**ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU**



**NIÊN LUẬN II**

**ỨNG DỤNG CHAT CƠ BẢN**

|  |  |
| --- | --- |
| **Sinh viên thực hiện:**  **Họ tên:** Phạm Minh Sáng  **MSSV:** B2110976  **Lớp:** DI21T9A2  **Ngành:** Mạng Máy Tính Và Truyền Thông Dữ Liệu  **Khóa:** 47 | **Giáo viên hướng dẫn:**  **Giảng viên:** TS. Hà Duy An |

**HỌC KỲ 2, 2023-2024**

**ĐÁNH GIÁ CỦA GIẢNG VIÊN**

**MỤC LỤC**

[Chương 1: Tổng quan 6](#_Toc165036239)

[I. Mô tả bài toán 6](#_Toc165036240)

[1. Mô tả 6](#_Toc165036241)

[2. Các chức năng chính 6](#_Toc165036242)

[3. Các yêu cầu phi chức năng 7](#_Toc165036243)

[4. Môi trường vận hành 7](#_Toc165036244)

[5. Các giả định phụ thuộc 7](#_Toc165036245)

[II. Mục tiêu cần đạt được 7](#_Toc165036246)

[III. Hướng giải quyết 8](#_Toc165036247)

[IV. Kế hoạch thực hiện 8](#_Toc165036248)

[Chương 2: Lý thuyết 9](#_Toc165036249)

[I. Khái niệm 9](#_Toc165036250)

[1. HTML 9](#_Toc165036251)

[2. CSS 9](#_Toc165036252)

[3. JavaScript 9](#_Toc165036253)

[4. Jquery 9](#_Toc165036254)

[5. Node.js 9](#_Toc165036255)

[6. ExpressJs 10](#_Toc165036256)

[7. EJS 10](#_Toc165036257)

[8. Socket.IO 10](#_Toc165036258)

[9. MySQL 11](#_Toc165036259)

[II. Ứng dụng 11](#_Toc165036260)

[Chương 3: Kết quả ứng dụng 12](#_Toc165036261)

[I. Thiết kế thành phần dữ liệu 12](#_Toc165036262)

[1. Sơ đồ hoạt vụ (Usecase) 12](#_Toc165036263)

[2. Mô hình mức quan niệm (CDM) 13](#_Toc165036264)

[3. Mô hình mức luận lý (LDM) 14](#_Toc165036265)

[4. Mô hình mức vật lý (PDM) 14](#_Toc165036266)

[II. Thiết kế giao diện 16](#_Toc165036267)

[1. Trang đăng ký 16](#_Toc165036268)

[2. Trang đăng nhập 16](#_Toc165036269)

[3. Trang chat 17](#_Toc165036270)

[Chương 4: Kết luận – đánh giá 20](#_Toc165036271)

[I. Kết quả đạt được 20](#_Toc165036272)

[II. Thu hoạch 20](#_Toc165036273)

[III. Ưu điểm 20](#_Toc165036274)

[IV. Hạn chế - nguyên nhân 20](#_Toc165036275)

[V. Hướng phát triển 21](#_Toc165036276)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Sơ đồ hoạt vụ (Usercase) 12](#_Toc164537664)

[Hình 2: Mô hình mức quan niệm (CDM) 13](file:///D:\Workspace\SourceCode\NLCS_ChatApp\Báo%20Cáo%20Niên%20Luận%20Cơ%20Sở%20-%20Phạm%20Minh%20Sáng%20-%20B2110976.docx#_Toc164537665)

[Hình 3: Mô hình mức vật lý (PDM) 14](file:///D:\Workspace\SourceCode\NLCS_ChatApp\Báo%20Cáo%20Niên%20Luận%20Cơ%20Sở%20-%20Phạm%20Minh%20Sáng%20-%20B2110976.docx#_Toc164537666)

[Hình 4: Trang đăng ký 16](#_Toc164537667)

[Hình 5: Trang đăng nhập 16](#_Toc164537668)

[Hình 6: Trang chat - Home 17](#_Toc164537669)

[Hình 7: Trang chat - Tìm kiếm người dùng để kết bạn 17](#_Toc164537670)

[Hình 8: Trang chat - Lời mời kết bạn 18](#_Toc164537671)

[Hình 9: Trang chat - Gửi và nhận tin nhắn 18](#_Toc164537672)

[Hình 10: Trang chat - Hiển thị thông báo 18](#_Toc164537673)

[Hình 11: Trang chat - Hiển thị có người dùng đang nhập tin 19](#_Toc164537674)

[Hình 12: Trang chat - Tìm kiếm người dùng muốn gửi tin 19](#_Toc164537675)

[Hình 13: Trang chat - Hiển thị kết quả tìm kiếm người dùng 19](#_Toc164537676)

**DANH MỤC BẢNG**

[Bảng 1: Kế hoạch thực hiện 8](#_Toc164537383)

[Bảng 2: PDM - USERS 15](#_Toc164537384)

[Bảng 3: PDM - FRIENDS 15](#_Toc164537385)

[Bảng 4: PDM - MESSAGES 15](#_Toc164537386)

# Tổng quan

## Mô tả bài toán

### Mô tả

Hiện nay, việc trao đổi thông tin, dữ liệu một cách nhanh chóng và tức thì là một trong những vấn đề thiết yếu của xã hội hiện nay. Do vậy, khi chúng ta trao đổi thông tin với nhau bằng văn bản trên mạng xã hội cũng cần được tức thì để hai bên có thể cập nhật được thông tin một cách tốt nhất và không phải chờ đợi quá lâu.

Với ứng dụng Chat cơ bản, lần đầu khi người dùng truy cập sẽ được dẫn đến trang đăng nhập, để người dùng có thể đăng nhập được vào ứng dụng Chat thì cần có một tài khoản cá nhân cho riêng mình với các thông tin cơ bản như: *tên tài khoản người dùng, mật khẩu, tình trạng trực tuyến*. Với việc bảo mật thông tin cá nhân của người dùng ngày càng quan trọng, cho nên mật khẩu của người dùng sẽ được mã hóa để tránh rò rỉ hoặc đánh cấp thông tin, mỗi tên tài khoản trong hệ thống tồn tại là duy nhất.

Sau khi đăng ký thành công một tài khoản, ứng dụng sẽ điều hướng đến trang đăng nhập, khi người dùng đã có cho mình một tài khoản cá nhân, họ sẽ tiến hành đăng nhập vào sử dụng ứng dụng Chat điền đầy đủ và đúng với những gì mà người dùng đã đăng ký trước đó.

Khi đăng nhập thành công vào ứng dụng một hộp thoại Chat hiện lên và dẫn đến phần thêm đoạn hội thoại với người dùng khác trong ứng dụng, lần đầu tiên người dùng sử dụng ứng dụng họ sẽ không tồn tại một hộp thoại nào. Lúc này người dùng sẽ tiến hành tìm và kết bạn với người dùng khác trong ứng dụng, chờ đợi người dùng được kết bạn đồng ý lời mời thì cả hai mới có thể bắt đầu tiến hành trò chuyện với nhau. Lời mời kết bạn sẽ được lưu vào bảng với thông tin như: *mã lời mời, người gửi kết bạn, người chấp nhận lời mời, tình trạng lời mời*.

Bắt đầu cuộc trò chuyện người dùng cần chọn vào hộp thoại của người cần nhận tin nhắn, sau đó người dùng sẽ nhập nội dung tin nhắn cần gửi đi vào ô nhập tin, nội dung tin nhắn sẽ được lưu vào bảng gồm: *mã tin nhắn, người gửi, người nhận, nội dung tin nhắn, thời gian, tình trạng nhận tin nhắn*. Trong lúc trò chuyện thì người dùng có thể thấy được là người kia có đang nhắn một đoạn tin nào đó không, khi có một tin nhắn mới hộp thoại của người dùng sẽ có một chấm đỏ hiển thị thông báo mới của người dùng khác gửi đến. Sau khi người dùng không sử dụng ứng dụng nữa họ có thể đăng xuất ra khỏi ứng dụng và sẽ được chuyển về trang đăng nhập.

### Các chức năng chính

* Đăng nhập.
* Đăng ký.
* Tìm kiếm người dùng muốn kết bạn.
* Gửi lời mời kết bạn và chấp nhận lời mời kết bạn.
* Tìm người dùng đã kết bạn.
* Chọn hộp thoại muốn nhắn tin.
* Gửi và nhận tin nhắn.
* Hiển thị thông báo tin nhắn chưa đọc.
* Đăng xuất.

### Các yêu cầu phi chức năng

* Giao diện thân thiện, hài hòa màu sắc, dễ dàng sử dụng.
* Phông chữ: sử dụng phông chữ Roboto tiếng Việt, tiêu chuẩn Unicode.
* Có tính mở rộng để thuận tiện cho việc phát triển hệ thống về sau.
* Hiệu quả về mặt tốc độ xử lý cập nhật, tra cứu thông tin trả về cho người dùng không quá 5 giây.
* Bảo mật tài khoản của người dùng bằng các giải thuật băm tránh các trường lộ, đánh cấp tài khoản.

### Môi trường vận hành

Máy tính cá nhân

* Phần cứng:
  + CPU: Intel Core i5 10300H
  + Ram: 16GB
  + SSD: 512GB
* Phần mềm:
  + Hệ điều hành Window 11
  + Visual Studio Code
  + Node.js 18.18.2
  + MySQL Workbench 8.0

### Các giả định phụ thuộc

* Máy chủ có thể bị nhiễm virus hoặc hacker tấn công làm hỏng dữ liệu.
* Phần cứng bị hỏng đột ngột
* Tài khoản người dùng có thể bị tấn công
* Nguồn điện không ổn định hoặc thiếu giải pháp cho nguồn điện dự phòng
* Đường truyền kết nối Internet hỏng hoặc không ổn định làm mất nối kết
* Các trình duyệt không hỗ trợ tương thích ứng dụng

## Mục tiêu cần đạt được

Mục tiêu tổng quan: Sau khi thực hiện đề tài có thể hiểu rõ được cách thức hoạt động của mô hình client – server, nghiên cứu về Node.js cũng như là hệ quản trị cơ sở dữ liệu MySQL. Bên cạnh đó là nghiên cứu về Socket.IO xử lý các tác vụ theo thời gian thực, cách phân tích bài toán, giải quyết vấn đề.

Mục tiêu cụ thể: Xây dựng được một ứng dụng web có thể kết bạn, trao đổi thông tin văn bản giữa hai người dùng với nhau theo thời gian thực trên nền tảng là Node.js kết hợp thư viện Socket.IO cùng với MySQL để lưu trữ dữ liệu.

## Hướng giải quyết

Sử dụng Node.js làm server để nhận thông tin từ người dùng như: thông tin tài khoản đăng nhập, danh sách bạn bè trao đổi tin nhắn trên ứng dụng, các đoạn tin nhắn của người dùng gửi đi, sau đó lưu trữ vào trong hệ quản trị cơ sở dữ liệu MySQL. Kết hợp sử dụng thư viện Socket.IO để có thể phản hồi tin nhắn ngược lại cho người dùng cũng như là người nhận được tin nhắn tức thì theo thời gian thực.

## Kế hoạch thực hiện

|  |  |  |
| --- | --- | --- |
| **Tuần** | **Công việc** | **Thời gian báo cáo** |
| 1 - 4 | Tìm hiểu công nghệ, phạm vi thực hiện. | Tuần 4 |
| 5 - 6 | Triển khai giao diện chat máy client bằng HTML, CSS. |  |
| 7 - 8 | Xây dựng server socket trao đổi tin nhắn giữa 2 client với nhau. |  |
| 9 | Xử lí sự kiện kết nối tên người dùng tới ứng dụng, thoát khỏi người dùng hiện tại, hiển thị số người dùng | Tuần 9 |
| 10 - 11 | Xử lý sự kiện người dùng tìm kiếm gửi lời mời, chấp nhận lời kết bạn, tìm kiếm người cần trao đổi tin nhắn |  |
| 12 - 13 | Sữa lỗi và bổ sung các chức năng. | Tuần 12 |
| 14 | Hoàn thiện demo chương trình và sửa lại lỗi nếu có. | Tuần 14 |
| 15 | Thiết kế slide báo cáo. |  |
| 16 | Làm văn bản in nên cuốn niên luận | Tuần 16 |
| 17 | Chuẩn bị các khâu báo cáo. |  |
| 18 | Hoàn thành báo cáo, trình bày kết quả đạt được. | Tuần 18 |

Bảng 1: Kế hoạch thực hiện

# Lý thuyết

## Khái niệm

### HTML

HTML (viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì.

### CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

### JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Mã lệnh được nhúng vào các trang web và được thực thi bởi trình duyệt web.

### Jquery

Jquery là thư viện JavaScript được thiết kế để đơn giản hóa việc lập trình giao diện web, cho phép tương tác các phần tử DOM trên HTML, kiểm các các hành động của người dùng một cách dễ dàng. Làm cho giao diện người dùng thân thiện hơn bởi các hiệu ứng phản hồi từ hệ thống cũng như việc lập trình AJAX trở nên đơn giản

### Node.js

Node.js là một môi trường JavaScript runtime mã nguồn mở và đa nền tảng. NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng C++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009. Node.js ra đời khi các developer đời đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

### ExpressJs

Express.js là một web application framework tối giản và linh hoạt, cung cấp một bộ tính năng mạnh mẽ để phát triển các ứng dụng di động và web dựa trên Node.js. Là một trong những web application framework phổ biến nhất trong hệ sinh thái Node.js. Express.js cung cấp tất cả các tính năng của một web application framework, như templating, xử lý tệp tĩnh, kết nối với cơ sở dữ liệu SQL và NoSQL.

### EJS

EJS viết tắt của “Embedded JavaScript,” là một ngôn ngữ mẫu (template language) được sử dụng trong phát triển ứng dụng web. Nó giúp phát triển các trang web động bằng cách cho phép tạo ra các mẫu HTML được kết hợp với mã JavaScript. EJS thường được sử dụng cùng với Node.js và các framework web như Express để tạo giao diện người dùng động dựa trên dữ liệu từ máy chủ.

### Socket.IO

Socket.IO là một thư viện mã nguồn mở trong Node.js được phát triển từ năm 2010. Mục đích nhằm tạo ra một môi trường giao tiếp thời gian thực và hai chiều giữa máy chủ và máy khách trên Internet. Nền tảng cho phép gửi và nhận dữ liệu một cách tức thì và tối ưu hóa việc sử dụng tài nguyên. Công nghệ được tích hợp với nhiều ngôn ngữ khác nhau như PHP, ASP.NET, Node.js.

Socket.IO xây dựng dựa trên Engine.IO và hỗ trợ việc thiết lập kết nối bảo mật thông qua sự hỗ trợ của các phương thức giao tiếp tốt hơn như Websocket. Ngoài ra, Socket.IO cũng cung cấp một số cơ chế bảo mật bổ sung, bao gồm:

* Bảo mật dựa trên TLS/SSL: Socket.IO hỗ trợ kết nối an toàn thông qua việc sử dụng giao thức HTTPS và WSS (WebSocket over TLS/SSL). Điều này có tác dụng mã hóa dữ liệu trên đường truyền và ngăn chặn các cuộc tấn công theo dõi hoặc thay đổi dữ liệu.
* Hỗ trợ cân bằng tải và proxy: Socket.IO có thể dễ dàng tích hợp với các cơ chế cân bằng tải và proxy để tăng cường bảo mật và khả năng chịu tải của hệ thống.
* Hỗ trợ cho tường lửa: Socket.IO cung cấp các cơ chế để vượt qua các tường lửa và phần mềm chống virus thông qua việc sử dụng các cổng mạng phổ biến như cổng 443 (HTTPS).

Cùng với một số tính năng đặc trưng khác của Socket.IO như:

* Cơ chế kết nối server tự động: tính năng tự động kết nối đến server giúp việc giao tiếp giữa client và server trở nên linh hoạt và dễ dàng hơn. Khi mất kết nối trong quá trình khởi chạy, Socket.IO sẽ liên tục cố gắng kết nối đến server cho đến khi kết nối lại thành công hoặc đạt đến ngưỡng thời gian chờ đặt trước.
* Mã hóa nhị phân: giúp người dùng có thể truyền các dữ liệu nhị phân như hình ảnh video hoặc âm thanh.
* Tiện ích tạo kênh và phòng.

Phương thức hoạt động: Socket.IO hoạt động bằng cách sử dụng mô hình giao thức "Long Polling" hoặc WebSockets để tạo và duy trì kết nối real time giữa client và server. Quy trình này hoạt động như sau:

* Trong trường hợp trình duyệt hoặc client không hỗ trợ WebSockets, Socket.IO sẽ sử dụng phương pháp long-polling. Khi client gửi yêu cầu tới server, server sẽ không trả về ngay lập tức mà duy trì kết nối. Hệ thống chỉ trả về khi có dữ liệu mới hoặc sau một khoảng thời gian nhất định. Sau đó, client sẽ gửi yêu cầu mới ngay lập tức để duy trì kết nối. Quá trình này tiếp diễn để tạo ra một luồng thông tin realtime giữa client và server qua HTTP.
* Trong trường hợp trình duyệt hoặc client hỗ trợ WebSockets, Socket.IO sẽ sử dụng giao thức này để tạo kết nối full-duplex giữa client và server. Điều này cho phép cả server và client gửi, nhận dữ liệu một cách tức thì mà không cần phải tạo các yêu cầu HTTP mới.

### MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc (SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle, hoạt động dựa trên mô hình tiêu chuẩn là Client (Máy khách) – Server (Máy chủ) . MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux, UNIX và Windows. MySQL thường được kết hợp với các ứng dụng web.

## Ứng dụng

Kết hợp HTML, CSS, JavaScript để làm một giao diện đăng ký, đăng nhập, Chat hoàn chỉnh cũng như là thân thiện với người dùng, có thể dễ dàng sử dụng khi tiếp xúc ứng dụng. Sử dụng mẫu trên vào EJS để kết hợp lấy dữ liệu trả về từ phía server. Tạo một server Node.js để xử lý thông tin người dùng gửi lên server cùng với đó là lưu trữ những thông tin ấy vào hệ quản trị cơ sở dữ liệu MySQL để dễ dàng truy xuất, tạo mới, sửa đổi. Trong quá trình người dùng trao đổi tin nhắn văn bản với người dùng khác thì sử dụng kết hợp Node.js và thư viện Socket.IO để việc gửi, nhận thông tin theo thời gian thực một cách nhanh chóng, hiệu quả.

# Kết quả ứng dụng

## Thiết kế thành phần dữ liệu

### Sơ đồ hoạt vụ (Usecase)



Hình 1: Sơ đồ hoạt vụ (Usercase)

### Mô hình mức quan niệm (CDM)

Hình 2: Mô hình mức quan niệm (CDM)

### Mô hình mức luận lý (LDM)

* USERS(**Username**, Password, isOnline)
* FRIENDS(***Friend\_request*, Friend\_accept***,* isAccept)
* MESSAGES(**ID\_Message**, *Sender,* Receiver, Created\_time, Message, Notification)

Các ràng buộc toàn vẹn tham chiếu

* FRIEND(Friend\_request) 🡪 USERS(Username)
* MESSAGES(Sender) 🡪 USERS(Username)

### Mô hình mức vật lý (PDM)

Hình 3: Mô hình mức vật lý (PDM)

Bảng USERS

Bảng 2: PDM - USERS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Tên thuộc tính* | *Kiểu dữ liệu* | *Khóa chính* | *Khóa ngoại* | *Giá trị mặc định* | *Identity* | *NN* | *Diễn giải* |
| Username | varchar(20) | x |  |  |  | x | Tên tài khoản của người dùng. |
| Password | varchar(200) |  |  |  |  | x | Mật khẩu của người dùng được mã hóa. |
| isOnline | smallint |  |  | ‘0’ |  | x | Trạng thái của người dùng khi sử dụng ứng dụng. ‘0’ là không trực tuyến, ‘1’ là đang trực tuyến. |

Bảng FRIENDS

Bảng 3: PDM - FRIENDS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Tên thuộc tính* | *Kiểu dữ liệu* | *Khóa chính* | *Khóa ngoại* | *Giá trị mặc định* | *Identity* | *NN* | *Diễn giải* |
| Friend\_request | varchar(20) | x | x |  |  | x | Tên người dùng đã gửi lời mời kết bạn. |
| Friend\_accept | varchar(20) | x |  |  |  | x | Tên người dùng được gửi lời mời kết bạn, chấp nhận lời mời kết bạn. |
| isAccept | smallint |  |  | ‘0’ |  | x | Trạng thái gửi yêu cầu kết bạn. ‘0’ là đã gửi nhưng chưa chấp nhận, ‘1’ là đã chấp nhận yêu cầu. |

Bảng MESSAGES

Bảng 4: PDM - MESSAGES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Tên thuộc tính* | *Kiểu dữ liệu* | *Khóa chính* | *Khóa ngoại* | *Giá trị mặc định* | *Identity* | *NN* | *Diễn giải* |
| ID\_Message | integer | x |  |  |  | x | Khóa chính của tin nhắn. |
| Sender | varchar(20) |  | x |  |  | x | Tên người dùng đã gửi tin nhắn. |
| Receiver | varchar(20) |  |  |  |  | x | Tên người dùng nhận được tin nhắn. |
| Created\_time | timestamp |  |  |  |  | x | Thời gian mà tin nhắn được gửi đi. |
| Message | long vachar |  |  |  |  | x | Nội dung tin nhắn. |
| Notification | smallint |  |  | ‘1’ |  | x | Trạng thái của tin nhắn. ‘1’ là tin nhắn chưa được đọc, ‘0’ tin nhắn đã đọc. |

## Thiết kế giao diện

### Trang đăng ký

Đăng ký tài khoản để sử dụng ứng dụng với các thông tin bao gồm: tên đăng nhập, mật khẩu và nhập lại mật khẩu lần nữa để xác nhận.

A screenshot of a computer screen

Description automatically generated

Hình 4: Trang đăng ký

### Trang đăng nhập

Đăng nhập bằng tài khoản mà người dùng đã đăng ký trước đó để có thể sử dụng ứng dụng và bắt đầu kết bạn, trao đổi tin nhắn với người dùng khác.

A screen shot of a login

Description automatically generated

Hình 5: Trang đăng nhập

### Trang chat

* Trang chat chính của ứng dụng với một tài khoản lần đầu tiên sử dụng ứng dụng. Tại trang này người dùng chọn tab home để kết bạn với người dùng khác trong hệ thống và bắt đầu trao đổi tin nhắn.

A screenshot of a computer

Description automatically generated

Hình 6: Trang chat - Home

* Tại đây người dùng có thể sử dụng chức năng tìm kiếm người cụ thể cần kết bạn dễ dàng hơn.

A screenshot of a computer

Description automatically generated

Hình 7: Trang chat - Tìm kiếm người dùng để kết bạn

* Khi có một người dùng muốn kết bạn sẽ hiển thị lên màn hình ứng dụng lời mời kết bạn của người đó. Để đồng ý kết bạn nhấn vào nút tích kế bên lời mời và có thể bắt đầu trao tin nhắn với nhau.

A screenshot of a computer

Description automatically generated

Hình 8: Trang chat - Lời mời kết bạn

* Bên tab Chats chọn người muốn nhận tin nhắn từ bạn và bắt đầu trao đổi những thông tin mà bạn muốn gửi đến cho người nhận.

A screenshot of a computer

Description automatically generated

Hình 9: Trang chat - Gửi và nhận tin nhắn

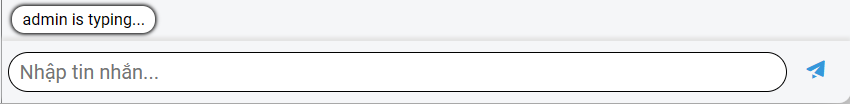
* Khi có một tin nhắn mới mà người dùng chưa đọc, sẽ có một chấm đỏ cạnh bên hộp thoại của người gửi tin nhắn đến để người dùng dễ dàng nhận biết.

A black text on a white background

Description automatically generated

Hình 10: Trang chat - Hiển thị thông báo

* Người dùng đang trong cuộc trò chuyện với một người dùng khác, khi người dùng đó đang nhập tin vào ô nhập tin, sẽ có một ô hiển thị là người dùng đó đang nhập tin nhắn.



Hình 11: Trang chat - Hiển thị có người dùng đang nhập tin

* Người dùng có thể sử dụng chức năng tìm kiếm người muốn gửi, trao đổi tin nhắn khi số lượng bạn bè quá nhiều.

A screenshot of a computer

Description automatically generated

Hình 12: Trang chat - Tìm kiếm người dùng muốn gửi tin

A screenshot of a computer

Description automatically generated

Hình 13: Trang chat - Hiển thị kết quả tìm kiếm người dùng

# Kết luận – đánh giá

## Kết quả đạt được

Sau khi xác định phạm vi nghiên cứu, áp dụng các kỹ thuật, kiến thức đã tìm hiểu, học hỏi được đã có thể xậy dựng được một ứng dụng Chat với các chức năng cơ bản như:

* Đăng ký được một tài khoản, có thể sử dụng tài khoản đã đăng ký để đăng nhập ứng dụng.
* Trao đổi tin nhắn giữa hai người dùng theo thời gian thực.
* Kết bạn, đồng ý kết bạn giữa hai người dùng để bắt đầu trao đổi tin nhắn.
* Thông báo khi có một tin nhắn mới mà người dùng chưa đọc.
* Thiết kế được một giao diện Chat cơ bản.

## Thu hoạch

Sau khi thực hiện đề tài bản thân đã học hỏi, tiếp thu được nhiều công nghệ và kiến thức mới có thể giúp ích cho bản thân trong việc nghiên cứu, học tập sau này cụ thể như:

* Học được cách sử dụng Node.js để làm một webserver.
* Sử dụng được thư viện Socket.IO để tạo trao đổi thông tin từ client này tới client khác theo thời gian thực .
* Thiết kế được giao diện người dùng động bằng HTML, CSS, JavaScript kết hợp với cả thư viện Jquery.
* Biết cách kết nối từ server tới hệ quản trị cơ sở dữ liệu MySQL để lưu thông tin người dùng, dữ liệu người dùng gửi đi.
* Tự thiết kế được một mô hình cơ sở dữ liệu chứa dữ liệu.
* Học hỏi được các cách giải quyết vấn đề trong việc trao đổi thông tin theo thời gian thực, cách truy xuất, sửa đổi các thông tin trong cơ sở dữ liệu.

## Ưu điểm

Ứng dụng đã xây được có được những ưu điểm như sau:

* Trao đổi thông tin văn bản giữa hai người dùng nhanh chóng theo thời gian thực.
* Hiển thị thông báo khi người dùng có tin nhắn từ người dùng khác chưa đọc.
* Thanh tìm kiếm trả về kết quả nhanh chóng cho người dùng.

## Hạn chế - nguyên nhân

Tuy đã cố gắng tìm hiểu các giải pháp, cách giải quyết vấn đề khác nhau để tối ưu, hoàn thiện ứng dụng nhưng vẫn còn tồn tại một số hạn chế chưa thể khắc phục để giúp quá trình sử dụng của người dùng thoải mái, mượt mà hơn, chẳng hạn như:

* Khi người dùng tải lại trang thì sẽ bị chuyển lại trang đăng nhập mà không ở lại trang Chat. Do sử dụng JavaScript để bắt sự kiện người dùng thoát khỏi trình duyệt sử dụng ứng dụng
* Giao diện còn đơn sơ, có thể chưa thể đáp ứng được hoàn toàn bộ các yêu cầu của tất cả người dùng.
* Cách truy xuất, cập nhật dữ liệu có thể chưa tối ưu một cách tốt nhất.
* Các hộp thoại chứa tin nhắn mới chưa được chèn lên trên đầu danh sách cách hộp thoại của người dùng trao đổi với người dùng khác.

## Hướng phát triển

Ứng dụng Chat cơ bản vẫn còn có thể phát triển thêm để trở thành một ứng dụng có thể áp dụng vào các dự án thực tế, chẳng hạn như:

* Sắp xếp những người gửi tin nhắn mới lên đầu danh sách.
* Xây dựng chức năng tạo nhóm cũng như là giao tiếp với nhứng người dùng trong cùng nhóm đó.
* Thêm chức năng video call để có thể giao tiếp trực quan hơn, thú vị hơn.
* Gửi các tin nhắn có chưa Sticker, icon vui nhộn tạo sự hứng thú, vui tươi, đa dạng cho cuộc trò chuyện.
* Gửi hình ảnh cho người khác để cuộc trò chuyện thêm trực quan, rõ ràng thông tin của người dùng.
* Phát triển ứng dụng trở thành ứng dụng đa nền tảng.

**TÀI LIỆU THAM KHẢO**

1. **Satheesh, Mithun, Bruno Joseph D'mello, and Jason Krol**.*Web development with MongoDB and NodeJs*. Chapter 1. Packt Publishing Ltd, 2015.
2. **Rai, Rohit**. *Socket.IO real-time web application development*. Chapter 2, 3 and 5. Packt Publishing, 2013.
3. **Node.js - Run JavaScript Everywhere**:<https://nodejs.org/en>
4. **Socket.IO**:[https://socket.io/](https://socket.io/docs/v4/)
5. **StackOverflow**: <https://stackoverflow.com/>