian lặp lại, và ghi chú. Ngoài ra, chức năng trong hình còn thể hiện tính năng đặc biệt là cho phép người dùng chọn những ngày đã

		hoàn thành thói quen, giúp ghi nhận lịch sử và theo dõi tiến độ thực hiện thói quen theo thời gian.
12	Xem danh sách tin nhắn	Cho phép người dùng nắm được dữ liệu về lịch sử nhắn tín của người dùng và chatbot
13	Nhắn tin với chatbot	Người dùng có thể tương tác bằng cách nhắn tin trực tiếp với chatbot và chat bot sẽ thực hiện trả lời lại tin nhắn người dùng thông qua phương thức stream liên tục dữ liệu được tạo ra
14	Đăng tải tệp tin	Người dùng thực hiện đăng tải một tập tin văn bản từ đó chatbot sẽ đưa ra các câu trả lời dựa trên nội dung bên trong tập tin
15	Gọi video	Trợ lý ảo có thể trực tiếp tương tác với người dùng thông qua giọng nói và quan sát được bối cảnh xung quanh của người dùng thông qua camera mà trong khi đó người dùng không cần thao tác gì trong quá trình gọi
16	Chia sẻ màn hình	Người dùng chia sẻ màn hình cho trợ lý ảo và thực hiện tương tác bằng giọng nói
17	Xem thiết lập cá nhân	Người dùng có thể xem tổng quan chi tiết cài đặt của tài khoản
18	Điều chỉnh chức năng cho chatbot	Thiết lập bật tắt các khả năng mà trợ lý ảo có thể sử dụng
19	Điều chỉnh hình nền	Cho phép thay đổi màn hình nền cho toàn bộ ứng dụng
20	Đăng tải hình nền	Cho phép người dùng đăng tải một tấm

		hình bất kỳ từ thiết bị để dùng làm hình nền
21	Liên kết với tài khoản Google	Liên Kết tài khoản người dùng hiện tại với tài khoản google để dùng cho chức năng đồng bộ dữ liệu với Google Calendar
22	Đăng xuất	Thoát khỏi trang chính của hệ thống và buộc người dùng phải đăng nhập lại trong lần tới

4.2.3. Mô tả chi tiết usecase

4.2.3.1. Đăng ký

Bảng 4.2 Đặc tả use case đăng ký

Usecase: Đăng ký	
Mô tả	Người dùng nhập họ tên, mật khẩu, email để tạo tài khoản trên hệ thống
Tác nhân	Người dùng
Điều kiện	Email chưa đăng ký trên hệ thống và thông tin thoả điều kiện hệ thống
Luồng sự kiện	1. Người dùng chọn tính năng Đăng ký ở màn hình Đăng nhập 2. Người dùng nhập thông tin trên màn hình Đăng ký 3. Hệ thống sẽ kiểm tra thông tin tài khoản đã tồn tại chưa 4. Hệ thống tự động đăng nhập và hiển thị giao diện trang thiết lập tác vụ.
Ngoại lệ	2a. Người dùng nhập lại mật khẩu không khớp với mật khẩu được đặt 3a. Email đã có người đăng ký trên hệ thống

4.2.3.2. Đăng nhập

Bảng 4.3 Đặc tả use case đăng nhập

Usecase: Đăng nhập	
Mô tả	Khi muốn tương tác trong hệ thống, người dùng cần phải đăng nhập. Tính năng đăng nhập hỗ trợ xác thức người dùng thông qua tài khoản và mật khẩu
Tác nhân	Người dùng
Điều kiện	Tài khoản và mật khẩu phải hợp lệ (đã tồn tại trong cơ sở dữ liệu).
Luồng sự kiện	1. Người dùng mở ứng dụng web. 2. Hệ thống hiển thị giao diện đăng nhập. 3. Người dùng nhập tên đăng nhập và mật khẩu. 4. Hệ thống sẽ kiểm trả thông tin tài khoản 5. Hệ thống ghi nhận thông tin đăng nhập của người dùng 6. Kết thúc Use-case
Ngoại lệ	4a. Hệ thống kiểm trả tài khoản người dùng có tồn tại trong hệ thống hay không 4a1. Hệ thống hiển thị thông báo “Người dùng không tồn tại” 4a2. Kết thúc Use-case 4b. Hệ thống kiểm trả tài khoản người dùng có mật khẩu không chính xác 4b1. Hệ thống hiển thị thông báo “Sai mật khẩu” 4b2. Kết thúc Use-case

4.2.3.3. Xem danh sách tác vụ

Bảng 4.4 Đặc tả use case xem danh sách tác vụ

Usecase: Xem danh sách tác vụ	
Mô tả	Người dùng có thể xem danh sách các tác vụ được phân loại theo ngày, trạng thái (hôm nay, quá hạn, sắp tới) hoặc theo tuần.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập
Luồng sự kiện	1. Người dùng chọn tính năng "Xem danh sách tác vụ" trên giao diện chính. 2. Hệ thống hiển thị danh sách các tác vụ được phân loại theo ngày, tuần hoặc trạng thái. 3. Người dùng có thể tương tác với danh sách tác vụ để xem chi tiết hoặc chỉnh sửa tác vụ. 4. Kết thúc Use-case.
Ngoại lệ	2a. Không có tác vụ nào được tạo. 2a1. Hệ thống hiển thị thông báo "Hiện không có tác vụ nào để hiển thị". 2a2. Kết thúc Use-case.

4.2.3.4. Tìm kiếm tác vụ

Bảng 4.5 Đặc tả use case tìm kiếm tác vụ

Usecase: Tìm kiếm tác vụ	
Mô tả	Người dùng có thể lọc ra danh sách các sản phẩm theo một khoản thời gian cụ thể, theo danh mục, màu tag mà tác vụ đó thuộc về.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài

	khoản được đăng nhập và có ít nhất một tác vụ được tạo trong hệ thống.
Luồng sự kiện	1. Người dùng chọn tính năng "Tìm kiếm tác vụ" trên giao diện chính. 2. Hệ thống hiển thị thanh tìm kiếm và các bộ lọc bổ sung (tiêu đề, màu tag, danh mục, khoảng thời gian). 3. Người dùng nhập từ khóa hoặc chọn tiêu chí tìm kiếm. 4. Hệ thống thực hiện tìm kiếm và hiển thị danh sách các tác vụ phù hợp. 5. Người dùng có thể chọn một tác vụ từ kết quả tìm kiếm để xem chi tiết hoặc chỉnh sửa. 6. Kết thúc Use-case.
Ngoại lệ	2a. Người dùng không nhập từ khóa hoặc chọn tiêu chí tìm kiếm. 2a1. Hệ thống hiển thị thông báo "Vui lòng nhập từ khóa hoặc chọn tiêu chí tìm kiếm." 2a2. Kết thúc Use-case. 4a. Không có tác vụ nào khớp với từ khóa hoặc tiêu chí tìm kiếm. 4a1. Hệ thống hiển thị thông báo "Không tìm thấy tác vụ nào phù hợp." 4a2. Kết thúc Use-case.

4.2.3.5. Xem chi tiết tác vụ

Bảng 4.6 Đặc tả use case xem chi tiết tác vụ

Usecase: Xem chi tiết tác vụ	
Mô tả	Người dùng có thể xem thông tin chi tiết của một tác vụ, bao

	gồm tiêu đề, màu tag, danh mục liên quan, thời hạn, ghi chú, trạng thái, danh sách tác vụ phụ, và tệp đính kèm.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và có ít nhất một tác vụ được tạo trong hệ thống.
Luồng sự kiện	1. Người dùng chọn một tác vụ từ danh sách tác vụ trên giao diện chính. 2. Hệ thống hiển thị đầy đủ thông tin chi tiết của tác vụ bao gồm: • Tiêu đề tác vụ. • Màu tag của tác vụ. • Danh mục liên quan (như sức khỏe, tài chính, v.v.). • Thời hạn hoàn thành của tác vụ. • Ghi chú đính kèm. • Danh sách các tác vụ phụ (nếu có). • Tệp đính kèm (nếu có). 3. Người dùng có thể chọn chỉnh sửa hoặc xóa tác vụ từ giao diện chi tiết. 4. Kết thúc Use-case.
	1a. Người dùng chọn một tác vụ đã bị xóa hoặc không tồn tại trong hệ thống. 1a1. Hệ thống hiển thị thông báo "Tác vụ không tồn tại hoặc đã bị xóa." 1a2. Kết thúc Use-case. 2a. Dữ liệu của tác vụ bị lỗi hoặc không đầy đủ. 2a1. Hệ thống hiển thị thông báo "Không thể tải thông tin chi tiết của tác vụ, vui lòng thử lại sau."

	2a2. Kết thúc Use-case.

4.2.3.6. Điều chỉnh tác vụ

Bảng 4.7 Đặc tả use case xem danh sách tác vụ

Usecase: Điều chỉnh tác vụ	
Mô tả	Người dùng có thể chỉnh sửa thông tin chi tiết của một tác vụ, bao gồm tiêu đề, màu tag, danh mục liên quan, thời hạn, ghi chú, trạng thái, danh sách tác vụ phụ, và tệp đính kèm để cập nhật theo nhu cầu.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và có ít nhất một tác vụ được tạo trong hệ thống.
Luồng sự kiện	1. Người dùng chọn một tác vụ từ danh sách tác vụ trên giao diện chính. 2. Hệ thống hiển thị giao diện chi tiết của tác vụ. 3. Người dùng chọn tính năng "Chỉnh sửa tác vụ". 4. Hệ thống hiển thị các trường thông tin có thể chỉnh sửa, bao gồm: • Tiêu đề tác vụ. • Màu tag của tác vụ. • Danh mục liên quan (như sức khỏe, tài chính, v.v.). • Thời hạn hoàn thành của tác vụ. • Ghi chú. • Danh sách các tác vụ phụ (thêm, xóa, hoặc chỉnh sửa từng tác vụ phụ). • Tệp đính kèm (thêm hoặc xóa). 5. Người dùng thực hiện các thay đổi cần thiết và lưu tác

	vụ. 6. Hệ thống cập nhật thông tin tác vụ trong cơ sở dữ liệu và hiển thị thông báo "Cập nhật thành công". 7. Kết thúc Use-case.
Ngoại lệ	3a. Người dùng hủy bỏ thao tác chỉnh sửa trước khi lưu thay đổi. 3a1. Hệ thống giữ nguyên thông tin ban đầu của tác vụ và quay lại giao diện chi tiết. 3a2. Kết thúc Use-case. 6a. Dữ liệu chỉnh sửa không hợp lệ hoặc không đủ (ví dụ: tiêu đề để trống). 6a1. Hệ thống hiển thị thông báo lỗi, yêu cầu người dùng nhập lại thông tin chính xác. 6a2. Người dùng sửa lại và lưu lại thông tin. 6b. Hệ thống gặp sự cố khi cập nhật dữ liệu. 6b1. Hệ thống hiển thị thông báo "Cập nhật thất bại, vui lòng thử lại". 6b2. Kết thúc Use-case.

4.2.3.7. Xem danh sách tác vụ phụ

Bảng 4.8 Đặc tả use case xem danh sách tác vụ

Usecase: Xem danh sách tác vụ phụ	
Mô tả	Người dùng có thể xem danh sách các tác vụ phụ liên quan đến một tác vụ chính, bao gồm thông tin về tên tác vụ phụ, trạng thái hoàn thành, và các tùy chọn chỉnh sửa hoặc xóa.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài

	khoản được đăng nhập và tác vụ chính có ít nhất một tác vụ phụ được liên kết.
Luồng sự kiện	1. Người dùng chọn một tác vụ chính từ danh sách tác vụ trên giao diện chính. 2. Hệ thống hiển thị giao diện chi tiết của tác vụ chính. 3. Người dùng chọn mục "Danh sách tác vụ phụ". 4. Hệ thống hiển thị danh sách các tác vụ phụ, bao gồm: • Tên tác vụ phụ. • Trạng thái hoàn thành (đã hoàn thành hoặc chưa hoàn thành). • Tùy chọn chỉnh sửa và xóa cho từng tác vụ phụ. 5. Người dùng có thể tương tác với danh sách tác vụ phụ để chỉnh sửa, xóa hoặc đánh dấu trạng thái hoàn thành. 6. Kết thúc Use-case.
Ngoại lệ	4a. Không có tác vụ phụ nào liên kết với tác vụ chính. 4a1. Hệ thống hiển thị thông báo "Hiện không có tác vụ phụ nào liên kết với tác vụ này". 4a2. Kết thúc Use-case. 4b. Dữ liệu của danh sách tác vụ phụ bị lỗi hoặc không thể tải. 4b1. Hệ thống hiển thị thông báo "Không thể tải danh sách tác vụ phụ, vui lòng thử lại sau". 4b2. Kết thúc Use-case.

4.2.3.8. Điều chỉnh tác vụ phụ

Bảng 4.9 Đặc tả use case xem danh sách tác vụ

Usecase: Điều chỉnh tác vụ phụ	
Mô tả	Người dùng có thể chỉnh sửa thông tin của một tác vụ phụ, bao gồm tên, trạng thái hoàn thành, và tùy chọn xóa hoặc thêm ghi chú, để quản lý các bước liên quan đến một tác vụ chính một cách chi tiết.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và tác vụ phụ tồn tại, được liên kết với một tác vụ chính.
Luồng sự kiện	1. Người dùng chọn một tác vụ chính từ danh sách tác vụ. 2. Hệ thống hiển thị giao diện chi tiết của tác vụ chính. 3. Người dùng chọn một tác vụ phụ từ danh sách tác vụ phụ. 4. Hệ thống hiển thị giao diện chỉnh sửa thông tin của tác vụ phụ, bao gồm: • Tên tác vụ phụ. • Trạng thái hoàn thành (đã hoàn thành hoặc chưa hoàn thành). • Tùy chọn thêm hoặc chỉnh sửa ghi chú. 5. Người dùng thực hiện thay đổi thông tin và nhấn "Lưu". 6. Hệ thống cập nhật thông tin của tác vụ phụ trong cơ sở dữ liệu và hiển thị thông báo "Cập nhật thành công". 7. Kết thúc Use-case.
Ngoại lệ	4a. Không có tác vụ phụ nào liên kết với tác vụ chính. 4a1. Hệ thống hiển thị thông báo "Hiện không có tác vụ phụ nào liên kết với tác vụ này". 4a2. Kết thúc Use-case. 4b. Dữ liệu của danh sách tác vụ phụ bị lỗi hoặc không thể

	tải. 4b1. Hệ thống hiển thị thông báo "Không thể tải danh sách tác vụ phụ, vui lòng thử lại sau". 4b2. Kết thúc Use-case.

4.2.3.9. Xem danh sách thói quen

Bảng 4.10 Đặc tả use case xem danh sách tác vụ

Usecase: Xem danh sách thói quen	
Mô tả	Người dùng có thể xem danh sách các thói quen của mình, được phân loại theo trạng thái hoạt động (hoạt động hoặc không hoạt động) và thời gian thực hiện, giúp quản lý và theo dõi việc thực hiện thói quen dễ dàng hơn.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và có ít nhất một thói quen được tạo trong hệ thống.
Luồng sự kiện	1. Người dùng chọn tính năng "Xem danh sách thói quen" trên giao diện chính. 2. Hệ thống hiển thị danh sách các thói quen, bao gồm: • Tên thói quen. • Trạng thái (hoạt động hoặc không hoạt động). • Thời gian thực hiện thói quen (hằng ngày, hằng tuần, hoặc thời gian cụ thể). • Các nhãn (tags) liên quan đến thói quen. 3. Người dùng có thể chọn một thói quen để xem thêm chi tiết hoặc chỉnh sửa. 4. Kết thúc Use-case.
Ngoại lệ	2a. Không có thói quen nào được tạo.

	2a1. Hệ thống hiển thị thông báo "Hiện không có thói quen nào để hiển thị". 2a2. Kết thúc Use-case. 2b. Dữ liệu thói quen bị lỗi hoặc không thể tải. 2b1. Hệ thống hiển thị thông báo "Không thể tải danh sách thói quen, vui lòng thử lại sau". 2b2. Kết thúc Use-case.

4.2.3.10. Xem chi tiết thói quen

Bảng 4.11 Đặc tả use case xem chi tiết thói quen

Usecase: Xem chi tiết thói quen	
Mô tả	Người dùng có thể xem thông tin chi tiết của một thói quen, bao gồm tên, màu tag, danh mục liên quan, trạng thái hoạt động, thời gian thực hiện (routine time), lịch sử ngày đã hoàn thành, ghi chú, và các thông tin bổ sung khác để theo dõi và quản lý thói quen hiệu quả.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và thói quen cần xem tồn tại trong hệ thống.
Luồng sự kiện	1. Người dùng chọn một thói quen từ danh sách thói quen trên giao diện chính. 2. Hệ thống hiển thị giao diện chi tiết của thói quen, bao gồm: • Tên thói quen. • Màu tag của thói quen. • Danh mục liên quan (như sức khỏe, tài chính, vui chơi, v.v.).

	• Trạng thái hoạt động (hoạt động hoặc không hoạt động). • Thời gian thực hiện thói quen (routine time). • Lịch sử ngày hoàn thành (các ngày đã đánh dấu là hoàn thành). • Ghi chú hoặc thông tin bổ sung. 3. Người dùng có thể thực hiện các thao tác như chỉnh sửa hoặc đánh dấu hoàn thành cho thói quen từ giao diện chi tiết. 4. Kết thúc Use-case.
Ngoại lệ	1a. Người dùng chọn một thói quen đã bị xóa hoặc không tồn tại trong hệ thống. 1a1. Hệ thống hiển thị thông báo "Thói quen không tồn tại hoặc đã bị xóa". 1a2. Kết thúc Use-case. 2a. Dữ liệu của thói quen bị lỗi hoặc không đầy đủ. 2a1. Hệ thống hiển thị thông báo "Không thể tải thông tin chi tiết của thói quen, vui lòng thử lại sau". 2a2. Kết thúc Use-case.

4.2.3.11. Điều chỉnh thói quen

Bảng 4.12 Đặc tả use case điều chỉnh thói quen

Usecase: Điều chỉnh thói quen	
Mô tả	Người dùng có thể chỉnh sửa thông tin chi tiết của một thói quen, bao gồm tên, màu tag, danh mục liên quan, trạng thái hoạt động, thời gian thực hiện (routine time), ghi chú, và lịch sử ngày hoàn thành, để cập nhật thói quen phù hợp với mục tiêu hiện tại.

Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhập và thói quen cần điều chỉnh tồn tại trong hệ thống.
Luồng sự kiện	1. Người dùng chọn một thói quen từ danh sách thói quen trên giao diện chính. 2. Hệ thống hiển thị giao diện chi tiết của thói quen. 3. Người dùng chọn tính năng "Chỉnh sửa thói quen". 4. Hệ thống hiển thị các trường thông tin có thể chỉnh sửa, bao gồm: • Tên thói quen. • Màu tag của thói quen. • Danh mục liên quan. • Trạng thái hoạt động (bật hoặc tắt). • Thời gian thực hiện (routine time). • Ghi chú hoặc thông tin bổ sung. • Lịch sử ngày hoàn thành (người dùng có thể thêm hoặc xóa ngày đã hoàn thành). 5. Người dùng thực hiện các thay đổi và nhấn "Lưu". 6. Hệ thống cập nhật thông tin thói quen trong cơ sở dữ liệu và hiển thị thông báo "Cập nhật thành công". 7. Kết thúc Use-case.
Ngoại lệ	4a. Người dùng không thực hiện bất kỳ thay đổi nào và hủy thao tác chỉnh sửa. 4a1. Hệ thống giữ nguyên thông tin ban đầu của thói quen và quay lại giao diện chi tiết. 4a2. Kết thúc Use-case. 5a. Thông tin chỉnh sửa không hợp lệ (ví dụ: tên để trống).

	5a1. Hệ thống hiển thị thông báo lỗi "Thông tin không hợp lệ, vui lòng nhập lại". 5a2. Người dùng sửa lại thông tin và nhấn "Lưu" lần nữa. 6a. Hệ thống gặp sự cố khi lưu thông tin. 6a1. Hệ thống hiển thị thông báo "Cập nhật thất bại, vui lòng thử lại". 6a2. Kết thúc Use-case.

4.2.3.12. Xem danh sách tin nhắn

Bảng 4.13 Đặc tả use case xem danh sách tin nhắn

Usecase: Xem danh sách tin nhắn	
Mô tả	Người dùng có thể xem danh sách các phiên trò chuyện đã thực hiện với chatbot, bao gồm thời gian bắt đầu của mỗi phiên, và có các tùy chọn quản lý như đổi tên hoặc xóa phiên trò chuyện.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhậpvà có ít nhất một phiên trò chuyện được lưu lại trong lịch sử
Luồng sự kiện	1. Người dùng vào trang tương tác với trợ lý ảo. 2. Hệ thống hiển thị danh sách các phiên trò chuyện, bao gồm: • Thời gian bắt đầu của từng phiên. • Tùy chọn quản lý (Rename, Delete) hiển thị dưới dạng menu khi nhấn nút ba chấm. 3. Người dùng chọn một mục từ danh sách: • Nếu nhấn vào tên phiên trò chuyện, nội dung chi tiết của phiên được hiển thị.

	• Nếu nhấn vào nút ba chấm, các tùy chọn Rename hoặc Delete xuất hiện. 4. Người dùng thực hiện một trong các thao tác: • Đổi tên: Hệ thống mở hộp thoại để người dùng nhập tên mới, sau đó lưu tên mới và cập nhật danh sách. • Xóa: Hệ thống yêu cầu xác nhận xóa, nếu xác nhận, phiên trò chuyện sẽ bị xóa khỏi danh sách. 5. Hệ thống cập nhật danh sách lịch sử trò chuyện và hiển thị thông báo thành công. 6. Kết thúc Use-case.
Ngoại lệ	4a. Người dùng không thực hiện bất kỳ thay đổi nào và hủy thao tác chỉnh sửa. 4a1. Hệ thống giữ nguyên thông tin ban đầu của thói quen và quay lại giao diện chi tiết. 4a2. Kết thúc Use-case. 5a. Thông tin chỉnh sửa không hợp lệ (ví dụ: tên để trống). 5a1. Hệ thống hiển thị thông báo lỗi "Thông tin không hợp lệ, vui lòng nhập lại". 5a2. Người dùng sửa lại thông tin và nhấn "Lưu" lần nữa. 6a. Hệ thống gặp sự cố khi lưu thông tin. 6a1. Hệ thống hiển thị thông báo "Cập nhật thất bại, vui lòng thử lại". 6a2. Kết thúc Use-case.

4.2.3.13. Nhắn tin với chatbot

Bảng 4.14 Đặc tả use case nhắn tin với chatbot

Usecase: Nhắn tin với chatbot	
Mô tả	Người dùng có thể trò chuyện với chatbot bằng cách gửi tin

	nhắn văn bản, nhận phản hồi tức thì, và sử dụng các tính năng bổ trợ như đính kèm tệp hoặc truy cập lịch sử trò chuyện.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện	1. Người dùng vào trang tương tác với trợ lý ảo. 2. Hệ thống hiển thị giao diện trò chuyện bao gồm: 3. Thanh nhập tin nhắn ở phía dưới. 4. Nút đính kèm tệp tin. 5. Danh sách các tin nhắn trong cuộc trò chuyện. 6. Lịch sử trò chuyện trong thanh bên phải (nếu có). 7. Người dùng nhập nội dung tin nhắn vào thanh nhập liệu và nhấn "Gửi". 8. Hệ thống gửi tin nhắn đến chatbot và hiển thị phản hồi từ chatbot trong giao diện trò chuyện. 9. Người dùng có thể tiếp tục gửi tin nhắn hoặc thực hiện các thao tác bổ trợ như: • Đính kèm tệp tin vào tin nhắn. • Xem lịch sử trò chuyện để truy cập lại các tin nhắn trước đó. 10. Kết thúc Use-case.
Ngoại lệ	3a. Người dùng gửi tin nhắn trống hoặc không hợp lệ. 3a1. Hệ thống hiển thị thông báo "Vui lòng nhập nội dung tin nhắn". 3a2. Kết thúc Use-case. 4a. Hệ thống không thể gửi tin nhắn do lỗi kết nối hoặc sự cố. 4a1. Hệ thống hiển thị thông báo "Không thể gửi tin nhắn, vui lòng thử lại".

	4a2. Kết thúc Use-case. 5a. Người dùng đính kèm tệp không hợp lệ hoặc vượt quá dung lượng cho phép. 5a1. Hệ thống hiển thị thông báo "Tệp đính kèm không hợp lệ". 5a2. Người dùng chọn tệp khác hoặc hủy thao tác.

4.2.3.14. Đăng tải tệp tin

Bảng 4.15 Đặc tả use case đăng tải tệp tin

Usecase: Đăng tải tệp tin	
Mô tả	Người dùng có thể đính kèm và gửi các tệp tin tài liệu(pdf, docx) trong cuộc trò chuyện với chatbot.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện	1. Người dùng chọn tính năng "Nhắn tin với chatbot" trên giao diện chính. 2. Hệ thống hiển thị giao diện trò chuyện, bao gồm tùy chọn đính kèm tệp tin. 3. Người dùng nhấn vào biểu tượng "Đính kèm tệp tin". 4. Hệ thống hiển thị giao diện chọn tệp từ thiết bị của người dùng. 5. Người dùng chọn một tệp tin từ thiết bị và nhấn "Mở". 6. Hệ thống kiểm tra định dạng và dung lượng của tệp tin: • Nếu hợp lệ: Tệp tin được hiển thị trong khu vực tin nhắn. • Nếu không hợp lệ: Hệ thống hiển thị thông báo lỗi. 7. Người dùng nhấn "Gửi" để gửi tệp tin đến chatbot. 8. Hệ thống gửi tệp tin thành công và hiển thị phản hồi từ chatbot (nếu có).

	9. Kết thúc Use-case.
Ngoại lệ	6a. Tệp tin không hợp lệ (ví dụ: định dạng không được hỗ trợ hoặc dung lượng vượt giới hạn). 6a1. Hệ thống hiển thị thông báo "Tệp không hợp lệ, vui lòng chọn tệp khác". 6a2. Người dùng chọn lại tệp tin hoặc hủy thao tác. 7a. Lỗi khi gửi tệp tin (do mất kết nối hoặc lỗi hệ thống). 7a1. Hệ thống hiển thị thông báo "Gửi tệp tin thất bại, vui lòng thử lại". 7a2. Người dùng thử gửi lại tệp tin hoặc kết thúc thao tác.

4.2.3.15. Gọi video

Bảng 4.16 Đặc tả use case gọi video

Usecase: Gọi video	
Mô tả	Người dùng có thể thực hiện cuộc gọi video trực tiếp với chatbot để giao tiếp bằng hình ảnh từ video và âm thanh trong thời gian thực.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống, có thiết bị hỗ trợ video.
Luồng sự kiện	1. Người dùng chọn tính năng "Gọi video" từ giao diện chính hoặc từ trong cửa sổ trò chuyện. 2. Hệ thống khởi tạo cuộc gọi video và kích hoạt camera cũng như micro của người dùng. 3. Giao diện cuộc gọi hiển thịv ideo trực tiếp từ camera của người dùng và các tùy chọn điều khiển, bao gồm: • Tắt/Mở micro: Bật hoặc tắt âm thanh của người dùng. • Tắt/Mở camera: Bật hoặc tắt video của người dùng.

	• Chia sẻ màn hình: Người dùng có thể chia sẻ nội dung màn hình của mình. • Kết thúc cuộc gọi: Dừng cuộc gọi video. 4. Người dùng thực hiện cuộc gọi video. 5. Khi kết thúc cuộc gọi, người dùng nhấn vào nút "Kết thúc", hoặc hệ thống tự động ngắt kết nối khi gặp sự cố hoặc khi cuộc gọi hoàn tất. 6. Kết thúc Use-case.
Ngoại lệ	2a. Camera hoặc micro không hoạt động hoặc không được cấp quyền. 2a1. Hệ thống hiển thị thông báo "Không thể truy cập camera/micro, vui lòng kiểm tra thiết bị hoặc cài đặt quyền". 2a2. Người dùng kiểm tra và khởi động lại thao tác gọi video. 4a. Kết nối mạng bị gián đoạn hoặc mất tín hiệu trong khi gọi. 4a1. Hệ thống hiển thị thông báo "Mất kết nối, đang cố gắng kết nối lại". 4a2. Nếu không thể kết nối lại, hệ thống hiển thị thông báo "Cuộc gọi đã bị gián đoạn" và kết thúc cuộc gọi.

4.2.3.16. Chia sẻ màn hình

Bảng 4.17 Đặc tả use case chia sẻ màn hình

Usecase: Chia sẻ màn hình	
Mô tả	Người dùng có thể chia sẻ nội dung màn hình của mình trong khi thực hiện cuộc gọi video hoặc trò chuyện, nhằm hỗ trợ trình bày thông tin, hướng dẫn cho trợ lý ảo để giải quyết vấn

	đề một cách trực quan hơn.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống, đang thực hiện cuộc gọi hoặc phiên trò chuyện.
Luồng sự kiện	1. Người dùng chọn tính năng "Chia sẻ màn hình" từ giao diện cuộc gọi video hoặc trò chuyện. 2. Người dùng chọn nội dung muốn chia sẻ (toàn bộ màn hình, cửa sổ ứng dụng, hoặc tab trình duyệt cụ thể). 3. Hệ thống bắt đầu hiển thị nội dung chia sẻ cho các bên tham gia cuộc gọi hoặc phiên trò chuyện. 4. Trong quá trình chia sẻ, người dùng có các tùy chọn: • Tạm dừng chia sẻ màn hình. • Dừng chia sẻ hoàn toàn. 5. Sau khi chia sẻ xong, hệ thống quay lại giao diện cuộc gọi hoặc trò chuyện bình thường. 6. Kết thúc Use-case.
Ngoại lệ	3a. Người dùng không cấp quyền chia sẻ màn hình. 3a1. Hệ thống hiển thị thông báo "Không thể chia sẻ màn hình vì quyền truy cập bị từ chối". 3a2. Kết thúc Use-case.

4.2.3.17. Điều chỉnh chức năng cho chatbot

Bảng 4.18 Đặc tả use case điều chỉnh chức năng cho chatbot

Usecase: Điều chỉnh chức năng cho chatbot	
Mô tả	Người dùng có thể tùy chỉnh các chức năng của chatbot bằng cách bật hoặc tắt từng công cụ (tools) cụ thể. Điều này cho phép cá nhân hóa các tính năng của chatbot theo nhu cầu sử dụng của người dùng, như quản lý thói quen, nhắc nhở công

	việc, duyệt web, hay truy vấn tài liệu.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện	1. Người dùng chọn tính năng "AI Setting" hoặc tương tự từ giao diện chính. 2. Hệ thống hiển thị danh sách các công cụ (tools) khả dụng, cùng với mô tả chức năng và trạng thái hiện tại (bật/tắt). 3. Người dùng thực hiện một hoặc nhiều thao tác: • Bật một công cụ bằng cách gạt công tắc sang trạng thái "ON". • Tắt một công cụ bằng cách gạt công tắc sang trạng thái "OFF". 4. Hệ thống lưu lại trạng thái mới của các công cụ và cập nhật cấu hình chatbot. 5. Người dùng có thể tiếp tục điều chỉnh hoặc thoát khỏi giao diện cài đặt. 6. Kết thúc Use-case.
Ngoại lệ	4a. Hệ thống không thể lưu trạng thái mới do lỗi kết nối hoặc hệ thống: Hệ thống hiển thị thông báo "Không thể lưu cài đặt, vui lòng thử lại". Người dùng có thể thử thực hiện lại thao tác hoặc thoát khỏi giao diện cài đặt.

4.2.3.18. Liên kết với tài khoản Google

Bảng 4.19 Đặc tả use case liên kết với tài khoản Google

Usecase: Liên kết với tài khoản Google

Mô tả	Người dùng có thể liên kết tài khoản Google của mình với hệ thống chatbot để đồng bộ hóa dữ liệu như lịch, tác vụ, thói quen, và sử dụng các công cụ hỗ trợ liên quan. Việc liên kết này giúp chatbot truy cập thông tin từ tài khoản Google để nâng cao trải nghiệm cá nhân hóa.
Tác nhân	Người dùng
Điều kiện	Hệ thống tương tác của người dùng đang ở trang thái có tài khoản được đăng nhậpvà có tài khoản Google sẵn sàng để liên kết.
Luồng sự kiện	1. Người dùng chọn tính năng "AI Setting" hoặc tương tự từ giao diện chính. 2. Người dùng chọn tùy chọn "Liên kết tài khoản Google" từ giao diện cài đặt. 3. Hệ thống hiển thị giao diện yêu cầu quyền truy cập tài khoản Google. 4. Người dùng đăng nhập vào tài khoản Google hoặc chọn một tài khoản Google đã được đăng nhập trước đó. 5. Google hiển thị danh sách các quyền mà chatbot yêu cầu (ví dụ: truy cập lịch, tác vụ, email). 6. Người dùng đồng ý cấp quyền và hoàn tất liên kết. 7. Hệ thống xác nhận liên kết thành công, hiển thị tài khoản Google đã được liên kết trên giao diện. 8. Kết thúc Use-case.
Ngoại lệ	4a. Người dùng hủy liên kết trước khi hoàn tất đăng nhập: Hệ thống hiển thị thông báo "Liên kết đã bị hủy". Người dùng quay lại giao diện cài đặt mà không thay đổi gì. 5a. Người dùng từ chối cấp quyền truy cập:

	Hệ thống hiển thị thông báo "Liên kết thất bại, bạn cần cấp quyền để hoàn tất liên kết". 6a. Lỗi kết nối hoặc tài khoản Google không khả dụng tại thời điểm liên kết: Hệ thống hiển thị thông báo "Không thể liên kết với tài khoản Google, vui lòng thử lại sau". 6b. Tài khoản Google đã được liên kết với một tài khoản khác trên hệ thống: Hệ thống hiển thị thông báo "Tài khoản Google này đã được liên kết với một tài khoản khác. Vui lòng sử dụng tài khoản khác". 8a. Đồng bộ hóa dữ liệu thất bại sau khi liên kết thành công: Hệ thống hiển thị thông báo "Liên kết thành công nhưng không thể đồng bộ dữ liệu, vui lòng kiểm tra kết nối mạng".

4.2.3.19. Điều chỉnh hình nền

Bảng 4.20 Đặc tả use case điều chỉnh hình nền

Usecase: Điều chỉnh hình nền	
Mô tả	Người dùng có thể tùy chỉnh hình nền của giao diện hệ thống bằng cách chọn hình ảnh từ máy tính, xóa hình nền hiện tại, hoặc chọn từ các hình nền có sẵn. Điều này giúp cá nhân hóa trải nghiệm sử dụng hệ thống.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện	1. Người dùng chọn mục "Hiển thị" từ giao diện cài đặt. 2. Hệ thống hiển thị giao diện tùy chỉnh hình nền, bao gồm:

	• Hình nền hiện tại (nếu có). • Tùy chọn "Không cần hình nền" để xóa hình nền hiện tại. • Tùy chọn "Chọn ảnh từ máy" để tải lên hình ảnh mới từ thiết bị của người dùng. • Các hình nền gợi ý hoặc hình nền đã được lưu trữ trước đó. 3. Người dùng thực hiện một trong các thao tác: • Xóa hình nền: Nhấn vào "Không cần hình nền". • Tải hình nền mới: Nhấn vào "Chọn ảnh từ máy", chọn hình ảnh từ thiết bị, và nhấn "Mở". • Chọn hình nền có sẵn: Nhấp vào một trong các hình nền gợi ý. 4. Hệ thống áp dụng thay đổi, hiển thị hình nền mới trên giao diện, và lưu cấu hình. 5. Kết thúc Use-case.
Ngoại lệ	3a. Người dùng không chọn hình nền hoặc hủy thao tác: Hệ thống giữ nguyên hình nền hiện tại và quay lại giao diện tùy chỉnh. 3b. Tệp hình ảnh tải lên không hợp lệ (định dạng sai hoặc kích thước vượt quá giới hạn): Hệ thống hiển thị thông báo "Tệp không hợp lệ, vui lòng chọn ảnh khác". Người dùng được yêu cầu tải lại hình ảnh đúng định dạng. 4a. Hệ thống gặp lỗi khi áp dụng hình nền mới: Hệ thống hiển thị thông báo "Không thể áp dụng hình nền, vui lòng thử lại".

	4b. Hình nền được tải lên nhưng không tương thích với kích thước hoặc hiển thị sai: Hệ thống hiển thị thông báo "Hình nền không tương thích, vui lòng chọn ảnh khác". 5a. Người dùng cố chọn lại hình nền đã bị xóa: Hệ thống hiển thị thông báo "Hình nền này không còn khả dụng".

4.2.3.20.Đăng tải hình nền

Bảng 4.21 Đặc tả use case đăng tải hình nền

Usecase: Đăng tải hình nền	
Mô tả	Người dùng có thể tải lên hình nền từ thiết bị cá nhân để sử dụng làm giao diện nền trong hệ thống. Tính năng này cho phép người dùng tùy chỉnh trải nghiệm trực quan của mình một cách linh hoạt và cá nhân hóa.
Tác nhân	Người dùng
Điều kiện	Người dùng đã đăng nhập vào hệ thống
Luồng sự kiện	1. Người dùng chọn mục "Hiển thị" từ giao diện cài đặt. 2. Hệ thống hiển thị giao diện quản lý hình nền với các tùy chọn. 3. Người dùng nhấn vào nút "Chọn ảnh từ máy". 4. Hệ thống mở trình duyệt tệp (file explorer) trên thiết bị của người dùng. 5. Người dùng chọn một tệp hình ảnh từ thiết bị và nhấn "Mở". 6. Hệ thống thực hiện kiểm tra tệp hình ảnh: • Định dạng (chỉ hỗ trợ JPG, PNG, v.v.).

	• Kích thước (không vượt quá dung lượng tối đa, ví dụ: 5MB). • Độ phân giải (đảm bảo phù hợp với giao diện). 7. Nếu tệp hợp lệ, hệ thống tải tệp lên máy chủ và lưu trữ tệp. 8. Hệ thống hiển thị thông báo "Tải lên thành công" và áp dụng hình nền mới trên giao diện. 9. Kết thúc Use-case.
Ngoại lệ	Tệp hình ảnh không hợp lệ (sai định dạng): Hệ thống hiển thị thông báo "Định dạng tệp không được hỗ trợ, vui lòng chọn ảnh JPG hoặc PNG". Người dùng quay lại bước 4 để chọn lại tệp. 6b. Tệp hình ảnh vượt quá dung lượng tối đa: Hệ thống hiển thị thông báo "Dung lượng tệp vượt quá giới hạn cho phép (5MB), vui lòng chọn ảnh khác". Người dùng quay lại bước 4 để chọn lại tệp. 7a. Lỗi khi tải lên máy chủ do kết nối mạng: Hệ thống hiển thị thông báo "Không thể tải lên hình nền, vui lòng kiểm tra kết nối và thử lại". Người dùng quay lại bước 5 để thử lại thao tác tải lên. 7b. Lỗi máy chủ khi lưu trữ hình ảnh: Hệ thống hiển thị thông báo "Có lỗi trong quá trình lưu trữ tệp, vui lòng thử lại sau". Người dùng quay lại bước 5 để thử lại thao tác tải lên. 8a. Người dùng hủy thao tác tải lên trước khi hoàn tất:

	Hệ thống hủy quá trình tải lên và giữ nguyên hình nền hiện tại. Người dùng quay lại giao diện quản lý hình nền mà không có thay đổi.

4.2.3.21.Đăng xuất

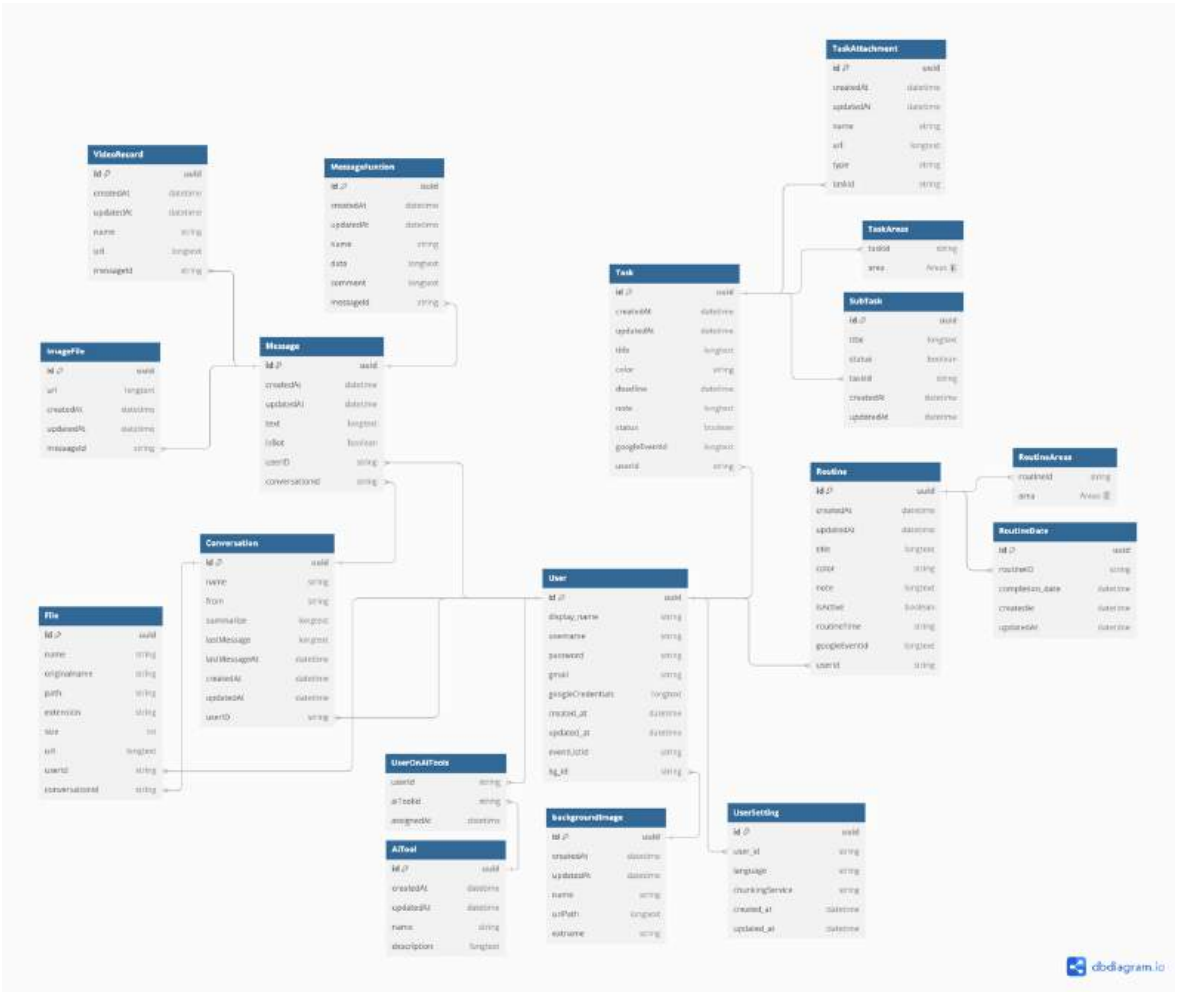
Bảng 4.22 Đặc tả use case đăng xuất

Usecase: Đăng xuất	
Mô tả	Người dùng có thể đăng xuất khỏi hệ thống để kết thúc phiên làm việc hiện tại, đảm bảo an toàn cho thông tin cá nhân và dữ liệu tài khoản.
Tác nhân	Người dùng
Điều kiện	Người dùng đang ở trạng thái đã đăng nhập vào hệ thốn
Luồng sự kiện	1. Người dùng nhấn vào tùy chọn "Đăng xuất" từ giao diện chính hoặc cài đặt. 2. Hệ thống hiển thị thông báo xác nhận "Bạn có chắc chắn muốn đăng xuất không?". 3. Người dùng chọn "Đồng ý" để xác nhận. 4. Hệ thống thực hiện các thao tác sau: • Xóa thông tin phiên làm việc hiện tại. • Hủy quyền truy cập của người dùng vào các tính năng hệ thống. • Xóa các cookie và thông tin lưu trữ liên quan (nếu có). 5. Hệ thống chuyển người dùng đến giao diện đăng nhập. 6. Kết thúc Use-case.
Ngoại lệ	2a. Người dùng chọn "Hủy" ở bước xác nhận: Hệ thống quay lại giao diện hiện tại mà không thực hiện

	đăng xuất. 4a. Lỗi kết nối hoặc lỗi hệ thống khi xử lý yêu cầu đăng xuất: Hệ thống hiển thị thông báo "Không thể đăng xuất, vui lòng thử lại". Người dùng vẫn ở trạng thái đăng nhập. 4b. Hệ thống không xóa được thông tin phiên làm việc (do lỗi cookie hoặc bộ nhớ): Hệ thống hiển thị thông báo "Đã xảy ra lỗi khi đăng xuất, vui lòng xóa bộ nhớ trình duyệt và thử lại".

4.3. Thiết kế cơ sở dữ liệu

4.3.1. sơ đồ cơ sở dữ liệu



Hình 4.13 Thiết kế cơ sở dữ liệu

4.3.2. Mô tả chi tiết các bảng

4.3.2.1. Bảng User

Bảng 4.23 Bảng mô tả dữ liệu của User

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi người dùng.

2	display_name	VARCHAR(100)	Tên hiển thị của người dùng.
3	username	VARCHAR(100)	Tên đăng nhập của người dùng.
4	password	VARCHAR(255)	Mật khẩu của người dùng, được mã hóa.
5	gmail	VARCHAR(100)	Địa chỉ email Gmail của người dùng, phải là duy nhất.
6	created_at	DATETIME	Ngày và giờ tạo tài khoản.
7	updated_at	DATETIME	Ngày và giờ cập nhật thông tin tài khoản.
8	bg_id	VARCHAR(255)	Tham chiếu đến bảng **backgroundImage** để liên kết hình nền cá nhân của người dùng.

4.3.2.2. Bảng Task

Bảng 4.24 Bảng mô tả dữ liệu của Task

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi nhiệm vụ.
2	title	VARCHAR(255)	Tiêu đề của nhiệm vụ.
3	color	VARCHAR(50)	Màu sắc liên quan đến nhiệm vụ (Ví dụ: xanh, đỏ,...).

4	deadline	DATETIME	Thời hạn hoàn thành nhiệm vụ.
5	note	LONGTEXT	Ghi chú bổ sung về nhiệm vụ.
6	status	BOOLEAN	Trạng thái hoàn thành của nhiệm vụ (Ví dụ: true = hoàn thành, false = chưa hoàn thành).
7	googleEventId	VARCHAR(255)	Mã sự kiện liên kết với Google Calendar (nếu có).
8	userId	VARCHAR(255)	Tham chiếu đến bảng User để xác định người sở hữu nhiệm vụ.

4.3.2.3. Bảng SubTask

Bảng 4.25 Bảng mô tả dữ liệu của SubTask

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi công việc con.
2	title	VARCHAR(255)	Tiêu đề của công việc con.
3	status	BOOLEAN	Trạng thái của công việc con (Ví dụ: true = hoàn thành, false = chưa hoàn thành).
4	taskId	VARCHAR(255)	Tham chiếu đến bảng Task để xác định công việc cha.
5	createdAt	DATETIME	Ngày và giờ tạo công việc con.

6	updatedAt	DATETIME	Ngày và giờ cập nhật công việc con.

4.3.2.4. Bảng TaskAreas

Bảng 4.26 Bảng mô tả dữ liệu của TaskAreas

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	taskId	VARCHAR(255)	Tham chiếu đến bảng **Task** để xác định nhiệm vụ liên quan đến khu vực này.
2	area	VARCHAR(255)	Khu vực hoặc lĩnh vực của nhiệm vụ (Ví dụ: công việc, cá nhân, học tập,...).

4.3.2.5. Bảng Routine

Bảng 4.27 Bảng mô tả dữ liệu của Routine

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi thói quen.
2	title	VARCHAR(255)	Tiêu đề của thói quen.
3	color	VARCHAR(50)	Màu sắc liên quan đến thói quen.
4	note	LONGTEXT	Ghi chú bổ sung về thói quen.
5	isActive	BOOLEAN	Trạng thái hoạt động của thói quen (Ví dụ: true = đang hoạt động, false =

			không hoạt động).
6	routineTime	DATETIME	Thời gian lặp lại của thói quen.
7	googleEventId	VARCHAR(255)	Mã sự kiện liên kết với Google Calendar (nếu có).
8	userId	VARCHAR(255)	Tham chiếu đến bảng User để xác định người sở hữu thói quen.

4.3.2.6. Bảng TaskAttachment

Bảng 4.28 Bảng mô tả dữ liệu của TaskAttachment

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi tệp đính kèm.
2	name	VARCHAR(255)	Tên của tệp đính kèm.
3	url	LONGTEXT	URL của tệp đính kèm.
4	type	VARCHAR(50)	Loại tệp đính kèm (Ví dụ: hình ảnh, tài liệu,...).
5	taskId	VARCHAR(255)	Tham chiếu đến bảng Task để xác định nhiệm vụ liên kết với tệp đính kèm.

4.3.2.7. Bảng Message

Bảng 4.29 Bảng mô tả dữ liệu của Message

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi tin nhắn.
2	text	LONGTEXT	Nội dung tin nhắn.
3	isBot	BOOLEAN	Tin nhắn có được gửi bởi bot không (true = bot, false = người dùng).
4	userId	VARCHAR(255)	Tham chiếu đến bảng User để xác định người gửi tin nhắn.
5	conversationId	VARCHAR(255)	Tham chiếu đến bảng Conversation để xác định cuộc trò chuyện chứa tin nhắn.

4.3.2.8. Bảng Conversation

Bảng 4.30 Bảng mô tả dữ liệu của Conversation

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi cuộc trò chuyện.
2	name	LONGTEXT	Tên của cuộc trò chuyện, có thể do người dùng đặt.
3	from	BOOLEAN	Nguồn bắt đầu cuộc trò chuyện.

4	summary	LONGTEXT	Tóm tắt nội dung chính của cuộc trò chuyện.
5	lastMessage	LONGTEXT	Nội dung tin nhắn cuối cùng trong cuộc trò chuyện.
6	lastMessageAt	DATETIME	Thời gian tin nhắn cuối cùng được gửi trong cuộc trò chuyện.
7	createdAt	DATETIME	Thời gian cuộc trò chuyện được tạo.
8	updatedAt	DATETIME	Thời gian cuộc trò chuyện được cập nhật lần cuối.
9	userID	VARCHAR(255)	Tham chiếu đến bảng User để xác định người sở hữu hoặc tham gia cuộc trò chuyện.

4.3.2.9. Bảng VideoRecord

Bảng 4.31 Bảng mô tả dữ liệu của VideoRecord

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi video.
2	createdAt	DATETIME	Ngày và giờ video được tạo.
3	updatedAt	DATETIME	Ngày và giờ video được cập nhật.

4	name	VARCHAR(255)	Tên của video.
5	url	VARCHAR(3000)	Đường dẫn URL đến video.
6	messageId	VARCHAR(255)	Tham chiếu đến bảng **Message** để xác định video thuộc tin nhắn nào.

4.3.2.10.Bảng ImageFile

Bảng 4.32 Bảng mô tả dữ liệu của ImageFile

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi hình ảnh.
2	url	VARCHAR(3000)	Đường dẫn URL đến hình ảnh.
3	createdAt	DATETIME	Ngày và giờ hình ảnh được tải lên.
4	updatedAt	DATETIME	Ngày và giờ hình ảnh được cập nhật.
5	messageId	VARCHAR(255)	Tham chiếu đến bảng **Message** để xác định hình ảnh thuộc tin nhắn nào.

4.3.2.11.Bảng MessageFunction

Bảng 4.33 Bảng mô tả dữ liệu của MessageFunction

STT	Thuộc tính	Kiểu dữ liệu	Mô tả

1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi hình nền.
2	createdAt	DATETIME	Ngày và giờ hình nền được tạo.
3	updatedAt	DATETIME	Ngày và giờ hình nền được cập nhật.
4	urlPath	VARCHAR(3000)	Đường dẫn URL đến hình nền.
5	extname	VARCHAR(50)	Phần mở rộng của tệp hình nền (Ví dụ: .jpg, .png,...).

4.3.2.12.Bảng MessageFunction

Bảng 4.34 Bảng mô tả dữ liệu của MessageFunction

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi chức năng tin nhắn.
2	createdAt	DATETIME	Ngày và giờ chức năng được tạo.
3	updatedAt	DATETIME	Ngày và giờ chức năng được cập nhật.
4	name	VARCHAR(255)	Tên của chức năng tin nhắn.
5	data	LONGTEXT	Dữ liệu liên quan đến chức năng.
6	comment	LONGTEXT	Ghi chú hoặc mô tả bổ sung về chức năng.

7	messageId	VARCHAR(255)	Tham chiếu đến bảng **Message** để xác định chức năng thuộc tin nhắn nào.

4.3.2.13.Bảng RoutineAreas

Bảng 4.35 Bảng mô tả dữ liệu của RoutineAreas

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	routineId	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi khu vực thói quen.
2	area	VARCHAR(255)	Khu vực hoặc lĩnh vực của thói quen.

4.3.2.14.Bảng RoutineDate

Bảng 4.36 Bảng mô tả dữ liệu của RoutineDate

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi ngày thực hiện thói quen.
2	routineId	VARCHAR(255)	Tham chiếu đến bảng **Routine** để xác định thói quen liên quan.
3	completion_date	DATETIME	Ngày hoàn thành thói quen.
4	createdAt	DATETIME	Ngày và giờ bản ghi được tạo.

5	updatedAt	DATETIME	Ngày và giờ bản ghi được cập nhật.

4.3.2.15.Bảng AiTool

Bảng 4.37 Bảng mô tả dữ liệu của AiTool

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi công cụ.
2	name	VARCHAR(255)	Tên của công cụ.
3	description	LONGTEXT	Mô tả về công cụ.
4	createdAt	DATETIME	Ngày và giờ công cụ được tạo.
5	updatedAt	DATETIME	Ngày và giờ công cụ được cập nhật.

4.3.2.16.Bảng UserOnAiTools

Bảng 4.38 Bảng mô tả dữ liệu của UserOnAiTools

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	userId	VARCHAR(255)	Tham chiếu đến bảng **User** để xác định người dùng sử dụng công cụ.
2	aToolId	VARCHAR(255)	Tham chiếu đến bảng **ATool** để xác định công cụ được sử dụng.
3	assignedAt	DATETIME	Ngày và giờ công cụ được gán cho người

			dùng.

4.3.2.17.Bảng UserSetting

Bảng 4.39 Bảng mô tả dữ liệu của UserSetting

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi thiết lập.
2	user_id	VARCHAR(255)	Tham chiếu đến bảng **User** để xác định người sở hữu thiết lập.
3	language	VARCHAR(50)	Ngôn ngữ được chọn trong thiết lập.
4	chunkingService	VARCHAR(255)	Dịch vụ phân đoạn hoặc chế độ xử lý.
5	created_at	DATETIME	Ngày và giờ thiết lập được tạo.
6	updated_at	DATETIME	Ngày và giờ thiết lập được cập nhật.

4.3.2.18.Bảng File

Bảng 4.40 Bảng mô tả dữ liệu của File

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi tệp tin.
2	name	VARCHAR(255)	Tên của tệp tin trong hệ thống.

3	originalName	VARCHAR(255)	Tên gốc của tệp tin do người dùng tải lên.
4	path	VARCHAR(3000)	Đường dẫn lưu trữ tệp tin trên hệ thống.
5	extension	VARCHAR(50)	Phần mở rộng của tệp tin (Ví dụ: .png, .jpg, .pdf,...).
6	size	BIGINT	Kích thước của tệp tin (tính bằng byte).
7	url	VARCHAR(3000)	Đường dẫn URL để truy cập tệp tin.
8	conversationId	VARCHAR(255)	Tham chiếu đến bảng **Conversation** để xác định tệp tin thuộc về cuộc trò chuyện nào.

Chương 5. CÀI ĐẶT VÀ KẾT QUẢ

5.1. Cài đặt ứng dụng

5.1.1. Cài đặt môi trường

Hệ thống sử dụng chromaDB thông qua docker được lấy từ trang document chỉnh thức của cơ sở dữ liệu vector này

```
docker pull chromadb/chroma

docker run -p 8000:8000 chromadb/chroma
```

Cài đặt Fast Api

```
pip install fastapi uvicorn
```

Cài đặt môi trường Nodejs theo đường link: https://nodejs.org/

5.1.2. Khởi tạo ứng dụng

Tạo một file .env để chứa các thông tin liên quan đến các API key hoặc là thông tin cơ sở dữ liệu như sau

```
GROQ_API_KEY=
OPENAI_API_KEY=
DEEPGRAM_API_KEY=

# Database configuration
MYSQL_ROOT_PASSWORD
MYSQL_DATABASE="raine_db"
MYSQL_USER=
MYSQL_PASSWORD=
DATABASE_URL=
REDIS_URL="redis://redis:6379"
SERVER_PORT= 9000
```

```
JWT_SECRET=
CLIENT_URL=  // Example"https://192.168.1.3:5173"
GOOGLE_API_KEY=
CLIENT_ID=
CLIENT_SECRET=
REDIRECT_URI="http://localhost:8001/api/v2/google/oauth2callback"
```

Sử dụng command sau để migrate dữ liệu schema lên database

```
npx prisma db push
```

Tiếp theo ta cần khởi tạo dữ liệu ban đầu cho hệ thống

```
mysql -u root -p raine_db < init.sql
```

Hình 5.1 Dữ liệu để khởi tạo cho cơ sở dữ liệu của hệ thống

Để khởi động hệ thống ta chạy lệnh để thực hiện kích hoạt các lệnh chạy của từng source code bên dưới

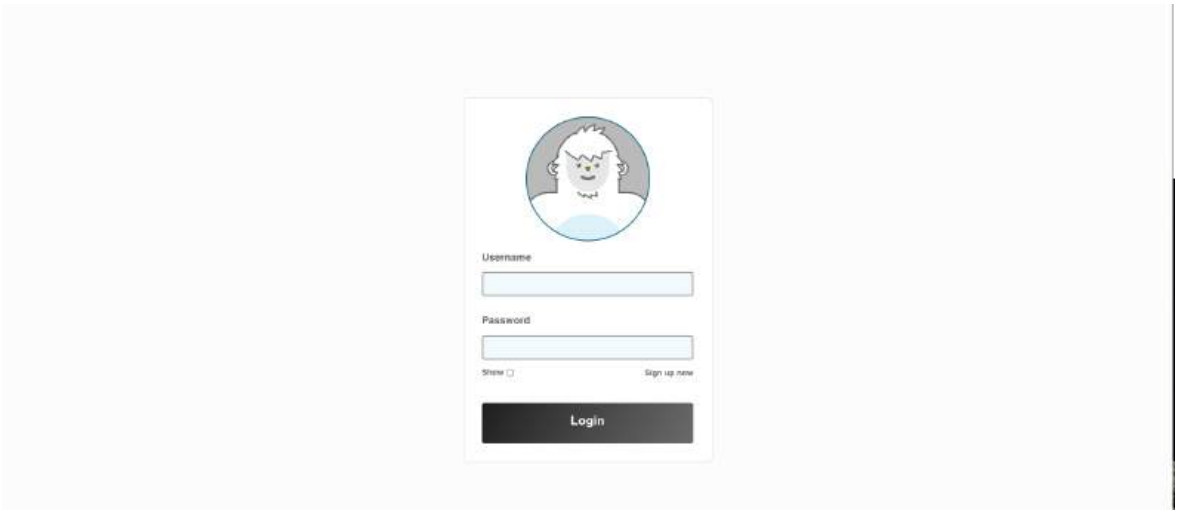
```
./run_all.sh
```

Hình 5.2 Nội dung script khởi động dự án

5.2. Kết quả các màn hình

5.2.1. Màn hình đăng nhập

Người dùng thực hiện đăng nhập vào hệ thống thông qua tên tài khoàn và mật khẩu. Khi nhập mật khẩu người dùng cũng có thể xem chi tiết những gì đã nhập bằng cách nhấn vào nút “Show” để hiển thị mật khẩu vừa nhập. Ngoài ra nếu muốn đăng ký tài khoản mới, người dùng có thể chọn “Sign up now”.



Hình 5.3 Màn hình đăng nhập

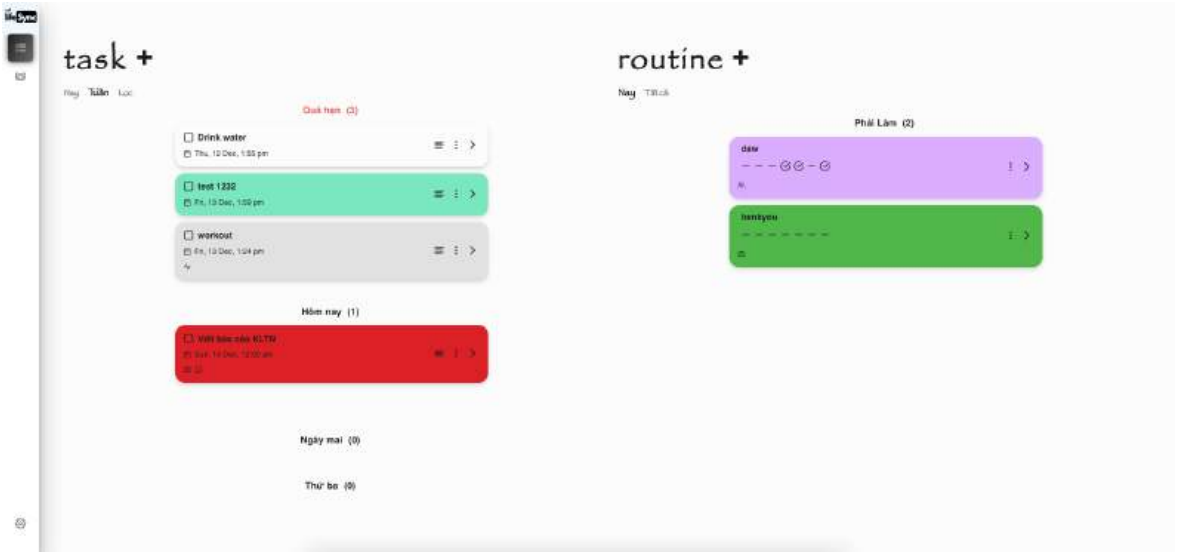
5.2.2. Màn hình đăng ký

Tại trạng đăng ký người dùng thực hiện tạo một tài khoản mới bằng tên tài khoản mật khẩu và nhập lại mật khẩu để xác thực thông tin vừa nhập trùng với mật khẩu đã nhập. Nếu muốn quay lại trang đăng nhập chọn “Back to Login”.

Hình 5.4 Màn hình đăng ký

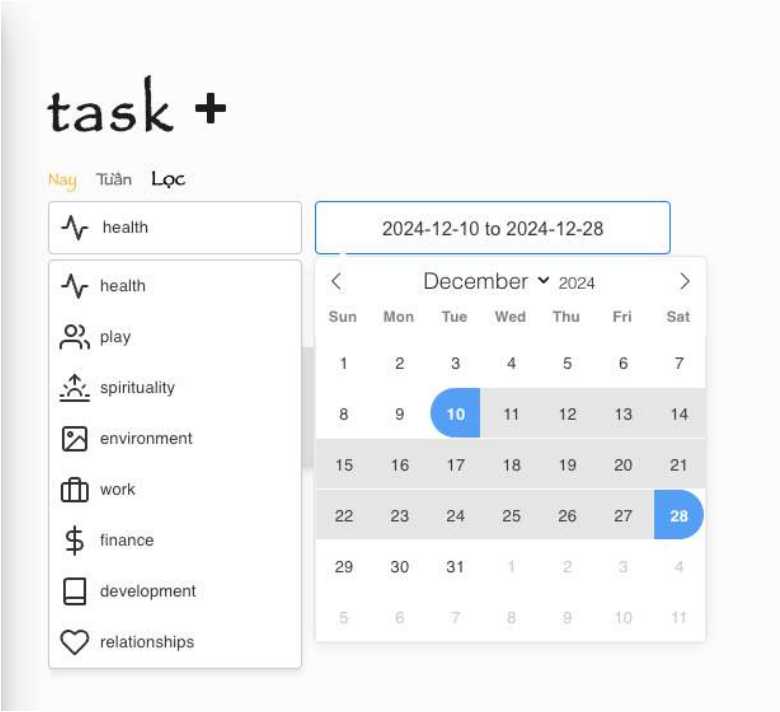
5.2.3. Màn hình quản lý tác vụ và thói quen

Tại màn hình quản lý tác vụ và thói quen được chia thành hai phần chính (**Task)** ở phía bên trái (**Routine)** ở phía bên phải. Phần **Task** hiển thị danh sách các tác vụ của người dùng, được phân loại rõ ràng theo các nhóm như Nay, Tuần và lọc để giúp người dùng tìm kiếm dữ liệu nhanh hơn. Phần **Routine** được chia ra theo mục cần làm và không cần phải làm



Hình 5.5Màn hình quản lý tác vụ và thói quen

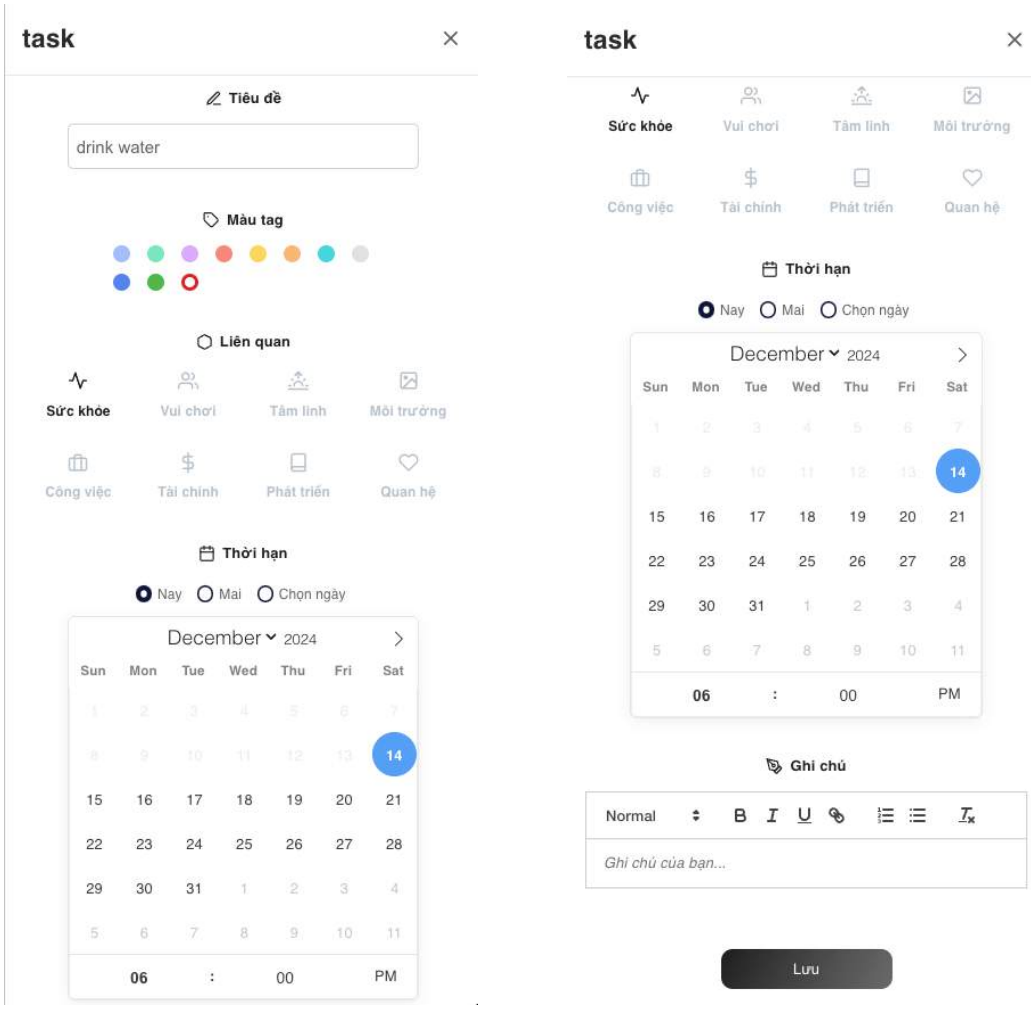
5.2.4. Màn hình Lọc t ác vụ



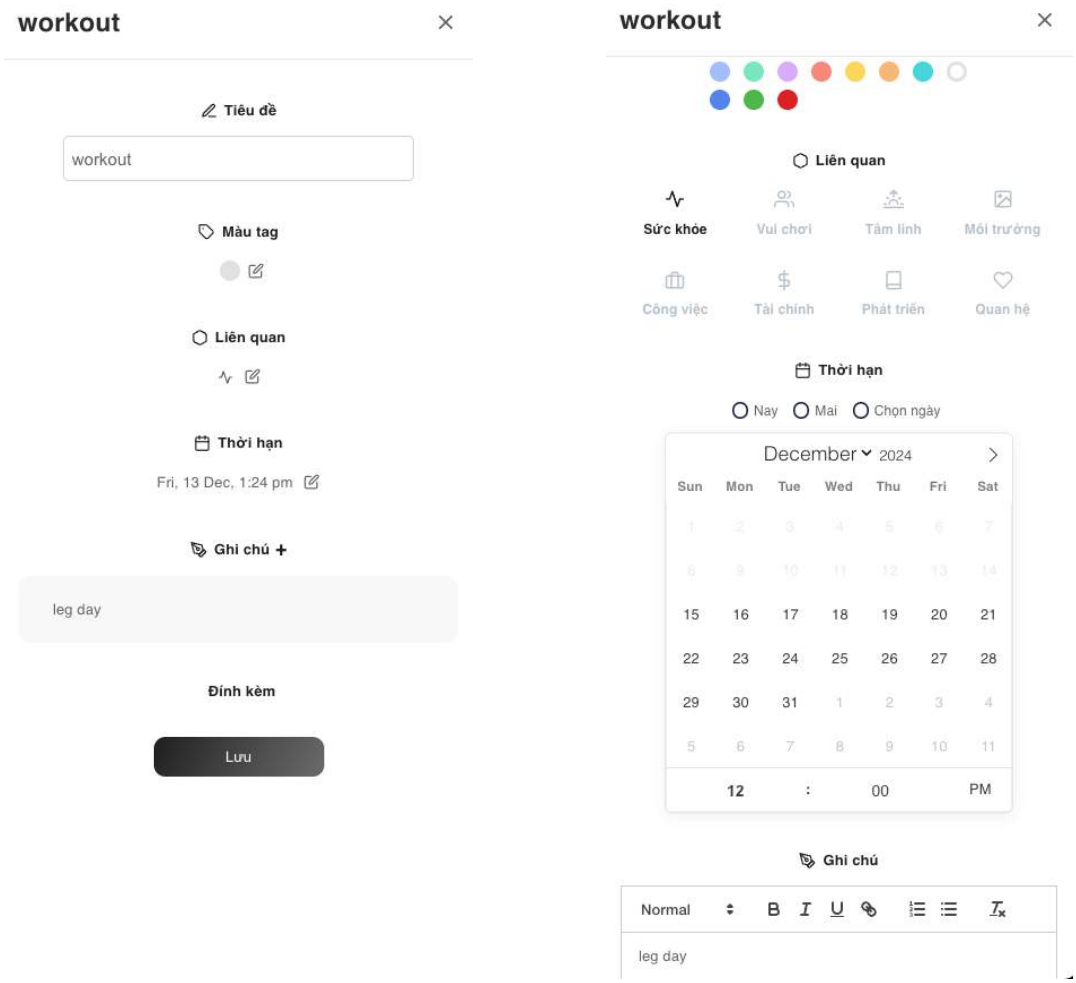
Hình 5.6 Màn hình chức năng tìm kiếm tác vụ

Màn hình hiển thị tính năng **lọc tác vụ** theo danh mục (health, play, work...) và khoảng thời gian cụ thể. Người dùng có thể chọn danh mục và thời gian từ lịch để tìm kiếm hoặc quản lý các tác vụ tương ứng.

5.2.5. Các Modal chức năng của tác vụ



Hình 5.7 Màn hình tạo mới tác vụ



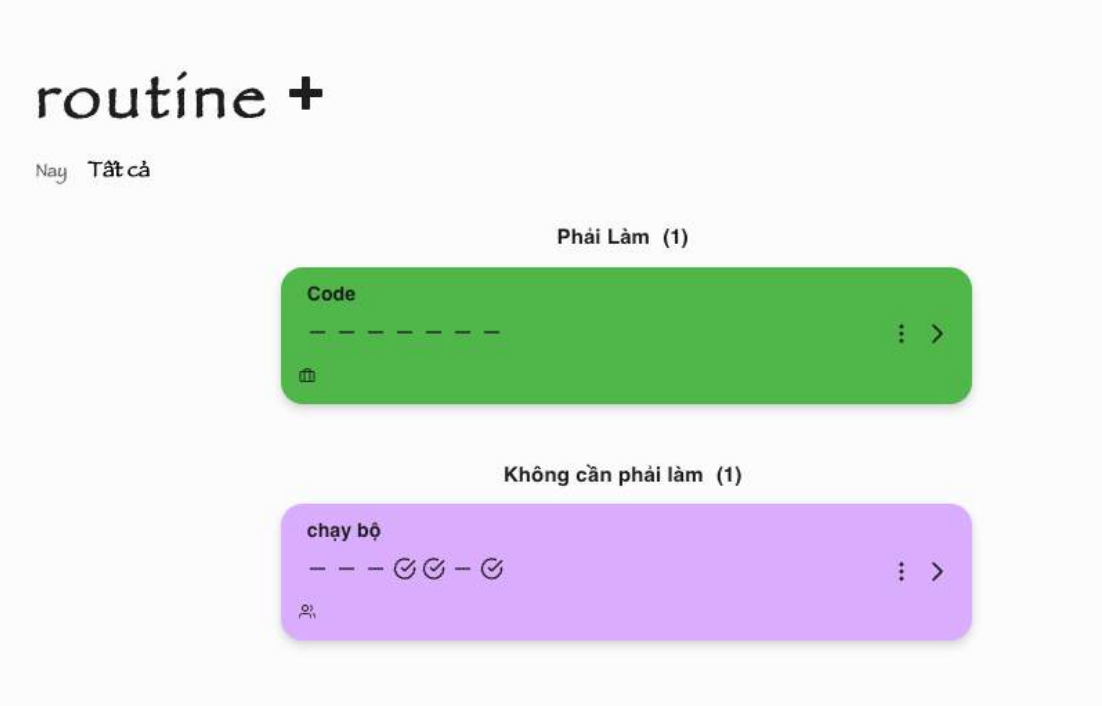
Hình 5.8 Màn hình chi tiết và chỉnh sửa tác vụ

Giao diện **tạo mớ và chi tiết tác vụ** cung cấp các trường nhập liệu và tùy chọn để người dùng có thể thêm một tác vụ mới và chỉnh sửa một cách chi tiết và hiệu quả. Người dùng nhập **tiêu đề** của tác vụ, chọn **màu tag** để phân loại, và liên kết tác vụ với một danh mục cụ thể như **Sức khỏe, Vui chơi, Tâm linh**, v.v. Ngoài ra, tính năng **Thời hạn** cho phép người dùng đặt ngày hoàn thành tác vụ (hôm nay, ngày mai hoặc chọn ngày cụ thể từ lịch).

Phần **Ghi chú** nằm phía dưới, hỗ trợ người dùng nhập các thông tin bổ sung cho tác vụ và có thể tuỳ chỉnh format của thông tin đó. Cuối cùng, người dùng có thể nhấn nút **Lưu** để hoàn tất việc tạo tác vụ mới hoặc chỉnh sửa. Giao diện

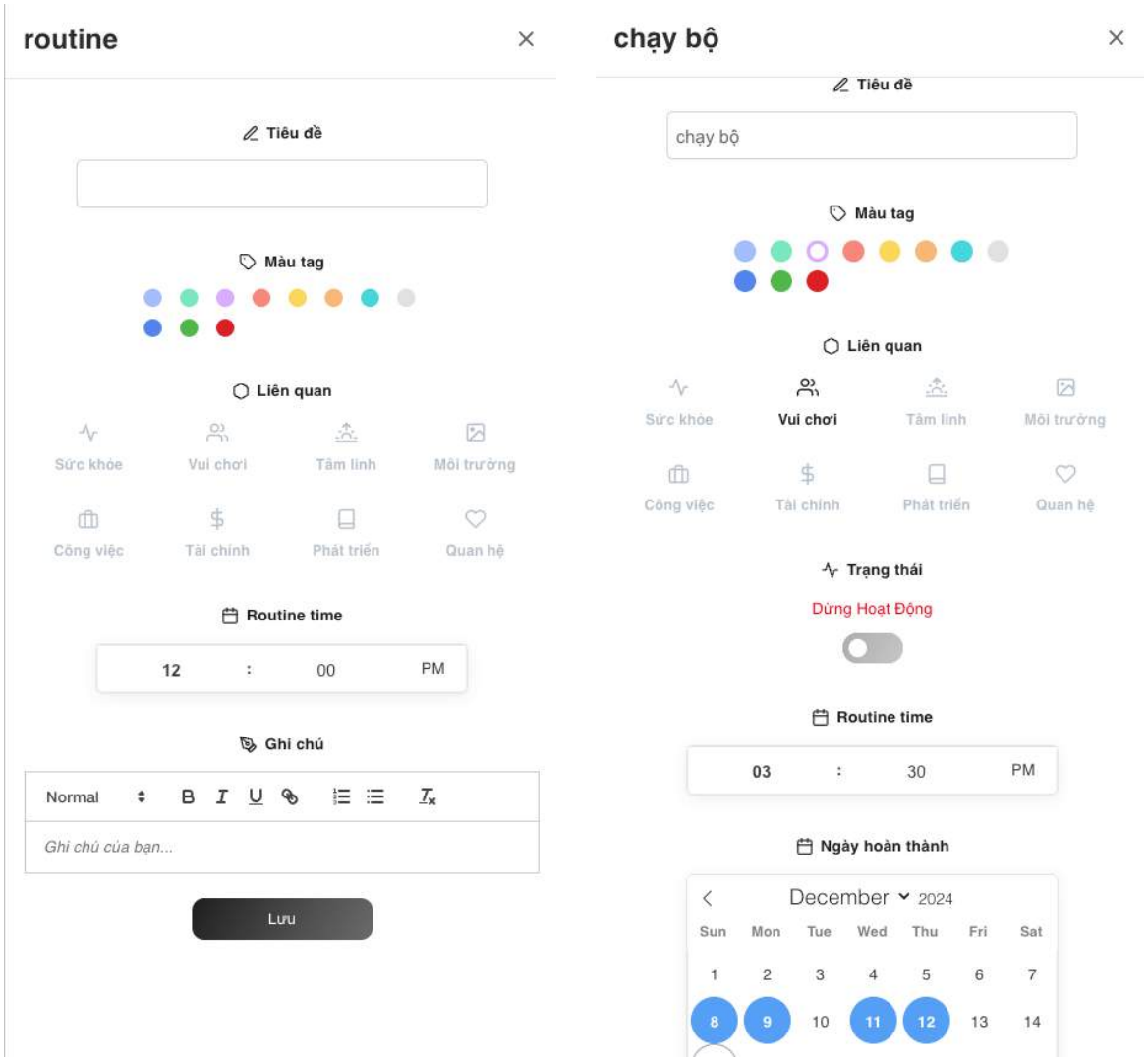
được thiết kế tối giản, trực quan và sử dụng các biểu tượng rõ ràng để hỗ trợ trải nghiệm người dùng tốt nhất.

5.2.6. Màn hình danh sách thói quen



Hình 5.9 Màn hình quản lý thói quen

5.2.7. Modal chức năng của thói quen



Hình 5.10 Màn hình chi tiết thêm mới và chỉnh sửa thói quen

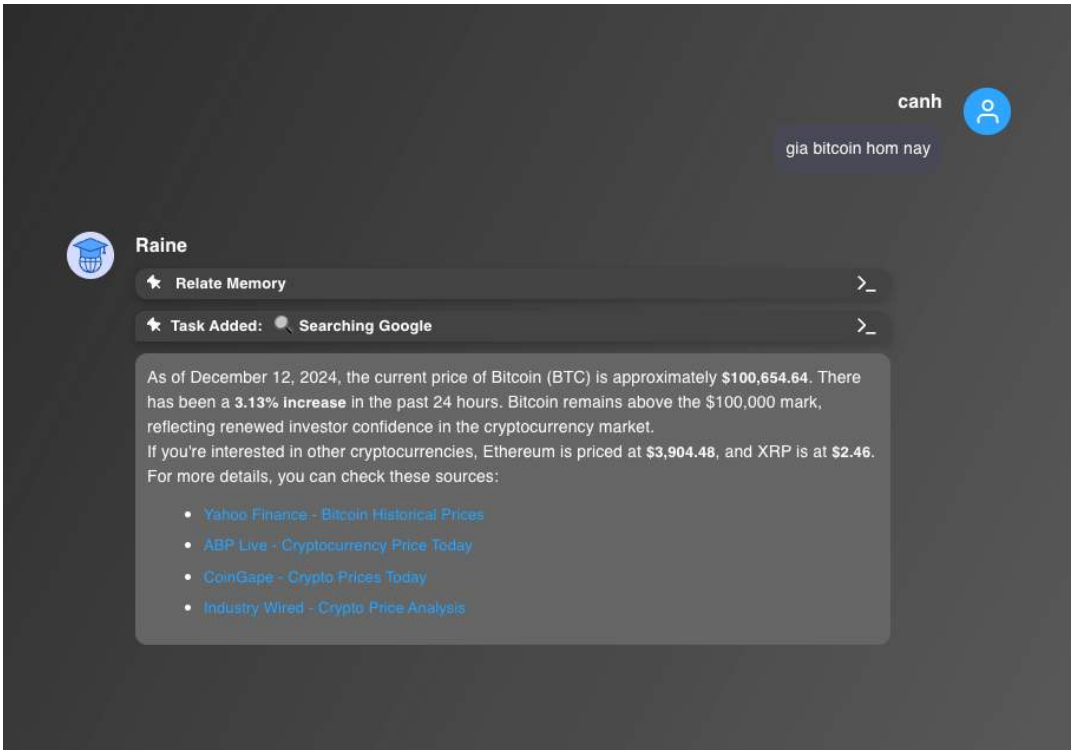
Giao diện modal **thêm mới** và **chỉnh sửa thói quen** cho phép người dùng nhập và điều chỉnh các thông tin liên quan đến thói quen cá nhân. Trong modal thêm mới, người dùng có thể nhập **tiêu đề**, chọn **màu tag** để phân loại, và liên kết thói quen với các danh mục như **Sức khỏe**, **Vui chơi**, **Công việc**, v.v. Người dùng cũng có thể đặt **Routine time** để xác định thời gian thực hiện thói quen trong ngày và thêm **ghi chú** để bổ sung thông tin chi tiết.

Trong modal chỉnh sửa, giao diện bổ sung thêm **Trạng thái** thói quen (ví dụ: **Đang Hoạt Động** hoặc **Dừng Hoạt Động**) và cho phép người dùng cập nhật

Ngày hoàn thành thông qua lịch chọn ngày. Các trường thông tin đều được thiết kế trực quan, với các tùy chọn như đổi màu tag, điều chỉnh thời gian và quản lý trạng thái hoạt động giúp người dùng theo dõi tiến độ thực hiện thói quen một cách hiệu quả.

5.2.8. Màn hình tương tác với trợ lý ảo

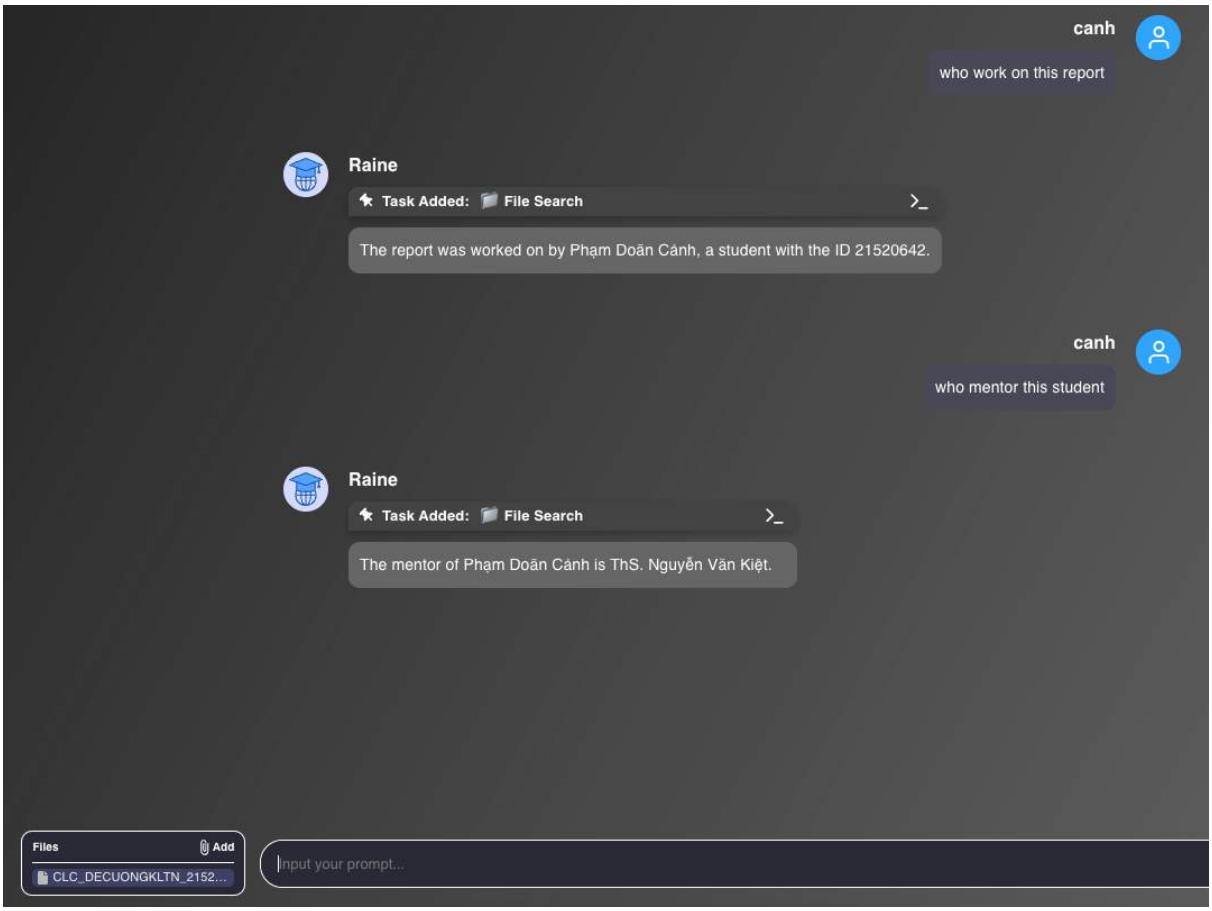
5.2.8.1. Hỏi đáp về thông tin thời gian thực



Hình 5.11 Hỏi đáp về thông tin từ internet

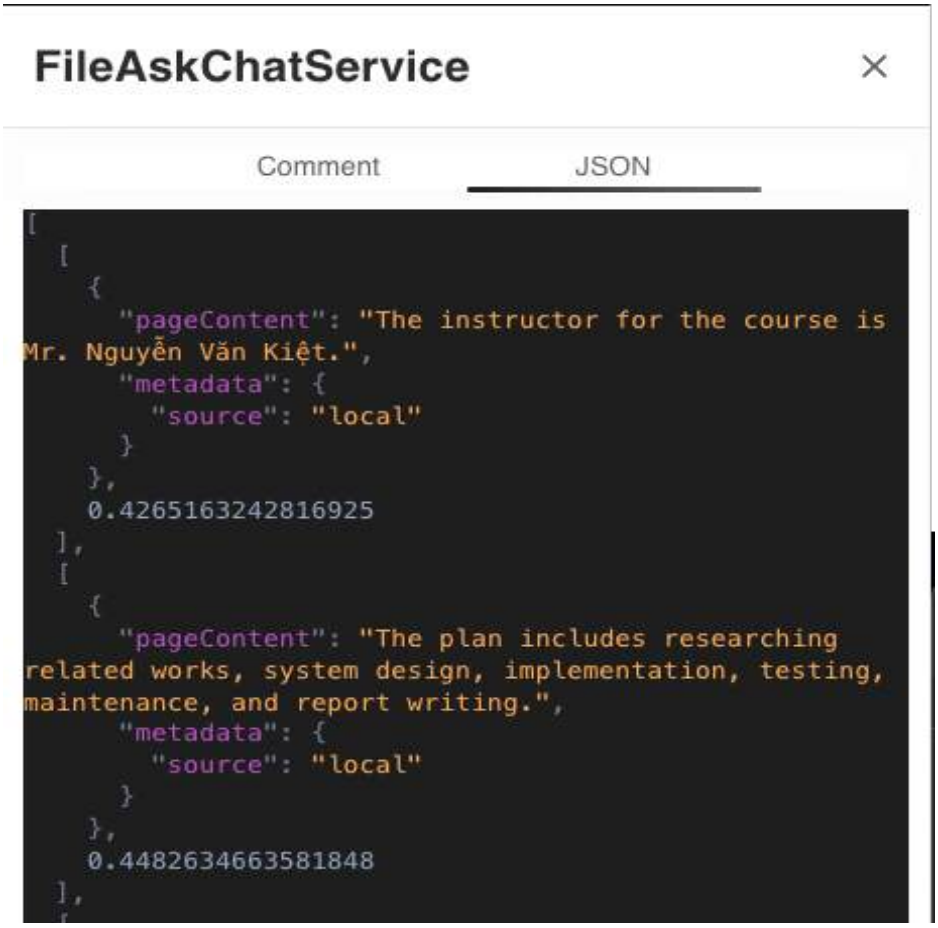
Trợ lý ảo hỗ trợ người dùng trong việc tìm kiếm thông tin từ nguồn internet, cho phép người dùng đặt câu hỏi về thông tin thời gian thực. Trong mỗi câu trả lời trợ lý ảo sẽ đính kèm các đường link có nội dung liên quan để người dùng tiện thao tác và theo dõi.

5.2.8.2. Hỏi đáp về nội dung trong tệp tin



Hình 5.12 Hỏi đáp về nội dung tệp tin

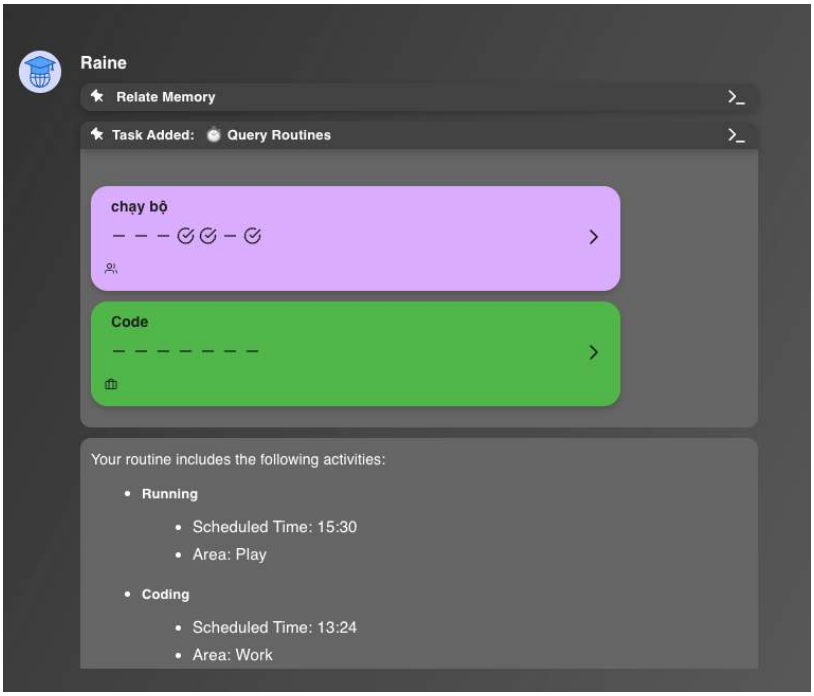
Cho phép người dùng đăng tải một tệp tin pdf hay word bất kỳ, từ đó trợ lý ảo sẽ thực hiện phân tích nội dung trong tệp tin đó và sẵn sàng trả lời bất cứ câu hỏi từ người dùng liên quan đến nội dung trong tệp tin được đăng tải.

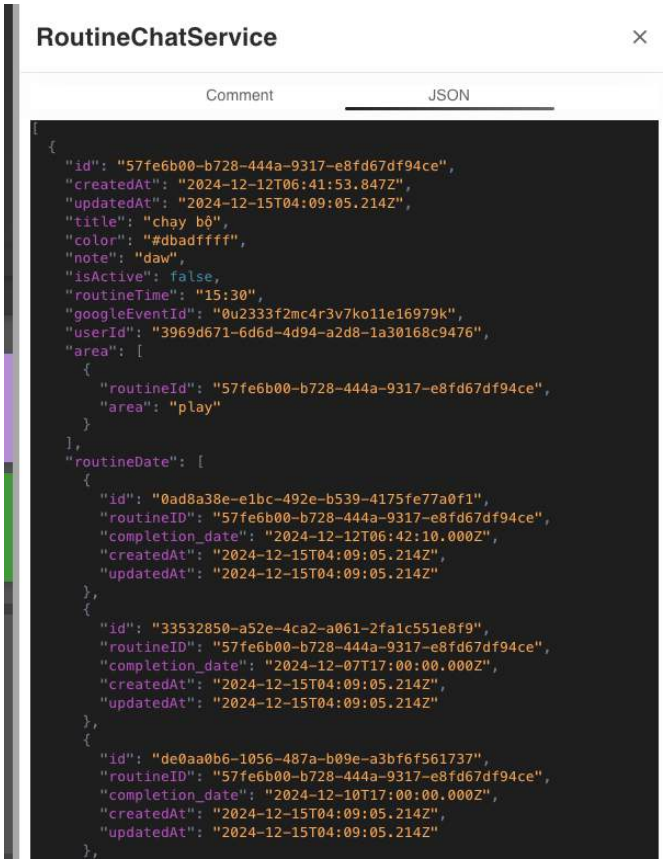


Hình 5.13 Chi tiết dữ liệu xử lý cho quá trình hỏi đáp file

Đồng thời hỗ trợ người dùng cho phép theo dõi các nội dung chi tiết mà trợ lý ảo phân tích ra từ trong tiệp tin dưới dạng dữ liệu JSON.

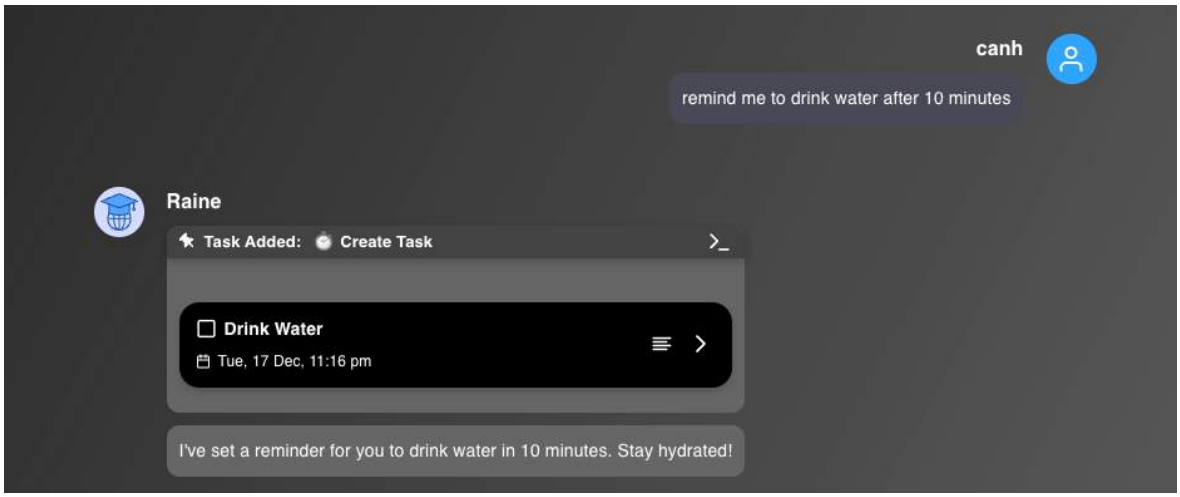
5.2.8.3. Hỏi đáp về các tác vụ và thói quen theo tài khoản



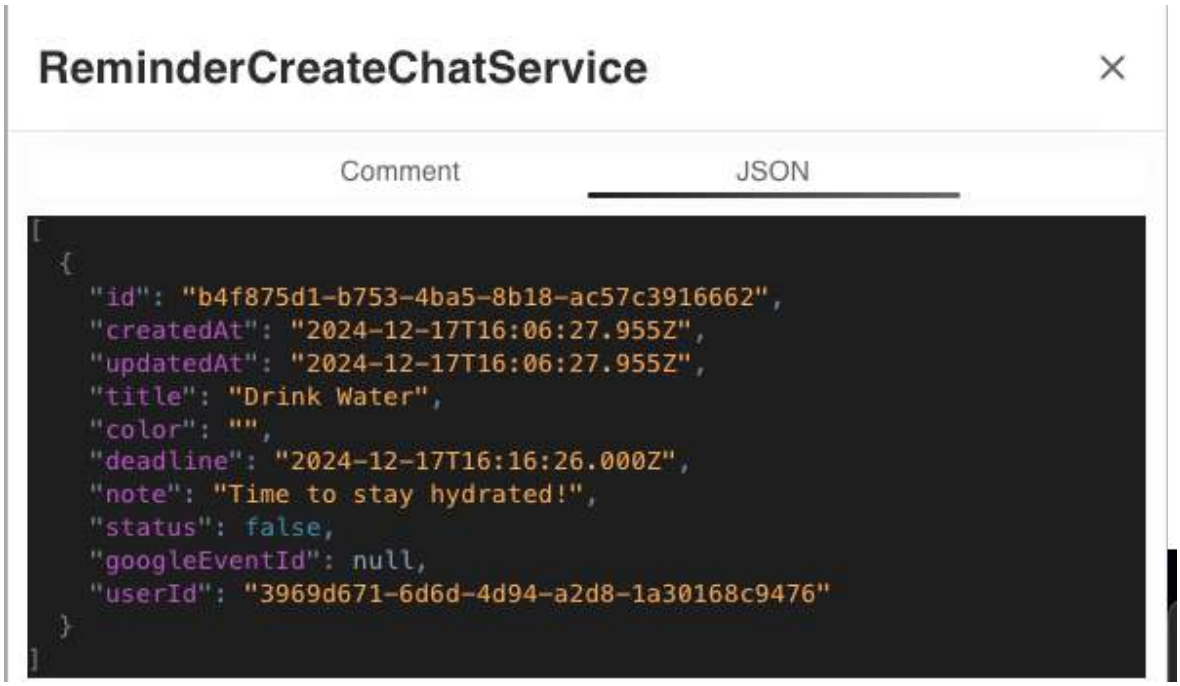


Hình 5.14 Hỏi đáp về và thói quen

Hỗ trợ người dùng hỏi đáp tương tác các dữ liệu tác vụ hay thói quen trong cơ sở dữ liệu của tài khoản người dùng hiện tại, đồng thời cho phép người dùng tương tác để tạo ra các tác vụ hay thói quen mới.

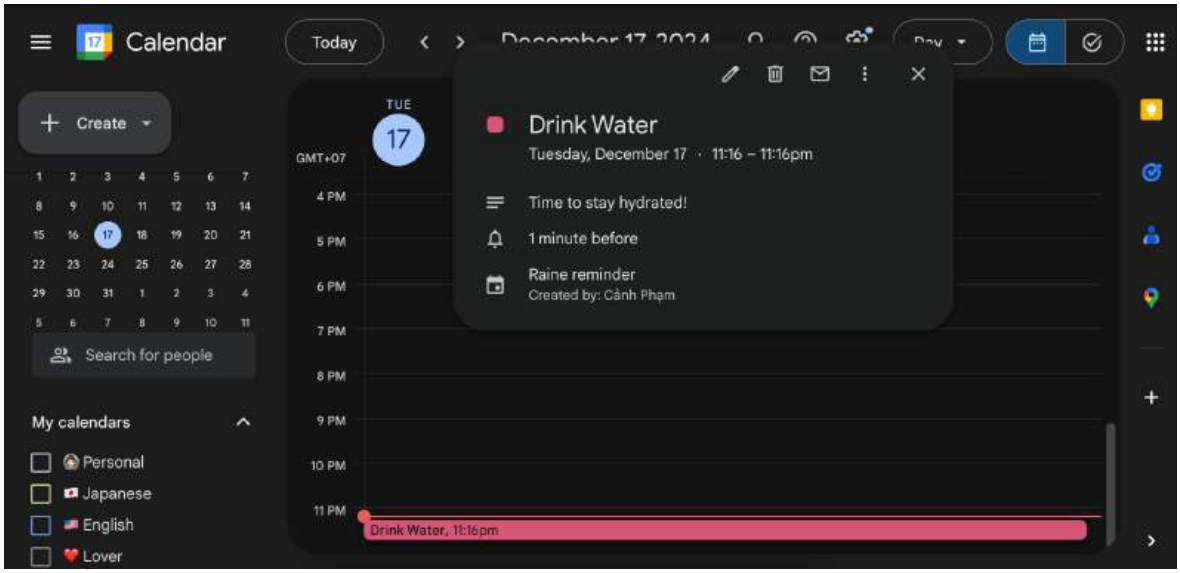


Hình 5.15 Hỏi đáp tạo tác vụ nhắc nhở



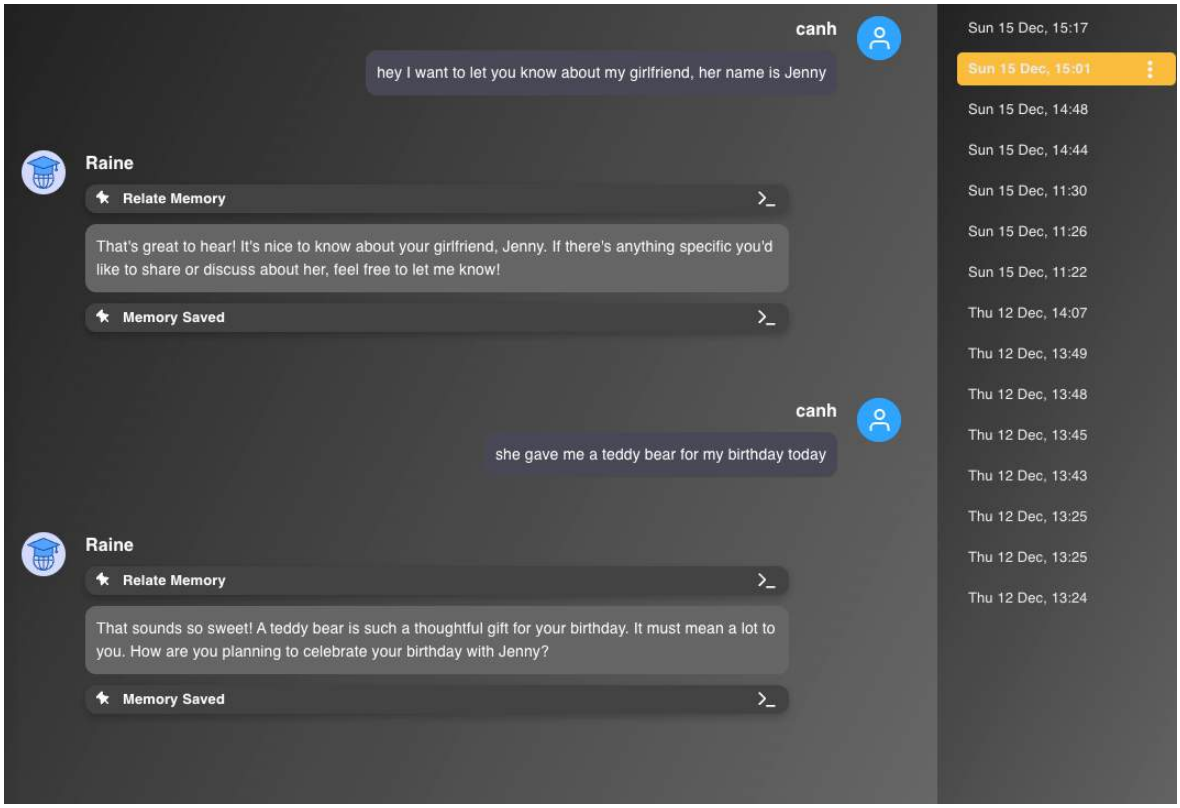
Hình 5.16 Dữ liệu được tạo tương ứng

Các dữ liệu khi được tạo ra cũng sẽ được tự động đồng bộ lên tài khoản Google Calendar tương ứng mà người dùng đã liên kết.



Hình 5.17 Tự động đồng bộ dự liệu lên tài khoản Google Calendar

5.2.8.4. Tận dụng bộ nhớ dài hạn



Hình 5.18 Phân tích và lưu trữ bộ nhớ dài hạn

Trong quá trình tương tác trợ lý ảo sẽ tự động thực hiện phân tích nội dung trong câu nói của người dùng để thực hiện xem xét có nên ghi nhớ thông tin từ câu nói người dùng hay không và nếu có sẽ phân tích ra nội dung cần ghi nhớ để thực hiện lưu trữ lại phục vụ cho sau này.

Hình 5.19 Thông tin được phân tích và lưu trữ dưới bộ nhớ dài hạn

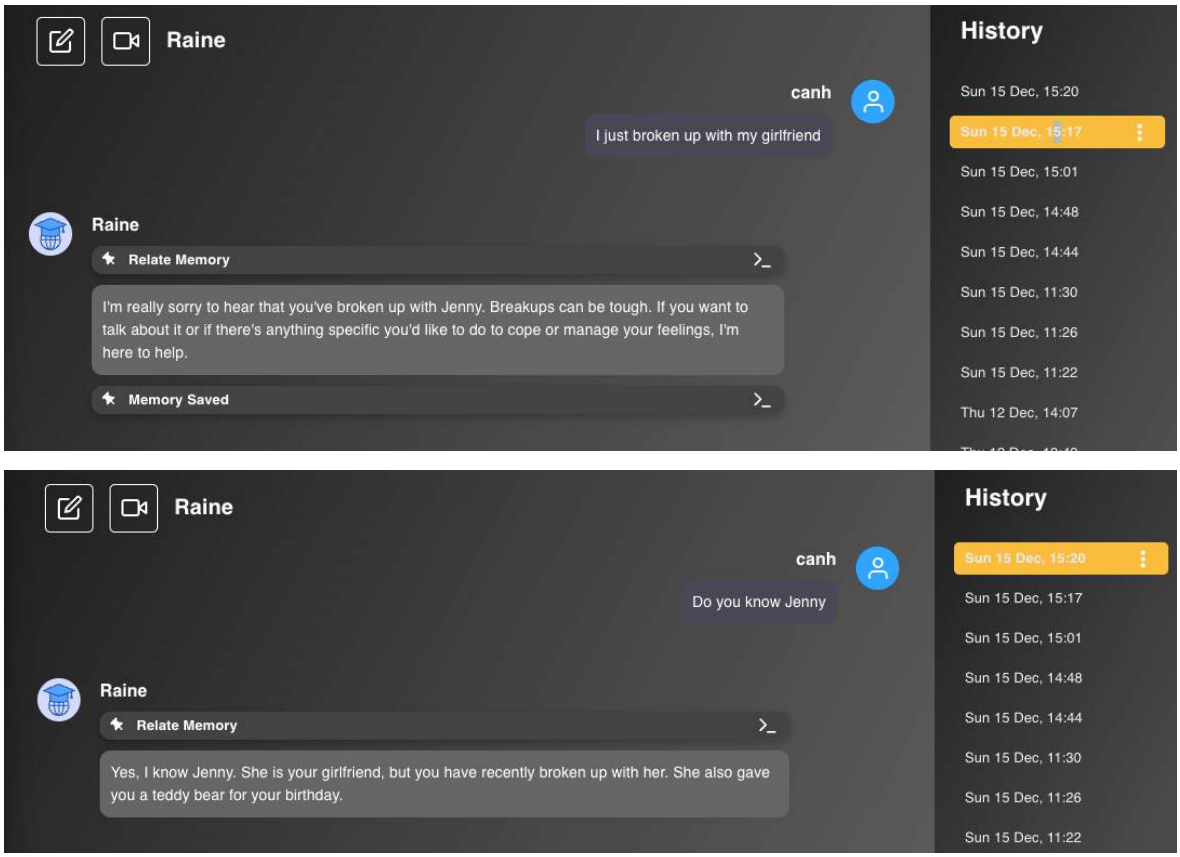
Các thông tin được phân tích ra sẽ bao gồm:

- Guide: Hướng dẫn về tóm tắt về chủ đề của nội dung
- Answer: câu trả lời cho nội dung đó

Và các tiêu chí để phục vụ cho quá trình phân tích sự tương đồng cho từng bộ nhớ:

- **Personal detail**: Xác định liệu nội dung có liên quan đến chi tiết cá nhân của người dùng hay không.
- **Relationship**: Phân tích mối quan hệ của người dùng với người khác được đề cập trong nội dung.
- **Favorite**: Nhận diện nội dung có liên quan đến sở thích hoặc điều yêu thích của người dùng.

- **Time-sensitive**: Đánh giá liệu thông tin trong nội dung có tính chất nhạy cảm về thời gian như ngày, giờ cụ thể hay không.
- **Context-relevant**: Xác định nội dung có phù hợp với ngữ cảnh của cuộc hội thoại hoặc yêu cầu hiện tại không.
- **Ai-actionable**: Kiểm tra xem nội dung có yêu cầu AI thực hiện hành động cụ thể như tìm kiếm thông tin, thiết lập lời nhắc, quản lý tác vụ hoặc cung cấp dữ liệu thời gian thực hay không.
- **Frequently-mentioned**: Xác định thông tin có được người dùng nhắc lại nhiều lần trong các tương tác trước đó hay không.

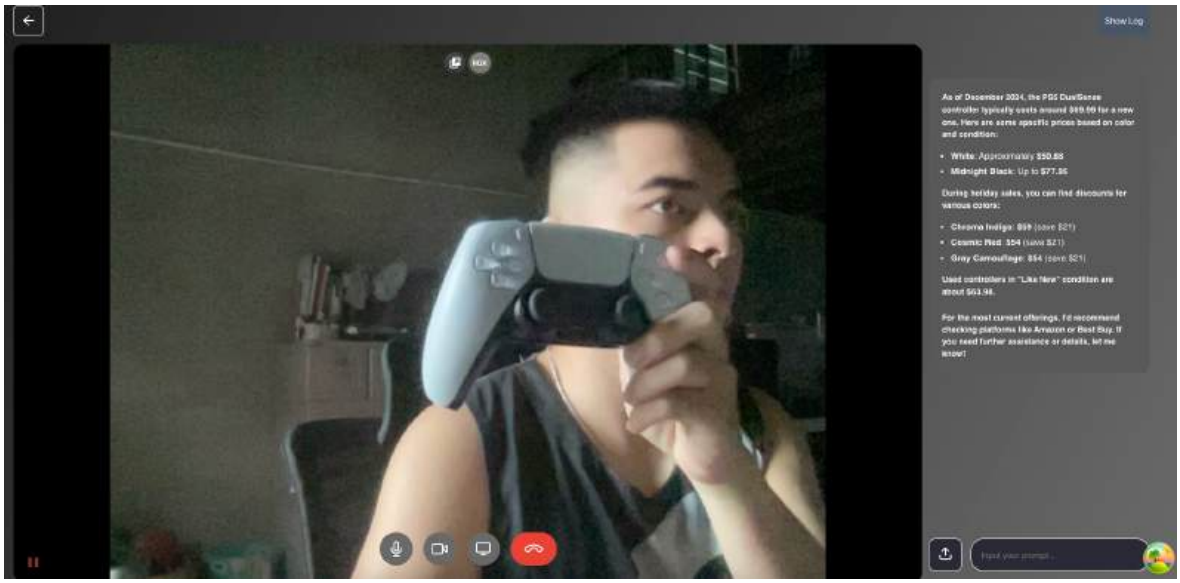


Hình 5.20 Phân tích bộ nhớ dài hạn để trả lời câu hỏi

Hình 5.21 Phân tích và cập nhật thông tin bộ nhớ

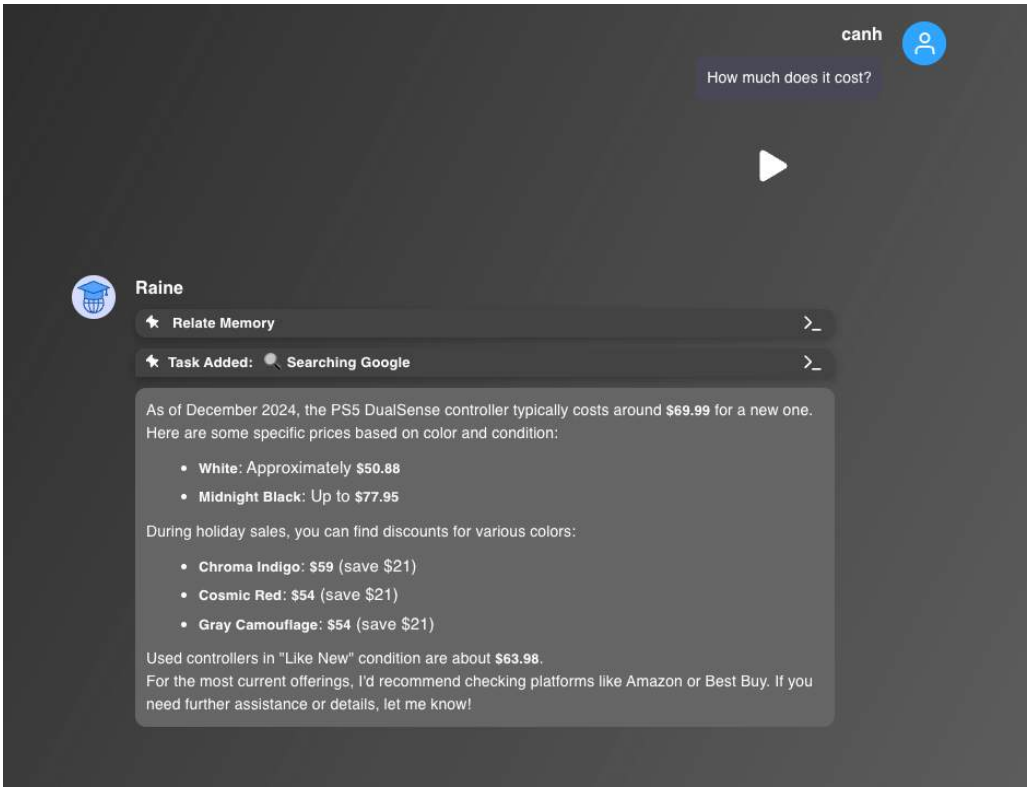
Hình 5.22 Dữ vào các thông tin đã được lưu trữ để trả lời

5.2.9. Màn hình gọi video với trợ lý ảo



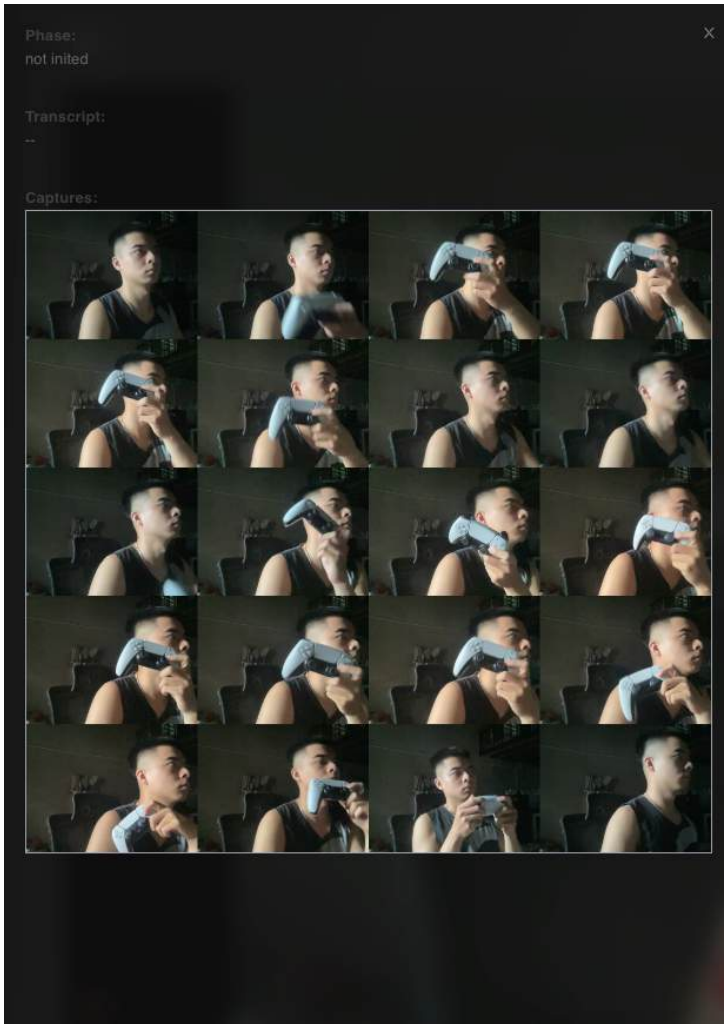
Hình 5.23 Màn hình gọi video

Giao diện hiển thị tính năng **gọi video** giữa người dùng và trợ lý ảo. Người dùng có thể tương tác với trợ lý qua giọng nói hoặc tin nhắn văn bản trong khung nhập liệu nằm ở góc dưới bên phải. Giao diện video còn bao gồm các điều khiển như **tắt/mở micro**, **tắt/mở video**, **chia sẻ màn hình** và **kết thúc cuộc gọi** để người dùng tùy chỉnh trải nghiệm trong quá trình trò chuyện.



Hình 5.24 Lịch sử lưu lại của cuộc gọi

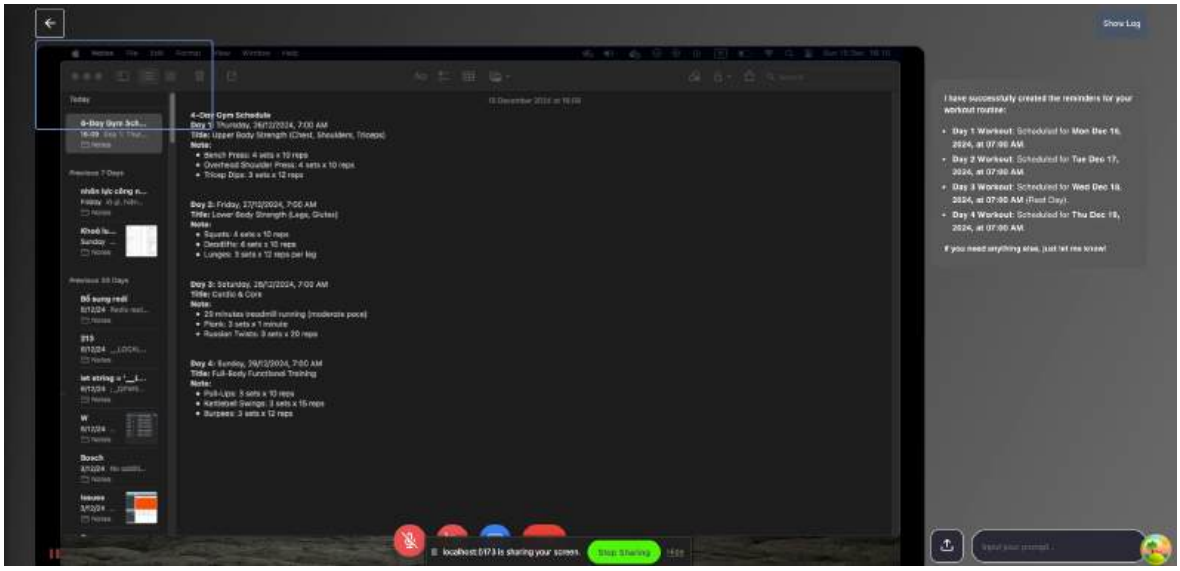
Trong quá trình gọi Video trợ lý ảo có thể thực hiện khả năng sử dụng các công cụ (Tools) để thực hiện các quá trình tìm kiếm hay truy xuất thông tin dữ liệu bên ngoài để đưa ra câu trả lời tốt hơn cho người dùng.



Hình 5.25 Màn hình hiển thị các frame video đã được ghi lại

Dữ liệu từng khung hình trong quá trình gọi sẽ liên lục được lưu lại sau mỗi giây để từ đó gộp lại thành một tấm ảnh lớn, giúp cho trợ lý ảo có thể dễ dàng tiếp cận bối cảnh của người dùng hơn thay vì một đoạn video dài

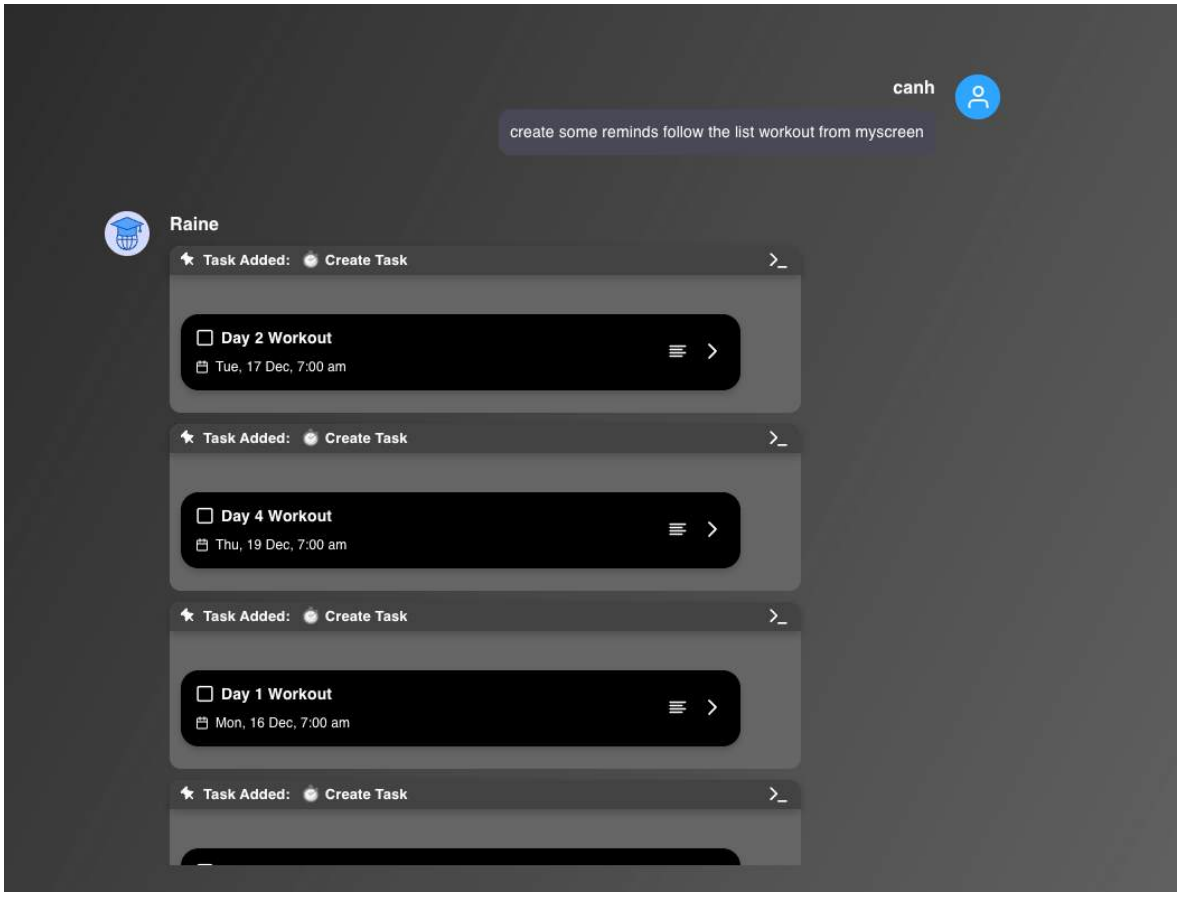
5.2.10. Màn hình chia sẻ màn hình với trợ lý ảo



Hình 5.26 Màn hình chia sẻ màn hình

Người dùng cũng có thể tương tác với trợ lý ảo thông qua khả năng chia sẻ màn hình, giúp trợ lý hiểu rõ hơn về ngữ cảnh và nội dung mà người dùng đang làm việc. Trong trường hợp này, người dùng chia sẻ nội dung liên quan đến **lịch trình tập luyện**. Trợ lý ảo phản hồi bằng cách tạo các **lời nhắc** cụ thể cho từng ngày, bao gồm ngày và thời gian chính xác (ví dụ: ngày 15, 16, 17, 19 tháng 12 năm 2024 lúc 07:00 sáng).

Giao diện cho thấy **thông báo xác nhận** từ trợ lý ảo nằm ở phía bên phải màn hình, đảm bảo người dùng nhận được thông tin rõ ràng và chính xác. Đồng thời, thanh điều khiển ở phía dưới cung cấp các chức năng như **dừng chia sẻ**, **ẩn chia sẻ**, và **kết thúc cuộc gọi**, mang lại trải nghiệm thuận tiện và hiệu quả khi cộng tác với trợ lý ảo.



Hình 5.27 Kết quả từ trợ lý ảo sau qúa trình chia sẻ màn hình

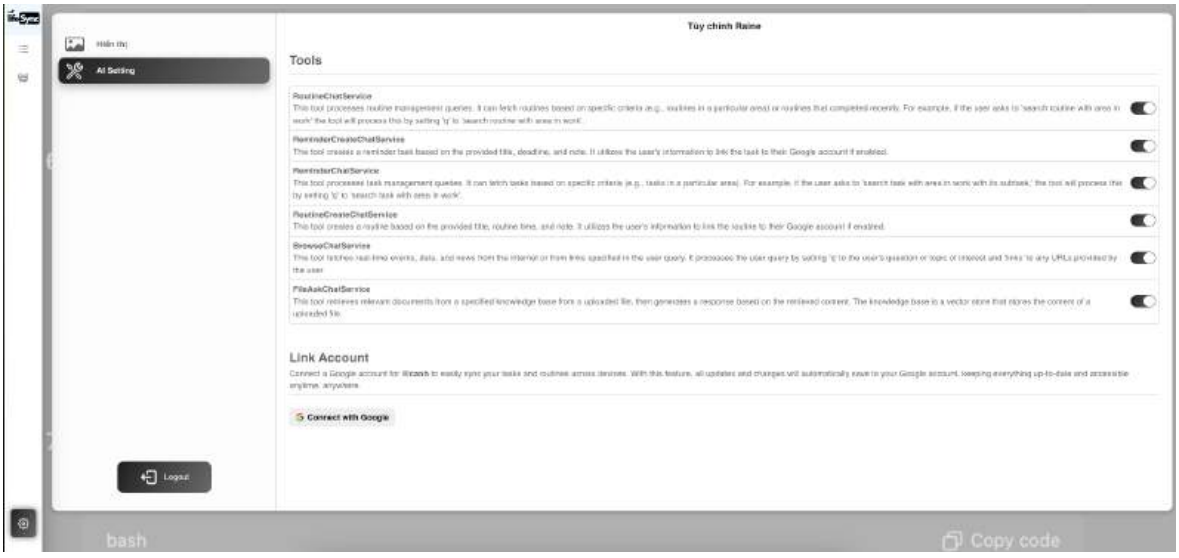
Trợ lý ảo có thể tương tác dựa trên nội dung được chia sẻ trên màn hình để thực hiện kích hoạt sử dụng công cụ tạo tác vụ để lên lịch trình cho người dùng.

5.2.11. Màn hình thiết lập hình nền

Hình 5.28 Màn hình thiết lập hình nền

Hệ thống hỗ trợ người dùng tính năng thiết lập hình nền cho toàn bộ hệ thống. Người dùng có thể sử dụng hình nền có sẵn trên ứng dụng hoặc cũng có thể đăng tải một ảnh bất kỳ dưới máy để làm hình nền cho ứng dụng

5.2.12. Màn hình thiết lập chức năng liên quan đến trợ lý ảo



Hình 5.29 màn hình thiết lập trợ lý ảo

Màn hình thiết lập chức năng cho trợ lý ảo cho phép người dùng có thể tuỳ chính bật tắt các công cụ (tool) mà trợ lý ảo có thể sử dụng. Điều này giúp cho việc hạn chế một số trường hợp trợ lý ảo sẽ tự động sử dụng một số công cụ không mong muốn trong quá trình tương tác. Để đảm bảo độ ổn định trong quá trình tương tác người dùng nên hạn chế bật một lần hết tất cả chức năng.

Chương 6. **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

6.1. Ưu điểm

Tận dụng từ các kỹ thuật tư duy chuỗi và các kỹ thuật xoay quanh truy xuất tăng cường trong việc áp dụng vào mô hình ngôn ngữ để phát triển một giải pháp ứng dụng tích hợp trợ lý ảo với khả năng phân tích, ghi nhớ và học hỏi từ quá trình tương tác với người dùng để đưa ra các quyết định và câu trả lời phù hợp nhất với từng người dùng. Trợ lý ảo cũng được tích hợp thêm khả năng sử dụng các công cụ (tool call) để lấy dữ liệu từ các nguồn khác ngoài dữ liệu đã được huấn luyện để trả lời cho người dùng như:

- Lấy dữ liệu mang tính thời gian thực từ nguồn internet thông qua quá trình dùng serper API để tìm kiếm thông tin theo từ khoá liên quan, từ đó sử dụng puppeteer để lấy các dữ liệu từ các đường dẫn tìm được và cuối cùng đưa vào langgraph để thực hiện đánh giá và tóm tắt nội dung trước khi trả về kết quả cho người dùng.
- Lấy dữ liệu liên quan đến các tệp tin đã được do người dùng cung cấp thông qua quá trình indexing tệp tin vào cơ sở dữ liệu vector bằng kỹ thuật agentic hoặc semantic trước đó và tìm kiếm top K nội dung liên quan từ cơ sở dữ liệu.
- Lấy hoặc cập nhật các dữ liệu về các lịch trình thói quen hay tác vụ liên quan đến tài khoản người dùng trong cơ sở dữ liệu thông qua việc phân tích tạo các câu truy vấn SQL tương ứng với yêu cầu của người dùng và kích hoạt vào cơ sở dữ liệu. Đồng thời các dữ liệu này sẽ được đồng bộ với tài khoản google calendar tương ứng mà người dùng đã liên kết để hỗ trợ khả năng đồng bộ trên nhiều thiết bị mà có đăng nhập cùng một tìa khoản google.

Ngoài ra, hệ thống còn cung cấp cho người dùng tính năng gọi video và chia sẻ màn hình trong quá trình tương tác với trợ lý ảo để giúp nâng cao trải nghiệm người dùng trong quá trình tương tác và đồng thời giúp trợ lý ảo có thể hiểu rõ

hơn về bối cảnh đang tương tác với người dùng từ đó có thể đưa ra các câu trả lời thiết thực và gần gủi hơn.

6.2. Khuyết điểm

Bên cạnh những tính năng đáng chú ý mà ứng dụng đã triển khai, song vẫn còn tồn tại một số khuyết điểm. Trong chức năng gọi điện bằng video và chia sẻ màn hình đối với trợ lý ảo, tốc độ phản hồi của trợ lý ảo vẫn còn là một hạn chế do quá trình xử lý riêng lẻ từng loại dữ liệu như chuyển từ giọng nói sang văn bản, xử lý hình ảnh đưa vào mô hình ngôn ngữ lớn chưa được tối ưu và chuyển từ văn bảng của kết quả cuối cùng sang giọng nói. Đồng thời đối với các câu hỏi của người dùng liên quan đến thông tin mang tính thời gian thực, hiện tại hệ thống cũng tiêu tốn khá nhiều thời gian để triển khai. việc lấy toàn bộ dữ liệu của một vài đường dẫn tìm được, trước khi đưa vào mô hình ngôn ngữ tóm tắt, đánh giá để đưa ra kết quả cuối cùng, do vậy tốc độ phản hồi đối với chức năng này cũng chưa được đáng kể đối với các câu hỏi đòi hỏi nhiều thông tin mang tính thời gian thực.

6.3. Hướng phát triển

Để có thể phát triển đề tài một cách tốt hơn nữa, tôi dự định định sẽ huấn luyện thêm một số mô hình ngôn ngữ nhỏ hơn để thực hiện cho một số chức năng hiện nay như chuyển đổi qua lại giữa giọng nói và văn bảng và ngược lại để tối ưu hơn về thời gian trả lời đối với chức năng gọi video và chia sẻ màn hình. Đồng thời tích hợp thêm các công nghệ liên quan đến thị giác máy tính và chức năng gọi điện video này để xây dựng thêm một số chức năng nhận diện bối cảnh xoay quành người dùng thay vì sử dụng mô hình ngôn ngữ lớn như hiện tại. Đồng thời nghiên cứu thêm một số kĩ thuật khác về RAG Graph để ứng dụng vào các chức năng tìm kiếm thông tin từ nguồn internet và tệp tin hiện nay.

TÀI LIỆU THAM KHẢO

[1] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao and Karthik Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," 3 Dec 2023.

[2] S. Siddesh, A. Ullas and B. Santosh, "Artificial intelligence based voice assistant," IEEE, 2020.

[3] D. T. Nguyễn, T. T. Phạm, D. N. Lê, T. V. Đỗ, T. X. Vũ, H. V. Nguyễn and T. N. Nguyễn, "NGHIÊN CỨU TRỢ LÝ ẢO ỨNG DỤNG TRÍ TUỆ NHÂN TẠO," KHOA HỌC & CÔNG NGHỆ, Hồ Chí Minh, 2022.

[4] Rohit Tamrakar and Niraj Wani, "Design and Development of CHATBOT: A Review," INTERNATIONAL CONFERENCE On “Latest Trends in Civil, Mechanical and Electrical Engineering”, 2021.

[5] J. Long, "Large Language Model Guided Tree-of-Thought," 15 May 2023.

[6] N. Rajkumar, Raymond Li and Dzmitry Bahdanau, "Evaluating the Text-to-SQL Capabilities of Large Language Models.," 15 Mar 2022.

[7] T. Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang and Dong Yu, "Dense X Retrieval: What Retrieval Granularity Should We Use?," 12/2023.