BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH TP. HỒ CHÍ MINH

NGHIÊN CỨU VÀ XÂY DỰNG MÔ HÌNH TIẾP NHẬN Ý KIẾN VÀ GIẢI ĐÁP THẮC MẮC CỦA SINH VIÊN TẠI TRƯỜNG ĐẠI HỌC

BÁO CÁO ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH TP HỒ CHÍ MINH

TRỊNH VIẾT KHẢI – 215122096

NGHIÊN CỨU VÀ XÂY DỰNG MÔ HÌNH TIẾP NHẬN Ý KIẾN VÀ GIẢI ĐÁP THẮC MẮC CỦA SINH VIÊN TẠI TRƯỜNG ĐẠI HỌC

BÁO CÁO ĐỀ TÀI NGHIÊN CỨU KHOA HỌC

GVHD: THS HOÀNG VĂN HIẾU

TP. Hồ Chí Minh – Năm 2024

THÔNG TIN ĐỀ TÀI

1. Tình hình chung:

Trong môi trường đại học, sinh viên thường gặp phải nhiều thắc mắc và khó khăn trong quá trình học tập, từ việc hiểu rõ nội dung môn học, đến việc tìm kiếm thông tin, các sự kiện workshop hay event, các bằng tin học hay tiếng anh, đóng học phí và đổi lớp như thế nào, hay thậm chí là xác định hướng nghiên cứu. Để giúp đỡ sinh viên, mô hình tiếp nhận ý kiến và giải đáp thắc mắc đã được phát triển và áp dụng trong các trường đại học.

Mô hình này thông qua việc tiếp nhận ý kiến từ sinh viên, bao gồm các câu hỏi, đề xuất, hay những vấn đề gặp phải trong quá trình học tập thông qua hệ thống trao đổi thông tin trực tuyến các sinh viên có thể gửi những thắc mắc của mình tới đội ngũ giảng viên, nhân viên tư vấn hoặc các chuyên gia trong lĩnh vực tương ứng. Đội ngũ chuyên gia và giảng viên sẽ tiếp nhận các ý kiến và thắc mắc này, và cung cấp giải đáp, hướng dẫn hoặc tư vấn phù hợp để giúp sinh viên vượt qua khó khăn. Mô hình này cũng có thể xây dựng cơ sở dữ liệu để lưu trữ, phân loại và tra cứu các câu hỏi và giải đáp đã được thực hiện trước đó, từ đó tạo ra một nguồn thông tin phong phú và dễ dàng truy cập cho sinh viên.

Tổng quan, nghiên cứu và xây dựng mô hình tiếp nhận ý kiến và giải đáp thắc mắc của sinh viên tại trường đại học mang lại nhiều lợi ích cho cả sinh viên và giảng viên. Nó tạo điều kiện thuận lợi cho việc chia sẻ thông tin, tương tác và hỗ trợ trong quá trình học tập, từ đó nâng cao chất lượng giảng dạy và sự thành công của sinh viên.

2. Lý do thực hiện đề tài:

Mục tiêu của mô hình là giúp cho các khoa, phòng ban tiếp nhận ý kiến và giải đáp thắc mắc, bên cạnh việc giúp sinh viên giải quyết các vấn đề trong quá trình học tập, còn tạo ra một môi trường giao tiếp tương tác giữa sinh viên và giảng viên. Khuyến khích sự tham gia và chia sẻ thông tin, đồng thời tạo ra một cầu nối để sinh viên có thể tận dụng kiến thức

và kinh nghiệm của giảng viên, đồng thời giúp giảng viên hiểu rõ hơn về nhu cầu và yêu cầu của sinh viên.

3. Mục tiêu đề tài:

- Mô hình ứng dụng là trang web hoặc thiết bị di động giúp giao tiếp tương tác trực tiếp giữa sinh viên và giảng viên.
- Sinh viên chỉ cần sử dụng email UEF để đăng nhập (sau đó có thể thiết lập tài khoản, mật khẩu) và hỏi đáp các giảng viên cùng khoa hoặc liên hệ các phòng ban trực tiếp. Bên cạnh đó người dùng có thể xem các bài viết về khoa đó khi họ đăng tải bài mới lên.
- Xây dựng website dạng blog gồm 3 vai trò chủ yếu là sinh viên, phòng ban và bên quản trị:
    - Về phía sinh viên: họ sẽ được sử dụng các tính năng có ở trên website ví dụ như giao tiếp tương tác trực tiếp với các giảng viên, phòng ban, xem các bài viết liên quan đến khoa, phòng ban,...
    - Về phía phòng ban: các thầy cô sẽ sử dụng để trả lời, giải đáp các thắc mắc của sinh viên, bên cạnh đó họ cũng thể đăng các bài viết về khoa lên trên đó
    - Về phía quản trị: có thể đăng nhập kiểm soát các thông số, số lượng người dùng, bài post, khoa, phòng ban và tùy chỉnh quyền của người dùng

4. Dự kiến kết quả đạt được:

- Hoàn thành các hình ảnh thiết kế về giao diện người dùng và trải nghiệm người dùng.
- Xây dựng hoàn tất sơ bộ website, đi kèm với các chức năng của website đã liệt kê ở trên.
- Tìm ra phân khúc phù hợp tích hợp công nghệ AI.
- Mở rộng dự án và có thể tiến hành đưa vào thực tế.

LỜI CẢM ƠN

Sau hơn nhiều tháng học tập và nghiên cứu đề tài, nhóm chúng em đã được mở mang kiến thức và được hiểu biết nhiều hơn về lĩnh vực công nghệ phần mềm qua sự quan tâm, hướng dẫn và chia sẽ tận tình của Thầy Hoàng Văn Hiếu. Nhờ sự quan tâm và hướng dẫn của Thầy, nhóm em đã có cơ hội tiếp cận với những khái niệm và kỹ năng quan trọng, từ đó phát triển và nâng cao khả năng chuyên môn của mình.

Với lòng cảm ơn sâu sắc nhất, nhóm em xin gửi tới Quý Thầy/ Cô trường Đại Học Kinh Tế - Tài Chính Thành Phố Hồ Chí Minh nói chung và khoa Công Nghệ Thông Tin nói riêng, những người đã trang bị hành trang kiến thức nền tảng cho đề tài này. Đặc biệt là Thầy Hoàng Văn Hiếu đã nhiệt tình mô tả, gợi ý hướng đi, và bổ sung cho chúng em thêm nhiều kiến thức cần thiết một cách chi tiết và cụ thể nhất, đóng góp những ý kiến quý báu cho việc hoàn thành đề tài. Em chân thành cảm ơn cô Võ Thị Kim Anh, thầy Nguyễn Minh Tuấn và thầy Nguyễn Hà Giang đã đưa ra những nhận xét, những phản biện để góp phần hoàn thiện bài báo cáo và đề tài của em hơn.

Một lần nữa, nhóm em xin gửi lời cảm ơn chân thành nhất đến Thầy Hoàng Văn Hiếu, các Thầy/Cô giảng viên trong khoa Công Nghệ Thông Tin vì sự đóng góp và hỗ trợ quý báu của mọi người trong hành trình học tập và nghiên cứu của nhóm em.

Nhóm xin chân thành cảm ơn!

Mục lục:

CHAPTER 1. CƠ SỞ LÝ THUYẾT

1. KHÁI NIỆM VỀ MÔ HÌNH GIẢI ĐÁP CHO SINH VIÊN
2. CÁC PHƯƠNG PHÁP GIẢI ĐÁP THÔNG THƯỜNG
3. LỢI ÍCH CỦA ỨNG DỤNG WEBSITE TƯƠNG TÁC TRỰC TIẾP

CHAPTER 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

1. PHÂN TÍCH YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG
   1.1. Xác định yêu cầu
   1.2. Xác định ưu tiên
   1.3 Mô tả dự án với các chức năng xác định
2. HƯỚNG THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG
   2.1. Mục tiêu thiết kế giao diện người dùng
   2.2. Nguyên tắc thiết kế giao diện người dùng
   2.3. Thiết kế màu sắc và giao diện
   2.4. Tương tác người dùng
3. MÔ PHỎNG CƠ SỞ DỮ LIỆU

CHAPTER 3. KIẾN TRÚC HỆ THỐNG VÀ CÔNG NGHỆ SỬ DỤNG

1. TỔNG QUAN VỀ HỆ THỐNG
   1.1. Mô tả tổng quan về hệ thống
   1.2. Mục tiêu và yêu cầu chung của hệ thống
2. CÁC CÔNG NGHỆ SỬ DỤNG
   2.1. Phần giao diện - Frontend
   2.2. Phần server - Backend chính
   2.3. Phần server - Backend riêng cho Chatbot

CHAPTER 4. XÂY DỰNG VÀ THỬ NGHIỆM ỨNG DỤNG

CHAPTER 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KHẢ NĂNG

2. Hạn chế

3. Hướng mở rộng và phát triển

Tài liệu tham khảo

CHAPTER 1. CƠ SỞ LÝ THUYẾT

1. KHÁI NIỆM VỀ MÔ HÌNH GIẢI ĐÁP CHO SINH VIÊN

Trong lĩnh vực giáo dục, mô hình giải đáp cho sinh viên là một hệ thống hoặc khuôn khổ tổ chức có mục tiêu cung cấp sự hỗ trợ và giải đáp các thắc mắc của sinh viên. Mô hình này tập trung vào việc tạo ra một môi trường tương tác trực tiếp giữa sinh viên và nguồn thông tin hoặc người giải đáp, nhằm đáp ứng nhanh chóng và hiệu quả những câu hỏi, vấn đề mà sinh viên gặp phải trong quá trình học tập và nghiên cứu.

Mô hình giải đáp cho sinh viên thường sử dụng các công nghệ và phương pháp truyền thông hiện đại, như website tương tác trực tiếp, để tạo ra một nền tảng năng động cho việc tương tác và giao tiếp giữa sinh viên và người giải đáp. Mục đích của mô hình này là cung cấp cho sinh viên sự hỗ trợ và giải đáp thông tin nhanh chóng, chính xác, từ đó giúp họ giải quyết các vấn đề và nâng cao hiệu suất học tập của mình. Bên cạnh đó, mô hình có thể tích hợp các tính năng như hệ thống câu hỏi và trả lời, diễn đàn trực tuyến, chat trực tiếp, hoặc giao diện tương tác thông qua ứng dụng di động. Điều này cho phép sinh viên truy cập và tương tác với nguồn thông tin và người giải đáp một cách thuận tiện và linh hoạt, không bị ràng buộc bởi thời gian và không gian.

2. CÁC PHƯƠNG PHÁP GIẢI ĐÁP THÔNG THƯỜNG

Trong quá trình giải đáp thắc mắc của sinh viên, có một số phương pháp thông thường được sử dụng để cung cấp sự hỗ trợ và giải đáp thông tin. Dưới đây là một số phương pháp phổ biến:

- Tham khảo tài liệu giáo trình: Sinh viên thường tìm kiếm câu trả lời trong sách giáo trình hoặc tài liệu học tập cung cấp bởi giảng viên. Đây là một phương pháp tự học thông thường, tuy nhiên, có thể đòi hỏi thời gian và cần sự hiểu biết để áp dụng kiến thức vào vấn đề cụ thể.
- Hỏi giảng viên hoặc trợ giảng: Sinh viên có thể tìm đến giảng viên hoặc trợ giảng để trao đổi và nhận được giải đáp trực tiếp từ người có kiến thức chuyên môn. Phương pháp này thường được sử dụng trong giờ học, buổi hướng dẫn hoặc thông qua việc hẹn lịch gặp riêng.

- Trao đổi với bạn bè: Sinh viên có thể thảo luận và trao đổi với bạn bè cùng khóa để giải quyết các thắc mắc. Nhờ vào sự chia sẻ kiến thức và kinh nghiệm, phương pháp này có thể mang lại lợi ích bổ sung và quan điểm đa dạng trong quá trình giải đáp.

Hình 1: Các sinh viên UEF thảo luận bài học cùng nhau

3. LỢI ÍCH CỦA ỨNG DỤNG WEBSITE TƯƠNG TÁC TRỰC TIẾP

Ứng dụng website tương tác trực tiếp trong mô hình giải đáp cho sinh viên mang lại nhiều lợi ích quan trọng:

- Tính tiện lợi và linh hoạt: Sinh viên có thể truy cập vào website tương tác trực tiếp từ bất kỳ thiết bị nào có kết nối internet, không bị ràng buộc bởi các giờ giới hạn hoặc địa điểm cụ thể. Điều này mang lại sự linh hoạt và tiện lợi cho sinh viên, cho phép họ truy cập và nhận được giải đáp mọi lúc, mọi nơi.
- Tận dụng công nghệ: Ứng dụng website tương tác trực tiếp sử dụng công nghệ để tạo ra một môi trường tương tác đa phương tiện. Sinh viên có thể sử dụng các tính năng như trò chuyện trực tiếp, gửi câu hỏi, chia sẻ tài liệu và thảo luận trực tuyến. Điều này giúp tăng cường trải nghiệm học tập và tạo ra một cách tiếp cận hiện đại và sáng tạo đối với việc giải quyết thắc mắc.
- Tiết kiệm thời gian và nỗ lực: Thay vì phải di chuyển đến nơi gặp mặt hoặc chờ đợi lịch hẹn, sinh viên có thể truy cập vào website tương tác trực tiếp và nhận được giải đáp ngay lập tức. Điều này giúp tiết kiệm thời gian và nỗ lực, cho phép sinh viên tập trung vào việc học tập và giải quyết các vấn đề khác trong cuộc sống.

- Ghi nhận và chia sẻ kiến thức: Ứng dụng website tương tác trực tiếp có thể lưu trữ và ghi nhận các câu hỏi và câu trả lời quan trọng. Sinh viên có thể dễ dàng tra cứu và truy cập lại thông tin đã được giải đáp trước đó.

Tổng quan, ứng dụng website tương tác trực tiếp trong giải đáp thông thường mang lại lợi ích vượt trội về tính tiện lợi, tương tác đa dạng và sử dụng công nghệ. Nó tạo ra một môi trường học tập hiện đại và linh hoạt, hỗ trợ sinh viên trong việc giải quyết thắc mắc và nâng cao hiệu suất học tập.

CHAPTER 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

1. PHÂN TÍCH YÊU CẦU CHỨC NĂNG CỦA HỆ THỐNG

Phân tích yêu cầu chức năng của hệ thống là một bước quan trọng trong quá trình thiết kế và triển khai hệ thống. Nó đảm bảo rằng hệ thống sẽ đáp ứng được các nhu cầu và mục tiêu của người sử dụng. Dưới đây là các giai đoạn chính trong quá trình phân tích yêu cầu chức năng:

1.1. Xác định yêu cầu

Trong giai đoạn này, nhóm phân tích yêu cầu tương tác với các bên liên quan để hiểu rõ các yêu cầu và mục tiêu của hệ thống. Điều này có thể bao gồm việc tiếp xúc trực tiếp với người dùng cuối, quản lý dự án, nhà phát triển và các bên liên quan khác để thu thập thông tin chi tiết về các chức năng, quy trình và yêu cầu kỹ thuật.

Trước khi tiến hành phát triển ứng dụng, nhóm đã đi thu thập được thông tin về nhu cầu hay mong muốn của người dùng cuối (sinh viên UEF) như sau:

Trò chuyện hay hẹn gặp trực tiếp là hình thức thường khá mất thời gian và có thể tốn thời gian đi lại, trao đổi. Phương thức này đã quá quen thuộc và đa phần người dùng thường không thích cách này, họ muốn giải quyết vấn đề ngay lập tức hoặc tiện lợi nhất có thể.

Bên cạnh đó, nhận thấy nhu cầu người dùng cần nguồn chia sẻ tài liệu mỗi khi đến kì thi hay bài kiểm tra, họ cần tài liệu để có thể ôn thi, một nơi mà sinh viên có thể dễ dàng tham khảo và chọn lọc tài liệu. Họ cũng muốn xem được tin tức mới từ giảng viên, khoa hay các phòng ban của mình để cập nhật thời gian sự kiện workshop, event hay có các thông báo mới được đăng tải.

Sau quá trình phân tích, nhóm đã tổng hợp lại các yêu cầu cần có như giao tiếp trực tiếp, ứng dụng dễ sử dụng, đẹp mắt, có thể tham khảo được bài viết hay thông tin từ các khoa, phòng ban. Tích hợp nguồn tài liệu có thể chia sẻ và công khai cho toàn bộ người dùng để mỗi khi đến kì thi hay bài kiểm tra có thể tham khảo, từ những ý tưởng đó nhóm đã tổng hợp lại và tiến hành đưa vào kế hoạch phát triển ứng dụng

Nhóm cũng đã nhận thấy được một vấn đề phát sinh sau quá trình tham khảo ý kiến của thầy cô và các bạn sinh viên UEF, nếu như vào ngày thứ 7 hoặc chủ nhật, hoặc cũng có thể là ngoài giờ làm việc, thầy cô sẽ không có mặt hay bận vào những ngày đó, vì vậy sinh

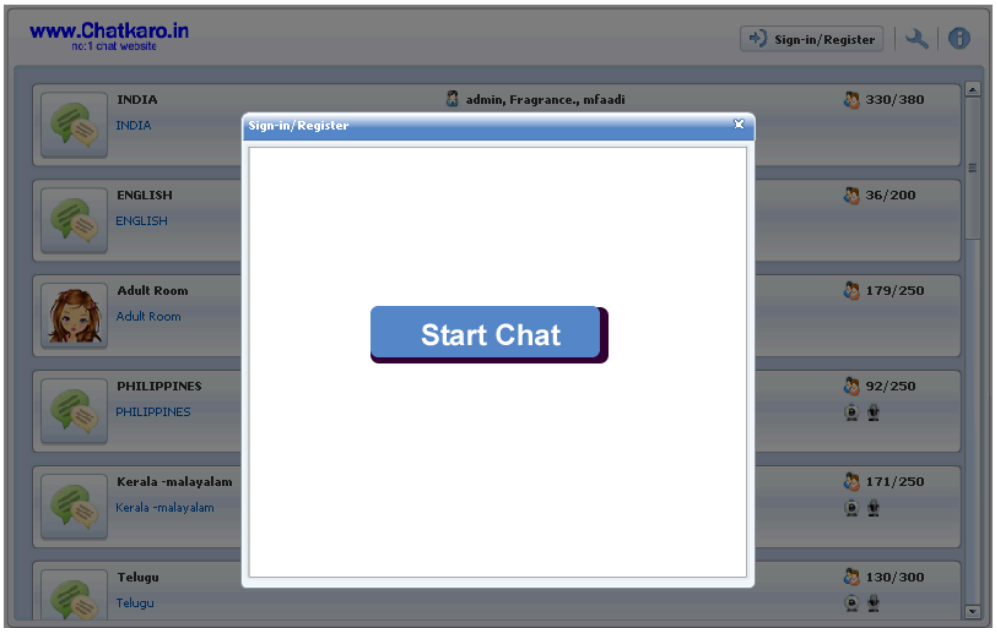
viên sẽ lại phải chờ đến thứ 2 hoặc vào giờ làm việc. Việc tích hợp 1 chatbot về những thông tin hay câu hỏi thường được hỏi bởi các bạn sinh viên sẽ giúp ứng dụng trở nên hữu ích hơn..

1.2. Xác định ưu tiên

Trước khi tiến hành triển khai, các yêu cầu chức năng cần được xác định ưu tiên để đảm bảo rằng các yêu cầu quan trọng nhất được ưu tiên cao hơn. Điều này giúp định rõ phạm vi và mục tiêu của dự án và nhóm trong các giai đoạn tiếp theo.

Đầu tiên là mục tiêu chính có thể là cung cấp một ứng dụng có chức năng chức năng giúp giao tiếp tương tác hiệu quả, thuận lợi giữa người dùng và các khoa, phòng ban liên quan, sau đó có thể là cung cấp khả năng chia sẻ tài liệu và xem các bài viết.

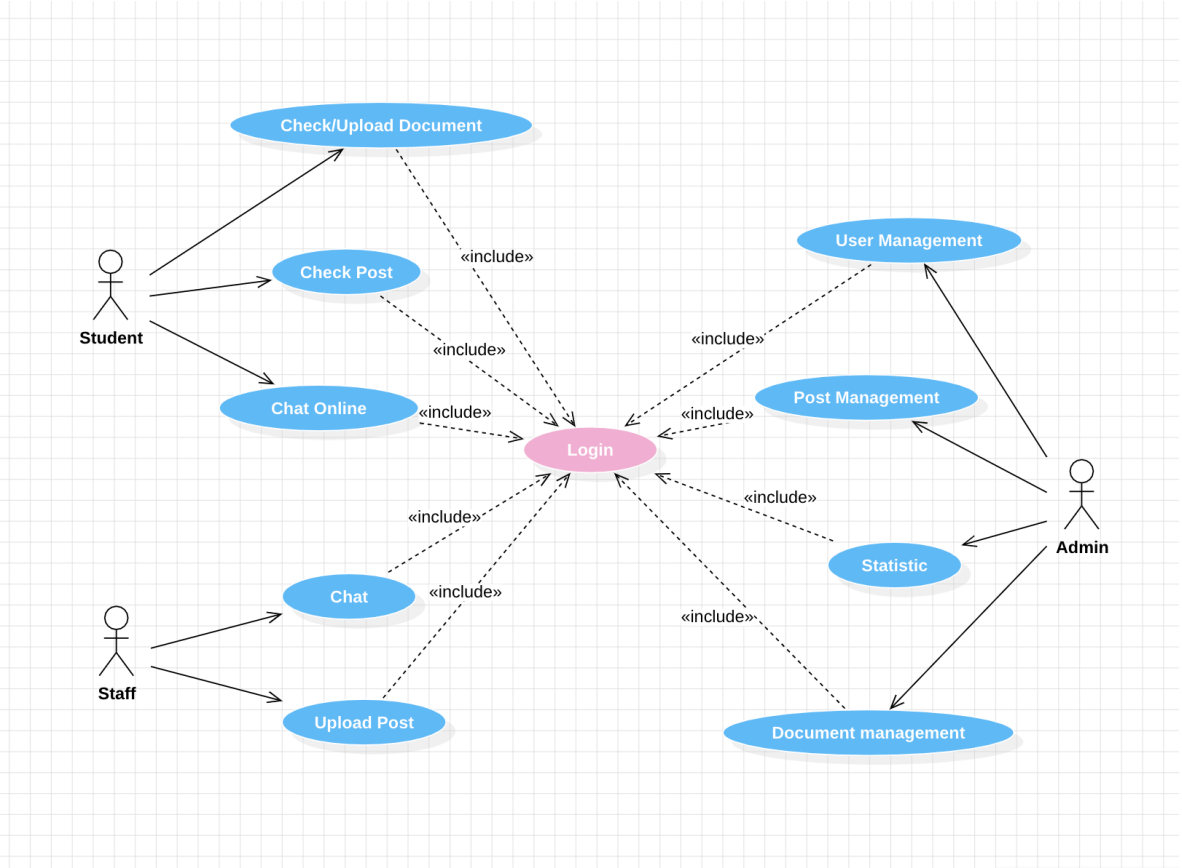
Dựa trên mục tiêu chính đã xác định, đánh giá mức độ ưu tiên của các chức năng Trong trường hợp này, với chức năng giao tiếp tương tác hiệu quả được xem là ưu tiên hàng đầu, nhóm xác định nó là mức ưu tiên cao nhất. Trong khi đó, chức năng chia sẻ tài liệu và xem bài viết có thể được xem là ưu tiên thấp hơn. Và như ở trên nhóm em đã trình bày, chatbot được bắt đầu nghiên cứu và thu thập dữ liệu để đưa vào thử nghiệm.



Hình 2: Chức năng tương tác trực tiếp được ưu tiên trước

1.3. Mô tả dự án với các chức năng xác định

Nhóm đã mô tả dự án với các chức năng chính cho mỗi người dùng qua sơ đồ use-case:



Hình 3: Sơ đồ Use-case với các actor chính

2. Hướng thiết kế giao diện người dùng

2.1. Mục tiêu thiết kế giao diện người dùng

Mục tiêu chung: Tạo ra một giao diện thân thiện, dễ sử dụng và hấp dẫn cho người dùng.

Mục tiêu cụ thể: Cung cấp trải nghiệm người dùng tốt, tăng tính tương tác và thuận tiện trong việc sử dụng ứng dụng.

2.2. Nguyên tắc thiết kế giao diện người dùng

Nguyên tắc 1: Đơn giản và rõ ràng.

Nguyên tắc 2: Nhất quán trong cả cấu trúc và giao diện.

Nguyên tắc 3: Trực quan và dễ hiểu.

Nguyên tắc 4: Tối ưu hóa trải nghiệm người dùng.

2.3. Thiết kế màu sắc và giao diện

Thiết kế màu sắc và giao diện:

Màu chủ đạo: [#990000] (sử dụng màu chủ đạo của UEF)

Màu nền: [#fff]

Màu sắc cho các thành phần: [#fff]

Kiểu chữ: [Times New Roman]

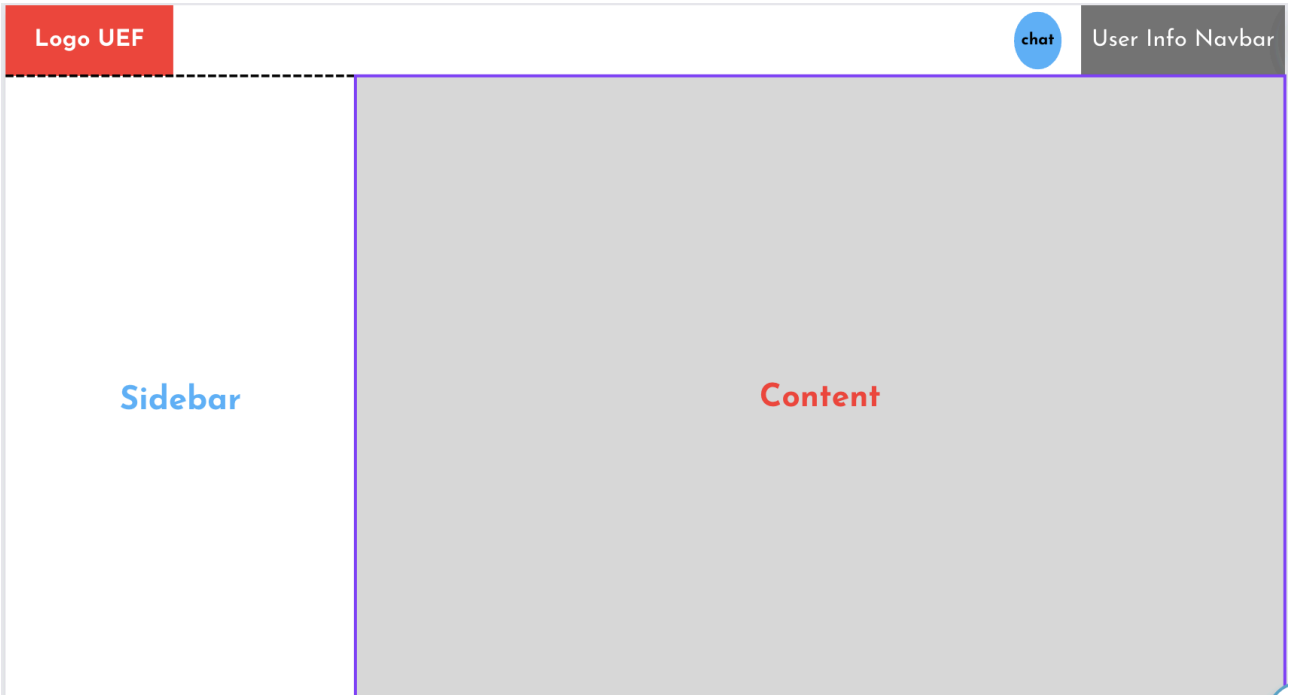
Cỡ chữ: [14]

2.4. Tương tác người dùng

Phản hồi giao diện: Hiệu ứng hover, nhấp chuột, trạng thái tương tác, v.v.

Cách thức tương tác: Nhập liệu, đăng nhập, tải dữ liệu, hình ảnh, v.v.

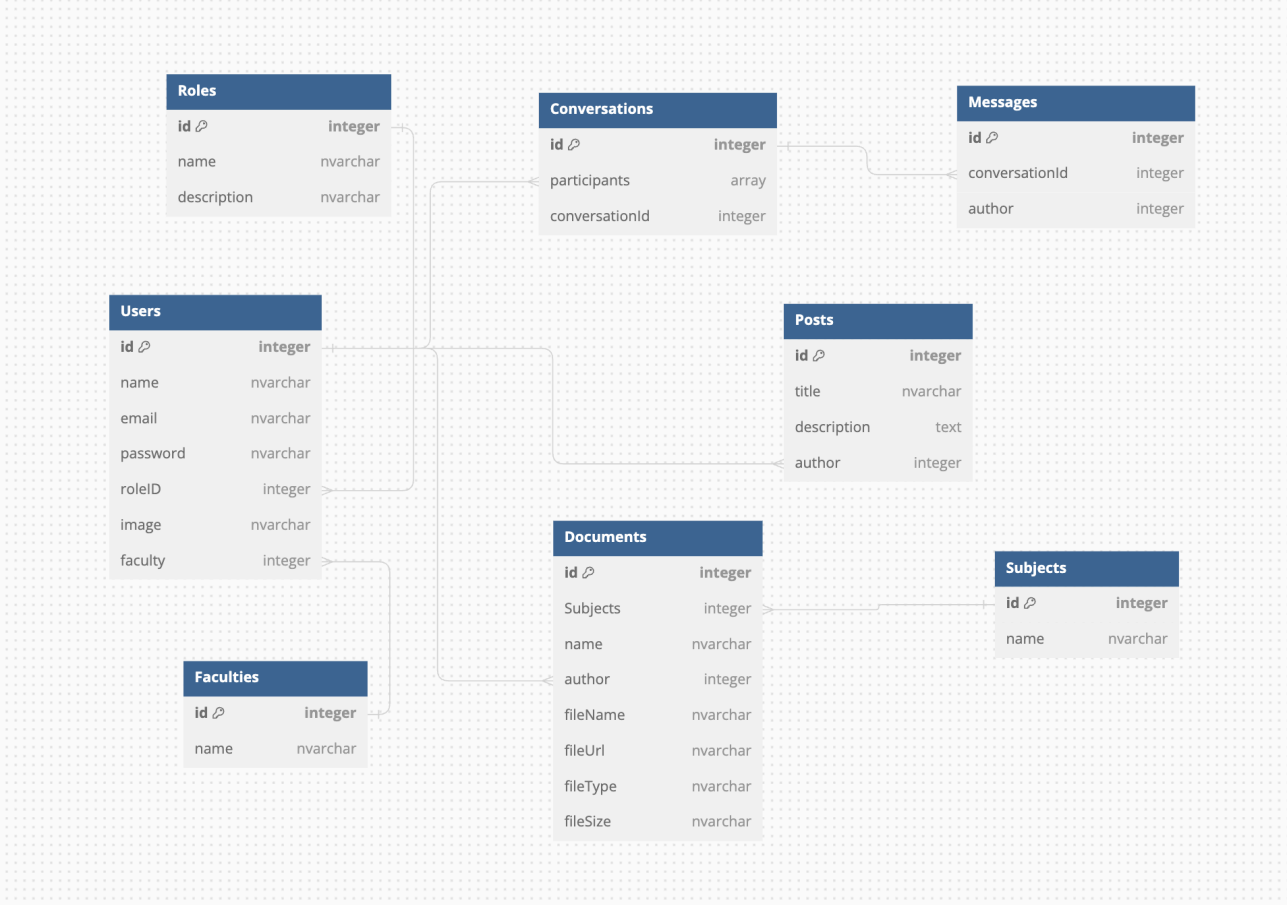
Mô phỏng giao diện người dùng:



Hình 4: Mô phỏng giao diện người dùng

3. MÔ PHỎNG CƠ SỞ DỮ LIỆU

Cơ sở dữ liệu gồm các bảng chính như Users, Roles, Conversations,..



Hình 5: Mô phỏng cơ sở dữ liệu

CHAPTER 3. KIẾN TRÚC HỆ THỐNG VÀ CÔNG NGHỆ SỬ DỤNG

1. TỔNG QUAN VỀ HỆ THỐNG

1.1. Mô tả tổng quan về hệ thống

Hệ thống mà nhóm xây dựng bao gồm Frontend được xây dựng bằng React Vite, cung cấp giao diện người dùng tương tác. Phần Backend được nhóm chia làm 2 phần, phần chính cũng như các chức năng chính của hệ thống xây dựng bằng NodeJS và ExpressJS, chịu trách nhiệm xử lý các yêu cầu và cung cấp API cho phía frontend, phần Backend thứ 2 cũng là chatbot được xây dựng bằng Python, chịu trách nhiệm xử lý các câu hỏi đến từ sinh viên, sau đó sử dụng Underthesea NLP Toolkit để phân tích từ thành các nhãn, truy vấn tới cơ sở dữ liệu, lấy dữ liệu được truy vấn và với câu hỏi ban đầu, sử dụng BERT để tự động tạo ra câu trả lời. Nhóm cũng sử dụng 2 cơ sở dữ liệu để thực hiện đề tài, đầu tiên là MongoDB, sử dụng ngôn ngữ truy vấn Mongoose để truy vấn và lưu trữ dữ liệu người dùng, tin nhắn, tài liệu và bài viết, cơ sở dữ liệu còn lại là Neo4j, sử dụng ngôn ngữ truy vấn Cypher để truy vấn và lưu trữ những câu hỏi, những thông tin hay chủ đề về trường UEF. Sử dụng nền tảng đám mây Cloudinary lưu trữ file và ảnh. Giao tiếp thời gian thực giữa client và server được thiết lập bằng Socket.IO, cho phép truyền tải dữ liệu theo thời gian thực và cập nhật trạng thái nhanh chóng.

1.2. Mục tiêu và yêu cầu chung của hệ thống

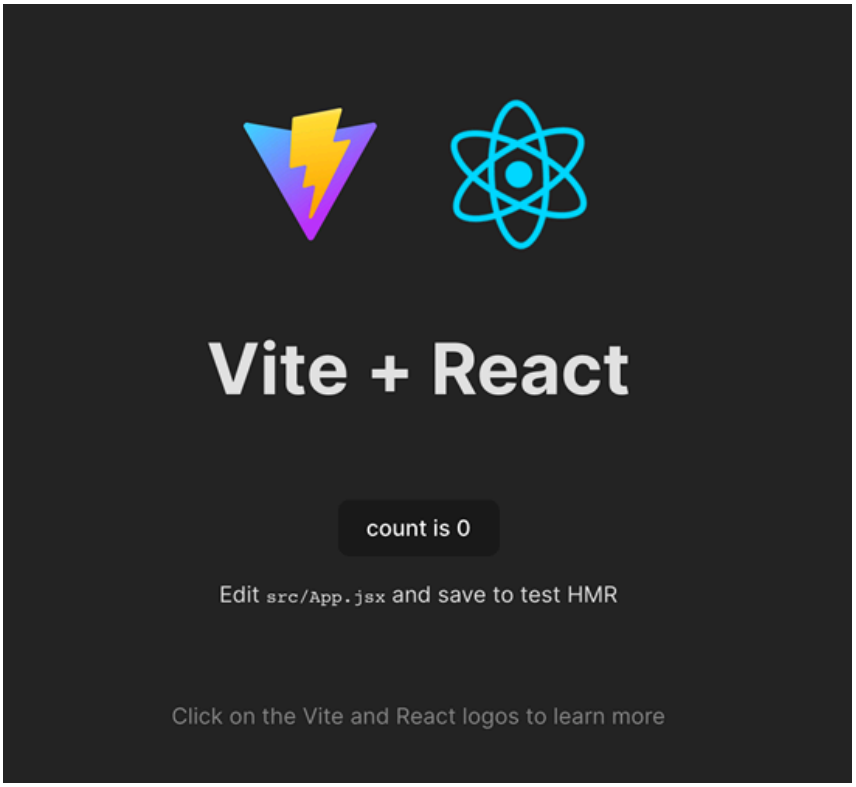
Mục tiêu của hệ thống nhóm là cung cấp một trải nghiệm giao tiếp tương tác trực tiếp, tải lên tài liệu và bài viết tốt cho người dùng. Hệ thống được nhóm xác định phải đảm bảo hiệu suất cao và khả năng mở rộng để có thể xử lý đồng thời nhiều yêu cầu từ người dùng. Đồng thời, hệ thống phải đảm bảo tính bảo mật cao, bao gồm xác thực người dùng và quản lý quyền truy cập. Trải nghiệm người dùng là một yếu tố quan trọng, vì vậy giao diện người dùng phải được thiết kế dễ sử dụng và hấp dẫn.

2. CÁC CÔNG NGHỆ SỬ DỤNG

2.1. Phần giao diện - Frontend

2.1.1. React Vite

React Vite là một công nghệ được sử dụng để xây dựng phần frontend của hệ thống. Nó là một công cụ phát triển hiệu suất cao, tập trung vào tăng tốc độ phát triển và trải nghiệm phát triển tốt nhất. React Vite kết hợp giữa React và Vite - một công cụ xây dựng ứng dụng phát triển nhanh (fast development tool) cho JavaScript và TypeScript. Với Vite, việc khởi tạo dự án và tải lại ứng dụng trở nên cực kỳ nhanh chóng nhờ cơ chế Hot Module Replacement (HMR). Điều này cho phép nhóm tập trung vào việc lập trình mà không mất thời gian chờ đợi quá lâu để xem kết quả thay đổi.



Hình 6: React + Vite

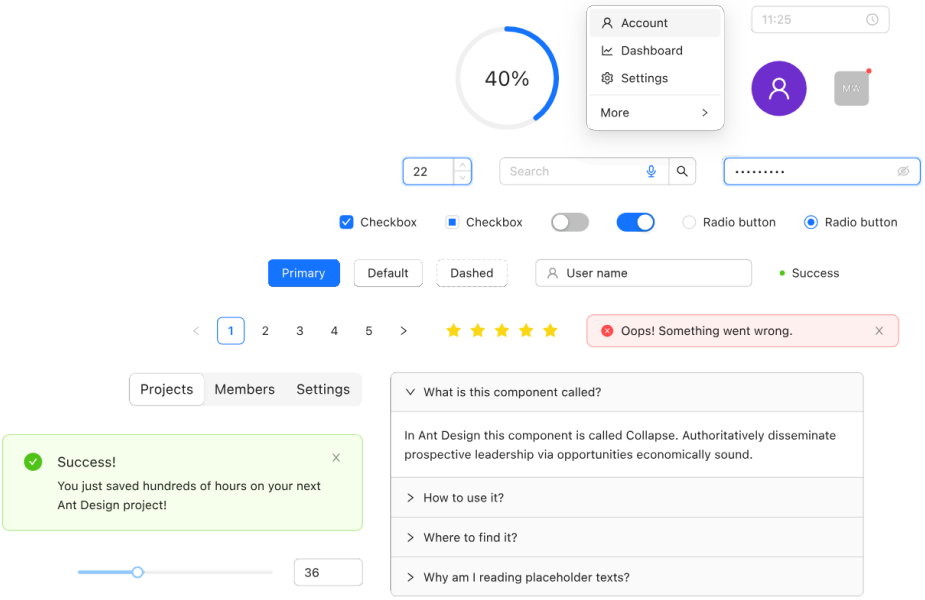
2.1.2. Redux Toolkit

Redux Toolkit là một thư viện hỗ trợ quản lý trạng thái ứng dụng trong React. Nó cung cấp một cách tiếp cận đơn giản và hiệu quả để làm việc với Redux, giảm bớt độ phức tạp và lặp lại khi làm việc với Redux truyền thống. Redux Toolkit giúp nhóm xây dựng các

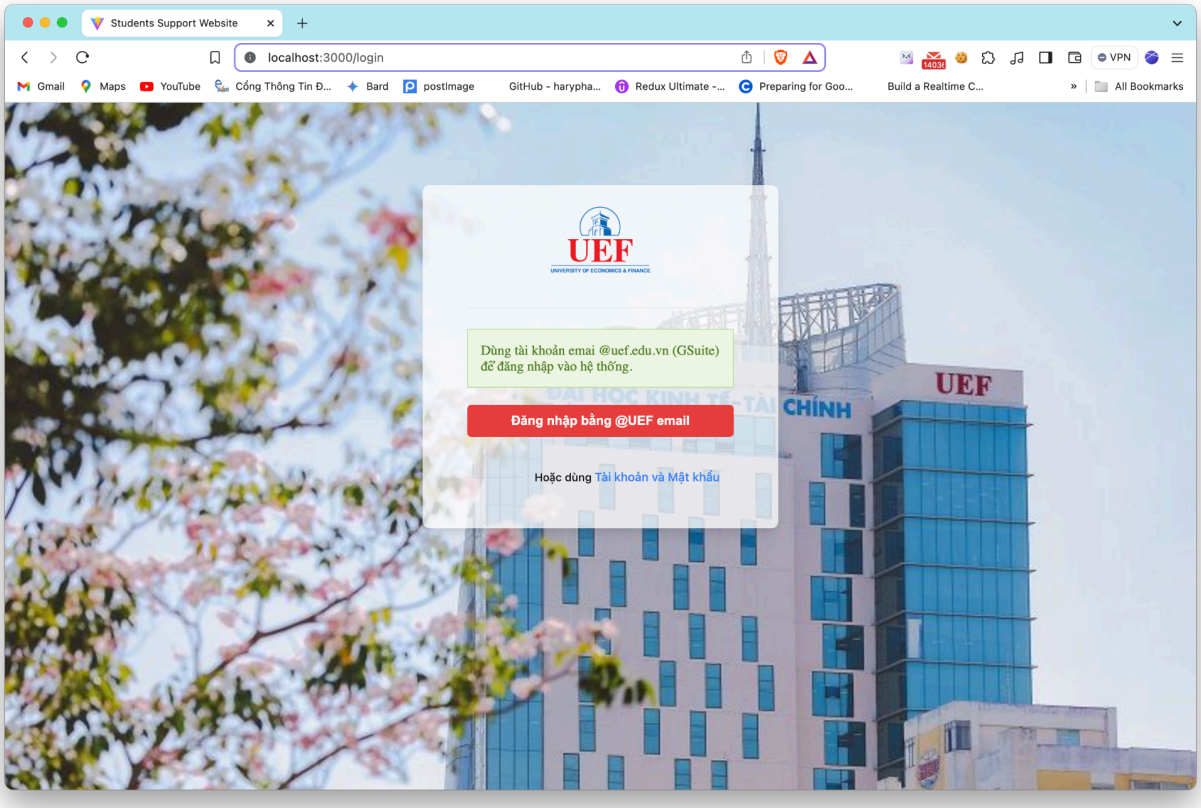
store Redux một cách dễ dàng thông qua API tiện ích như createSlice và createAsyncThunk, giúp việc định nghĩa reducers và actions trở nên ngắn gọn và dễ đọc hơn. Nó cũng tích hợp sẵn Redux DevTools Extension, giúp nhóm theo dõi và gỡ lỗi trạng thái ứng dụng một cách dễ dàng. Nhóm đã sử dụng để theo dõi trạng thái đăng nhập và tải lại các cuộc trò chuyện trước đó cho người dùng.

2.1.3. Ant Design

Cuối cùng là Ant Design, một thư viện giao diện người dùng (UI) đẹp và chuyên nghiệp cho React. Nó cung cấp một bộ sưu tập các thành phần UI đã được thiết kế sẵn, giúp nhóm xây dựng giao diện người dùng nhanh chóng và dễ dàng. Ant Design đã được tối ưu hóa cho việc phát triển ứng dụng React, với mục tiêu cung cấp trải nghiệm người dùng tốt nhất và tuân thủ các nguyên tắc thiết kế UI tốt nhất. Thư viện này cung cấp rất nhiều thành phần UI, bao gồm nút, biểu mẫu, bảng, menu, thanh điều hướng và rất nhiều thành phần khác. Bằng cách sử dụng Ant Design, nhóm có thể tạo ra giao diện người dùng chuyên nghiệp, tương thích với các trình duyệt và thiết bị khác nhau, mà không cần tiêu tốn nhiều thời gian và công sức trong việc thiết kế từ đầu.



Hình 7: Ant Design giúp xây dựng UI nhanh chóng hơn



Hình 8: Nhóm đã sử dụng Ant Design để có thể thiết kế giao diện nhanh chóng hơn

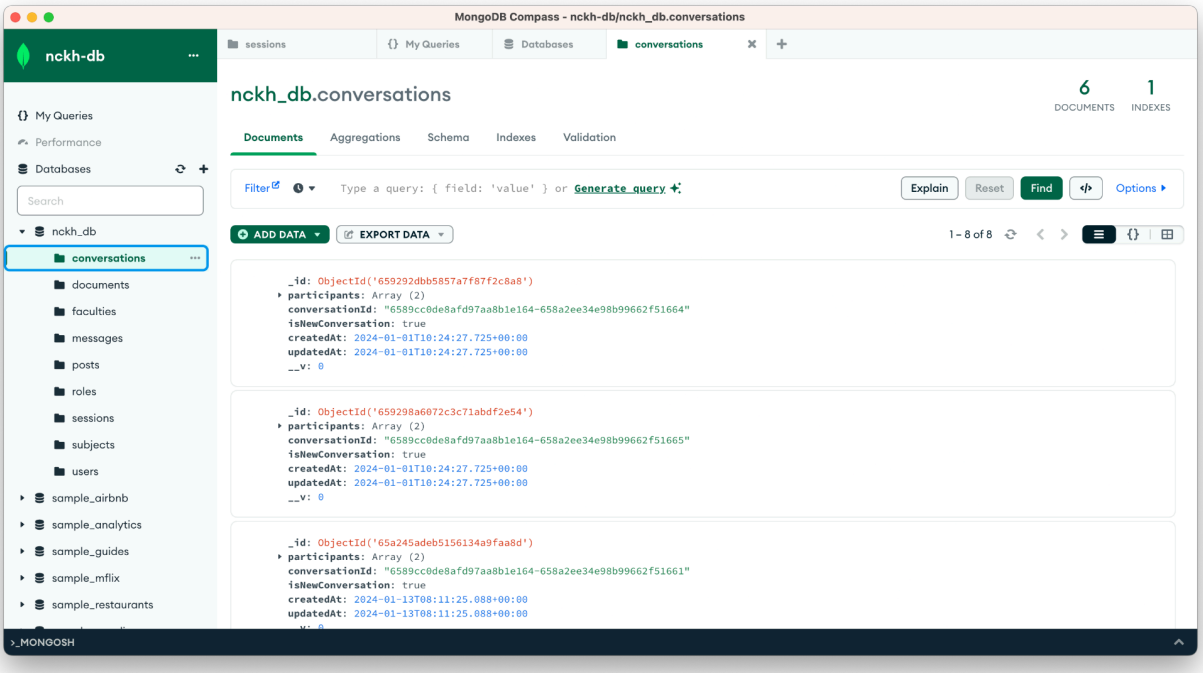
2.2. Phần server - Backend chính

2.2.1. NodeJS và ExpressJS

NodeJS là một môi trường chạy mã JavaScript phía máy chủ và ExpressJS là một framework web xây dựng trên NodeJS đã quá quen thuộc với đa số developer nên nhóm quyết định sử dụng để xây dựng ứng dụng backend. Với NodeJS, việc viết mã JavaScript trên cả frontend và backend trở nên thống nhất và dễ dàng. Node.js có kiến trúc dựa trên sự kiện (event-driven) và không chặn (non-blocking), cho phép xử lý đồng thời hàng ngàn kết nối, giúp ứng dụng xử lý tải cao một cách hiệu quả. Bên cạnh đó ExpressJS cung cấp các công cụ và tính năng để xử lý yêu cầu HTTP, định tuyến (routing), quản lý middleware và tạo các API RESTful. Nó là một framework mạnh mẽ, linh hoạt và dễ sử dụng để xây dựng backend cho ứng dụng web.

2.2.2. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) và nó được sử dụng làm cơ sở dữ liệu cho dự án. Nhóm sử dụng ngôn ngữ truy vấn Mongoose để tương tác với MongoDB từ ứng dụng của mình. Mongoose là một thư viện ODM (Object Data Modeling) cho MongoDB, cung cấp các cơ chế để định nghĩa các mô hình dữ liệu, tạo, đọc, cập nhật và xóa dữ liệu trong MongoDB một cách dễ dàng.



Hình 9: Hệ quản trị cơ sở dữ liệu phi quan hệ MongoDB

2.2.3. Mongoose

Mongoose là một thư viện mô hình hóa đối tượng cho NodeJS và MongoDB, được thiết kế để giúp nhóm dễ dàng và linh hoạt tương tác với cơ sở dữ liệu MongoDB. Được xây dựng trên ngôn ngữ JavaScript, Mongoose cung cấp cho nhóm một cách tiếp cận đơn giản và đáng tin cậy để quản lý và tương tác với các bản ghi trong MongoDB.

Dưới đây là một số đặc điểm và lợi ích chính của Mongoose:

Mô hình hóa đối tượng: Mongoose cho phép nhóm mô hình hóa các đối tượng dữ liệu của mình thành các đối tượng JavaScript. Bên cạnh đó cũng có thể định nghĩa các lược đồ (schema) để xác định kiểu dữ liệu, quy tắc và quan hệ giữa các trường dữ liệu.

```
const postSchema = new mongoose.Schema({
    title: {
        type: String,
        required: true},
    description: {
        type: String,
        required: true },
    author: {
        type: Schema.Types.ObjectId,
        ref: 'User',
        required: true }
}, { timestamps: true });
const Post = mongoose.model('Post', postSchema);
module.exports = Post;
```

Hình 10: Lược đồ (schema) của Post

Tương tác dễ dàng với MongoDB: Mongoose cung cấp các phương thức và API dễ sử dụng để thực hiện các hoạt động cơ bản như tìm kiếm, thêm, cập nhật và xóa dữ liệu trong MongoDB. Nhóm sử dụng các phương thức như find, findOne, save, update, remove,... để tương tác với cơ sở dữ liệu một cách đơn giản và tiện lợi.

Populating (Kết hợp dữ liệu): Mongoose cung cấp tính năng Populating cho phép nhóm có thể kết hợp dữ liệu từ nhiều bảng (collection) trong MongoDB. Bằng cách sử dụng phương thức populate, lấy dữ liệu liên quan từ các bảng khác nhau và đồng thời truy xuất thông tin chi tiết về các quan hệ giữa chúng.

```
let posts = await Post.find({
        author: data.authorId,
        _id: { $lt: data.lastPostId }
    })
        .populate('author', 'name image')
        .sort({ createdAt: -1 })
        .limit(limitNumberOfFacultyPost)
        .select({
            __v: 0,
            __t: 0,
        });
```

Hình 11: Sử dụng populate để có thể lấy ra dễ dàng 2 trường name và image của author

2.2.4. Cloudinary

Cloudinary là một dịch vụ lưu trữ hình ảnh và quản lý tài nguyên đa phương tiện trên đám mây. Nhóm đã sử dụng thư viện multer-storage-cloudinary để tương tác với Cloudinary từ backend. Thư viện này giúp nhóm lưu trữ, quản lý các tệp tin và hình ảnh tải lên từ các file trong tin nhắn, hình ảnh của người dùng hay các file tài liệu tại Document-Sharing được tải lên trên Cloudinary một cách dễ dàng và hiệu quả.

Node.js image upload

The following Node.js method uploads an image to the cloud:

Node.js

```
function upload(file, options).then(callback)
```

For example, uploading a local image file named 'my_image.jpg':

Node.js

```
cloudinary.v2.uploader
  .upload("/home/my_image.jpg")
  .then(result=>console.log(result));
```

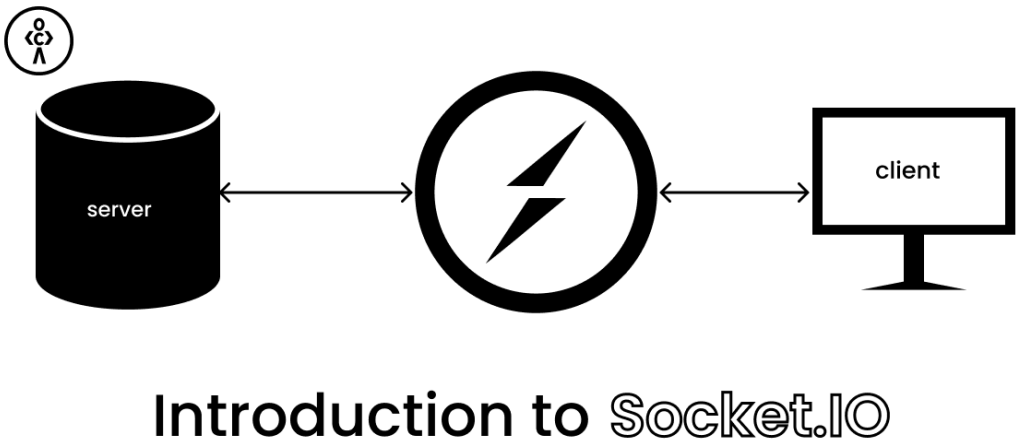
Hình 12: Upload cơ bản với NodeJS

2.2.5. Passport và Google OAuth

Passport là một middleware xác thực phổ biến trong NodeJS và được sử dụng để xác thực và quản lý quá trình đăng nhập trong dự án của nhóm. Passport hỗ trợ nhiều chiến lược xác thực khác nhau, bao gồm Google OAuth. Google OAuth là một phương thức xác thực dựa trên OAuth 2.0 sử dụng tài khoản Google. Với Google OAuth, người dùng có thể đăng nhập vào ứng dụng bằng tài khoản Google của họ, mà không cần tạo ra một cặp tên người dùng/mật khẩu mới. Việc sử dụng Passport và Google OAuth giúp bảo mật và thuận tiện cho người dùng, đồng thời cung cấp cơ chế xác thực mạnh mẽ cho ứng dụng. Nhóm đã thêm các parameter để giới hạn người dùng chỉ có thể đăng nhập bằng email có định dạng uef.edu.vn.

2.2.6. Socket.IO

Socket.IO là một thư viện JavaScript mạnh mẽ cho phép giao tiếp hai chiều (bidirectional) và thời gian thực giữa máy chủ và máy khách trong ứng dụng web. Nó cung cấp một cơ chế giao tiếp thông qua các kết nối socket, cho phép truyền dữ liệu tức thì và gửi sự kiện từ máy chủ đến tất cả các máy khách đã kết nối. Với Socket.IO, nhóm có thể xây dựng các tính năng như chat trực tiếp, cập nhật dữ liệu tức thì và thông báo thời gian thực, đem đến tính năng cũng như chức năng được ưu tiên nhất của mô hình một cách dễ dàng hơn. Thư viện này hỗ trợ nhiều trình duyệt và nền tảng, giúp tạo ra trải nghiệm giao tiếp mượt mà và nhanh chóng cho người dùng.



Hình 13: Socket.IO là một thư viện JS cho phép giao tiếp hai chiều

Client

```
socket.emit('hello', 'world');
```

Server

```
io.on('connection', (socket) => {
  socket.on('hello', (arg) => {
    console.log(arg); // 'world'
  });
});
```

Hình 14: Cơ bản về socket với phương thức emit truyền data từ client

2.3. Phần server - Backend riêng cho Chatbot UEF

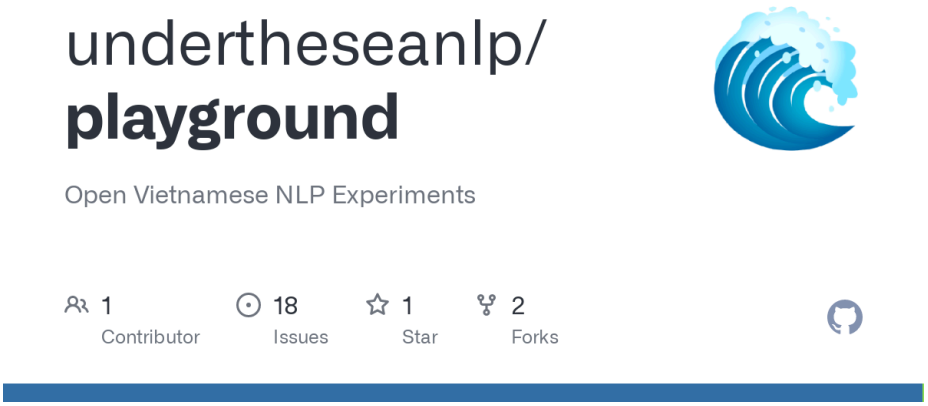
2.3.1. Python và Flask

Python là ngôn ngữ lập trình mạnh mẽ và dễ học. Nó được sử dụng trong phần backend của Chatbot UEF. Flask là một framework web được xây dựng bằng Python. Nó giúp xây dựng các ứng dụng web nhanh chóng và linh hoạt. Flask được sử dụng để xây dựng API cho Chatbot UEF, cho phép giao tiếp với phần frontend, cung cấp dữ liệu và chức năng cho ứng dụng.

2.3.2. BeautifulSoup và Underthesea NLP Toolkit

BeautifulSoup là một thư viện Python phổ biến được sử dụng để phân tích và trích xuất dữ liệu từ các tài liệu HTML và XML. Nó cung cấp các phương pháp đơn giản và linh hoạt để tạo ra cây cú pháp từ tài liệu và điều hướng qua các phần tử trong cây để tìm kiếm và trích xuất thông tin. Cũng là bước tiền xử lý để làm sạch dữ liệu, sau đó sử dụng để tổ chức lưu trữ vào cơ sở dữ liệu.

Underthesea NLP Toolkit là một bộ công cụ xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) dành cho tiếng Việt. Được phát triển dựa trên Python, Underthesea cung cấp các tính năng và chức năng để phân tích, xử lý và trích xuất thông tin từ văn bản tiếng Việt.

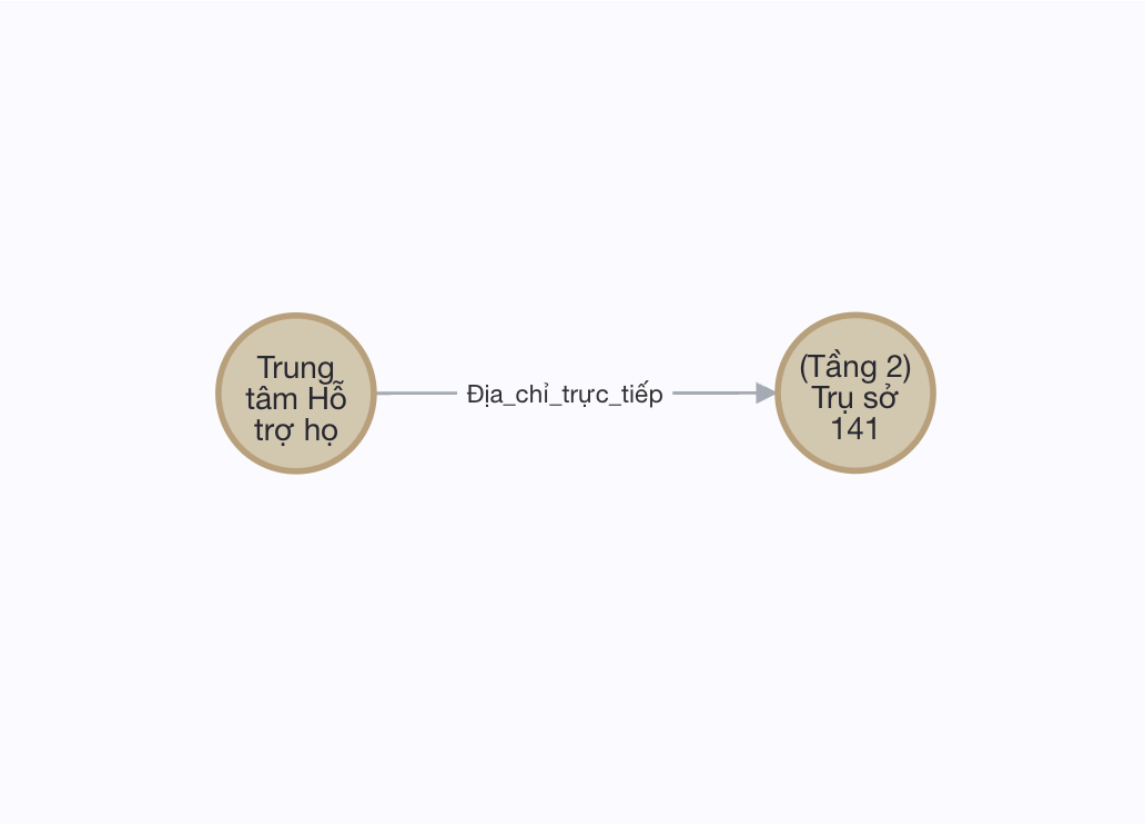


Hình 15: Sử dụng Underthesea NLP Toolkit để nhận dạng từ khoá đầu vào

2.3.3. Neo4j và Ngôn ngữ truy vấn Cypher

Neo4j là một hệ quản trị cơ sở dữ liệu đồ thị. Nó cho phép lưu trữ và truy vấn dữ liệu dưới dạng đồ thị, với các nút và mối quan hệ giữa chúng. Neo4j là hệ quản trị cơ sở dữ liệu theo dạng cây. Neo4j không dùng câu truy vấn thông thường để truy xuất như các cơ sở dữ liệu truyền thống, mà thay vào đó Neo4j sẽ sử dụng một dạng truy vấn gọi là Cypher.

Ngôn ngữ truy vấn Cypher là ngôn ngữ truy vấn đặc biệt được sử dụng để truy vấn dữ liệu trong Neo4j. Nó cho phép truy vấn và tìm kiếm các mối quan hệ và thông tin trong cơ sở dữ liệu đồ thị. Cypher có cấu trúc tương tự như SQL, các câu query được xây dựng từ nhiều mệnh đề khác nhau. Các mệnh đề là chuỗi liên kết với nhau, cấu trúc xây dựng dựa trên ngôn ngữ Tiếng Anh với ngữ nghĩa thuận tiện cho người thao tác với ngôn ngữ, điều này giúp cho việc viết và đọc các câu query cũng dễ dàng hơn. Một số mệnh đề sử dụng trong cơ sở dữ liệu đồ thị như match, where, return, create, set, remove, merge,…



Hình 16: Ví dụ cơ bản về 2 node liên kết trong cơ sở dữ liệu đồ thị

Để xây dựng đồ trị tri thức cũng như là nơi lưu trữ toàn bộ cơ sở dữ liệu của hệ thống Chatbot liên quan đến thông tin trường UEF hay chủ yếu là những câu hỏi của sinh viên UEF dưới dạng đồ thị.

2.3.4. BERT

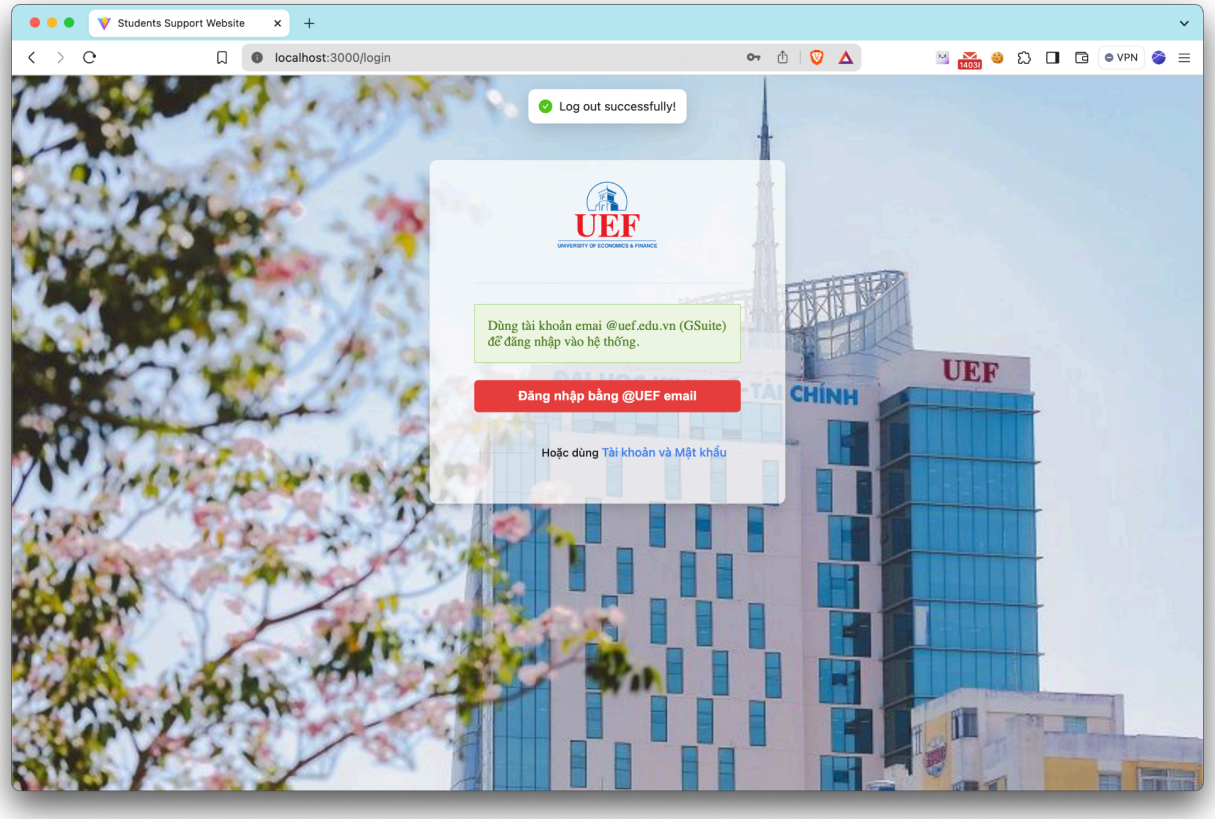
BERT (Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ dựa trên mạng neural transformer và là một trong những tiến bộ quan trọng nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). BERT đã đạt được những thành tựu đáng kể trong các tác vụ như phân loại văn bản, trích xuất thông tin và hiểu ý nghĩa của văn bản.

BERT là một mô hình học sâu đặc biệt được đào tạo trên một lượng lớn dữ liệu văn bản. Mô hình BERT sử dụng kiến trúc transformer, một loại mạng neural đặc biệt trong việc xử lý dữ liệu tuần tự. Mô hình BERT có khả năng hiểu được ngữ cảnh và sự phụ thuộc từ ngữ hai chiều trong văn bản nhờ vào kiến trúc transformer. Điều này cho phép BERT có khả năng xử lý các tác vụ NLP phức tạp như hiểu ý nghĩa của văn bản, tạo câu hỏi và trả lời, phân loại văn bản và nhiều tác vụ khác.

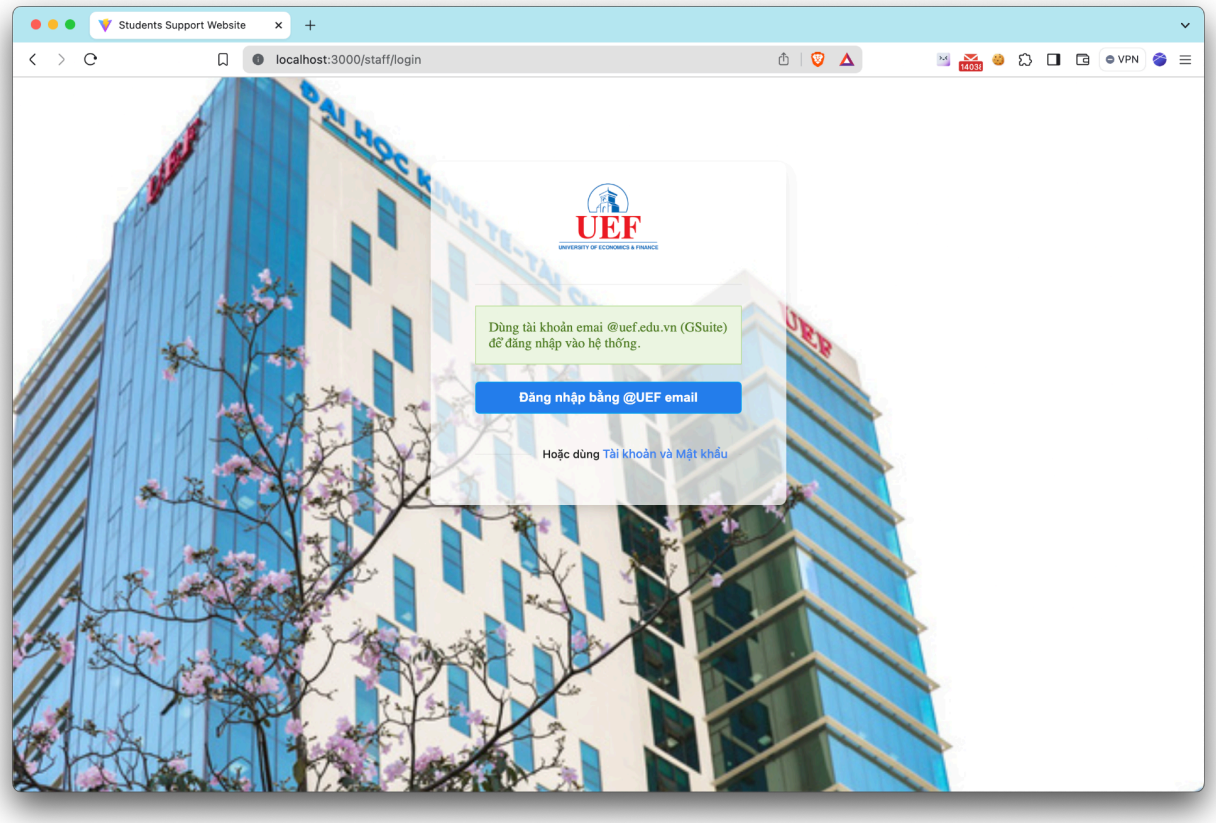
CHAPTER 4. XÂY DỰNG VÀ THỬ NGHIỆM ỨNG DỤNG

Trong phần này, nhóm sẽ trình bày về quá trình xây dựng mô hình tiếp nhận ý kiến và giải đáp thắc mắc cho sinh viên. Dựa trên nhu cầu người dùng và các công nghệ sử dụng đã trình bày ở phía trên, nhóm tiến hành phát triển các chức năng được ưu tiên trước.

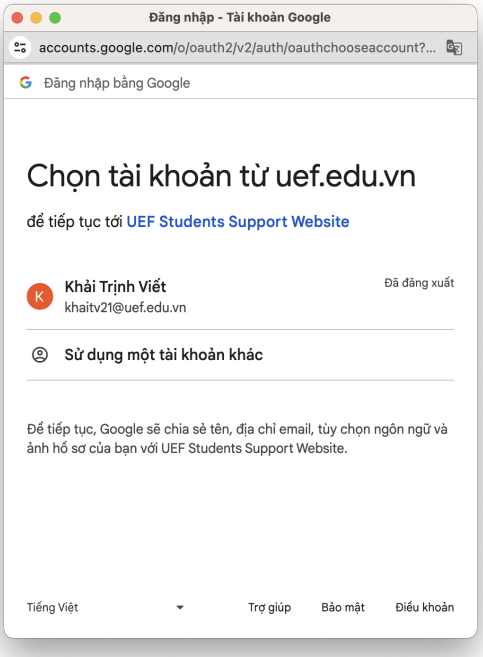
Về chức năng Login, nhóm sử dụng Google OAuth để thực hiện, đăng ký Credentials tại Google Cloud Console, sử dụng GoogleStrategy của thư viện passport-google-oauth20 để kiểm soát đăng nhập và giới hạn email với đuôi là uef.edu.vn. Tại backend, nhóm cũng tích hợp JSON Web Token để dễ dàng xác thực người dùng và truyền thông tin xác thực giữa các bên trong ứng dụng. Khi người dùng đăng nhập thành công, họ sẽ nhận được một JWT, và sau đó, họ có thể gửi JWT đó trong các yêu cầu tiếp theo để chứng minh xác thực của mình. Máy chủ có thể kiểm tra chữ ký trong JWT để xác thực nguồn gốc của nó và truy cập thông tin trong Payload.



Hình 17: Trang đăng nhập của sinh viên



Hình 18: Trang đăng nhập của phòng ban, khoa và giảng viên

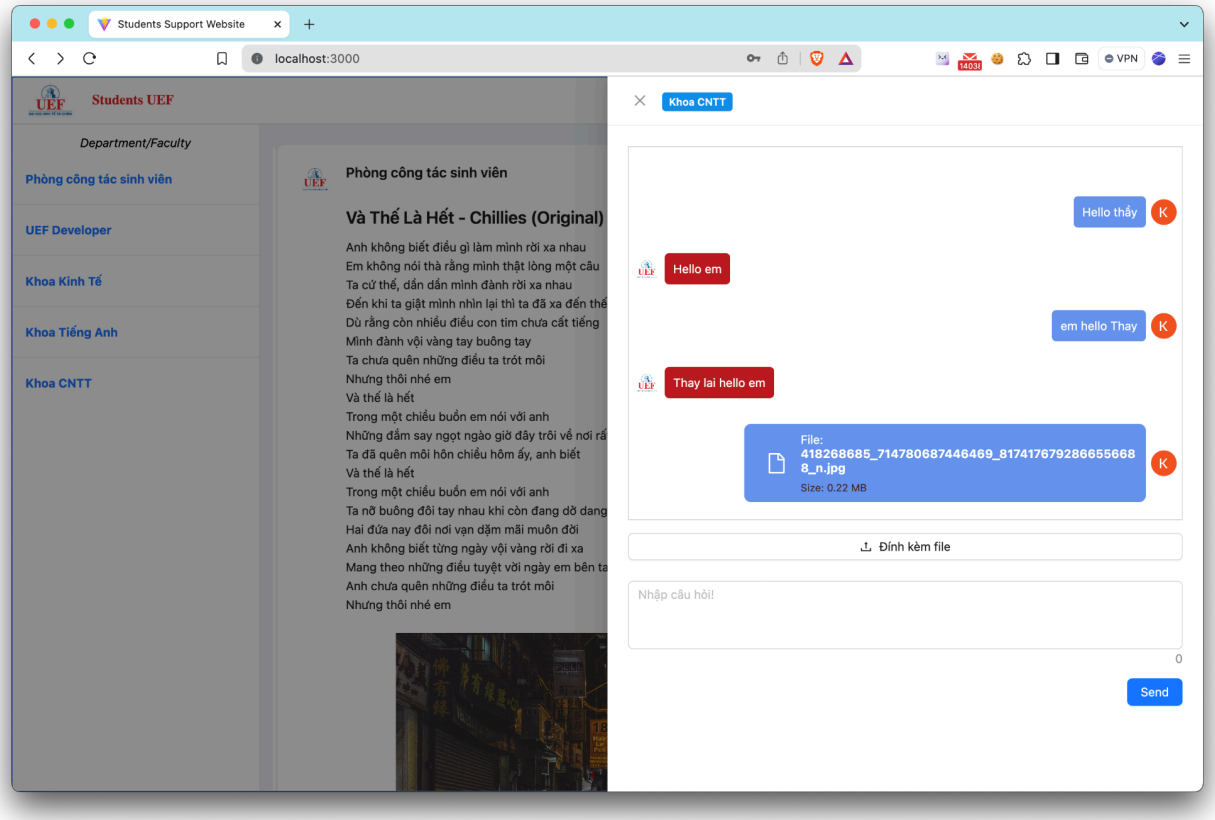


Hình 19: Sử dụng đăng nhập google cho ứng dụng

Tiếp đến là Users, sử dụng Mongoose truy vấn và MongoDB làm nơi lưu trữ dữ liệu chính của các users, mỗi user sẽ có các trường chủ yếu như name, email, roleID, image. Dựa vào lợi thế của cơ sở dữ liệu NoSQL là cấu trúc dữ liệu linh hoạt, các trường được thay đổi tuỳ ý với từng loại user, trong đó đối với các khoa, phòng ban hoặc giảng viên sẽ thêm khoa hay chuyên môn mà họ đảm nhận, đối với sinh viên, họ sẽ phải thêm mã số sinh viên và khoa mà họ đang thuộc về.

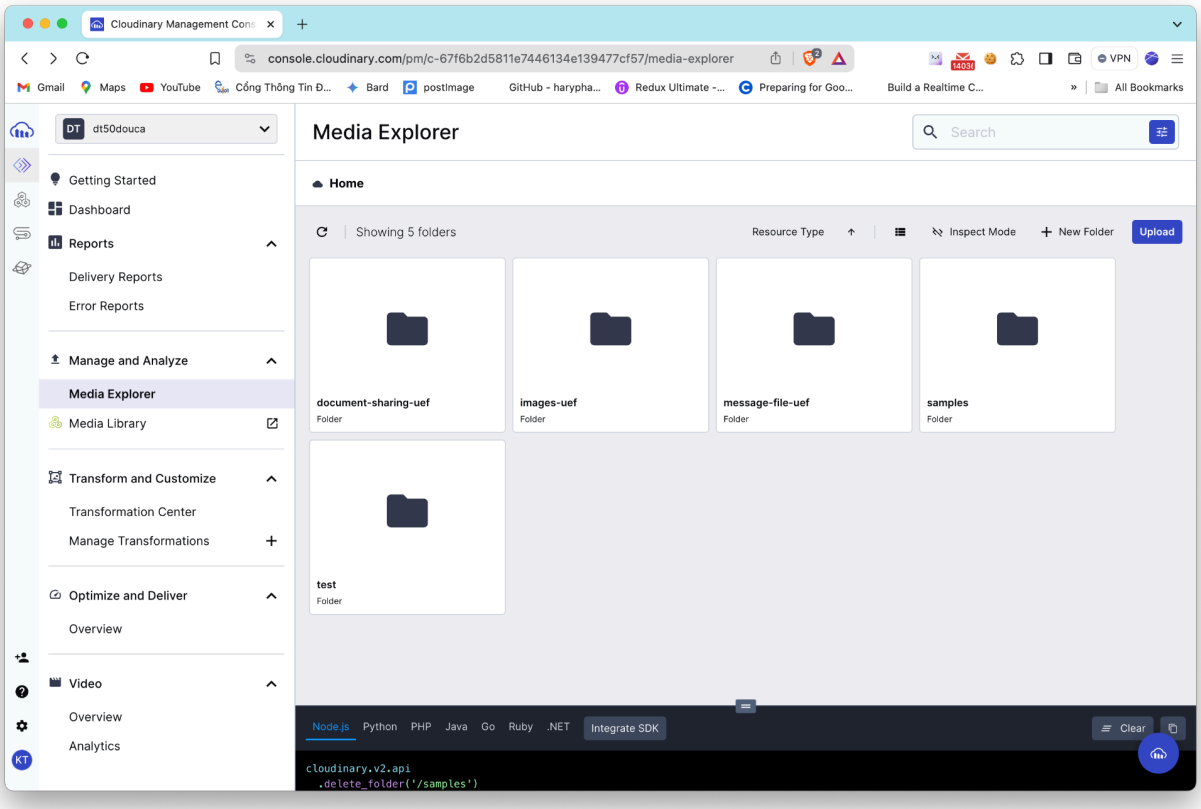
Về cuộc trò chuyện giữa giảng viên, phòng ban hoặc khoa và sinh viên, nhóm thiết lập nhanh với thư viện socket.io tại backend NodeJS và socket.io-client tại frontend React, khi chưa có cuộc trò chuyện nào sẽ thực hiện kết nối để mở ra cuộc trò chuyện cũng như là phương thức join của thư viện.

Sau đó hai bên sẽ dễ dàng trao đổi thông tin với thời gian thực, bên cạnh đó ngoài việc gửi tin nhắn thông thường, nhóm cũng đã thêm chức năng đính kèm file để sinh viên hay giảng viên dễ dàng gửi các tài liệu về các thông tin liên quan. File đính kèm tải lên được giới hạn các loại file như zip, pdf, docx, excel hay hình ảnh với dung lượng tối đa là 5MB.



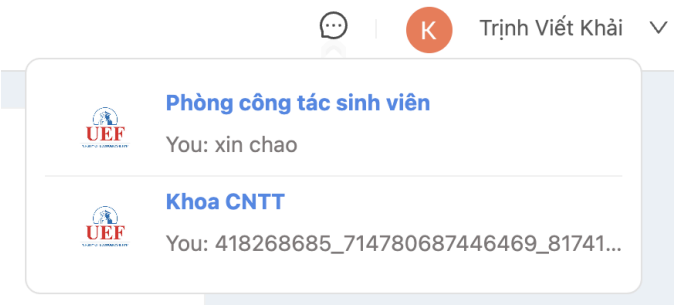
Hình 20: Sinh viên và giảng viên có thể nhắn tin và gửi tệp đính kèm

File hay hình ảnh đều được lưu trữ tại Cloudinary, nhóm đã tiến hành tìm hiểu và tích hợp thư viện cloudinary, là một thư viện phía máy chủ và phía client được cung cấp bởi dịch vụ lưu trữ đám mây Cloudinary. Thư viện này cung cấp các phương thức và công cụ để tương tác với dịch vụ Cloudinary, quản lý và xử lý các tệp đa phương tiện như hình ảnh và video. Từ đó giúp tăng hiệu suất của ứng dụng nhờ khả năng lưu trữ tốt hơn thay vì lưu tất cả dưới dạng Base64 xuống cơ sở dữ liệu.

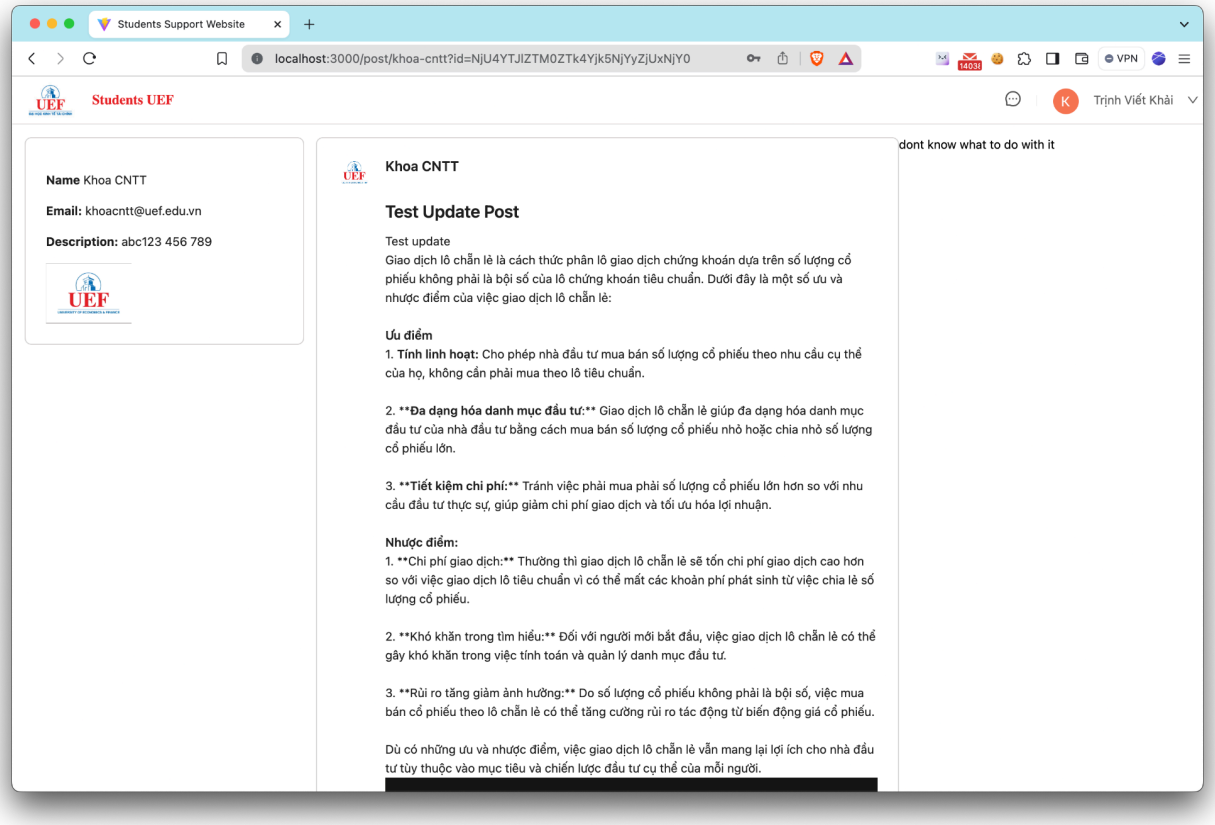


Hình 21: Cloudinary - Dịch vụ lưu trữ đa phương tiện trên đám mây

Người dùng cũng có thể xem lại lịch sử cuộc trò chuyện nhờ thư viện Redux-Toolkit và kết hợp với MongoDB lưu trữ lịch sử cuộc trò chuyện theo conversation_id, tải lại các cuộc trò chuyện khi đăng nhập lại vào ứng dụng.

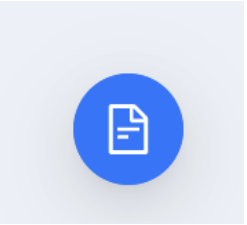


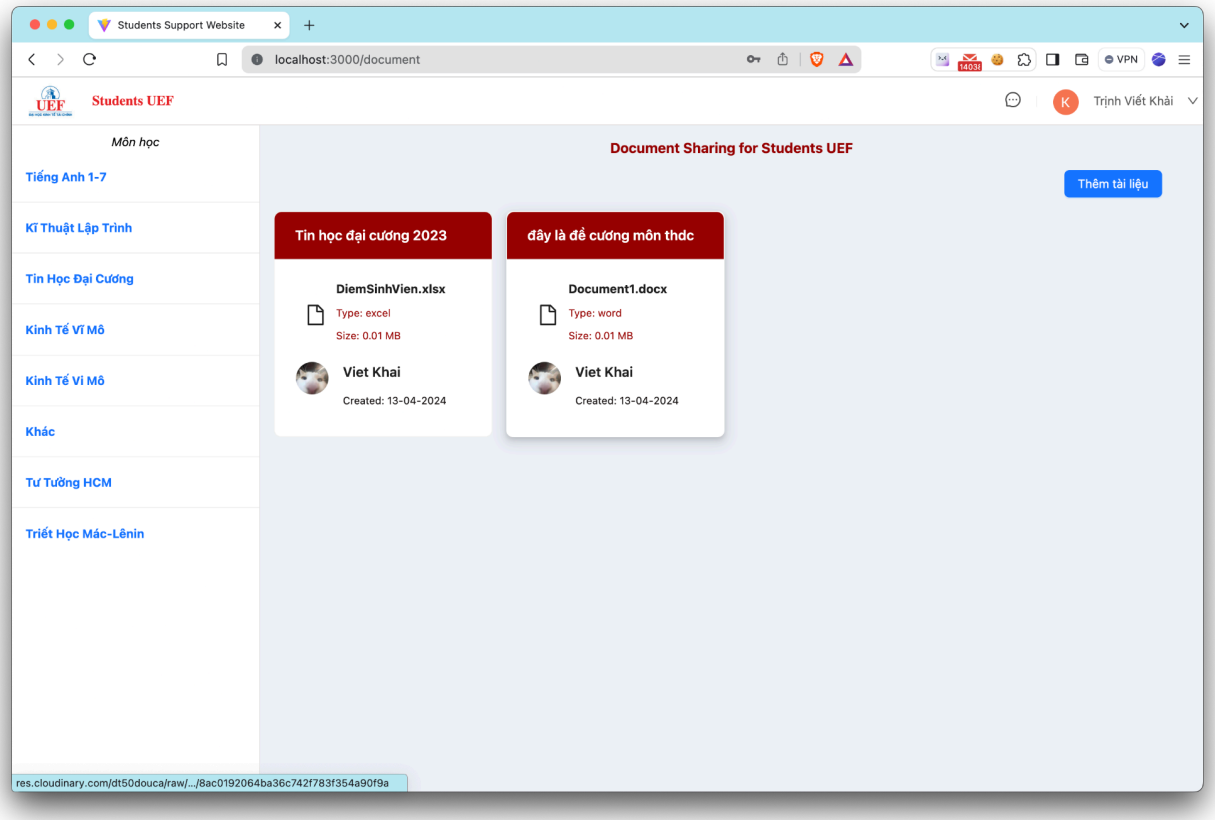
Phần Posts - bài viết của khoa và phòng ban. Mỗi phòng ban hay khoa đều có thể đăng tải các bài viết mới nhất về các sự kiện workshop hay các thông tin liên quan, sinh viên có thể click vào tên của khoa hoặc phòng ban trên thanh chat để tới trang thông tin của phòng ban đó.



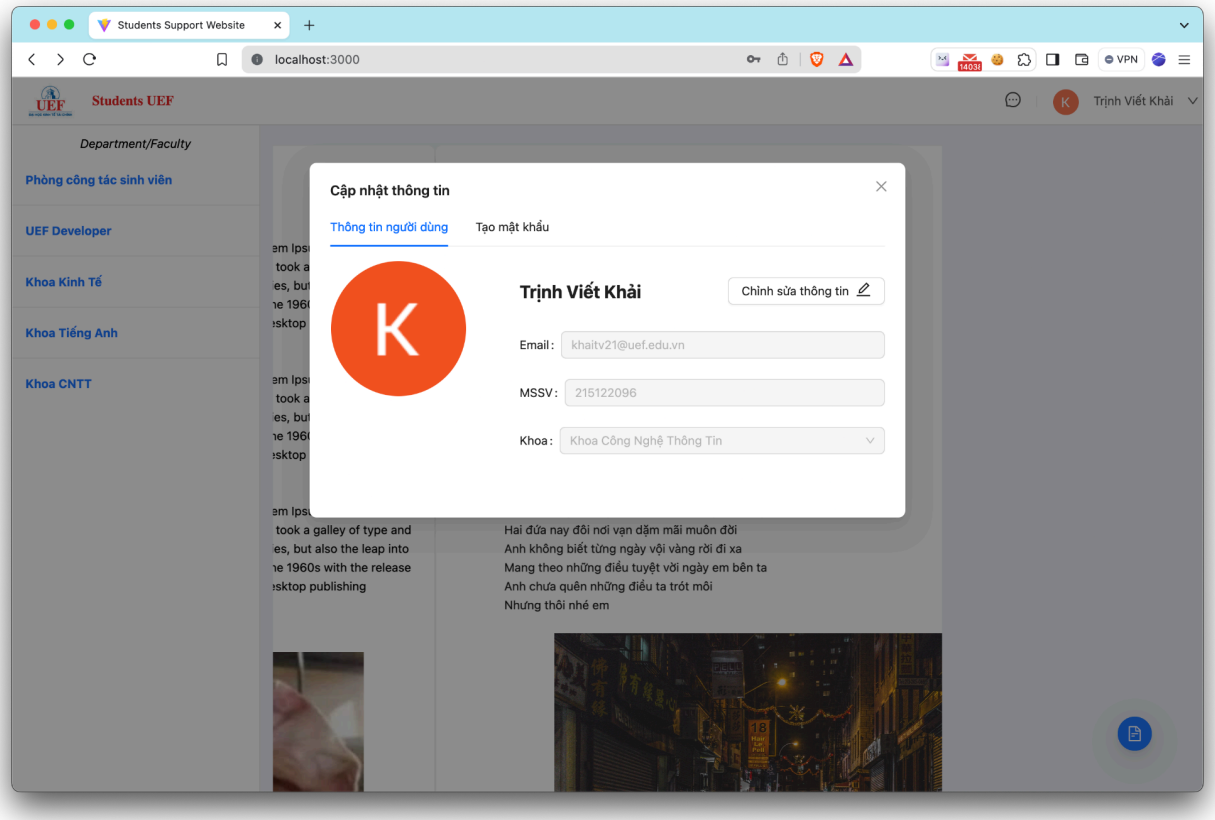
Hình 22: Trang thông tin của mỗi phòng ban/khoa sau khi sinh viên click vào

Sinh viên có thể click vào biểu tượng tại góc màn hình để lựa chọn chuyển trang đến Document-Sharing dành riêng cho sinh viên, tại đây nhóm cũng sử dụng thư viện Cloudinary để lưu trữ các file và phân loại theo các môn học để tải tài liệu. Sinh viên có thể click vào các môn học và xem các tài liệu tại môn học đó.



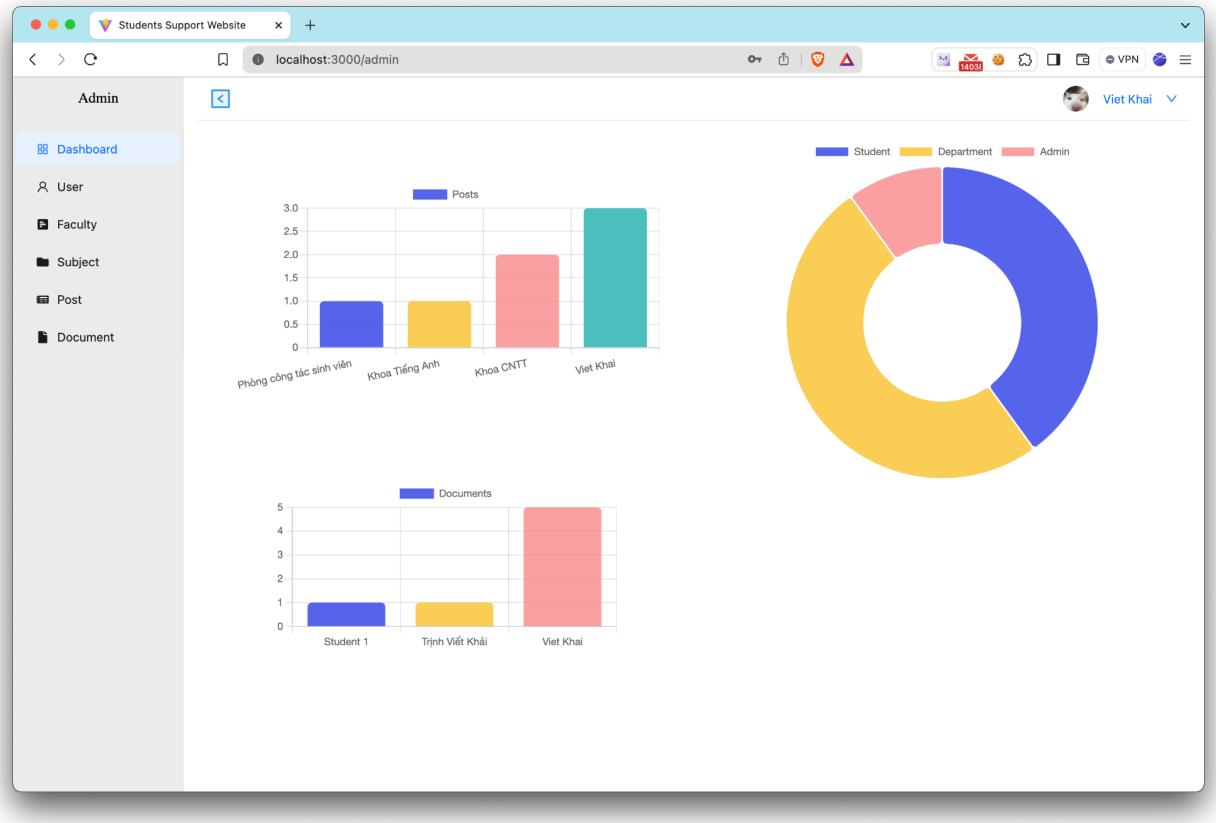


Hình 23: Trang Document-Sharing để sinh viên có thể chia sẻ tài liệu học tập



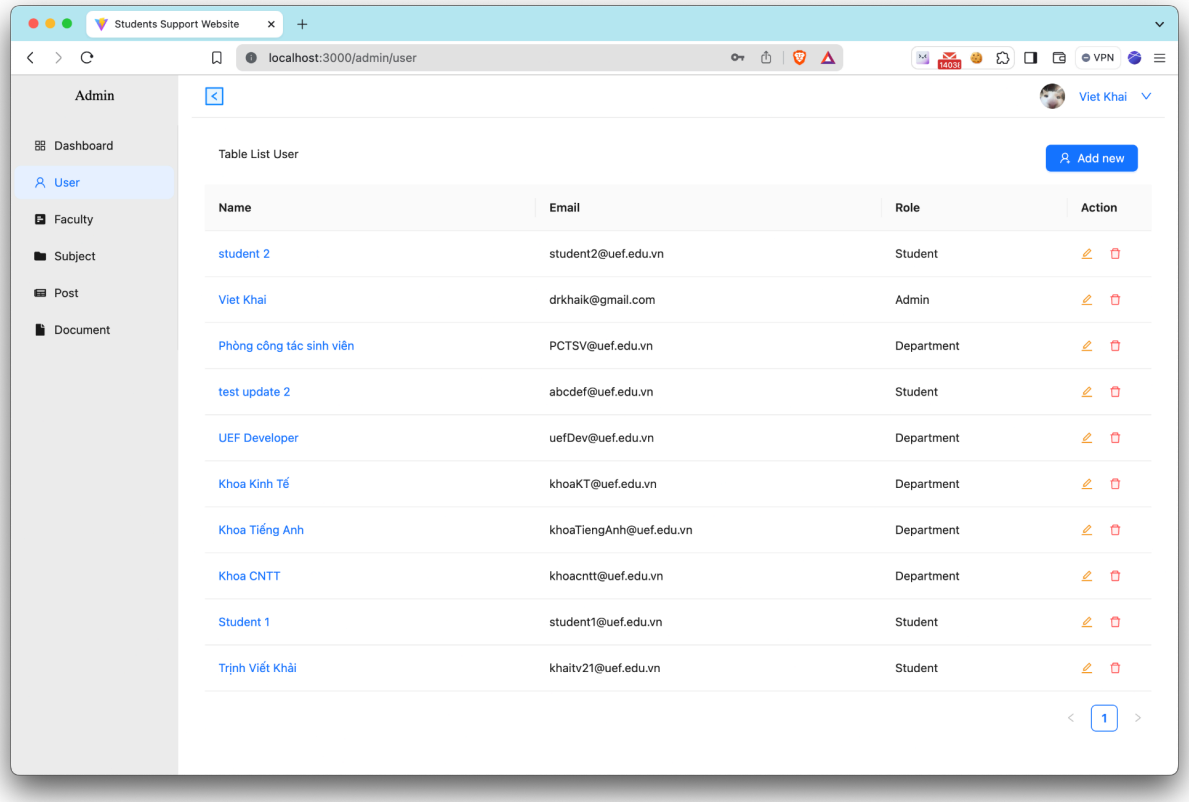
Hình 24: Tuỳ chọn thay đổi mật khẩu dành cho người dùng

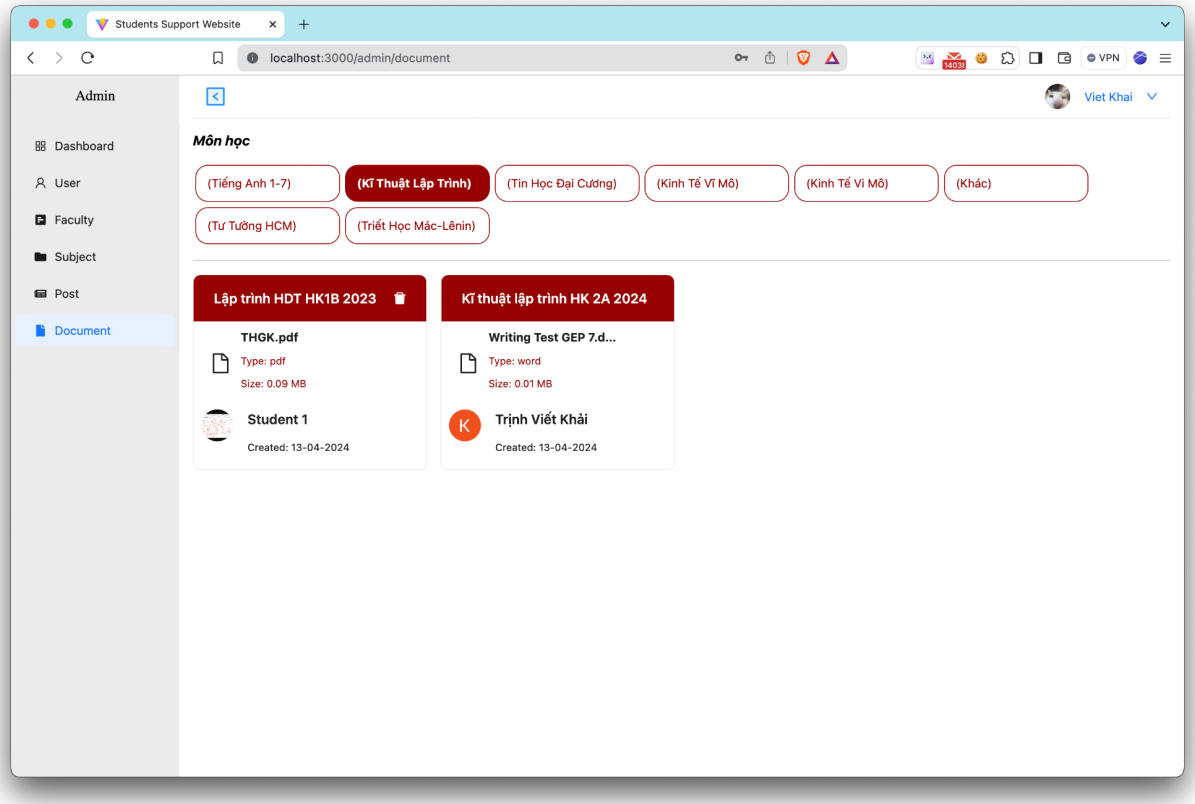
Về phần Admin, nhóm đã sử dụng thư viện React-chart để có thể tiến hành xây nhanh giao diện trang thống kê.



Hình 25: Giao diện trang Dashboard dành cho Admin

Admin cũng có thể kiểm soát Users, Posts hay Documents,...





Hình 26, 27: Giao diện trang Users, Document,... dành cho Admin

Tiếp theo là Chatbot, phần Backend được xây dựng bằng Python - framework Flask và phát triển dưới dạng API để linh hoạt trong việc tích hợp trên nhiều nền tảng, Client vẫn là React, và được tích hợp vào biểu tượng tại góc màn hình.

Quá trình thu thập dữ liệu nhóm sử dụng module requests để request đến các địa chỉ trang web (trang chứa nội dung cần thu thập), tất cả các nội dung trang web sẽ được trả về lưu vào một đối tượng. Sau đó thư viện BeautifulSoup để loại bỏ các thẻ html và lấy ra các dữ liệu cần thiết.

Hình 28: Sử dụng BeautifulSoup và các thư viện đi kèm để làm sạch dữ liệu

Nhận dạng từ khóa của dữ liệu đầu vào (câu hỏi) là một trong những bước xử lý vô cùng quan trọng trong hệ thống chatbot. Vì bước nhận dạng từ khóa sai dẫn đến việc truy vấn dữ liệu sai và dẫn đến dữ liệu đưa vào model BERT không chính xác.

Bước nhận dạng từ khóa của luận văn, nhóm sử dụng kỹ thuật Named Entity Recognition (NER) của Underthesea NLP Toolkit. Đây là một là bộ module Python mã nguồn mở, giúp hỗ trợ xử lý các tác vụ NLP trong tiếng Việt, tác giả của bộ toolkit này là ông Vũ Anh và các cộng sự. Bộ toolkit này cung cấp các API để áp dụng các mô hình được huấn luyện trước vào các văn bản tiếng Việt cho các tác vụ NLP như: Word Segmentation, POS tagging, Named Entity Recognition, v.v... Underthesea được dựa trên những thư viện Deep Learning nổi tiếng là Pytorch, để dễ dàng huấn luyện các mô hình Deep learning và có thể thí nghiệm với các phương pháp mới sử dụng Underthesea. Underthesea NLP Toolkit được sử dụng trong Chatbot UEF để xử lý và hiểu các câu, yêu cầu từ người dùng.

Ví dụ với câu hỏi: "**Điểm trung bình tích lũy như nào là bằng giỏi?**". Sau khi xử lý, nhận được kết quả trả về với từ khoá chính của câu: "**trung bình tích lũy**".

Từ những từ khóa tìm được trong câu hỏi mà người dùng đưa vào, nhóm sử dụng ngôn ngữ truy vấn Cypher để tạo ra câu truy vấn với điều kiện là các từ khóa. Câu truy vấn được đưa vào module tìm kiếm của hệ thống để tìm ra các node và mối liên kết có liên quan từ đồ thị tri thức. Ví dụ từ khóa "**trung bình tích lũy**", câu truy vấn nhóm sử dụng như sau:

match (s)-[r]-(o)

where s.name contains 'trung bình tích lũy'

return s,r,o limit 20

Trong đó:

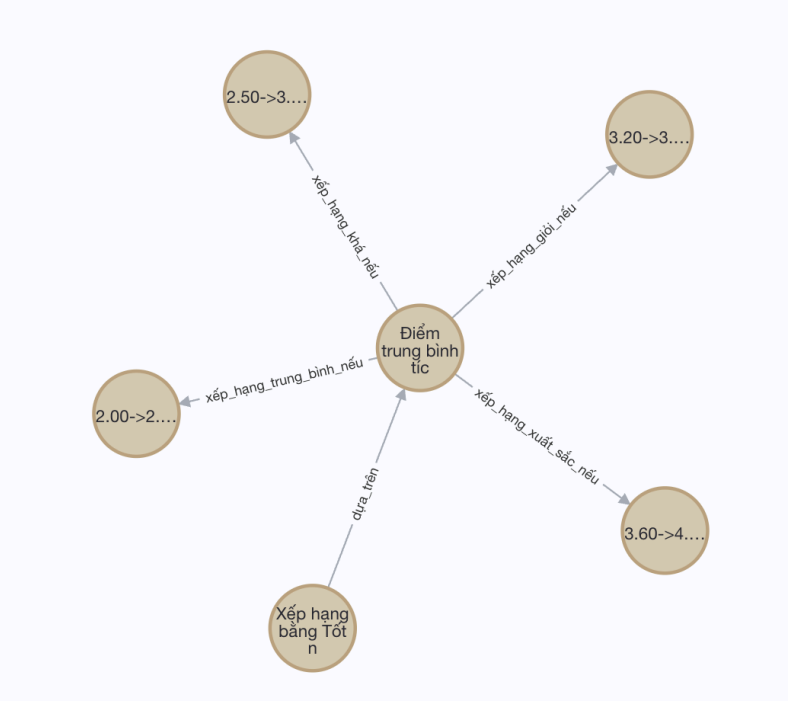
- **match (s)-[r]-(o)**: lấy ra những node và liên kết (subject, relationship và object).

- **where s.name contains 'trung bình tích lũy'**: tìm những node có chứa từ khóa "**trung bình tích lũy**".

- **return (s)-[r]-(o)**: trả về kết quả bao gồm node và các liên kết.

- **limit 20**: giới hạn số node trả về mỗi lần truy vấn.

Sau khi thực hiện câu truy vấn trên ta nhận được kết quả như sau:



Hình 29: Kết quả trả về dưới dạng đồ thị

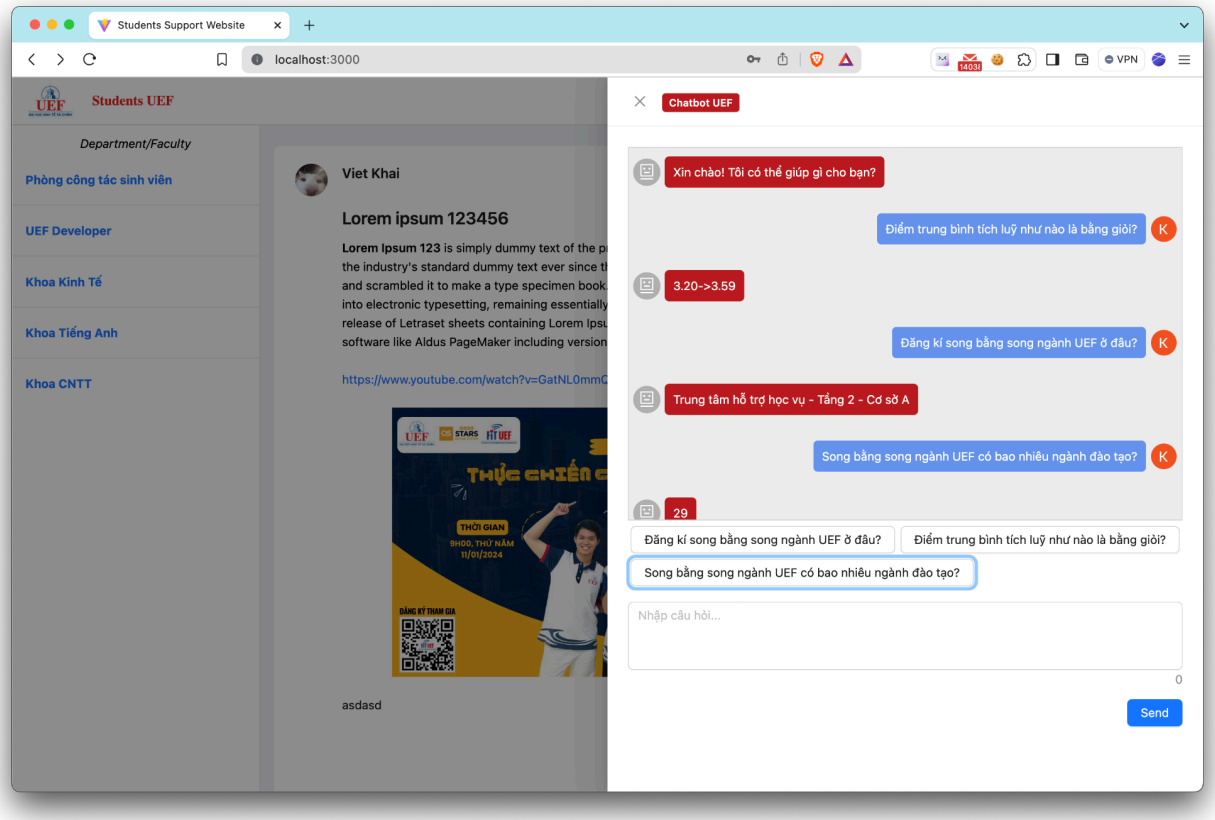
Từ đó ghép các kết quả truy vấn, ta có được một đoạn văn gồm những thông tin liền kề với node và liên kết tương ứng. Đoạn văn có dạng như sau: "**Điểm trung bình tích lũy xếp hạng xuất sắc nếu 3.60->4.00. Điểm trung bình tích lũy xếp hạng giỏi nếu 3.20->3.59. Điểm trung bình tích lũy xếp hạng khá nếu 2.50->3.19. Điểm trung bình**

tích lũy xếp hạng trung bình nếu 2.00->2.49. Điểm trung bình tích lũy theo quy định của Xếp hạng bằng Tốt nghiệp. Điểm trung bình tích lũy dựa trên Xếp hạng bằng Tốt nghiệp. ”

Sau khi có cả đoạn văn và câu hỏi của người dùng, nhóm tiến tiếp đến bước phân tích và ứng dụng thuật toán BERT. Trong phần này, nhóm đã sử dụng mô hình bert-multi-cased-finetuned-xquadv1 của tác giả Manual Romero để thực hiện dự đoán kết quả hỏi đáp trong Chatbot. Mô hình được phát triển bởi Google và đã được tinh chỉnh dựa trên các tập dữ liệu SQuAD v1.1 nhằm xử lý các tác vụ hỏi đáp BERT trên 11 ngôn ngữ khác nhau trong đó có tiếng Việt. Nhóm cũng sử dụng "question-answering" Pipeline từ thư viện Transformers. Đây là một Pipeline được xây dựng sẵn cho tác vụ trả lời câu hỏi. Nó kết hợp giữa mô hình BERT đã được fine-tune cho trả lời câu hỏi đa ngôn ngữ ("mrm8488/bert-multi-cased-finetuned-xquadv1") và tokenizer tương ứng ("mrm8488/bert-multi-cased-finetuned-xquadv1").

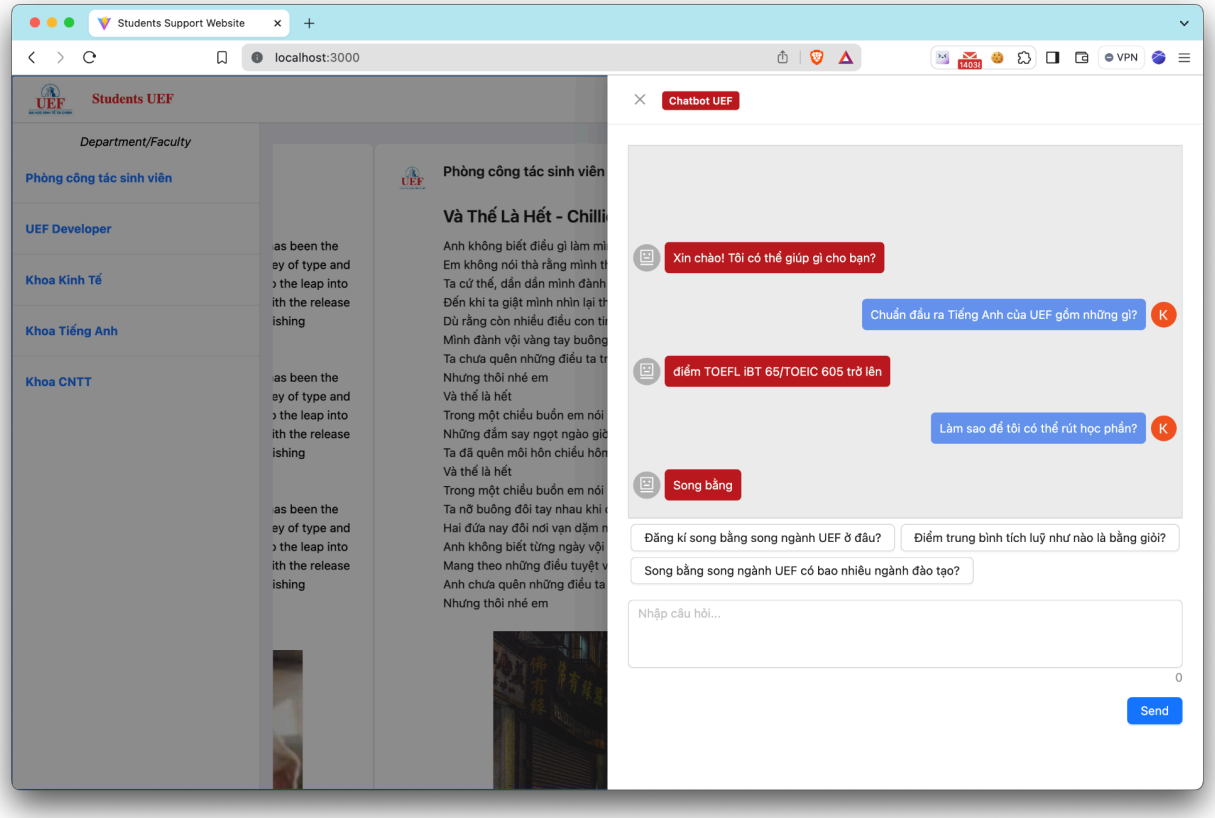
Khi sử dụng Pipeline "question-answering", từ kết quả truy vấn lên cơ sở dữ liệu trả về kết hợp cùng câu hỏi của người dùng, Pipeline sẽ sử dụng mô hình BERT để dự đoán câu trả lời cho câu hỏi dựa trên ngữ cảnh đã cho. Kết quả sẽ là một đối tượng có chứa thông tin về câu trả lời, vị trí trong đoạn văn và điểm độ tin cậy của câu trả lời.

Sau khi tích hợp Chatbot vào thực nghiệm trên dự án, thông qua API để nhận câu hỏi và phản hồi đáp án từ phía người dùng.



Hình 30: Một vài câu hỏi mà nhóm đã chuẩn bị về chủ đề nằm trong cơ sở tri thức

Qua đánh giá sơ bộ chatbot mà nhóm đã thực hiện, vì thời gian gấp rút và ý tưởng được bắt đầu vào 30/3/2024 (sau khi nộp báo cáo lần đầu và được giảng viên hướng dẫn là thầy Hoàng Văn Hiếu đề xuất ý tưởng phân khúc tích hợp AI), thời gian để thu thập dữ liệu khiến cho cơ sở tri thức đa dạng và đáp ứng được mọi câu hỏi là không thể. Cơ sở tri thức hiện tại là khoảng 188 nodes và 523 liên kết, nên có nhiều câu hỏi chatbot không thể trả lời đầy đủ và chính xác.



Hình 31 Dữ liệu không đủ khiến chatbot trả lời sai hoặc không đủ

CHAPTER 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KHẢ NĂNG

Trong dự án này, nhóm đã hoàn thành việc xây dựng một ứng dụng web đa chức năng với khả năng chat online, upload tài liệu và bài viết. Sinh viên có thể sử dụng ứng dụng để tương tác và cập nhật tin tức với các bên liên quan như khoa và phòng ban, họ cũng có thể tải và chia sẻ các bộ tài liệu về các môn học cùng nhau.

2. HẠN CHẾ

Một trong những nhược điểm của ứng dụng là tính bảo mật. Vì thời gian gấp rút và kiến thức còn hạn chế, để đưa ứng dụng vào thực tế, nhóm cần phải xem xét kĩ đến bảo mật thông tin cho người dùng. Bên cạnh đó là responsive cho ứng dụng, nhóm vẫn chưa giải quyết vấn đề về thiết kế nội dung có thể hiển thị tương thích trên nhiều loại thiết bị khác nhau.

3. HƯỚNG MỞ RỘNG VÀ PHÁT TRIỂN

Về hướng phát triển của ứng dụng nhằm cung cấp một nền tảng mạnh mẽ giúp dự án có thể đưa vào thực tế. Dưới đây là một số hướng phát triển tiềm năng mà nhóm đã nghĩ ra cho dự án:

- Tăng cường bảo mật: tăng cường thêm tính bảo mật, áp dụng các biện pháp bảo mật kỹ thuật và quản lý hiệu quả để có thể giảm thiểu rủi ro và bảo vệ thông tin quan trọng của người dùng và hệ thống khỏi các cuộc tấn công và vi phạm bảo mật.
- Tích hợp thêm tính năng: Có thể mở rộng dự án bằng cách tích hợp các tính năng bổ sung như tương tác thảo luận và tích hợp chatbot giúp trả lời tự động khi các phòng ban và khoa vắng mặt. Điều này sẽ tăng cường tính năng và giá trị của ứng dụng.
- Tối ưu hóa hiệu suất: Để đáp ứng được lượng truy cập lớn và tăng cường hiệu suất, có thể tối ưu hóa hệ thống bằng cách sử dụng các công nghệ như bộ đệm, tải cân bằng tải và lưu trữ tệp tin hiệu quả hơn.

- Mở rộng vào các nền tảng khác: Ngoài việc phát triển trang web, có thể xây dựng phiên bản ứng dụng di động cho Android và iOS, để mở rộng phạm vi sử dụng của ứng dụng.
- Cải thiện giao diện người dùng: Tạo ra một giao diện người dùng hấp dẫn, tương tác và dễ sử dụng để cung cấp trải nghiệm tốt hơn cho người dùng cuối.
- Kiểm thử và tối ưu hóa: Tiếp tục thực hiện kiểm thử, tối ưu và sửa lỗi để cải thiện tính ổn định và đáng tin cậy của ứng dụng. Điều này có thể bao gồm việc tăng cường bảo mật, xử lý các trường hợp đặc biệt và tối ưu hóa mã nguồn.
- Tích hợp các dịch vụ bên thứ ba: Nếu cần, có thể tích hợp các dịch vụ bên thứ ba như email thông báo, SMS thông báo và tích hợp mạng xã hội để tạo ra một trải nghiệm tích cực và linh hoạt hơn cho người dùng.
- Thu thập dữ liệu và những câu hỏi thường hay hỏi tại tầng 2 - Trung tâm hỗ trợ học vụ, những vấn đề xoay quanh sinh viên UEF và đưa vào cơ sở dữ liệu, khiến cơ sở tri thức đầy đủ và đa dạng hơn, tạo ra Chatbot hữu ích hơn với người dùng.

[1] [Trực tuyến]: https://socket.io/get-started/chat

[2] [Trực tuyến]: https://cloudinary.com/documentation/upload_images

[3] [Trực tuyến]: https://redux-toolkit.js.org/tutorials/quick-start

[4] [Trực tuyến]: https://developers.google.com/identity/protocols/oauth2/web-server

[5] [Trực tuyến]: https://ant.design/docs/react/getting-started

TÀI LIỆU THAM KHẢO

[1] [SOCKET.IO] : https://socket.io/get-started/chat

[2] [CLOUDINARY] : https://cloudinary.com/documentation/upload_images

[3] [REDUX] : https://redux-toolkit.js.org/tutorials/quick-start

[4] [GOOGE CONSOLE] : https://developers.google.com/identity/protocols/oauth2/web-server

[5] [ANT DESIGN] : https://ant.design/docs/react/getting-started

[6] [MONGOOSE ODM] : https://mongoosejs.com/docs/guide.html

[7] [NEO4J] : https://neo4j.com/docs/python-manual/current/

[8] [UNDERTHESEA] : https://underthesea.readthedocs.io/en/latest/readme.html

[9] [BERT MODEL] : https://huggingface.co/mrm8488/bert-multi-cased-finetuned-xquadv1