

BÁO CÁO ĐỒ ÁN MÔN HỌC

Discord-Lite

Ngành: **CÔNG NGHỆ THÔNG TIN**

Tên học phần: **LẬP TRÌNH MẠNG MÁY TÍNH**

Giảng viên hướng dẫn : Đoàn Khải Hiếu

Sinh viên thực hiện : 4

Dương Công Phước 2280602494

Nguyễn Thành Thi 2280603023

Huỳnh Tấn Phát 2280602291

Lê Quốc Việt 2280603663

Lớp: 22DTHA1

TP. Hồ Chí Minh, 2025

BÁO CÁO ĐỒ ÁN MÔN HỌC

Discord-Lite

Ngành: **CÔNG NGHỆ THÔNG TIN**

Tên học phần: **LẬP TRÌNH MẠNG MÁY TÍNH**

Giảng viên hướng dẫn : Đoàn Khải Hiếu

Sinh viên thực hiện : 4

Dương Công Phước 2280602494

Nguyễn Thành Thi 2280603023

Huỳnh Tân Phát 2280602291

Lê Quốc Việt 2280603663

Lớp: 22DTHA1

TP. Hồ Chí Minh, 2025

LỜI CAM ĐOAN

Chúng em xin cam đoan rằng đồ án này là kết quả làm việc của bản thân và nhóm. Em không sao chép nội dung từ bất kỳ nguồn tài liệu nào mà không trích dẫn. Mọi bài viết hoặc dữ liệu tham khảo nếu có đã được chân thực ghi rõ trong danh mục tài liệu tham khảo. Em cam kết tuân thủ các quy định về đạo đức nghiên cứu và học thuật. Chúng em xin chịu hoàn toàn trách nhiệm với nội dung và kết quả của đồ án này, đồng thời sẵn sàng giải trình nếu có bất kỳ yêu cầu nào liên quan đến tính minh bạch và trung thực của đồ án.

MỤC LỤC

LỜI CAM ĐOAN	III
MỤC LỤC	1
DANH MỤC HÌNH ẢNH, HÌNH VẼ VÀ ĐỒ THỊ.....	2
CHƯƠNG 1: TỔNG QUAN	3
1.1 GIỚI THIỆU VỀ ĐỀ TÀI.....	3
1.2 NHIỆM VỤ ĐỒ ÁN	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 KHÁI NIỆM VÀ PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ	4
2.1.1 Khái niệm về các công nghệ sử dụng	4
2.1.2 Phương pháp giải quyết vấn đề	4
2.2 XÂY DỰNG MÔ HÌNH VÀ GIẢI PHÁP	4
2.2.1. Mô hình tổng quan hệ thống.....	4
2.2.2. Giải pháp kỹ thuật	5
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM	6
3.1 KẾT QUẢ NGHIÊN CỨU VÀ THỰC NGHIỆM HỆ THỐNG	6
3.2 SƠ LƯỢC PHẦN MỀM.....	6
3.3 DEMO CHƯƠNG TRÌNH	7
CHƯƠNG 4 : KẾT LUẬN VÀ KIẾN NGHỊ	9
4.1 KẾT LUẬN	16
4.2 KIẾN NGHỊ	16
TÀI LIỆU THAM KHẢO.....	18

DANH MỤC HÌNH ẢNH, HÌNH VẼ VÀ ĐỒ THỊ

HÌNH 3.3.1: TRANG ĐĂNG NHẬP	7
HÌNH 3.3.2: TRANG ĐĂNG KÝ	7
HÌNH 3.3.3: TRANG QUÊN MẬT KHẨU	8
HÌNH 3.3.4: TRANG CHỦ KHI ĐĂNG NHẬP VÀ JOIN PHÒNG	8
HÌNH 3.3.5: TRANG KẾT BẠN	10
HÌNH 3.3.6: NÚT ĐỂ TẠO RA SERVER TRÒ CHUYỆN	10
HÌNH 3.3.7: NÚT ĐỂ TẠO RA CODE MỜI VÀO SERVER TRÒ CHUYỆN	11
HÌNH 3.3.8: NÚT ĐỂ THAM GIA VÀO SERVER BẰNG CODE	12
HÌNH 3.3.9: PHẦN CẬP NHẬT TRẠNG THÁI HOẠT ĐỘNG VÀ CHỈNH SỬA HỒ SƠ NGƯỜI DÙNG	13
HÌNH 3.3.10: DEMO PHẦN CHAT TRẢ LỜI NGƯỜI DÙNG VÀ THẢ EMOJI	13
HÌNH 3.3.11: DEMO PHẦN CHAT GỬI CÁC LOẠI TỆP, VOICE, HÌNH ẢNH ,...	14
HÌNH 3.3.12: DEMO STREAM MÀN HÌNH	15

Chương 1: Tổng quan

1.1 Giới thiệu về đề tài

Hiện nay thời đại công nghệ thông tin đang phát triển rất mạnh mẽ, các ứng dụng giao tiếp trực tuyến đóng vai trò quan trọng trong việc kết nối con người với nhau. Vậy nên, Discord là một trong những nền tảng tiêu biểu được sử dụng rộng rãi nhờ khả năng tạo kênh trò chuyện, chia sẻ tệp tin và tương tác thời gian thực. Từ thực tế đó, nhóm chúng em thực hiện đề tài “Discord-Lite” một phiên bản thu gọn mô phỏng các chức năng cơ bản của Discord, tập trung vào giao tiếp mạng client-server, trao đổi dữ liệu thời gian thực, và quản lý người dùng.

1.2 Nhiệm vụ đồ án

Đồ án nhóm tập trung nghiên cứu và xây dựng một hệ thống trò chuyện trực tuyến hoạt động theo mô hình client-server với các chức năng cơ bản, cụ thể như sau:

- Xây dựng máy chủ (Server): Quản lý kết nối từ nhiều client cùng lúc, lưu trữ danh sách người dùng đang hoạt động, tiếp nhận và chuyển tiếp tin nhắn đến các client tương ứng.
- Phát triển ứng dụng khách (Client): Cung cấp giao diện đơn giản cho phép người dùng đăng nhập, gửi và nhận tin nhắn theo thời gian thực. Hỗ trợ gửi tin cá nhân và nhóm tin trong phòng.
- Quản lý dữ liệu người dùng: Thực hiện đăng ký tài khoản, đăng nhập và kiểm tra thông tin.
- Nghiên cứu, áp dụng các công nghệ mạng: Lập trình socket TCP/IP, xử lý đa luồng cho việc giao tiếp song song, và truyền dữ liệu theo mô hình Client–Server.

1.3 Tính cấp thiết và lý do hình thành đề tài

Trong xu hướng số hóa và làm việc trực tuyến hiện nay, các ứng dụng trò chuyện và trao đổi thông tin qua Internet đã trở thành công cụ không thể thiếu đối với nhiều người. Việc nghiên cứu, thiết kế và xây dựng một mô hình thu gọn như Discord-Lite giúp nhóm hiểu sâu hơn về:

- Cách thiết lập và duy trì kết nối mạng giữa nhiều thiết bị trong cùng một hệ thống.
- Cơ chế gửi – nhận dữ liệu thời gian thực và xử lý đồng bộ giữa các client.
- Ứng dụng các kiến thức lý thuyết về socket, và giao thức mạng vào sản phẩm thực tế.

Chương 2: Cơ sở Lý thuyết

2.1 Khái niệm và Phương Pháp giải quyết vấn đề

2.1.1 Khái niệm về các công nghệ sử dụng

Trong quá trình xây dựng hệ thống Discord Clone, nhóm sử dụng nhiều công nghệ hiện đại nhằm đảm bảo hiệu năng, tính bảo mật và khả năng mở rộng của ứng dụng. Một số công nghệ chính bao gồm:

- Node.js: Là môi trường chạy JavaScript phía server, giúp xử lý nhiều kết nối đồng thời một cách hiệu quả, phù hợp cho ứng dụng thời gian thực như chat, voice, video.
- Express.js: Framework giúp xây dựng API RESTful nhanh chóng, hỗ trợ tốt cho việc triển khai các route, middleware và xử lý yêu cầu HTTP.
- ReactJS: Thư viện JavaScript phía client để xây dựng giao diện người dùng động, linh hoạt và dễ bảo trì.
- MongoDB: Hệ quản trị cơ sở dữ liệu NoSQL lưu trữ dữ liệu dưới dạng JSON linh hoạt, phù hợp cho các đối tượng như người dùng, server, channel, tin nhắn, file.
- JWT (JSON Web Token): Dùng để xác thực và phân quyền người dùng an toàn, tránh rò rỉ thông tin đăng nhập.
- Bcrypt: Thư viện mã hóa mật khẩu, giúp tăng tính bảo mật trong quá trình đăng ký và đăng nhập.
- WebSocket: Giao thức giúp truyền dữ liệu hai chiều giữa client và server theo thời gian thực, phục vụ chức năng chat và thông báo.
- WebRTC: Công nghệ truyền thông peer-to-peer hỗ trợ gọi thoại, video và chia sẻ màn hình giữa các người dùng.
- SMTP (Simple Mail Transfer Protocol): Dùng để gửi email khôi phục mật khẩu cho người dùng.
- Cloud Storage (nếu có): Lưu trữ và phân phối các tệp tin (file, hình ảnh, voice message, video stream).

2.1.2 Phương pháp giải quyết vấn đề

Để mô phỏng và tái tạo các chức năng của Discord, nhóm áp dụng mô hình client-server, trong đó:

- Server đảm nhiệm vai trò trung tâm, xử lý toàn bộ logic nghiệp vụ, xác thực, và truyền dữ liệu.
- Client đóng vai trò giao diện, giúp người dùng tương tác với hệ thống thông qua các hành động như chat, gọi, chia sẻ file, v.v.
- Quá trình phát triển được thực hiện theo mô hình chia nhỏ module, mỗi thành viên phụ trách một phần chức năng (authentication, chat, voice/video, database, giao diện, báo cáo...) để dễ kiểm thử và tích hợp.

2.2 Xây dựng mô hình và giải pháp

2.2.1. Mô hình tổng quan hệ thống

Hệ thống được thiết kế theo kiến trúc Client - Server, với hai thành phần chính:

a. Mô tả các chức năng của Server

- Trong hệ thống này, Server đảm nhận vai trò xử lý chính, bao gồm:
- Xác thực và phân quyền người dùng: Sử dụng JWT để quản lý phiên đăng nhập, đảm bảo chỉ người dùng hợp lệ mới được truy cập.
- Cung cấp API quản lý: Bao gồm các API cho người dùng (User), server, channel, tin nhắn và file.

- Xử lý dữ liệu: Thực hiện CRUD cho các đối tượng như server, channel, và message; đồng thời quản lý trạng thái hoạt động của người dùng.
- Truyền thông thời gian thực: Sử dụng WebSocket để cập nhật tin nhắn, thông báo, và trạng thái online/offline của bạn bè hoặc thành viên trong server.
- Truyền tải và streaming: Hỗ trợ upload/download file, voice, và media streaming thông qua WebRTC.
- Quản lý cuộc gọi: Điều phối signaling giữa các client để thực hiện cuộc gọi voice/video, chia sẻ màn hình và live stream.
- Gửi mail tự động: Sử dụng SMTP để gửi email xác nhận hoặc khôi phục mật khẩu.

b. Mô tả chức năng của Client

- Client là phần giao diện người dùng (UI) chạy trên trình duyệt web, cung cấp các chức năng chính như:
- Đăng ký, đăng nhập, đăng xuất thông qua API xác thực của server.
- Quản lý server và channel: Tạo, chỉnh sửa, xóa hoặc tham gia server và channel.
- Giao tiếp và trò chuyện:
- Gửi và nhận tin nhắn văn bản, biểu tượng cảm xúc, file, voice.
- Nhận thông báo real-time khi có tin nhắn hoặc cuộc gọi đến.
- Thả react cho tin nhắn.
- Voice/Video Call và Streaming:
- Gọi thoại và video qua WebRTC.
- Bật/tắt mic, camera, chia sẻ màn hình hoặc live stream.
- Giao diện thân thiện và responsive:
- Hỗ trợ kéo-thả file để upload.
- Hiển thị danh sách bạn bè, trạng thái online/offline.
- Thông báo đẩy khi có tin nhắn hoặc cuộc gọi mới.

2.2.2. Giải pháp kỹ thuật

- Frontend: Sử dụng ReactJS để xây dựng giao diện động, kết hợp với CSS hoặc framework như Tailwind để tối ưu hiển thị.
- Backend: Node.js + Express để triển khai REST API và WebSocket Server.
- Database: MongoDB lưu trữ toàn bộ dữ liệu người dùng, tin nhắn, server, channel.
- Bảo mật: Kết hợp bcrypt (mã hóa mật khẩu) và JWT (token-based authentication).
- Realtime Communication: WebSocket và WebRTC giúp truyền tải dữ liệu thời gian thực, hỗ trợ nhắn tin, gọi thoại, video, và streaming.

Chương 3: Kết quả thực nghiệm

3.1 Kết quả nghiên cứu và thực nghiệm hệ thống

Trong quá trình thực hiện đề tài "Discord Lite," nhóm đã tiến hành nghiên cứu các mô hình lý thuyết về hệ thống giao tiếp trực tuyến và áp dụng mô hình UML để thiết kế hệ thống. Cụ thể, hệ thống được xây dựng dựa trên các nghiệp vụ chính như xác thực và phân quyền người dùng, quản lý server, channel, tin nhắn, file, upload/download, streaming media, cùng với hỗ trợ chat real-time qua WebSocket và voice/video call qua WebRTC. Các chức năng này được mô tả thông qua biểu đồ UML như Use Case Diagram, giúp xác định rõ ràng các yêu cầu nghiệp vụ và quy trình hoạt động của hệ thống.

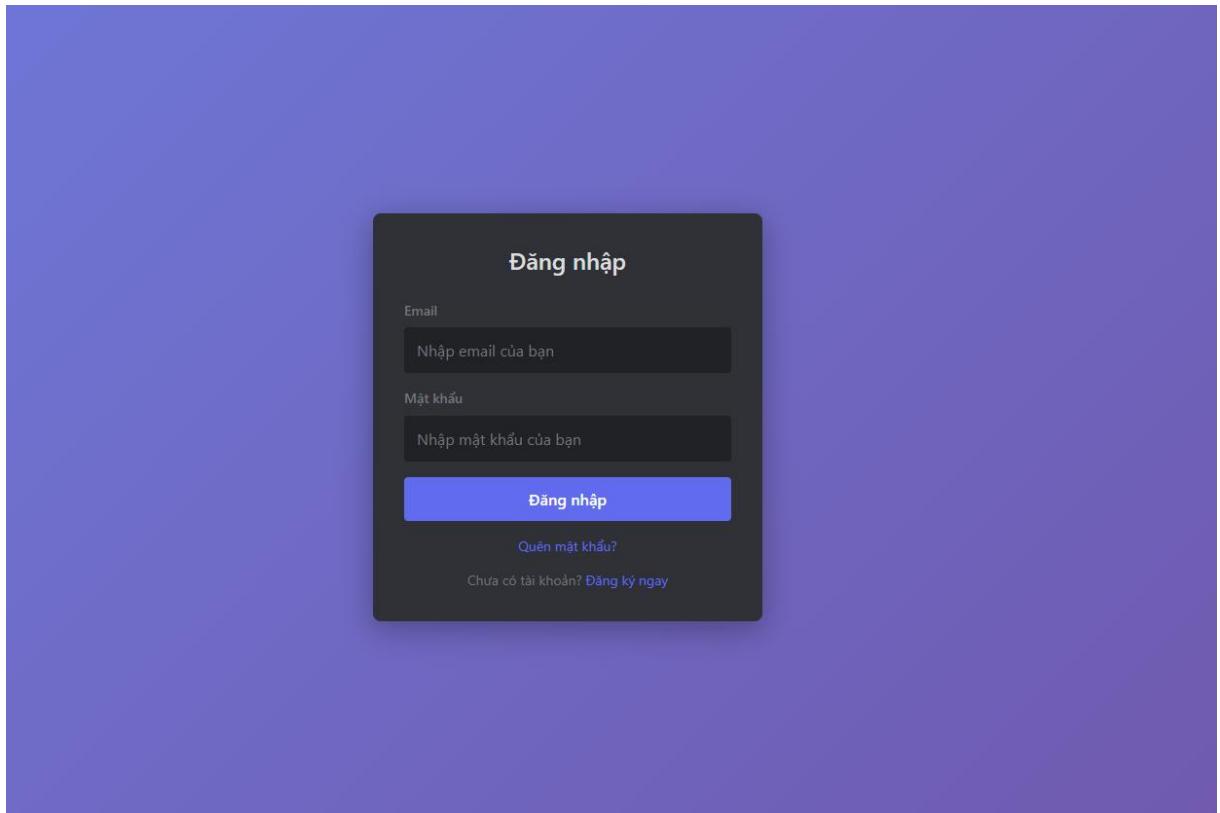
Về thực nghiệm, nhóm đã triển khai các chức năng chính như xác thực người dùng (bao gồm đăng ký, đăng nhập, quên mật khẩu), quản lý thông tin người dùng, server, channel, tin nhắn, file với các thao tác CRUD, cùng với các tính năng upload/download file, streaming media, và hỗ trợ voice/video call qua WebRTC. Hệ thống sử dụng WebSocket để đảm bảo chat và thông báo real-time hoạt động mượt mà. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng tốt các yêu cầu về tốc độ, bảo mật, và khả năng xử lý dữ liệu thời gian thực, đồng thời đảm bảo dễ dàng mở rộng và tích hợp trong tương lai.

3.2 Sơ lược phần mềm

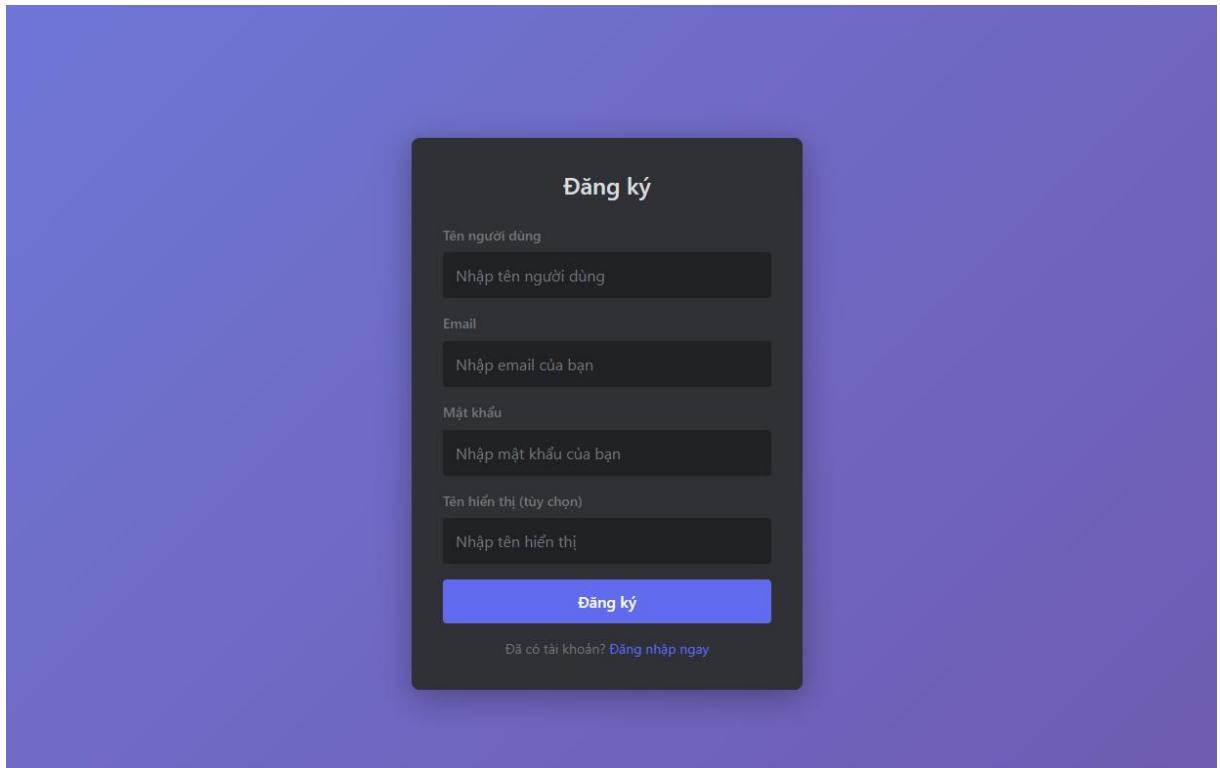
Phần mềm "Discord Lite" được thiết kế để hỗ trợ giao tiếp trực tuyến, giúp người dùng tương tác trong các cộng đồng thông qua server và channel. Phần mềm có các tính năng chính sau:

- Tính năng xác thực và phân quyền:** Hỗ trợ đăng ký, đăng nhập, quên mật khẩu, và phân quyền người dùng (Admin, thành viên).
- Quản lý người dùng, server, channel, tin nhắn, và file:** Bao gồm các thao tác thêm, xóa, sửa, tìm kiếm thông tin liên quan.
- Quản lý trang chủ:** Cung cấp thông kê chính xác về số lượng người dùng, server, channel, tin nhắn, và file được chia sẻ trong hệ thống.
- Quản lý giao tiếp thời gian thực:** Hỗ trợ chat real-time qua WebSocket, thông báo tức thời, và voice/video call thông qua WebRTC với cơ chế signaling và điều phối media streaming.
- Quản lý file và media:** Hỗ trợ upload/download file, streaming media, và xem chi tiết lịch sử chia sẻ file.

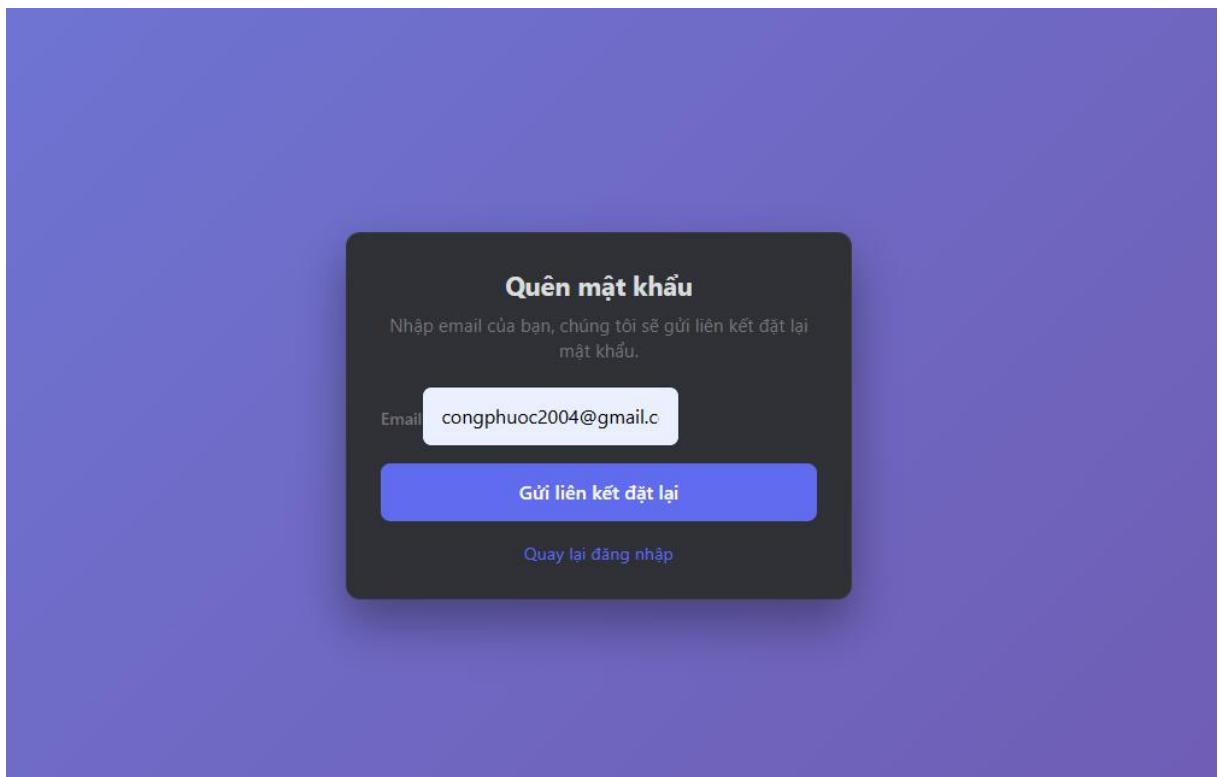
3.3 Demo chương trình



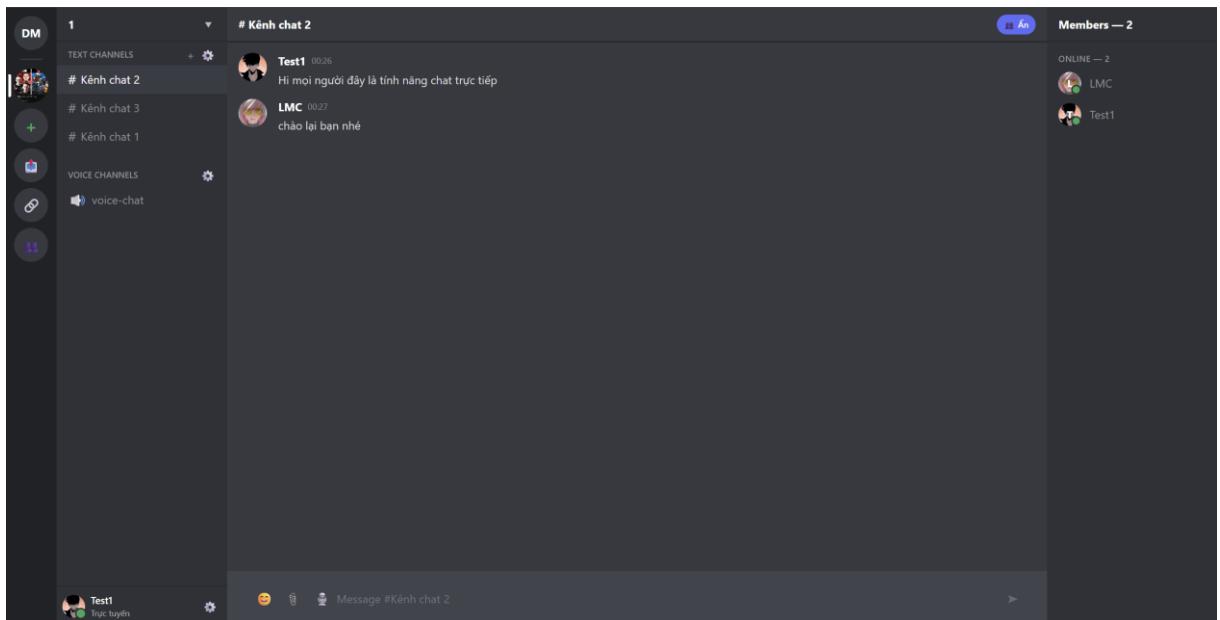
Hình 3.3.1: Trang đăng nhập



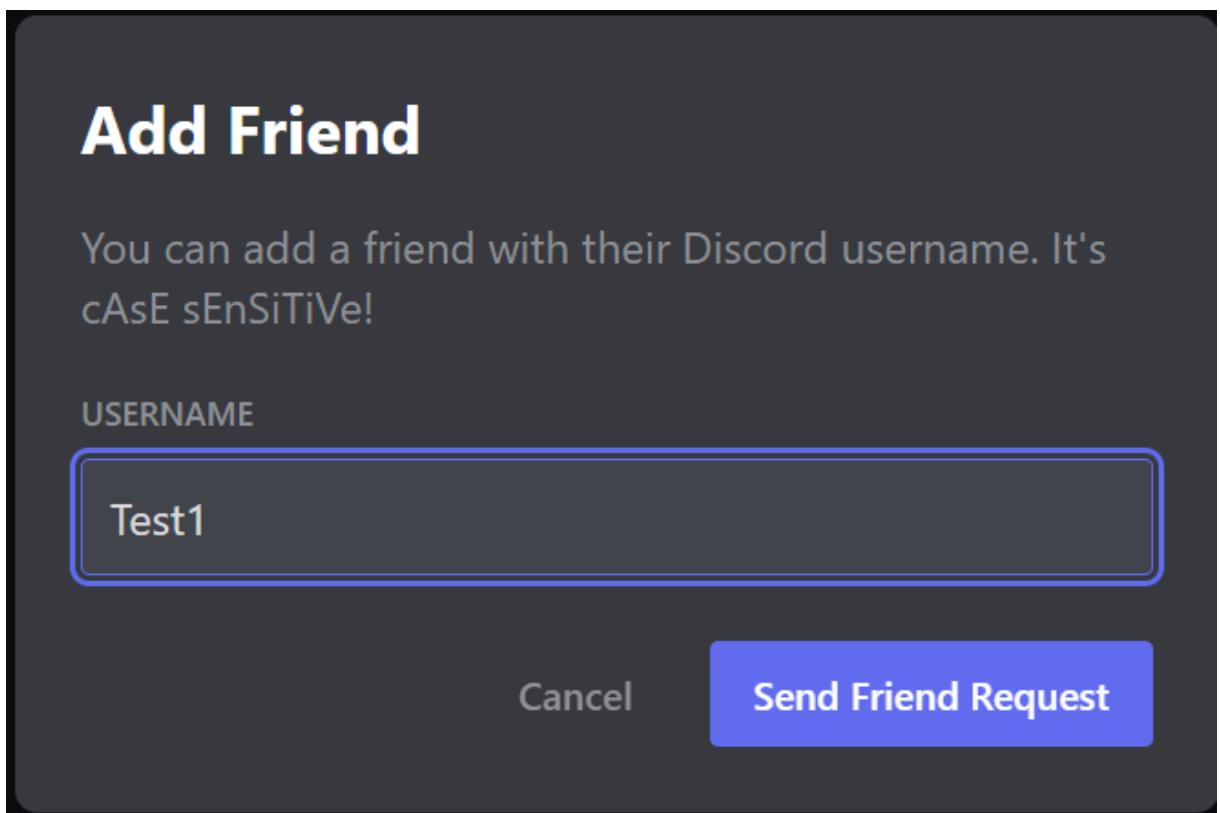
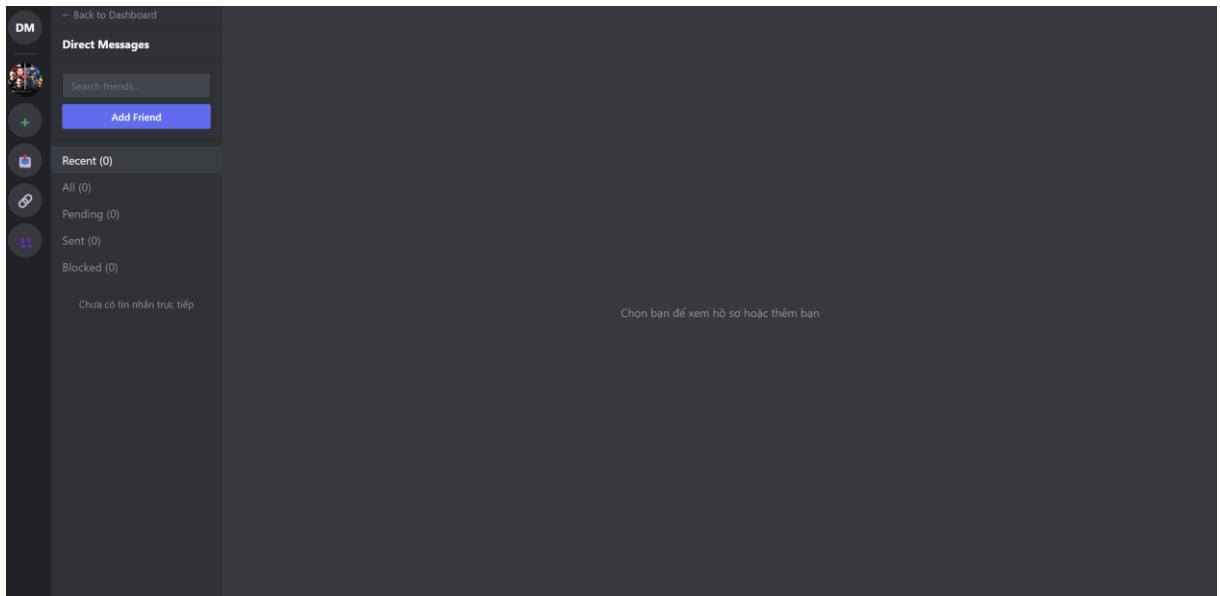
Hình 3.3.2: Trang đăng ký

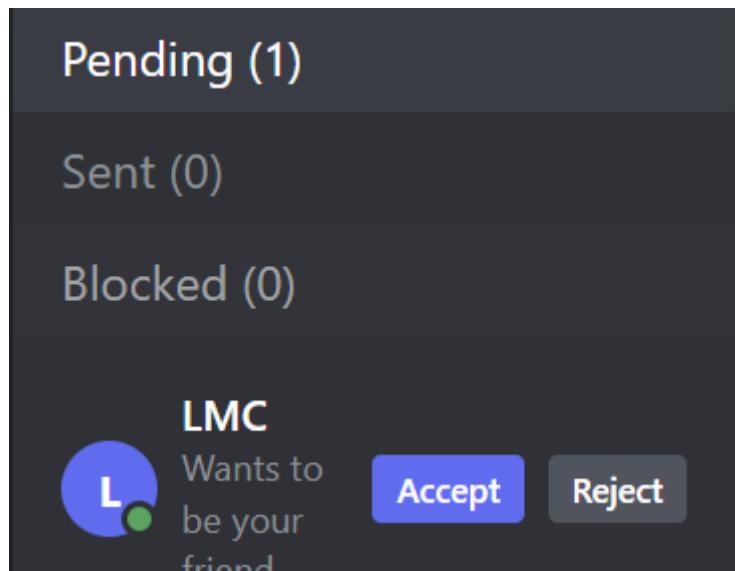


Hình 3.3.3: Trang quên mật khẩu

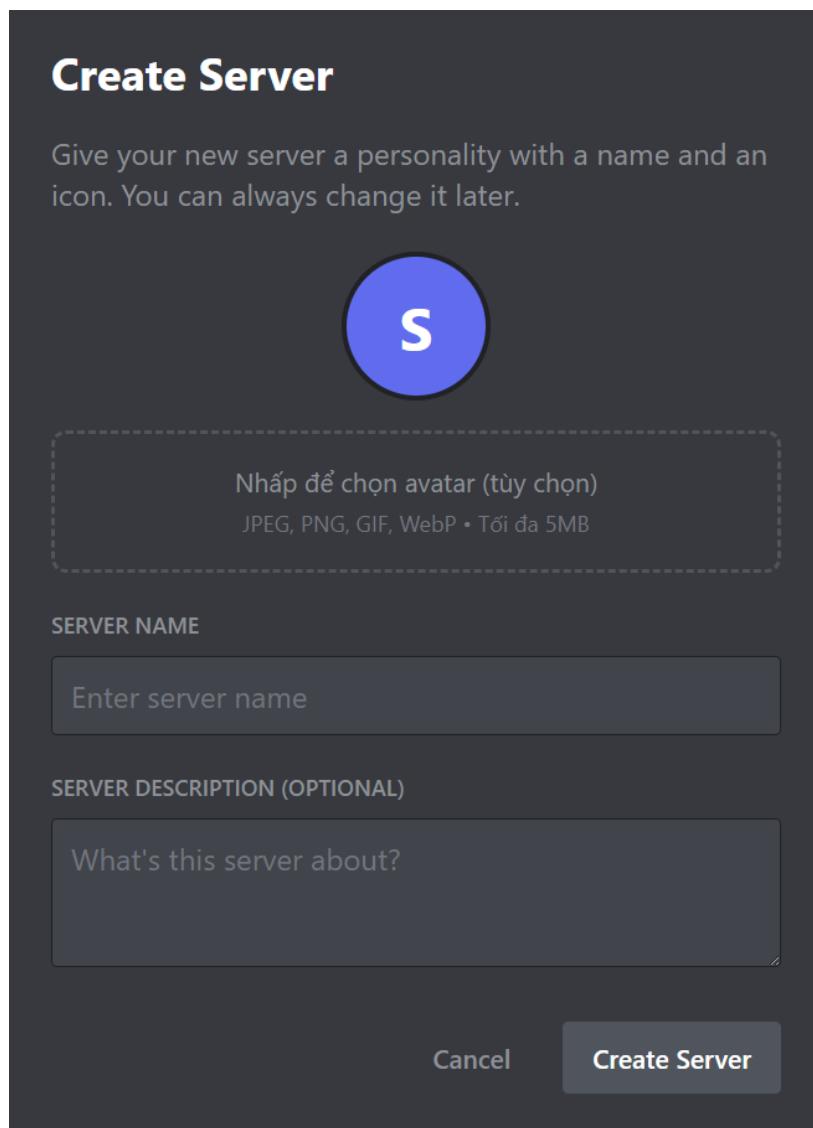


Hình 3.3.4: Trang chủ khi đăng nhập và join phòng





Hình 3.3.5: Trang kết bạn



Hình 3.3.6: Nút để tạo ra server trò chuyện

Invite People to 1

Create an invite link to share with your friends. They'll be able to join your server instantly.

Cancel

Generate Invite Link

Invite People to 1

Create an invite link to share with your friends. They'll be able to join your server instantly.

INVITE CODE

WT5MVVPV

Copy

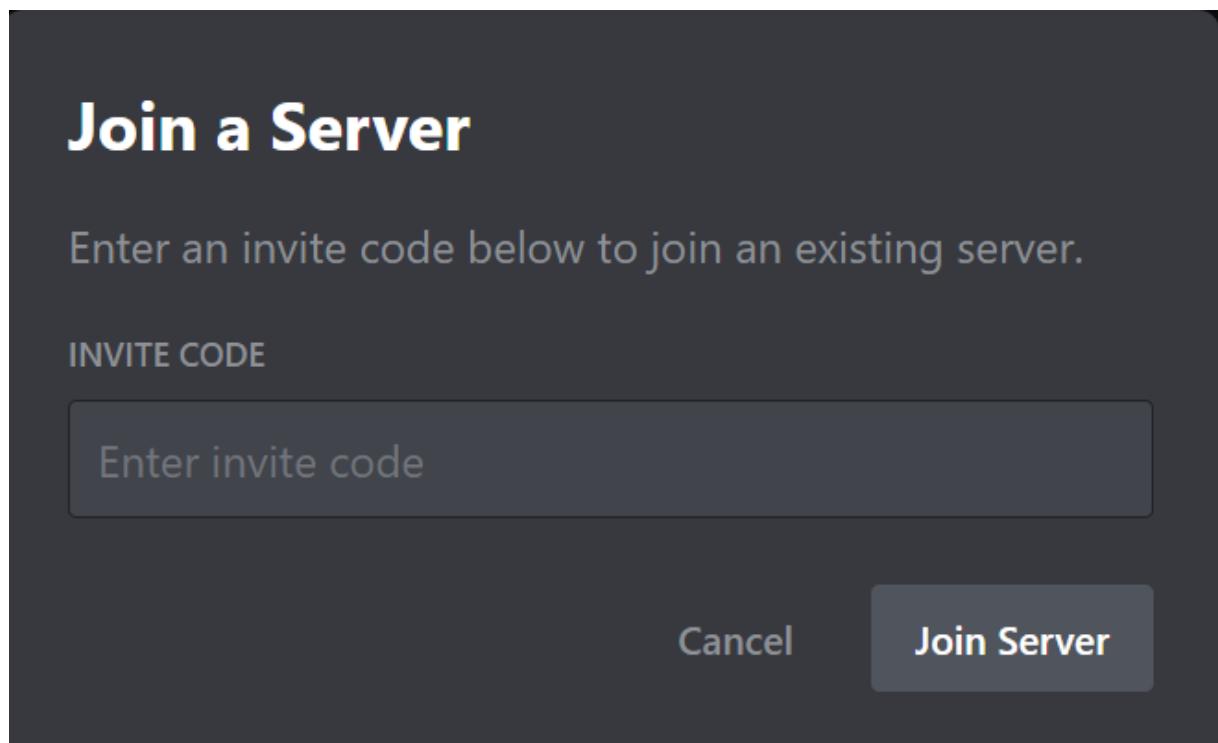
INVITE LINK

<http://localhost:3001/join/WT5MVVPV>

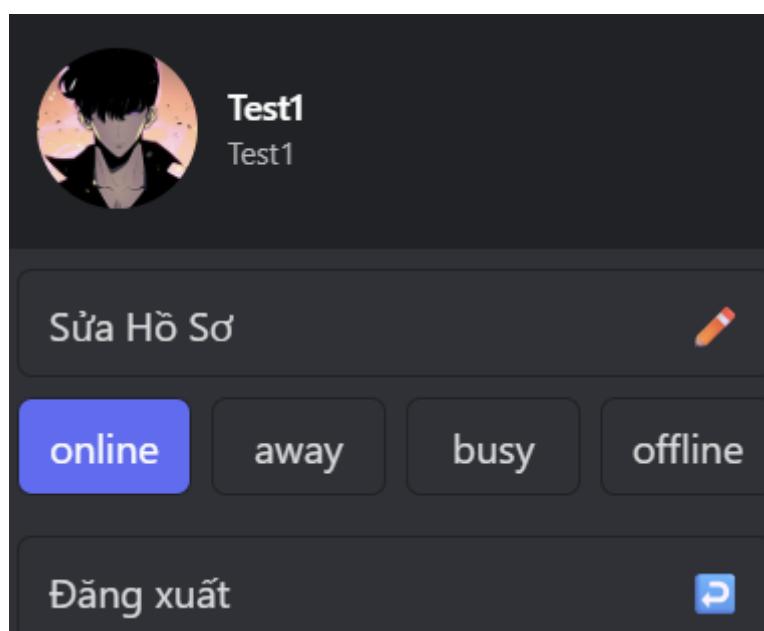
Copy Link

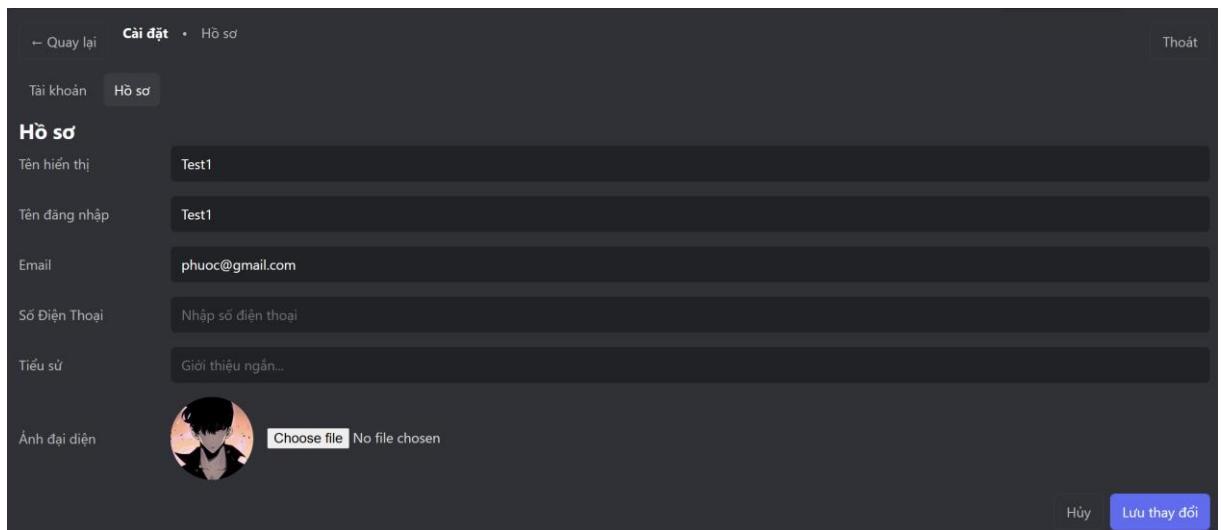
Done

Hình 3.3.7: Nút để tạo ra code mời vào server trò chuyện

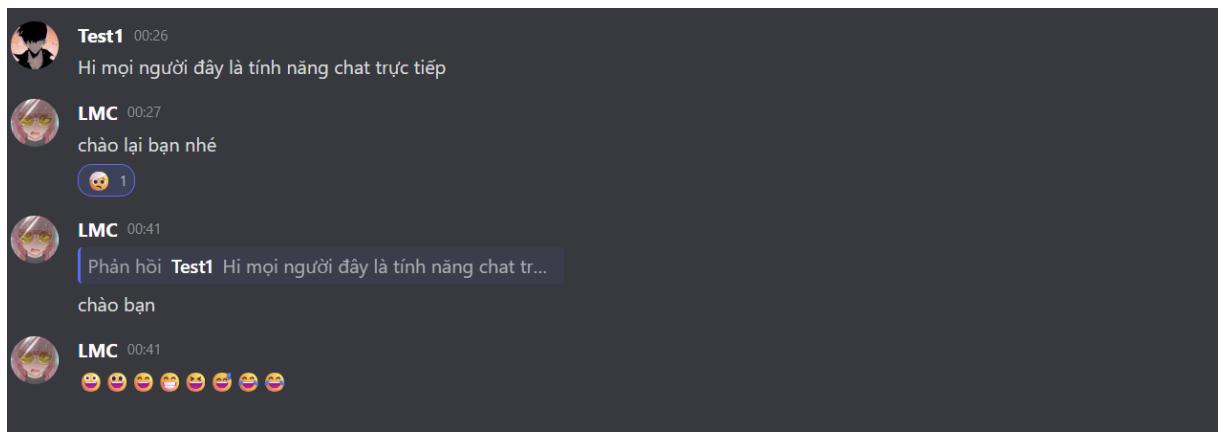


Hình 3.3.8: Nút để tham gia vào server bằng code

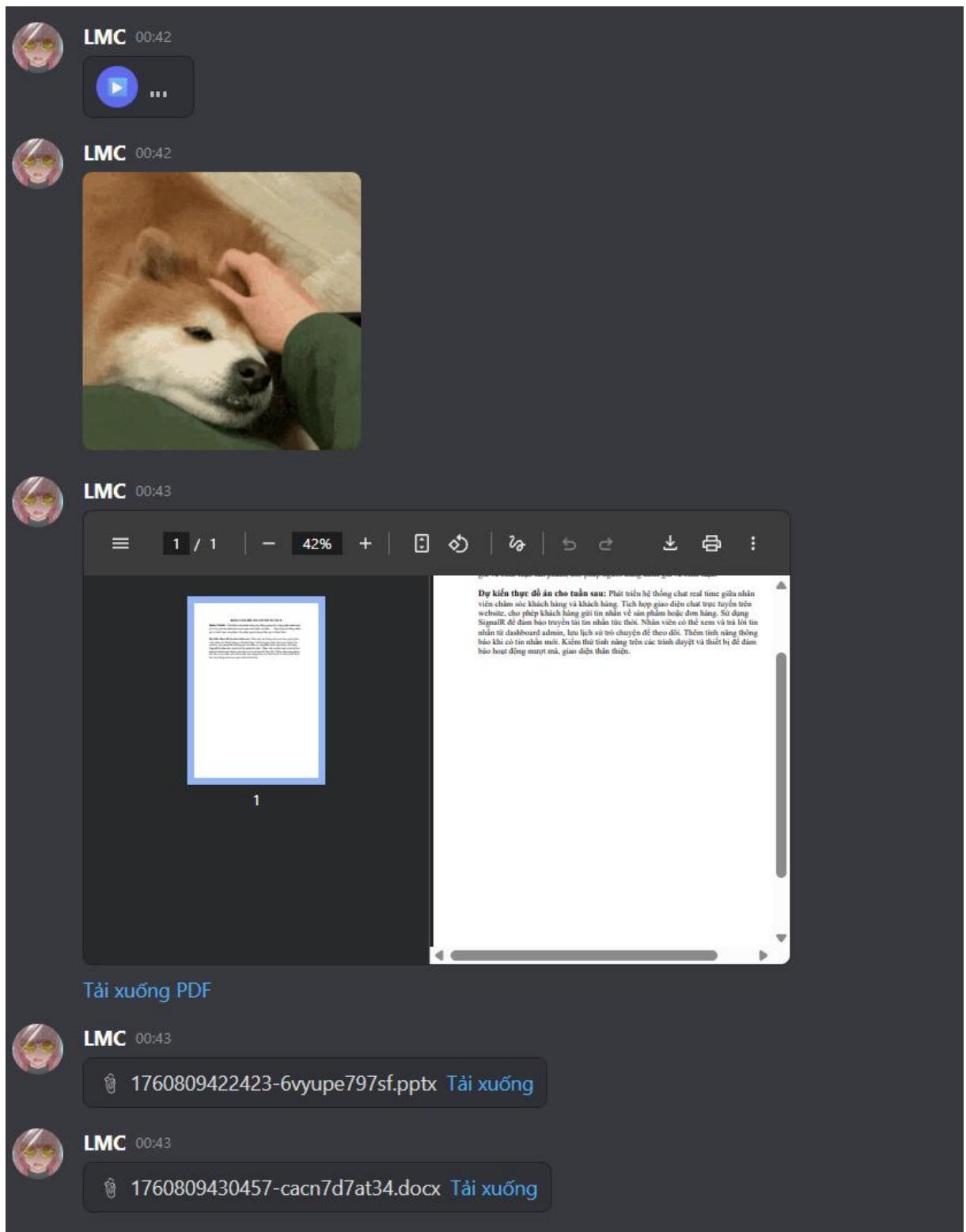




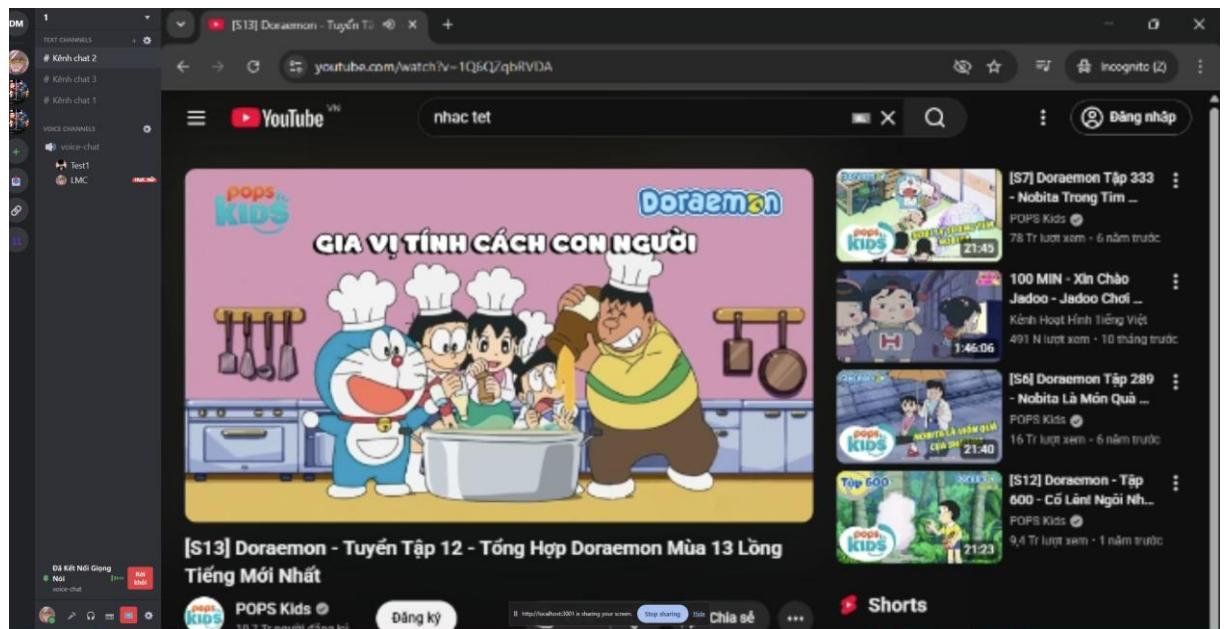
Hình 3.3.9: Phần cập nhật trạng thái hoạt động và chỉnh sửa hồ sơ người dùng



Hình 3.3.10: Demo phần chat trả lời người dùng và thả emoji



Hình 3.3.11: Demo phần chat gửi các loại tệp, voice, hình ảnh ,...



Hình 3.3.12: Demo stream màn hình

CHƯƠNG 4 : KẾT LUẬN VÀ KIẾN NGHỊ

4.1 Kết luận

Dự án "Discord Lite" đã hoàn thành mục tiêu đề ra, cung cấp một hệ thống giao tiếp trực tuyến hiệu quả với các chức năng quan trọng như xác thực và phân quyền người dùng, quản lý server, channel, tin nhắn, file, upload/download và streaming media. Hệ thống được xây dựng trên nền tảng server backend, sử dụng API để quản lý dữ liệu, WebSocket cho chat và thông báo real-time, đồng thời hỗ trợ signaling cho voice/video call qua WebRTC, đảm bảo tính chính xác và hiệu quả trong xử lý thông tin giao tiếp.

Hệ thống mang lại nhiều lợi ích thực tiễn như: Tự động hóa các quy trình giao tiếp: Giảm thiểu độ trễ, nâng cao tính bảo mật và chuyên nghiệp trong quản lý cộng đồng trực tuyến. Thông kê và quản lý dữ liệu chi tiết: Hỗ trợ theo dõi hoạt động người dùng, quản lý file/media, và xử lý thông báo thời gian thực một cách tiện lợi. Giao diện thân thiện: Dễ sử dụng và phù hợp với các đối tượng người dùng khác nhau, từ cá nhân đến nhóm lớn. Khả năng mở rộng: Sẵn sàng bổ sung các tính năng nâng cao trong tương lai. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng tốt các yêu cầu đề ra, đồng thời mở ra tiềm năng ứng dụng rộng rãi trong các nền tảng giao tiếp trực tuyến khác.

4.2 Kiến nghị

Mở rộng tính năng:

Tích hợp quản lý bot tự động để hỗ trợ moderation và các lệnh tùy chỉnh.

Bổ sung chức năng chia sẻ màn hình hoặc tích hợp với các dịch vụ bên thứ ba nhằm tăng tính tiện lợi cho người dùng.

Triển khai trên môi trường đa nền tảng:

Nghiên cứu và triển khai phiên bản di động hoặc ứng dụng desktop, giúp hệ thống có thể hoạt động mượt mà trên nhiều thiết bị.

Đảm bảo tính bảo mật khi xử lý dữ liệu nhạy cảm của người dùng, như mã hóa end-to-end cho voice/video call. Tối ưu hóa hiệu năng: Thực hiện kiểm thử trên nhiều môi trường mạng để cải thiện tốc độ và độ ổn định của WebSocket và WebRTC.

Tối ưu hóa server backend để giảm thời gian xử lý API và lưu trữ file/media.

Áp dụng trí tuệ nhân tạo:

Nghiên cứu tích hợp AI để phân tích dữ liệu chat, dự đoán hành vi người dùng, và gợi ý các tính năng cá nhân hóa hoặc tự động hóa moderation hiệu quả.

Dự án đã đáp ứng được các yêu cầu cơ bản và đặt nền móng vững chắc cho những cải tiến trong tương lai. Với các kiến nghị trên, hệ thống có thể được nâng cấp để phục vụ tốt hơn cho nhu cầu giao tiếp của người dùng và tối ưu hóa hiệu quả hoạt động.

TÀI LIỆU THAM KHẢO

- [1] <https://monamedia.co/entity-framework-la-gi/>
- [2]<https://bizflycloud.vn/tin-tuc/visual-studio-la-gi-nhung-tinh-nang-can-thiet-cua-visual-studio-20210324175835691.htm>
- [3]https://vi.wikipedia.org/wiki/Visual_Studio
- [4]<https://mso.vn/microsoft-sql-server-management-studio/>