|  |  |
| --- | --- |
| TRƯỜNG ĐH GIAO THÔNG VẬN TẢI PHÂN HIỆU TP.HCM  Bộ Môn Công Nghệ Thông Tin | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc Lập – Tự Do -Hạnh Phúc |

**ĐỀ CƯƠNG CHI TIẾT ĐỒ ÁN TỐT NGHIỆP**

1. **Thông tin Sinh viên:**

Họ tên : Nguyễn Ngọc Minh Nhật Mã sinh viên : 6151071078

Lớp : CQ.61.CNTT Hệ : Cử nhân

Ngành đào tạo : Công nghệ thông tin Khoá : 61

Email : nnminhnhat317@gmail.com Số điện thoại : 0931 331 405

1. **Thông tin Giảng viên hướng dẫn:**

Họ tên : Trần Phong Nhã Học vị : Thạc sĩ

Email : tpnha@utc2.edu.vn Số điện thoại : 0906 761 014

Đơn vị công tác: Bộ môn Công nghệ thông tin Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

**NỘI DUNG**

**I. Tên đề tài**

Xây dựng hệ thống web đặt lịch khám bệnh kết hợp chatbot và kỹ thuật RAG hỗ trợ tư vấn.

**II. Giới thiệu**

*Giới thiệu về bối cảnh và cần thiết của đề tài.*

Trong bối cảnh chuyển đổi số ngày càng mạnh mẽ, các cơ sở y tế và bệnh viện luôn tìm kiếm những giải pháp công nghệ nhằm nâng cao hiệu suất làm việc và tối ưu hóa quy trình vận hành của hệ thống y tế online nhằm giúp bệnh nhân dễ dàng tiếp cận. Một trong những khó khăn mà bệnh nhân gặp phải là tìm kiếm thông tin nhanh chóng và chính xác về chính sách, quy trình làm việc cũng như các vấn đề về liên quan của các cơ sở y tế và bác sĩ. Việc xây dựng hệ thống đặt lịch khám tập trung giúp tổng hợp nhiều nguồn thông tin từ nhiều cơ sở y tế cho bệnh nhân nhằm phần góp phần làm đơn giản đi phần nào quá trình tiếp cận của bệnh nhân để tìm kiếm, so sánh giữa nhiều cơ sở y tế và các bác sĩ.

Chatbot đã trở thành một giải pháp phổ biến giúp tự động hóa quá trình phản hồi, giảm tải công việc cho bộ phận nhân sự và hỗ trợ bệnh nhân một cách nhanh chóng, hiệu quả. Đặc biệt, với sự phát triển của công nghệ Retrieval-Augmented Generation (RAG), chatbot có thể kết hợp khả năng tìm kiếm thông tin từ nguồn dữ liệu nội bộ với mô hình ngôn ngữ tiên tiến, mang đến câu trả lời chính xác và phù hợp theo ngữ cảnh.

Xuất phát từ nhu cầu thực tiễn, hệ thống đặt lịch khám với tiêu chí đơn giản hóa quy trình tìm kiếm thông tin về các cơ sở y tế của bệnh nhân và giảm tải công việc cho bộ phận nhân sự bằng cách tự động hóa việc giải đáp các câu hỏi thường gặp của bệnh nhân. Việc áp dụng mô hình RAG giúp cải thiện chatbot cung cấp câu trả lời chính xác, phù hợp với thông tin của các cơ sở y tế dựa trên nguồn dữ liệu nội bộ để tạo nên một hệ thống linh hoạt và thông minh. Hơn nữa, hệ thống chatbot này có khả năng mở rộng và dễ dàng tích hợp vào hệ thống đặt lịch khám, giúp tổ chức phát triển thêm các tính năng mới khi cần thiết.

*Mục tiêu của đề tài.*

Mục tiêu của đề tài là nghiên cứu và xây dựng hệ thống đặt lịch khám bệnh kết hợp chatbot cùng kỹ thuật Retrieval-Augmented Generation (RAG), xây dựng hệ thống có khả năng tìm kiếm và tổng hợp thông tin từ dữ liệu nội bộ để giải đáp thắc mắc và hỗ trợ tìm kiếm thông tin. Hệ thống đặt lịch khám sẽ được thiết kế nhằm đảm bảo giúp bệnh nhân dễ dàng tiếp cận tìm kiếm, so sánh các cơ sở y tế với nhau cùng với chatbot hỗ trợ trả lời câu hỏi và nhận phản hồi một cách nhanh chóng. Điều này sẽ giúp giảm thiểu sự phụ thuộc vào bộ phận nhân sự hỗ trợ, đồng thời nâng cao hiệu quả xử lý thông tin. Với đề tài này, hệ thống đặt lịch khám bệnh không chỉ giúp bệnh nhân dễ dàng tiếp cận đặt lịch khám online mà còn tạo ra môi trường hiện đại, hỗ trợ bệnh nhân một cách hiệu quả và chuyên nghiệp.

*Phạm vi và đối tượng nghiên cứu.*

Đề tài nghiên cứu này tập trung vào hệ thống đặt lịch khám bệnh và việc ứng dụng kỹ thuật Retrieval-Augmented Generation (RAG) kết hợp với API của LLM để xây dựng chatbot, được tích hợp trên nền tảng web trong hệ thống đặt lịch khám. Các mô hình LLM đã chứng tỏ sự mạnh mẽ trong việc hiểu ngữ cảnh câu hỏi, sinh câu trả lời tự nhiên và tạo văn bản mượt mà nhờ vào khả năng xử lý ngôn ngữ tự nhiên (NLP) tiên tiến. Với mô hình LLM đã được huấn luyện trên một lượng dữ liệu khổng lồ, cho phép nó tự động hiểu và phản hồi các câu hỏi mà không cần sự can thiệp của các phương pháp NLP thủ công. Chính vì thế, việc sử dụng các mô hình ngôn ngữ này trong hệ thống chatbot sẽ giúp tối ưu hóa thời gian và tập trung hơn vào việc tìm hiểu kỹ thuật RAG. Với sự kết hợp này đề tài sẽ tạo ra một hệ thống đặt lịch khám và kết hợp chatbot tương tác một cách linh hoạt, tự nhiên cho bệnh nhân.

Đối tượng nghiên cứu chính của đề tài là bệnh nhân, đặc biệt là những người thường xuyên cần hỗ trợ tư vấn về thông tin các cơ sở y tế hay muốn chủ động thời gian khám bệnh. Đề tài sẽ tập trung vào việc nghiên cứu và phát triển hệ thống đặt lịch khám và các công nghệ liên quan đến RAG, đồng thời xây dựng một hệ thống chatbot thông minh để nâng cao hiệu quả hỗ trợ bệnh nhân dễ dàng.

*Các chức năng dự kiến của đề tài:*

Cho phép người dùng lựa chọn thông tin bác sĩ, chuyên khoa, cơ sở y tế. Đặt lịch khám bệnh theo các khung giờ linh hoạt.

Quản trị viên và bác sĩ: quản lý thông tin cá nhân, thông tin bác sĩ, thời gian làm việc của bác sĩ, lịch hẹn thăm khám của bệnh nhân.

Chatbot hỗ trợ tìm kiếm bác sĩ hoặc phòng khám, đặt lịch khám.

**III. Cơ sở lý thuyết**

*Các khái niệm và lý thuyết liên quan đến đề tài.*

**1. HTML, CSS, JS**

HTML, CSS và JavaScript là ba công nghệ nền tảng trong phát triển web, giúp tạo nên các trang web hiện đại và tương tác. HTML (HyperText Markup Language) đóng vai trò là bộ khung của trang web, giúp xác định các thành phần như tiêu đề, đoạn văn, hình ảnh và liên kết. Trong khi đó, CSS (Cascading Style Sheets) chịu trách nhiệm định dạng và thiết kế giao diện, giúp trang web trở nên trực quan và đẹp mắt hơn bằng cách thay đổi màu sắc, kiểu chữ, bố cục và hiệu ứng hiển thị. Để làm cho trang web trở nên sống động và có khả năng tương tác, JavaScript (JS) được sử dụng để xử lý các sự kiện, cập nhật nội dung động, gửi và nhận dữ liệu từ máy chủ mà không cần tải lại trang. Sự kết hợp của ba công nghệ này cho phép xây dựng từ những trang web tĩnh đơn giản đến các ứng dụng web động phức tạp, mang lại trải nghiệm tốt hơn cho người dùng.

**1. ReactJS là gì ?**

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Meta (Facebook), giúp xây dựng giao diện người dùng (UI) một cách nhanh chóng và hiệu quả. ReactJS được thiết kế theo hướng component-based, tức là chia giao diện thành các thành phần nhỏ, có thể tái sử dụng, giúp tăng khả năng bảo trì và mở rộng ứng dụng.

Một trong những điểm mạnh của ReactJS là Virtual DOM (Document Object Model ảo), giúp cập nhật giao diện nhanh hơn bằng cách chỉ thay đổi những phần tử cần thiết thay vì render lại toàn bộ trang. ReactJS cũng hỗ trợ One-way Data Binding (ràng buộc dữ liệu một chiều), giúp kiểm soát luồng dữ liệu tốt hơn và làm cho ứng dụng dễ quản lý hơn.

ReactJS còn nổi bật với khả năng tích hợp tốt với TypeScript, Redux, React Query, và nhiều thư viện hỗ trợ khác, giúp xây dựng các ứng dụng phức tạp từ web đơn giản đến các hệ thống quy mô lớn. React cũng có thể sử dụng để phát triển ứng dụng di động thông qua React Native.

Nhờ tính linh hoạt, hiệu suất cao và hệ sinh thái rộng lớn, ReactJS hiện là một trong những công nghệ frontend phổ biến nhất, được sử dụng trong nhiều dự án lớn như Facebook, Instagram, Airbnb, Netflix và nhiều công ty công nghệ khác.

**2. Spring Boot và Spring Data JPA**

Spring là một Java framework rất lớn và khổng lồ nắm vai trò quan trọng trong phát triển phần mềm, tuy nhiên chúng vẫn chứa đựng những nhược điểm nhất định pring có nhược điểm là cấu hình (config) dự án quá phức tạp, phải làm rất nhiều thứ cho dù đó là chương trình đơn giản nhất. Chính vì lý do này, Spring Boot đã ra đời nhằm cải thiện những nhược điểm này và hứa hẹn mang đến nhiều sản phẩm phần mềm hoàn hảo trong tương lai.

Spring Boot là một trong số các module của Spring framework chuyên cung cấp các tính năng RAD (Rapid Application Development) cho phép tạo ra và phát triển các ứng dụng độc lập dựa trên Spring một cách nhanh chóng như ứng dụng web, microservices hay hệ thống backend. Spring Boot ra đời với mục đích loại bỏ những cấu hình phức tạp của Spring, nó không yêu cầu cấu hình XML và nâng cao năng suất cho các nhà phát triển. Một số đặc điểm nổi bật của Spring Boot như: tự động cấu hình (Auto Configuration), tích hợp nhanh các công nghệ như Spring Data JPA, Spring Security, Spring Web, Spring Cloud, quản lý phụ thuộc dễ dàng với Maven hoặc Gradle. Tóm lại, Spring Boot là một lựa chọn phổ biến cho các hệ thống backend hiện đại, giúp các lập trình viên xây dựng ứng dụng dễ dàng, nhanh chóng và linh hoạt hơn.

Spring Data JPA là một module giúp tương tác với cơ sở dữ liệu dễ dàng hơn mà không cần viết SQL thủ công. Nó hoạt động trên nền JPA (Java Persistence API) và hỗ trợ ORM (Object-Relational Mapping). Các thành phần tiêu chuẩn cần có để cấu hình Spring Data JPA gồm *Entity*: Lớp đại diện cho bảng trong cơ sở dữ liệu. *Repository*: Interface giúp thực hiện các thao tác CRUD tương tác với database (Create, Read, Update, Delete). *Service*: Thực hiện các logic phức tạp, nó giúp tách biệt logic xử lý nghiệp vụ (business logic) khỏi controller, giúp mã nguồn dễ bảo trì và tái sử dụng hơn. *Controller*: controller trong Spring JPA xử lý các request từ client, gọi service để thao tác dữ liệu, và trả về response, cung cấp các API tương tác frontend.

**3. MySql**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển bởi Oracle Corporation. Nó sử dụng ngôn ngữ SQL (Structured Query Language) để quản lý và truy vấn dữ liệu. MySQL có hiệu suất cao, dễ sử dụng và hỗ trợ nhiều nền tảng, giúp nó trở thành lựa chọn phổ biến cho các ứng dụng web, từ các hệ thống nhỏ đến các nền tảng lớn như Facebook, Twitter (X).

Trong phát triển backend, MySQL thường được sử dụng kết hợp với Spring Boot và JPA để quản lý dữ liệu một cách hiệu quả. Ngoài ra, MySQL còn cung cấp các cơ chế bảo mật mạnh mẽ, hỗ trợ xác thực người dùng, mã hóa dữ liệu và sao lưu phục hồi, giúp đảm bảo an toàn thông tin. Hệ thống này cũng hỗ trợ giao dịch với cơ chế ACID (Atomicity, Consistency, Isolation, Durability), giúp duy trì tính toàn vẹn của dữ liệu. Bên cạnh đó, MySQL có khả năng tích hợp tốt với các ngôn ngữ lập trình phổ biến như Java, Python, PHP, cũng như làm việc hiệu quả với các framework như Spring Boot, Django, Laravel.

**4. Phương pháp xác thực JWT**

Token-based Authentication sử dụng token để xác thực người dùng thay vì lưu trữ phiên trên máy chủ. Khi người dùng đăng nhập, hệ thống cấp một token để người dùng gửi kèm trong mỗi request tiếp theo (thường là JWT - JSON WebToken). Token này được mã hóa và có thể chứa thông tin về người dùng. Không cần lưu trạng thái trên máy chủ. Giảm tải cho máy chủ trong việc quản lý phiên.

JWT (JSON Web Token) là một phương pháp xác thực không trạng thái, trong đó token chứa thông tin người dùng được mã hóa và có thể xác minh mà không cần truy vấn cơ sở dữ liệu. JWT là một tiêu chuẩn mã hóa và xác thực thông tin giữa hai bên client và server dưới dạng chuỗi JSON. JWT được sử dụng phổ biến trong xác thực người dùng (authentication).

Cấu trúc JWT gồm 3 phần chính được ngăn cách bằng dấu chấm (.): HEADER.PAYLOAD.SIGNATURE  
 (Ví dụ:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

.

eyJ1c2VybmFtZSI6ImFkbWluIiwicm9sZSI6IkFETUlOIn0

.

JH1Qh2o4X3Jv5K3a3KLN4d23mP0eXjGpQ

)

+ Header chứa loại thuật toán mã hóa (alg) và kiểu token (typ):

{

"alg": "HS256",

"typ": "JWT"

}

+ Payload chứa dữ liệu của người dùng (claims) tùy chỉnh:  
 {

"username": "admin",

"role": "ADMIN",

"exp": 1713443200

}

+ Signature là mã hóa bảo mật giúp JWT không bị giả mạo, signature được mã hóa từ việc dùng thuật toán HMACSHA256 mã hóa bao gồm 3 thành phần: header đã mã hóa base64, payload đã mã hóa base64, và một secret key:

HMACSHA256( base64UrlEncode(header) + "." + base64UrlEncode(payload), secret\_key )

**5. Chatbot và kỹ thuật RAG**

Để có thể tạo ra chatbot một cách nhanh chóng mà chỉ cần hiểu cơ bản các kiến thức về Machine Learning (ML), Deep Learning (DL), hay các mô hình LLM thì với cách gọi API ChatGPT và tích hợp vào hệ thống là một lựa chọn tốt. API ChatGPT là một dịch vụ mạnh mẽ được phát triển bởi OpenAI, sử dụng mô hình ngôn ngữ tiên tiến như GPT-4 để xử lý ngôn ngữ tự nhiên. ChatGPT có khả năng hiểu ngữ cảnh câu hỏi, sinh câu trả lời tự nhiên và tạo văn bản một cách mượt mà. Với sự huấn luyện trên một lượng dữ liệu khổng lồ, ChatGPT có thể tự động phản hồi các truy vấn của người dùng mà không cần phải áp dụng thêm các phương pháp xử lý ngôn ngữ thủ công, mang đến khả năng giao tiếp tự nhiên và linh hoạt trong nhiều lĩnh vực ứng dụng.

Retrieval-Augmented Generation (RAG) là một kỹ thuật kết hợp giữa khả năng tìm kiếm thông tin (retrieval) và khả năng sinh ngôn ngữ (generation). RAG cho phép hệ thống truy xuất thông tin từ các nguồn dữ liệu bên ngoài (như cơ sở dữ liệu, tài liệu nội bộ) và sử dụng mô hình ngôn ngữ để tạo ra câu trả lời hoặc văn bản phù hợp. Kỹ thuật này giúp cải thiện độ chính xác và tính kịp thời của các câu trả lời, đặc biệt trong các ứng dụng chatbot, nơi việc cung cấp thông tin nhanh chóng và chính xác là rất quan trọng.

*Các nghiên cứu, bài báo, tài liệu tham khảo có liên quan đến đề tài.*

Bài viết của công ty izisolution: <https://izisolution.vn/ai-chatbot-trong-quan-ly-nhan-su-co-hoi-va-thach-thuc/?utm_source=chatgpt.com>

Bài blog của một trung tâm tin học: <https://csc.edu.vn/tin-tuc/Blog-chia-se/kham-pha-rag-huong-dan-xay-dung-chatbot-voi-rag-8433?utm_source=chatgpt.com>

**IV. Phương pháp nghiên cứu**

Phương pháp thu thập tài liệu: được áp dụng để thu thập thông tin từ các nguồn học thuật, tài liệu chuyên ngành về thông tin nghiệp vụ y tế online, chatbot, kỹ thuật RAG. Việc nghiên cứu các công trình trước đây giúp xây dựng nền tảng lý thuyết vững chắc và xác định hướng tiếp cận phù hợp.

Phương pháp thử nghiệm và kiểm thử: Sau khi xây dựng hệ thống đặt lịch khám kết hợp chatbot sẽ thực hiện các thử nghiệm với nhiều tình huống thực tế khác nhau. Chatbot sẽ được đánh giá dựa trên các tiêu chí như độ chính xác của câu trả lời, tốc độ phản hồi, khả năng hiểu ngữ cảnh và sự hài lòng của người dùng.

**V. Kết quả dự kiến**

Sau khi hoàn thành đề tài, hệ thống đặt lịch khám kết hợp chatbot hỗ trợ bệnh nhân dự kiến sẽ đạt được nhiều kết quả quan trọng, góp phần nâng cao hiệu quả mô hình hoạt động y tế online. Trước tiên, một chatbot thông minh cùng với kỹ thuật Retrieval-Augmented Generation (RAG) sẽ được phát triển và triển khai trên nền tảng web của hệ thống đặt lịch khám bệnh. Hệ thống được thiết kế giao diện dễ dùng giúp người dùng và người bận rộn dễ dàng theo dõi và đặt lịch theo khung giờ hợp lý theo nhu cầu cá nhân. Chatbot sẽ có khả năng tự động tìm kiếm, trích xuất thông tin từ dữ liệu nội bộ của cơ sở y tế và cung cấp câu trả lời chính xác, nhanh chóng theo từng ngữ cảnh cụ thể.

Bên cạnh đó, hệ thống đặt lịch khám tích hợp chatbot được kỳ vọng sẽ đáp ứng được nhu cầu chủ động thời gian thăm khám của bệnh nhân, giúp họ giảm thiểu thời gian phải chờ đợi nếu thăm khám trực tiếp tại cơ sở y tế. Nhờ vào khả năng phản hồi tự động và chính xác, chatbot sẽ góp phần nâng cao trải nghiệm người dùng, giúp bệnh nhân dễ dàng tiếp cận thông tin mà không cần chờ đợi lâu.

Tóm lại, kết quả mong đợi sau khi hoàn thành đề tài là việc xây dựng một hệ thống đặt lịch khám kết hợp chatbot thông minh hướng đến việc tạo ra một hệ thống hỗ trợ bệnh nhân tốt hơn, giúp bệnh viện tối ưu quy trình làm việc, tiết kiệm thời gian và nâng cao sự hài lòng của bệnh nhân.

**VI. Đóng góp của đề tài**

*Trình bày các đóng góp của đề tài đối với lĩnh vực nghiên cứu và thực tiễn.*

Về mặt nghiên cứu, đề tài đóng góp vào việc ứng dụng kỹ thuật Retrieval-Augmented Generation (RAG) kết hợp chatbot trong việc xây dựng hệ thống hỗ trợ đặt lịch khám bệnh. Đây là một hướng tiếp cận mới, giúp chatbot không chỉ dựa vào tri thức có sẵn của mô hình ngôn ngữ mà còn có khả năng tìm kiếm, trích xuất và tổng hợp thông tin từ cơ sở dữ liệu nội bộ của hệ thống y tế, như thông tin bác sĩ, chuyên khoa, khung giờ khám và quy trình đặt lịch. Điều này đảm bảo độ chính xác, tính cập nhật và khả năng tư vấn phù hợp với nhu cầu thực tế của bệnh nhân. Ngoài ra, đề tài còn đánh giá hiệu quả của chatbot khi triển khai trong môi trường đặt lịch khám trực tuyến, qua đó cung cấp cơ sở cho các nghiên cứu tiếp theo về tối ưu hóa chatbot trong lĩnh vực y tế điện tử. Kết quả nghiên cứu cũng có thể mở rộng ứng dụng cho các lĩnh vực khác như hỗ trợ khách hàng trong chăm sóc sức khỏe, tư vấn y khoa tự động hoặc quản lý thông tin bệnh viện thông minh.

Về mặt thực tiễn, hệ thống đặt lích khám kết hợp chatbot giúp tự động hóa quá trình hỗ trợ bệnh nhân, giảm tải công việc cho nhân viên tổng đài và nâng cao trải nghiệm người dùng. Bệnh nhân có thể tra cứu thông tin bác sĩ, chuyên khoa, thời gian trống, quy trình khám, hoặc đặt lịch trực tiếp thông qua hội thoại tự nhiên với chatbot, mà không cần thao tác phức tạp hay chờ đợi hỗ trợ từ nhân viên. Điều này không chỉ tiết kiệm thời gian và tối ưu hóa nguồn lực cho cơ sở y tế, mà còn tăng tính chủ động và hài lòng của bệnh nhân, góp phần xây dựng hệ thống khám chữa bệnh hiện đại, thân thiện và thông minh. Như vậy, đề tài không chỉ mang ý nghĩa khoa học khi nghiên cứu việc ứng dụng RAG và chatbot cho hệ thống trực tuyến trong lĩnh vực y tế, mà còn có giá trị thực tiễn cao khi hỗ trợ chuyển đổi số trong chăm sóc sức khỏe, giúp nâng cao hiệu quả vận hành, tối ưu quy trình đặt lịch và cải thiện chất lượng dịch vụ y tế.

**VII. Cấu trúc đồ án**

Mô tả chi tiết về cấu trúc của đồ án, bao gồm các chương, phần, mục.

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

**1.1. Mục 1 chương 1**

1.1.1 Mục 1 chấm 1 chương 1

1.1.2 Mục 1 chấm 2 chương 1  
**1.2 Mục 2 chương 1**

**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1. Mục 1 chương 2**

2.1.1 Mục 1 chấm 1 chương 2  
**2.2 Mục 2 chương 2**

**CHƯƠNG 3: XÂY DỰNG MÔ HÌNH RAG**

**CHƯƠNG 4: XÂY DỰNG HỆ THỐNG**

**CHƯƠNG 5: KẾT LUẬN**

**VIII. Tài liệu tham khảo**

Liệt kê các tài liệu tham khảo được sử dụng trong quá trình nghiên cứu và viết đồ án.

**IX. Kế Hoạch thực hiện và tiến độ nghiên cứu**

Thời gian và nội dung công công việc theo tuần.

|  |  |  |
| --- | --- | --- |
| **Thời gian** | **Nội dung công việc** | **Ghi chú** |
| Tuần 1 (20/10 - 26/10) | Chọn đề tài, xây dựng đề cương. |  |
| Tuần 2 (27/10 – 02/11) | Xây dựng giao diện website |  |
| Tuần 3 (03/11 – 09/11) | Xây dựng giao diện website |  |
| Tuần 4 (10/11 -16/11) | Xây dựng logic website |  |
| Tuần 5 (17/11 -23/11) | Xây dựng logic website |  |
| Tuần 6 (24/11 - 30/11) | Tìm hiểu RAG và chatbot |  |
| Tuần 7 (01/12 - 07/12) | Tìm hiểu RAG và chatbot |  |
| Tuần 8 (08/12 - 14/12) | Áp dụng RAG vào hệ thống |  |
| Tuần 9 (15/12 - 21/12) | Áp dụng RAG vào hệ thống |  |
| Tuần 10 (22/12 - 28/12) | Kiểm thử và hoàn thiện hệ thống |  |
| Tuần 11 (29/12 - 04/01) | Kiểm thử và hoàn thiện hệ thống |  |
| Tuần 12 (05/01 - 11/01) | Hoàn thành báo cáo đề tài |  |
| Tuần 13 (12/01 - 18/01) | Hoàn thiện đề tài |  |

Tiến độ công việc

|  |  |
| --- | --- |
| **Thời gian** | **Tiến độ** |
|  |  |

Github:

|  |  |  |
| --- | --- | --- |
|  |  | ……ngày….tháng….năm…. |
| **Trưởng Bộ Môn** | **Ý kiến của GVHD** | **Sinh viên thực hiện** |
| **Nguyễn Văn A** | **Nguyễn Văn B** | **Nguyễn Văn C** |