



KHOA CÔNG NGHỆ THÔNG TIN
CSE703016 – GIAO DIỆN NGƯỜI MÁY

BÁO CÁO DỰ ÁN CUỐI KỲ

Tên dự án:	Xây dựng ứng dụng chat trực tuyến
Nhóm thực hiện:	Nhóm 03
Thành viên:	Trương Việt Anh - 20010695
	Bùi Văn Tuấn
	Nguyễn Tuấn Sơn
	Phan Văn Tình
	Phạm Mạnh Hùng

I. Giới thiệu dự án

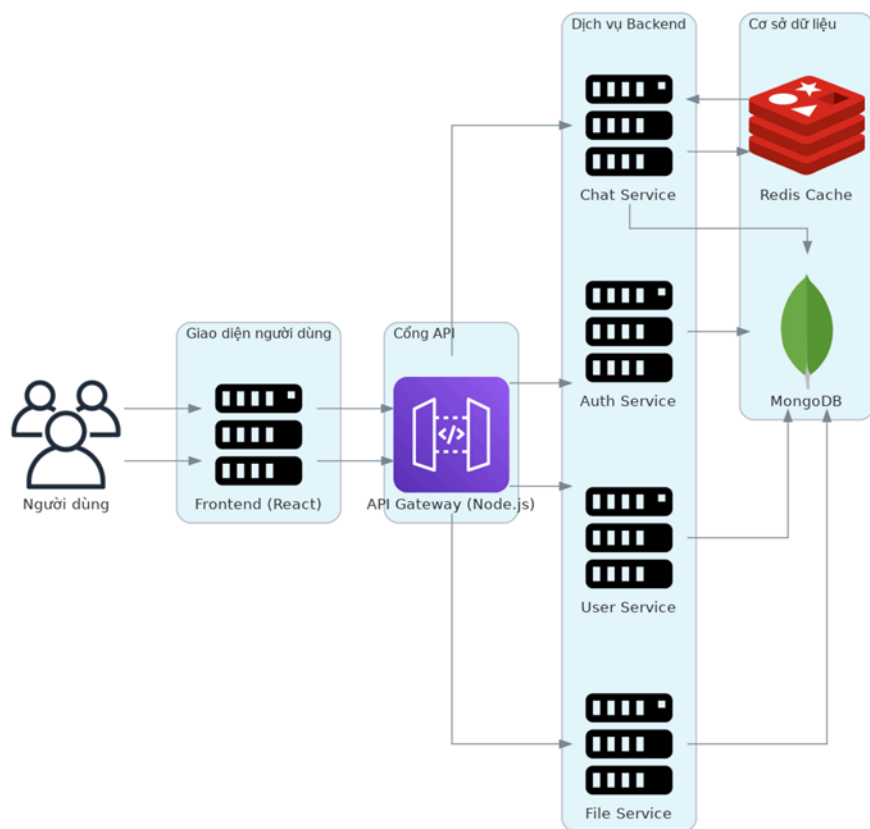
Dự án "Website ứng dụng chat trực tuyến" được xây dựng với mục tiêu cung cấp một nền tảng giao tiếp trực tiếp giữa người dùng và các nhân viên hỗ trợ khách hàng, giúp tối ưu hóa quá trình trao đổi thông tin và giải quyết các vấn đề của khách hàng một cách nhanh chóng và hiệu quả. Website này được phát triển bằng hai công nghệ chính: ReactJS cho phần giao diện người dùng (frontend) và NodeJS cho phần máy chủ (backend).

Trong bối cảnh công việc ngày càng đòi hỏi sự kết nối liên tục giữa các thành viên trong nhóm, đặc biệt là các nhóm làm việc từ xa và các tổ chức công nghệ, việc sử dụng nhiều ứng dụng khác nhau để giao tiếp, quản lý công việc và theo dõi tiến độ dự án đã trở thành một vấn đề phổ biến. Điều này không chỉ gây ra sự phân tán thông tin mà còn làm mất thời gian chuyển đổi giữa các ứng dụng, ảnh hưởng đến hiệu quả công việc. Để giải quyết vấn đề này, chúng tôi phát triển ứng dụng nhắn tin trực tuyến tích hợp các công cụ làm việc như GitHub, Trello, Jira,... vào trong chính nền tảng chat. Điều này giúp người dùng có thể nhận thông báo, quản lý công việc và cập nhật tiến độ dự án ngay trong ứng dụng nhắn tin mà không cần phải chuyển đổi qua lại giữa các phần mềm khác nhau. Tính năng này không chỉ tối ưu hóa quy trình làm việc mà còn giúp giảm thiểu sự phân tán thông tin, tạo điều kiện thuận lợi cho việc phối hợp và làm việc nhóm hiệu quả hơn.

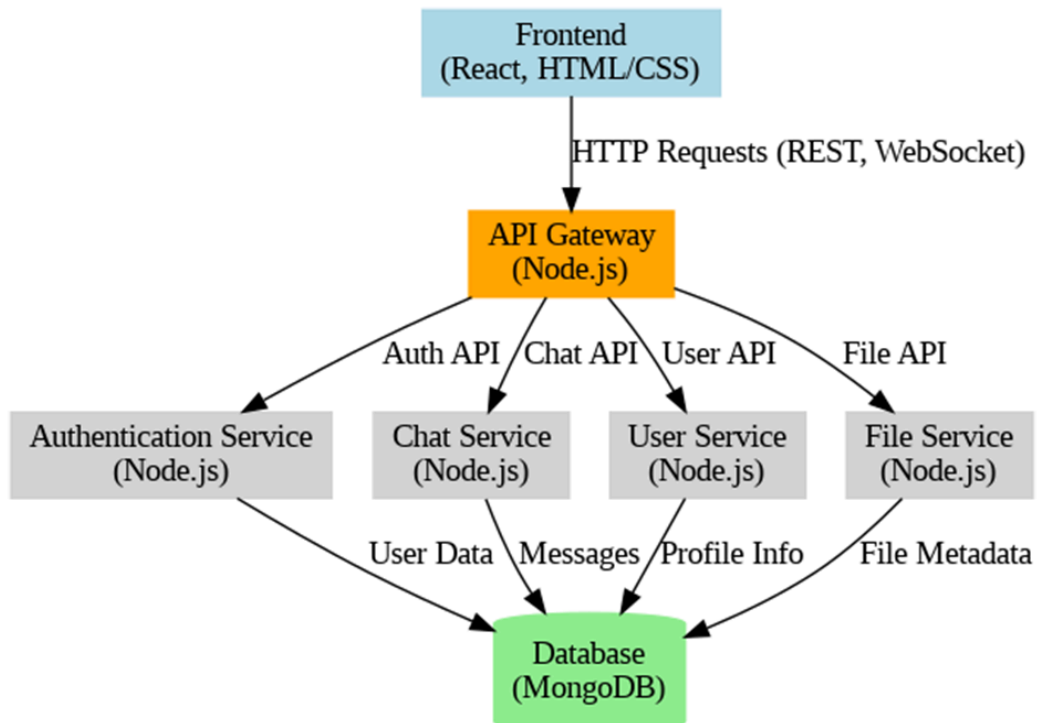
Mục tiêu của dự án là xây dựng một nền tảng nhắn tin không chỉ phục vụ cho việc trao đổi thông tin mà còn hỗ trợ tối ưu công tác quản lý công việc, dự án và giao tiếp nhóm một cách trực quan, tiết kiệm thời gian và nâng cao hiệu quả làm việc.

II. Cấu trúc chương trình

a. Sơ đồ kiến trúc của chương trình



Sơ đồ kiến trúc phần mềm chat Wave



b. Mô tả kiến trúc

1. Frontend (Giao diện người dùng):

Mô tả: Đây là phần người dùng tương tác trực tiếp qua trình duyệt hoặc ứng dụng di động.

Công nghệ: Sử dụng React và HTML/CSS để xây dựng giao diện người dùng.

Chức năng:

- Đăng ký, đăng nhập tài khoản.
- Chat thời gian thực (gửi/nhận tin nhắn, tải tệp).
- Kết bạn, tạo nhóm chat.
- Chỉnh sửa thông tin cá nhân.

Giao tiếp: Kết nối với API Gateway thông qua HTTP REST API hoặc WebSocket để nhận/gửi dữ liệu.

2. API Gateway:

Mô tả: Cầu nối giữa Frontend và Backend, định tuyến các yêu cầu đến đúng dịch vụ.

Công nghệ: Xây dựng bằng Node.js.

Chức năng:

- Tiếp nhận yêu cầu từ Frontend và chuyển tiếp đến các dịch vụ Backend.
- Xử lý giao thức REST cho các API hoặc WebSocket để hỗ trợ chat thời gian thực.
- Cung cấp cơ chế bảo mật cho các API như xác thực (token-based authentication).

3. Backend Services (Dịch vụ xử lý):

Các dịch vụ xử lý được triển khai theo kiến trúc microservices, mỗi dịch vụ đảm nhiệm một chức năng riêng biệt.

+ **Authentication Service:**

Mô tả: Xử lý xác thực và quản lý người dùng.

Chức năng:

- Đăng nhập/đăng ký tài khoản.
- Quản lý phiên làm việc bằng JWT (JSON Web Token).

+ **Chat Service:**

Mô tả: Xử lý tất cả các hoạt động liên quan đến gửi và nhận tin nhắn.

Chức năng:

- Gửi và nhận tin nhắn thời gian thực.
- Hỗ trợ gửi tệp tin (hình ảnh, video).
- Quản lý nhóm chat.

+ **User Service:**

Mô tả: Quản lý thông tin hồ sơ người dùng.

Chức năng:

- Cập nhật thông tin cá nhân (ảnh đại diện, trạng thái).
- Quản lý danh sách bạn bè.

+ **File Service:**

Mô tả: Quản lý việc tải lên và lưu trữ các tệp.

Chức năng:

- Lưu trữ metadata của tệp.
- Tích hợp với hệ thống lưu trữ đám mây (có thể là AWS S3 hoặc Google Cloud Storage).

4. Database (Cơ sở dữ liệu):

Công nghệ: Sử dụng MongoDB (NoSQL) để lưu trữ dữ liệu.

Chức năng:

- Lưu trữ thông tin người dùng, nhóm chat, và danh sách bạn bè.
- Lưu trữ tin nhắn và tệp tin metadata.
- Khả năng mở rộng tốt để xử lý lượng lớn dữ liệu và người dùng.

5. Các cơ chế giao tiếp:

- REST API: Sử dụng để thực hiện các thao tác như đăng nhập, đăng ký, lấy thông tin người dùng.
- WebSocket: Cung cấp giao tiếp thời gian thực cho việc gửi và nhận tin nhắn.
- Database Communication: Tất cả các dịch vụ backend đều giao tiếp với MongoDB để lưu trữ và truy vấn dữ liệu.

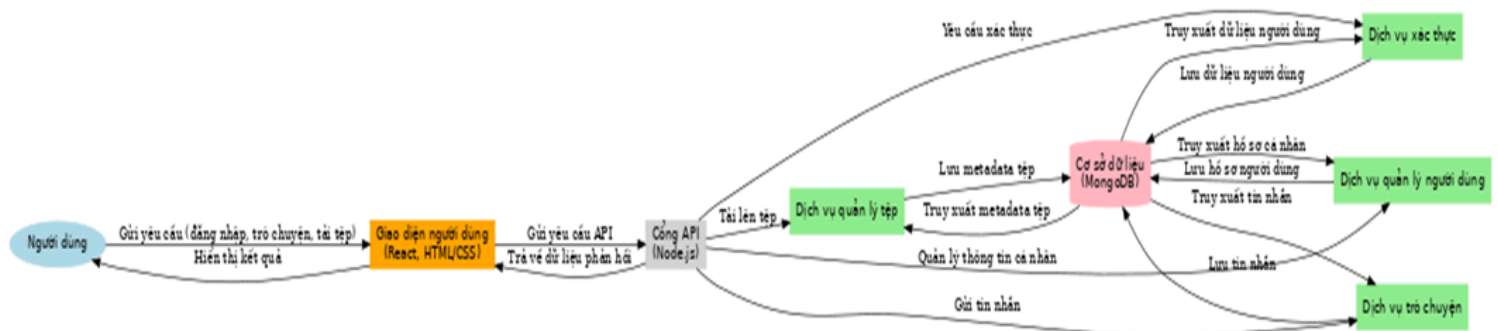
Luồng hoạt động chính:

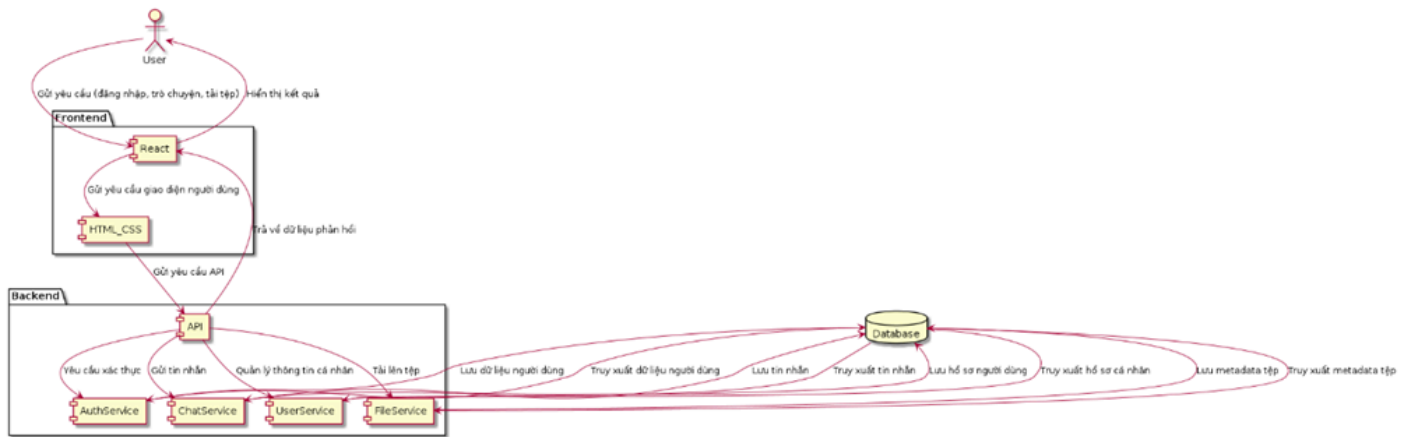
1. Đăng nhập/Đăng ký:
 - Người dùng gửi yêu cầu đến API Gateway.
 - API Gateway chuyển tiếp yêu cầu đến Authentication Service để xử lý.
 - Authentication Service kiểm tra thông tin, tạo token và lưu vào MongoDB.
2. Chat thời gian thực:
 - Người dùng gửi tin nhắn qua WebSocket đến API Gateway.
 - API Gateway chuyển tin nhắn đến Chat Service.
 - Chat Service lưu trữ tin nhắn vào MongoDB và gửi tin nhắn cho các thành viên trong nhóm.
3. Quản lý bạn bè và nhóm:
 - Người dùng gửi yêu cầu thêm bạn hoặc tạo nhóm thông qua API Gateway.
 - User Service xử lý yêu cầu và cập nhật vào MongoDB.

Tính năng nổi bật:

- Khả năng mở rộng: Kiến trúc microservices cho phép dễ dàng mở rộng từng thành phần.
- Giao tiếp thời gian thực: WebSocket đảm bảo việc trao đổi tin nhắn mượt mà.
- Bảo mật: Sử dụng JWT cho xác thực và HTTPS cho truyền dữ liệu an toàn.

c. Sơ đồ luồng dữ liệu chương trình và mô tả





1. Người dùng (User): Người dùng tương tác với giao diện người dùng (Frontend) để thực hiện các chức năng như đăng nhập, gửi tin nhắn, tải tệp, chỉnh sửa thông tin cá nhân, v.v.

2. Giao diện người dùng (Frontend)

Chức năng:

- + Hiển thị các thông tin và các tùy chọn mà người dùng có thể tương tác (giao diện đăng nhập, chat, gửi tệp, quản lý hồ sơ cá nhân).
- + Lấy thông tin từ người dùng và chuyển các yêu cầu (dữ liệu) tới cổng API.

Luồng dữ liệu:

- + Người dùng gửi yêu cầu đăng nhập, gửi tin nhắn, tải tệp hoặc chỉnh sửa thông tin cá nhân qua giao diện frontend.
- + Gửi yêu cầu API để thực hiện các hành động cần thiết.

3. Cổng API (API Gateway)

Chức năng:

- + Cổng API chịu trách nhiệm nhận các yêu cầu từ giao diện người dùng, xử lý chúng và chuyển tiếp các yêu cầu này đến các dịch vụ backend tương ứng.
- + Cổng API sẽ phân phối các yêu cầu đến các dịch vụ như dịch vụ xác thực, trò chuyện, quản lý người dùng và quản lý tệp.

Luồng dữ liệu:

- + Nhận các yêu cầu API từ frontend, bao gồm các hành động như yêu cầu xác thực người dùng, gửi tin nhắn, tải lên tệp, và quản lý hồ sơ người dùng.

4. Dịch vụ xác thực (AuthService)

Chức năng:

- + Xử lý các yêu cầu xác thực, bao gồm đăng nhập và đăng ký người dùng.
- + Kiểm tra thông tin người dùng với cơ sở dữ liệu (ví dụ, xác thực mật khẩu, mã xác thực).

Luồng dữ liệu:

- + Nhận yêu cầu xác thực từ cổng API.
- + Truy xuất và lưu trữ dữ liệu người dùng vào cơ sở dữ liệu MongoDB.
- + Trả về kết quả xác thực (thành công hoặc thất bại) về cho cổng API.

5. Dịch vụ trò chuyện (ChatService)

Chức năng:

- + Xử lý các yêu cầu liên quan đến việc gửi và nhận tin nhắn.
- + Lưu trữ tin nhắn trong cơ sở dữ liệu MongoDB.

Luồng dữ liệu:

- + Nhận các yêu cầu gửi tin nhắn từ API.
- + Lưu trữ các tin nhắn vào cơ sở dữ liệu.
- + Truy xuất và trả lại tin nhắn khi có yêu cầu từ frontend.

6. Dịch vụ quản lý người dùng (UserService)

Chức năng:

- + Quản lý hồ sơ người dùng, bao gồm việc chỉnh sửa thông tin cá nhân của người dùng.
- + Cập nhật thông tin người dùng trong cơ sở dữ liệu MongoDB.

Luồng dữ liệu:

- + Nhận yêu cầu cập nhật thông tin người dùng từ API.
- + Lưu trữ và truy xuất thông tin người dùng từ cơ sở dữ liệu.
- + Cung cấp dữ liệu cho frontend khi có yêu cầu.

7. Cơ sở dữ liệu (MongoDB)

Chức năng:

- + Lưu trữ dữ liệu người dùng (hồ sơ người dùng), tin nhắn, và metadata của tệp.

Luồng dữ liệu:

- + Dữ liệu người dùng, tin nhắn, hồ sơ và metadata của tệp được lưu trữ và truy xuất từ cơ sở dữ liệu MongoDB.
- + Dịch vụ xác thực, dịch vụ trò chuyện, dịch vụ người dùng, và dịch vụ tệp sẽ truy xuất và lưu trữ dữ liệu vào cơ sở dữ liệu.

8. Luồng dữ liệu tổng quan

Quá trình đăng nhập:

- + Người dùng gửi thông tin đăng nhập (tên người dùng, mật khẩu) qua giao diện frontend.
- + Frontend gửi yêu cầu này đến API.
- + API chuyển tiếp yêu cầu đến dịch vụ xác thực.
- + Dịch vụ xác thực truy xuất dữ liệu người dùng từ cơ sở dữ liệu và trả kết quả (thành công hoặc thất bại) về API.
- + API gửi kết quả trả về cho frontend và người dùng được thông báo.

Quá trình gửi tin nhắn:

- + Người dùng nhập tin nhắn qua giao diện frontend.
- + Frontend gửi tin nhắn đến API.
- + API chuyển tiếp tin nhắn đến dịch vụ trò chuyện.
- + Dịch vụ trò chuyện lưu tin nhắn vào cơ sở dữ liệu.
- + Tin nhắn được hiển thị trên giao diện người dùng

d. Cấu trúc CSDL

Trong hệ thống chat trực tuyến của chúng tôi, cơ sở dữ liệu được thiết kế sử dụng MongoDB — một hệ quản trị cơ sở dữ liệu NoSQL. Cấu trúc cơ sở dữ liệu bao gồm ba collection chính: **User**, **Messages**, và **Group**. Mỗi collection này lưu trữ dữ liệu cụ thể và có mối quan hệ với nhau để hỗ trợ các chức năng như quản lý người dùng, tin nhắn, và nhóm.

Collection User chứa thông tin chi tiết của người dùng trong hệ thống. Mỗi tài liệu (document) trong collection này tương ứng với một người dùng và lưu trữ các thông tin như tên người dùng, email, mật khẩu, số điện thoại, và các mối quan hệ của người dùng (bạn bè, yêu cầu kết bạn, nhóm tham gia, v.v.).

- **_id**: Mỗi người dùng trong MongoDB đều có một trường **_id** duy nhất, giúp phân biệt các người dùng.
- **username, email, password, phone**: Thông tin cơ bản của người dùng. email và phone phải là duy nhất, do đó được chỉ định là unique.
- **bio, avatarImage, nickname**: Các trường bổ sung cho phép người dùng thêm mô tả cá nhân và ảnh đại diện.
- **friends**: Mảng chứa các ObjectId tham chiếu đến các người dùng khác, tạo mối quan hệ **n-n** (nhiều người dùng có thể có nhiều bạn bè và ngược lại).
- **friendRequests**: Mảng chứa các ObjectId tham chiếu đến các người dùng mà người này đã gửi yêu cầu kết bạn.
- **groups**: Mảng chứa các ObjectId tham chiếu đến các nhóm mà người dùng tham gia.
- **groupInvitations**: Mảng chứa các ObjectId tham chiếu đến các nhóm mà người dùng nhận lời mời tham gia.

Collection Messages lưu trữ thông tin về các tin nhắn trong hệ thống. Mỗi tin nhắn có thể được gửi giữa hai người dùng (tin nhắn cá nhân) hoặc trong một nhóm (tin nhắn nhóm). Tin nhắn có thể bao gồm văn bản và phương tiện (hình

ảnh, video, v.v.), và mỗi tin nhắn có trạng thái để xác định tình trạng của nó (đã gửi, đã nhận, đã đọc).

- **_id**: Mỗi tin nhắn có một **_id** duy nhất trong MongoDB.
- **message.text**: Nội dung văn bản của tin nhắn.
- **message.media**: Đường dẫn đến tệp phương tiện đính kèm (nếu có).
- **from**: Tham chiếu đến User — người gửi tin nhắn.
- **to**: Tham chiếu đến User — người nhận tin nhắn (nếu đây là tin nhắn cá nhân).
- **group**: Tham chiếu đến Group — nhóm mà tin nhắn thuộc về (nếu là tin nhắn nhóm).
- **status**: Trạng thái của tin nhắn, có thể là "sent" (đã gửi), "delivered" (đã nhận), hoặc "read" (đã đọc).
- **createdAt và updatedAt**: Thời gian gửi tin nhắn và thời gian cập nhật tin nhắn (do timestamps: true trong schema).

Collection Group lưu trữ thông tin về các nhóm trong hệ thống. Mỗi nhóm có thể có nhiều thành viên và một người quản trị. Mỗi nhóm có thể chứa nhiều tin nhắn, và mỗi người dùng có thể tham gia nhiều nhóm.

- **_id**: Mỗi nhóm có một **_id** duy nhất trong MongoDB.
- **name**: Tên của nhóm.
- **members**: Mảng chứa các ObjectId tham chiếu đến những người dùng trong nhóm.
- **admin**: Tham chiếu đến User — người quản trị của nhóm.
- **createdAt**: Thời gian tạo nhóm.

Các collection trong cơ sở dữ liệu của hệ thống chat có các mối quan hệ rõ ràng giữa chúng:

1. **Mối quan hệ giữa User và Messages:**

- Một người dùng có thể gửi nhiều tin nhắn, tạo ra mối quan hệ **1-n** (một người dùng có thể có nhiều tin nhắn).
- Trường **from** trong Messages tham chiếu đến **_id** của người gửi trong User.
- Trường **to** trong Messages tham chiếu đến **_id** của người nhận trong User (nếu tin nhắn cá nhân).

2. **Mối quan hệ giữa User và Group:**

- Một người dùng có thể tham gia nhiều nhóm, tạo ra mối quan hệ **n-n** (một người dùng có thể tham gia nhiều nhóm, và mỗi nhóm có thể có nhiều người dùng).
- Trường **groups** trong User lưu trữ các ObjectId tham chiếu đến các nhóm mà người dùng tham gia.
- Trường **members** trong Group lưu trữ các ObjectId tham chiếu đến các người dùng là thành viên của nhóm.

3. **Mối quan hệ giữa Messages và Group:**

- Một tin nhắn có thể được gửi đến một nhóm, tạo ra mối quan hệ **n-1** (nhiều tin nhắn có thể thuộc về một nhóm).

- Trường group trong Messages tham chiếu đến _id của nhóm.

III. Kết quả và đánh giá

1. Kết quả hoạt động

- Hệ thống đăng ký và đăng nhập người dùng hoạt động ổn định nếu không có lỗi khi nhập thông tin người dùng (email, username, mật khẩu). Các thông báo lỗi rõ ràng, ví dụ: email đã tồn tại, mật khẩu không hợp lệ, v.v., giúp người dùng dễ dàng hiểu được vấn đề khi đăng ký hoặc đăng nhập.
- Người dùng có thể dễ dàng gửi yêu cầu kết bạn, chấp nhận hoặc từ chối yêu cầu kết bạn. Đây là một chức năng cơ bản của hệ thống, giúp tạo mối quan hệ giữa các người dùng. Giao diện người dùng có thể hiển thị các yêu cầu kết bạn và danh sách bạn bè dễ dàng, nâng cao trải nghiệm người dùng.
- Chức năng nhắn tin giữa các người dùng hoạt động mượt mà, bao gồm các tính năng như gửi tin nhắn văn bản và phương tiện (hình ảnh, video). Các tin nhắn được hiển thị theo thứ tự thời gian.
- Tính năng nhắn tin trong nhóm giúp người dùng trao đổi thông tin với nhiều người cùng lúc. Mỗi nhóm có thể có nhiều thành viên và tin nhắn được gửi trong nhóm có thể được phân biệt với tin nhắn cá nhân thông qua các trường nhóm (group) trong cơ sở dữ liệu. Các trạng thái tin nhắn như "đã gửi", "đã nhận", và "đã đọc" hoạt động hiệu quả giúp người dùng biết được tình trạng của tin nhắn.
- Người dùng có thể tạo nhóm mới, tham gia các nhóm hoặc rời khỏi nhóm. Các nhóm có thể có người quản trị để kiểm soát các quyền của thành viên trong nhóm. Tính năng mời người dùng tham gia nhóm và chấp nhận lời mời cũng được triển khai tốt, người dùng có thể dễ dàng nhận và trả lời lời mời nhóm.
- Giao diện người dùng: Giao diện phải dễ sử dụng và trực quan, giúp người dùng dễ dàng điều hướng qua các chức năng của website như nhắn tin, thêm bạn, tạo nhóm, v.v. Nếu giao diện rõ ràng và dễ dàng tùy chỉnh, người dùng sẽ có trải nghiệm tốt hơn.
- Trải nghiệm người dùng: Hệ thống cần cung cấp phản hồi nhanh chóng với các hành động người dùng, ví dụ như gửi tin nhắn, nhận thông báo về yêu cầu kết bạn, hay tham gia nhóm. Trải nghiệm này giúp người dùng không cảm thấy bị gián đoạn.

2. Đánh giá so với đặc tả

+ Đăng ký và đăng nhập người dùng

- Yêu cầu: Hệ thống phải cho phép người dùng đăng ký tài khoản mới, đăng nhập và xác thực thông tin người dùng (username, email, mật khẩu).
- Kết quả thực tế: Hệ thống đã triển khai đầy đủ chức năng đăng ký và đăng nhập. Người dùng có thể đăng ký bằng email, username và mật

khẩu, với các kiểm tra đầu vào hợp lý như yêu cầu mật khẩu ít nhất 8 ký tự và username có độ dài từ 3 đến 20 ký tự. Mật khẩu được mã hóa, đảm bảo an toàn cho người dùng.

+ Quản lý người dùng và bạn bè

- Yêu cầu: Hệ thống phải hỗ trợ người dùng gửi yêu cầu kết bạn, chấp nhận hoặc từ chối yêu cầu kết bạn, và quản lý danh sách bạn bè.
- Kết quả thực tế: Hệ thống đáp ứng đầy đủ yêu cầu này, cho phép người dùng gửi yêu cầu kết bạn và quản lý bạn bè thông qua các danh sách và trạng thái kết bạn. Tuy nhiên, có thể cải thiện giao diện người dùng để làm rõ hơn trạng thái yêu cầu kết bạn.

+ Nhắn tin cá nhân và nhóm

- Yêu cầu: Hệ thống phải hỗ trợ nhắn tin cá nhân giữa hai người dùng và nhắn tin trong nhóm.
- Kết quả thực tế: Hệ thống nhắn tin cá nhân hoạt động tốt và cũng hỗ trợ nhắn tin nhóm. Người dùng có thể gửi tin nhắn văn bản và phương tiện (hình ảnh/video) trong các cuộc trò chuyện cá nhân hoặc nhóm. Mỗi tin nhắn có trạng thái (sent, delivered, read) để theo dõi tình trạng giao tiếp.

+ Quản lý nhóm và lời mời tham gia nhóm

- Yêu cầu: Hệ thống phải cho phép người dùng tạo nhóm mới, tham gia nhóm và mời người khác tham gia nhóm.
- Kết quả thực tế: Hệ thống đã triển khai tính năng tạo nhóm và mời người dùng tham gia nhóm một cách đầy đủ. Người dùng có thể tạo nhóm, mời bạn bè vào nhóm và chấp nhận lời mời. Mỗi nhóm có thể có nhiều thành viên và người quản trị để kiểm soát quyền truy cập.

+ Tính năng thời gian thực (Real-time)

- Yêu cầu: Hệ thống phải hỗ trợ nhắn tin thời gian thực mà không cần tải lại trang.
- Kết quả thực tế: Hệ thống sử dụng Socket.IO để hỗ trợ nhắn tin thời gian thực giữa người dùng, đảm bảo các tin nhắn được gửi và nhận ngay lập tức mà không cần tải lại trang, đáp ứng yêu cầu này.

+ Giao diện người dùng

- Yêu cầu: Giao diện người dùng phải đơn giản, dễ sử dụng và phản hồi nhanh chóng.
- Kết quả thực tế: Giao diện người dùng cơ bản đã được xây dựng, nhưng có thể cải thiện để dễ sử dụng hơn, đặc biệt trên các thiết bị di động. Tốc độ phản hồi và trải nghiệm người dùng cần được tối ưu thêm.

3. Đánh giá kết quả test

a. Kiểm Thử API của User Controller

Mục Tiêu Kiểm Thử: Kiểm tra tính năng liên quan đến người dùng như đăng ký, đăng nhập, và quản lý tài khoản người dùng.

Test 1: Đăng Ký Người Dùng Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 201 và thông báo "User created successfully".
- Đánh giá: Kiểm thử thành công. API thực hiện đúng chức năng tạo người dùng mới và trả về thông báo hợp lý.

Test 2: Đăng Ký Người Dùng Với Dữ Liệu Bị Thiếu

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Missing required fields".
- Đánh giá: Kiểm thử thành công. API xử lý tình huống thiếu dữ liệu nhập vào và trả về mã lỗi 400.

Test 3: Đăng Nhập Người Dùng Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 200 và token xác thực người dùng.
- Đánh giá: Kiểm thử thành công. API xử lý yêu cầu đăng nhập và trả về kết quả hợp lý khi người dùng đã đăng ký và thông tin hợp lệ.

Test 4: Đăng Nhập Với Thông Tin Sai

- Kết quả: Thành công. API trả về mã lỗi 401 và thông báo "Invalid credentials".
- Đánh giá: Kiểm thử thành công. API xử lý tình huống đăng nhập với thông tin không chính xác và bảo vệ hệ thống khỏi các truy cập trái phép.

b. Kiểm Thử API của Message Controller

Mục Tiêu Kiểm Thử: Kiểm tra tính năng liên quan đến tin nhắn giữa các người dùng như gửi, nhận, và xử lý tin nhắn.

Test 1: Thêm Tin Nhắn Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 200 và thông báo "Message added successfully".
- Đánh giá: Kiểm thử thành công. API xử lý đúng yêu cầu gửi tin nhắn và trả về kết quả hợp lệ.

Test 2: Thêm Tin Nhắn Thiếu Trường Bắt Buộc

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Missing required fields".

- Đánh giá: Kiểm thử thành công. API có khả năng kiểm tra tính đầy đủ của dữ liệu và phản hồi hợp lý khi thiếu trường bắt buộc.

Test 3: Xử Lý Lỗi Server Khi Thêm Tin Nhắn

- Kết quả: Thành công. API trả về mã lỗi 500 và thông báo "Failed to add message to the database".
- Đánh giá: Kiểm thử thành công. API xử lý lỗi khi có sự cố trong quá trình tương tác với cơ sở dữ liệu, giúp bảo vệ hệ thống và cung cấp phản hồi rõ ràng cho người dùng.

Test 4: Lấy Tin Nhắn Thành Công

- Kết quả: Thành công. API trả về danh sách tin nhắn với đúng dữ liệu.
- Đánh giá: Kiểm thử thành công. API trả về tin nhắn giữa hai người dùng đúng cách và đáp ứng yêu cầu của người dùng.

Test 5: Lấy Tin Nhắn Với Dữ Liệu Không Hợp Lệ

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Invalid 'from' or 'to' ObjectId".
- Đánh giá: Kiểm thử thành công. API xử lý tình huống ID người dùng không hợp lệ và phản hồi lỗi đúng cách.

c. Kiểm Thử API của Group Controller

Mục Tiêu Kiểm Thử: Kiểm tra các tính năng liên quan đến nhóm như tạo nhóm, gửi lời mời, và từ chối lời mời tham gia nhóm.

Test 1: Tạo Nhóm Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 200 và thông báo "Nhóm đã được tạo thành công".
- Đánh giá: Kiểm thử thành công. API thực hiện chính xác yêu cầu tạo nhóm và trả về kết quả hợp lý, dữ liệu nhóm được trả về đầy đủ và chính xác.

Test 2: Thiếu Trường **name** Khi Tạo Nhóm

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Tên nhóm là bắt buộc".
- Đánh giá: Kiểm thử thành công. API xử lý tốt tình huống thiếu trường bắt buộc và trả về thông báo lỗi rõ ràng.

Test 3: Trường **members** Không Phải Mảng Khi Tạo Nhóm

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Cần có ít nhất một thành viên trong nhóm".

- Đánh giá: Kiểm thử thành công. API có khả năng kiểm tra tính hợp lệ của dữ liệu nhập vào, giúp bảo vệ dữ liệu và đảm bảo tính đúng đắn của nhóm.

Test 4: Từ Chối Lời Mời Nhóm Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 200 và thông báo "Group invitation declined".
- Đánh giá: Kiểm thử thành công. API xử lý đúng yêu cầu từ chối lời mời tham gia nhóm và trả về kết quả hợp lý.

Test 5: Người Dùng Không Tìm Thấy Khi Từ Chối Lời Mời

- Kết quả: Thành công. API trả về mã lỗi 404 và thông báo "User not found".
- Đánh giá: Kiểm thử thành công. API xử lý đúng tình huống người dùng không tồn tại và cung cấp thông báo lỗi rõ ràng.

Test 6: Gửi Lời Mời Nhóm Thành Công

- Kết quả: Thành công. API trả về mã trạng thái 200 và thông báo "Group invitation sent".
- Đánh giá: Kiểm thử thành công. API xử lý đúng yêu cầu gửi lời mời tham gia nhóm và phản hồi hợp lý khi người gửi là thành viên hợp lệ.

Test 7: Nhóm Không Tìm Thấy Khi Gửi Lời Mời

- Kết quả: Thành công. API trả về mã lỗi 404 và thông báo "Group not found".
- Đánh giá: Kiểm thử thành công. API kiểm tra sự tồn tại của nhóm trước khi gửi lời mời và đảm bảo các yêu cầu được xử lý chính xác.

Test 8: Người Gửi Không Phải Thành Viên Khi Gửi Lời Mời

- Kết quả: Thành công. API trả về mã lỗi 400 và thông báo "Sender is not a member of this group".
- Đánh giá: Kiểm thử thành công. API có cơ chế kiểm tra quyền hạn của người gửi trước khi gửi lời mời, giúp đảm bảo tính bảo mật và chính xác.

d. Tổng Quan Đánh Giá Tất Cả Kiểm Thử

- Độ Chính Xác: Các API của User, Message, và Group Controller đều thực hiện đúng chức năng của mình. Các kiểm thử đều trả về kết quả chính xác và thông báo phù hợp cho từng tình huống, đảm bảo hệ thống hoạt động như mong đợi.
- Xử Lý Lỗi: Các trường hợp lỗi như thiếu dữ liệu, dữ liệu sai định dạng, nhóm không tồn tại, người dùng không có quyền đều được hệ thống xử lý tốt với mã lỗi phù hợp và thông báo chi tiết.

- **Tính Bảo Mật:** Các API bảo vệ tốt tính bảo mật của hệ thống bằng cách kiểm tra quyền truy cập của người dùng (ví dụ: người gửi phải là thành viên nhóm để gửi lời mời). Điều này giúp ngăn ngừa các hành vi truy cập trái phép và bảo vệ dữ liệu người dùng.
- **Độ Bao Quát của Kiểm Thử:** Các kiểm thử đã bao quát đầy đủ các tình huống có thể xảy ra trong các API của User, Message, và Group Controller. Từ các tình huống thành công đến các lỗi phổ biến, hệ thống đều được kiểm thử toàn diện.

IV. Kết luận và hướng phát triển

1. Kết luận

Qua quá trình xây dựng và kiểm thử hệ thống chat trực tuyến, đề tài đã hoàn thành các mục tiêu đã đề ra, bao gồm việc triển khai các tính năng cơ bản như đăng ký người dùng, đăng nhập, gửi và nhận tin nhắn cá nhân và nhóm, cũng như các chức năng quản lý nhóm. Các mô-đun chính của hệ thống như User, Message và Group đã được phát triển và kiểm tra kỹ lưỡng. Các bài kiểm tra được thực hiện bằng cách sử dụng các công cụ như Supertest và Jest, giúp kiểm tra các chức năng quan trọng như đăng ký người dùng, gửi tin nhắn, và mời tham gia nhóm.

Kết quả của các bài kiểm tra cho thấy hệ thống hoạt động ổn định, với khả năng xử lý các tình huống thường gặp, bao gồm lỗi khi nhập thiếu thông tin hoặc khi dữ liệu không hợp lệ. Hệ thống còn có khả năng phát hiện và xử lý các lỗi phía server một cách mượt mà, đồng thời đảm bảo tính chính xác của các chức năng chính như đăng ký người dùng, gửi tin nhắn và quản lý nhóm.

2. Hướng phát triển

Dù hệ thống đã hoàn thiện các tính năng cơ bản, vẫn còn nhiều hướng phát triển để mở rộng và nâng cao hiệu quả hoạt động. Trong tương lai, có thể bổ sung các tính năng như chat video và gọi thoại để tăng tính tương tác giữa người dùng.

Cải thiện giao diện người dùng (UI/UX) cũng là một hướng phát triển cần thiết, giúp tăng trải nghiệm người dùng. Cuối cùng, hệ thống có thể được mở rộng để hỗ trợ đa nền tảng, giúp người dùng có thể truy cập từ nhiều thiết bị và nền tảng khác nhau như ứng dụng di động hoặc desktop.

V. Phụ lục

Bảng phân chia công việc các thành viên

Công việc chính	Công việc chi tiết	Mô tả	Thành viên
1. Xác định và phân tích yêu cầu	- Thu thập yêu cầu	Tổ chức cuộc họp để thu thập yêu cầu chức năng và phi chức năng cho hệ thống chat.	Cả nhóm
	- Xác định tính năng chính của hệ thống (Đăng ký, đăng nhập, gửi/nhận tin nhắn, quản lý nhóm, v.v.)	Xác định tất cả các tính năng cơ bản và các yêu cầu về hiệu suất, bảo mật, khả năng mở rộng, và khả năng sử dụng của hệ thống.	Cả nhóm
	- Phân tích yêu cầu hiệu suất và khả năng mở rộng	Đánh giá khối lượng dữ liệu cần xử lý, số lượng người dùng dự kiến, và khả năng mở rộng hệ thống.	Cả nhóm
2. Phân tích thiết kế hệ thống	- Phân tích hệ thống và lựa chọn công nghệ (Node.js, MongoDB, Express, Socket.IO, v.v.)	Quyết định các công nghệ sẽ sử dụng trong hệ thống, bao gồm backend, database, và frontend.	Cả nhóm
	- Xây dựng Use Case Diagram	Tạo sơ đồ sử dụng để mô tả các hành động chính của người dùng và hệ thống.	Tuân, Tình
	- Xây dựng Sequence Diagram	Tạo sơ đồ tuần tự để mô tả luồng dữ liệu giữa các thành phần trong hệ thống (ví dụ: luồng giữa người dùng và server khi gửi tin nhắn).	Son, Hùng
	- Đặc tả hệ thống và các yêu cầu API	Xác định các yêu cầu API cho các chức năng như đăng ký, đăng nhập, gửi tin nhắn, tạo nhóm, v.v. Tạo tài liệu đặc tả chi tiết API cho các endpoint.	Việt Anh
3. Xây dựng giao diện Figma	- Thiết kế giao diện người dùng (UI)	Tạo các wireframe, mockup cho các màn hình chính như trang chủ, trang đăng ký, trang đăng nhập, và giao diện tin nhắn.	Tình, Tuấn

	- Lựa chọn màu sắc, font chữ, và các yếu tố thiết kế UI	Xác định các yếu tố thiết kế như bảng màu, font chữ, và kiểu dáng chung cho website.	Tình
	- Thiết kế UX (trải nghiệm người dùng)	Tạo các prototype với Figma để mô phỏng trải nghiệm người dùng, giúp cải thiện khả năng sử dụng và tối ưu hóa luồng người dùng.	Tuân
4. Code các chức năng	- Thiết lập môi trường phát triển (IDE, framework, các công cụ hỗ trợ)	Cài đặt và cấu hình môi trường phát triển với các công cụ như Visual Studio Code, Node.js, npm, v.v.	Cả nhóm
	- Xây dựng cấu trúc cơ sở dữ liệu (MongoDB schema)	Thiết kế và triển khai các schema MongoDB cho các mô hình người dùng, tin nhắn, nhóm, v.v.	Việt Anh
	- Phát triển API backend cho các chức năng như đăng ký, đăng nhập, gửi tin nhắn, quản lý nhóm	Xây dựng các endpoint API với Express cho các tính năng như đăng ký, đăng nhập, gửi tin nhắn, tạo nhóm, v.v.	Việt Anh
	- Tạo và tích hợp các chức năng người dùng (đăng ký, đăng nhập, thay đổi thông tin cá nhân, v.v.)	Xây dựng các chức năng cho phép người dùng đăng ký, đăng nhập, và quản lý thông tin cá nhân.	Son
	- Phát triển chức năng gửi tin nhắn (text, media, hình ảnh, v.v.)	Tạo chức năng gửi và nhận tin nhắn dưới dạng văn bản, hình ảnh, và các loại media khác.	Hùng
	- Xây dựng chức năng quản lý nhóm (tạo nhóm, mời người tham gia, từ chối lời mời)	Phát triển các chức năng liên quan đến quản lý nhóm như tạo nhóm, mời thành viên, từ chối lời mời.	Tuân
5. Testing và đánh giá	- Viết test case cho các chức năng chính (đăng ký, đăng nhập, gửi tin nhắn, quản lý nhóm, v.v.)	Tạo các test case cho các tính năng quan trọng để kiểm tra sự hoạt động chính xác của hệ thống.	Cả nhóm

	- Sử dụng Jest và Supertest để kiểm tra API	Sử dụng các công cụ kiểm thử như Jest và Supertest để kiểm tra các API backend và chức năng của hệ thống.	Hùng
	- Kiểm tra tính năng chat thời gian thực bằng cách sử dụng unit tests và integration tests	Kiểm tra tính năng chat thời gian thực thông qua các bài kiểm tra đơn vị và kiểm tra tích hợp, đảm bảo mọi tin nhắn được gửi và nhận đúng cách.	Son
	- Đánh giá và sửa lỗi sau kiểm thử	Phân tích kết quả kiểm thử, sửa các lỗi phát sinh và tối ưu hóa lại các chức năng để cải thiện hiệu suất và bảo mật.	Cả nhóm