



Bao cao ngon ngu kich ban

Công nghệ phần mềm (Đại học Điện lực)



Scan to open on Studocu

**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC
PHẦN NGÔN NGỮ KỊCH BẢN**

ĐỀ TÀI: XÂY DỰNG CHATBOX

Sinh Viên Thực Hiện	: NGUYỄN TÙNG LÂM
Giảng Viên Hướng Dẫn	: ĐỖ ĐỨC CƯỜNG
Khoa	: CÔNG NGHỆ THÔNG TIN
Chuyên Ngành	: CÔNG NGHỆ PHẦN MỀM
Lớp	: D15CNPM5
Khóa	: 2020 – 2025

PHIẾU CHẤM ĐIỂM

Sinh viên thực hiện:

Họ và tên	Chữ ký	Ghi chú
NGUYỄN TÙNG LÂM 20810310479		

Giảng viên chấm:

Họ và tên	Chữ ký	Ghi chú
Giảng viên chấm 1 :		
Giảng viên chấm 2 :		

MỤC LỤC

Tran

	g
LỜI MỞ ĐẦU.....	1
CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG, XÁC LẬP DỰ ÁN.....	2
1. Mục Đích.....	2
2. Khảo Sát Hiện Trạng: Yêu Cầu Chức Năng:.....	2
3. Kế Hoạch Dự Án:.....	2
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	4
2.1 Biểu đồ Use case của hệ thống.....	4
2.2 Biểu đồ Sequence (Biểu đồ tuần tự).....	4
2.2.1 Biểu đồ trình tự đăng nhập.....	4
2.2.2 Biểu đồ trình tự đăng ký.....	5
2.3. Biểu đồ Class (Biểu đồ lớp).....	6
CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG.....	7
3.1. Tổng quan về Javascript.....	7
3.1.1. Định nghĩa.....	7
3.1.2. Ưu điểm, nhược điểm.....	7
3.1.3. Ứng dụng.....	9
3.2. Tổng quan về NodeJS, Express.....	11
3.2.1. Định nghĩa.....	11
3.2.2. Ưu điểm, nhược điểm.....	11
3.2.3. Cài đặt NodeJS.....	13
3.3. Tổng quan về MongoDB.....	14
3.3.1. Định nghĩa.....	14
3.3.2. Ưu điểm, nhược điểm.....	14
3.3.3. Cài đặt MongoDB.....	15
3.4 Tổng quan về Vite + React.....	15
3.4.1 Định Nghĩa.....	15
3.4.2 Ưu điểm và Nhược điểm.....	16
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM.....	18
4.1 Cấu trúc thư mục.....	18

4.2 Các API.....	19
4.2.1 API Đăng nhập, Đăng ký.....	19
4.2.2 Chats API.....	20
4.2.3 Message API.....	21
4.3 Giao diện Front-end.....	22
4.3.1 Giao diện đăng nhập.....	22
4.3.2 Giao diện đăng ký.....	22
4.3.3 Giao diện trang chủ.....	23
4.3.4 Giao diện trang chủ và thông báo.....	23
KẾT LUẬN.....	24
TÀI LIỆU THAM KHẢO.....	25

LỜI MỞ ĐẦU

Ngày nay thế giới đang phát triển công nghệ 4.0 , trào lưu kinh doanh qua mạng ngày càng trở lên phổ biến, internet phát triển kéo theo nhiều dịch vụ phát triển theo đi cùng. Và việc tiếp cận với internet hiện nay không còn mấy xa lạ với con người được trở thành phương tiện truyền thông được nhiều người sử dụng nhất trên thế giới. Không chỉ dùng internet để trao đổi thông tin, đọc báo, giải trí....mọi người còn dùng internet để chia sẻ những khoảnh khắc, địa điểm du lịch tuyệt vời mà mình được trải nghiệm.

Du lịch là tất cả về khám phá những địa điểm mới, văn hóa, ẩm thực, nghi lễ và phong cách sống. Chúng ta cũng đi du lịch bởi vì khoảng cách và sự khác biệt kích thích sự tò mò của tất cả, điều mà chúng ta không thể quan sát bằng cách ngồi ở nhà. Bản thân việc đi du lịch có những lợi thế, vì nó khiến người ta quên đi những lo lắng, buồn rầu và thất vọng.

Vì vậy chúng em xin được chọn đề tài: “*XÂY DỰNG CHATBOX*”.

CHƯƠNG 1: KHẢO SÁT HIỆN TRẠNG, XÁC LẬP DỰ ÁN

1. Mục Đích

Dự án nhắn tin trên nền tảng web nhằm cung cấp một trải nghiệm giao tiếp linh hoạt và thuận tiện, đồng thời đảm bảo tính riêng tư và an toàn cho người dùng.

2. Khảo Sát Hiện Trạng:

Yêu Cầu Chức Năng:

- **Gửi Tin Nhắn:** Khả năng gửi và nhận tin nhắn văn bản, hình ảnh, và tệp đính kèm.
- **Bảo Mật:** Bảo vệ thông tin người dùng và tin nhắn trên kênh truyền.
- **Giao Diện Người Dùng: Dễ Sử Dụng:** Giao diện thân thiện, đơn giản và dễ sử dụng.
- **Tích Hợp Emoji:** Tính năng tăng tính tương tác và giải trí

Bảo Mật và Quản Lý Tài Khoản:

- **Xác Thực An Toàn:** Đăng nhập bằng các phương tiện xác thực an toàn.
- **Quản Lý Tài Khoản:** Khả năng quản lý thông tin cá nhân và danh bạ.

Hiệu Suất:

- **Tính Mượt Mà:** Đảm bảo ứng dụng hoạt động mượt mà và nhanh chóng trên các trình duyệt web phổ biến.

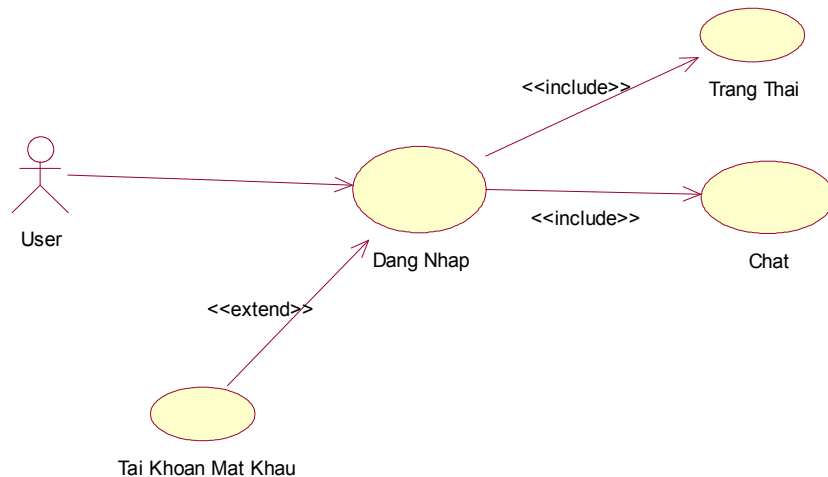
3. Kế Hoạch Dự Án:

- **Phân Tích Yêu Cầu:** Xác định yêu cầu cụ thể của ứng dụng, bao gồm chức năng, yêu cầu bảo mật, và trải nghiệm người dùng.
- **Thiết Kế Giao Diện và Trải Nghiệm Người Dùng (UI/UX):** Tạo wireframes và mô hình giao diện người dùng dựa trên yêu cầu phân tích. Thiết kế giao diện người dùng dễ sử dụng và thân thiện.
- **Phát Triển Ứng Dụng Web:** Sử dụng framework như React hoặc Vue.js cho việc xây dựng ứng dụng web hiện đại. Tối ưu hóa mã nguồn và đảm bảo tương thích trên các trình duyệt phổ biến.

- **Kiểm Thử:** Thực hiện kiểm thử chức năng và bảo mật trước khi triển khai. Tích hợp kiểm thử người dùng để thu thập phản hồi và đánh giá trải nghiệm.
- **Triển Khai và Duy Trì:** Triển khai ứng dụng trên môi trường web (hosting) và cập nhật định kỳ. Duy trì để đảm bảo tính ổn định và bảo mật.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

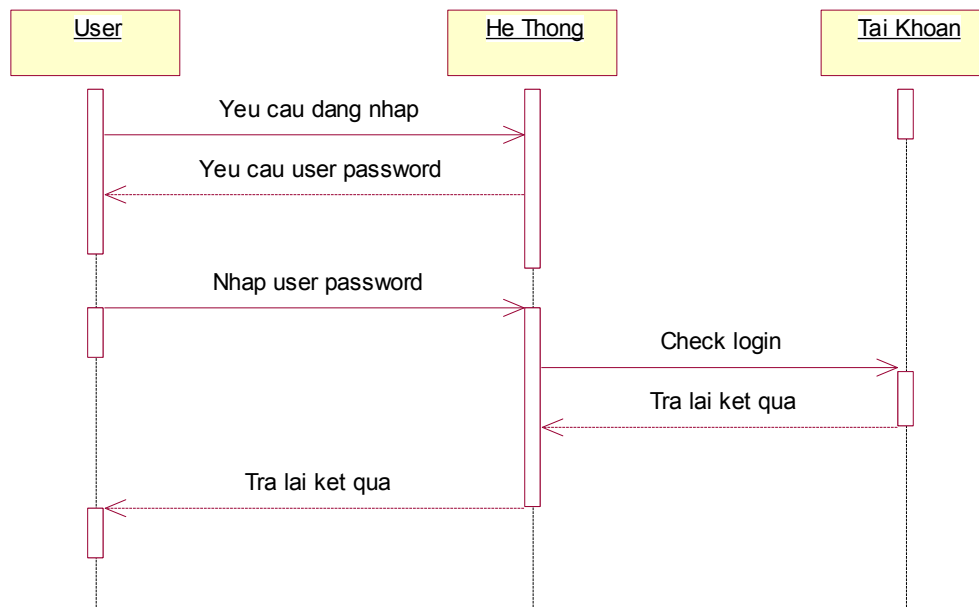
2.1 Biểu đồ Use case của hệ thống



Hình 2.1: Biểu đồ usecase của hệ thống

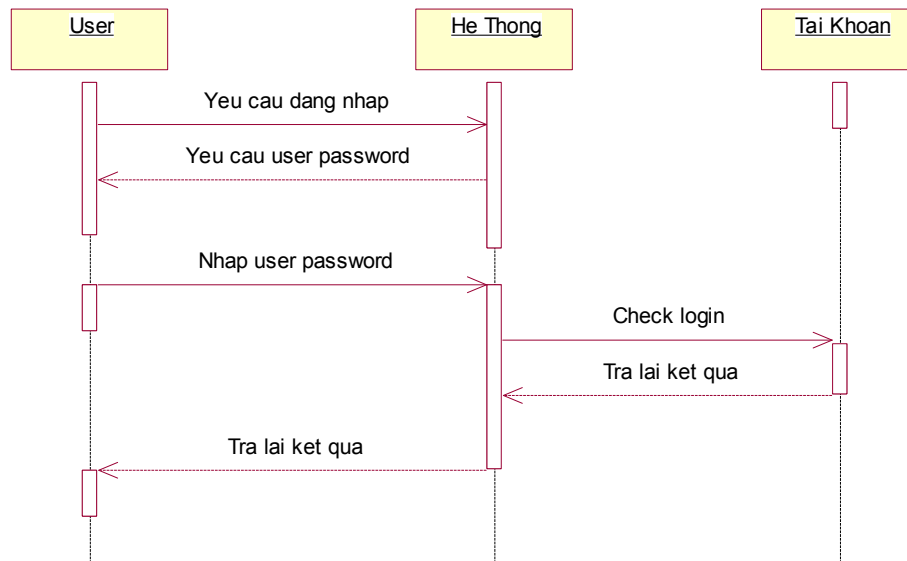
2.2 Biểu đồ Sequence (Biểu đồ tuần tự)

2.2.1 Biểu đồ trình tự đăng nhập



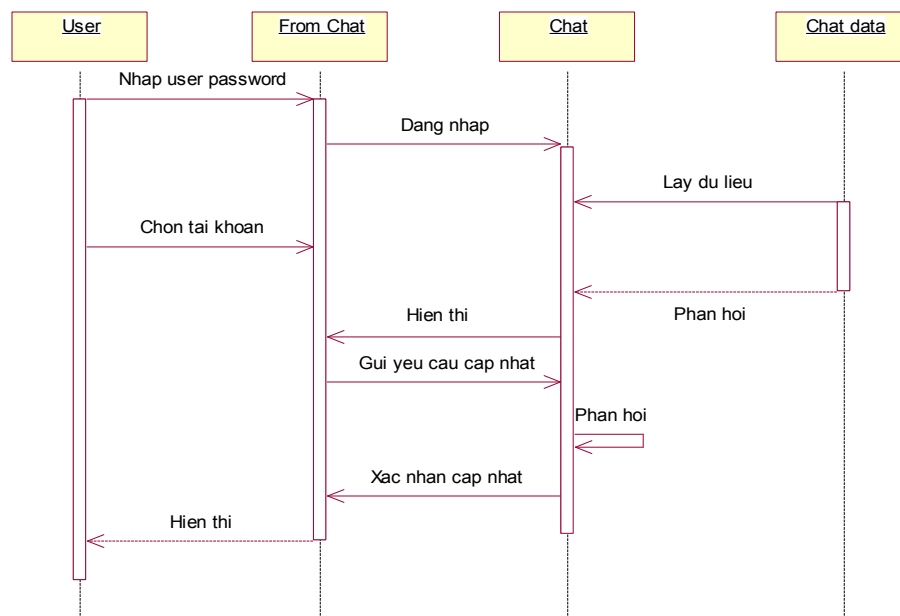
Hình 2.2: Biểu đồ trình tự Đăng nhập

2.2.2 Biểu đồ trình tự đăng ký



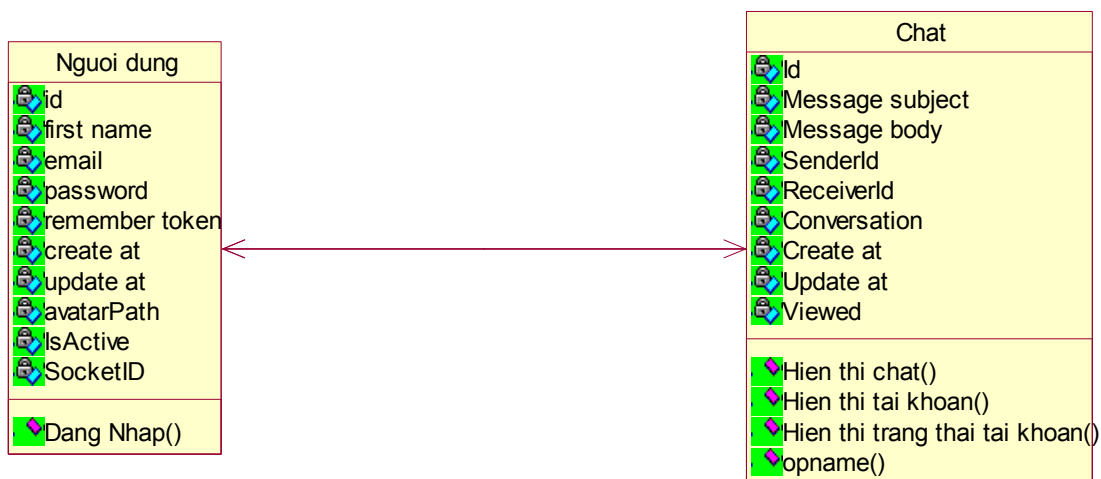
Hình 2.3: Biểu đồ trình tự Đăng ký

2.2.3 Biểu đồ trình tự Chat



Hình 2.4: Biểu đồ trình tự của Chat

2.3. Biểu đồ Class (Biểu đồ lớp)



Hình 2.5 Biểu đồ Class

CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG

3.1. Tổng quan về Javascript

3.1.1. Định nghĩa

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phân tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

3.1.2. Ưu điểm, nhược điểm

*Ưu Điểm

Dễ học và triển khai:

JavaScript là một ngôn ngữ dễ học, đặc biệt là đối với những người mới học lập trình.

Không cần cài đặt hoặc biên dịch trước, có thể thực thi trực tiếp trên trình duyệt.

Phát triển web động:

JavaScript là ngôn ngữ chủ yếu cho việc tạo ra các trang web động và tương tác cao.

Dùng để thay đổi nội dung trang web mà không cần tải lại toàn bộ trang.

Đa nhiệm và bất đồng bộ:

Hỗ trợ các chức năng đa nhiệm thông qua sự hỗ trợ của sự bất đồng bộ (asynchronous) thông qua các callback và promises.

Hỗ trợ mạnh mẽ cho các thư viện và framework:

Có nhiều thư viện và framework mạnh mẽ như React, Angular, Vue.js được xây dựng trên nền tảng JavaScript.

Chạy trên mọi trình duyệt:

JavaScript được hỗ trợ trên hầu hết các trình duyệt hiện đại, giúp đảm bảo tính tương thích đa nền tảng.

Nhược Điểm*Bảo mật:**

JavaScript chạy trên trình duyệt của người dùng, điều này có thể tạo ra các vấn đề về bảo mật như mã độc (malware) và tấn công Cross- Site Scripting (XSS).

Thực thi chậm trên một số thiết bị có tài nguyên hạn chế:

Trong môi trường thiết bị có tài nguyên hạn chế, việc thực thi mã JavaScript có thể gây ra hiệu suất kém.

Single-threaded nature:

JavaScript là ngôn ngữ đơn luồng, điều này có nghĩa là nó chỉ có thể thực hiện một nhiệm vụ tại một thời điểm. Điều này có thể tạo ra vấn đề với các tác vụ đòi hỏi nhiều xử lý đồng thời.

Quản lý dự án phức tạp:

Trong các dự án lớn, quản lý mã nguồn JavaScript có thể trở nên khó khăn vì nó không cung cấp cấu trúc chặt chẽ như một số ngôn ngữ khác.

Không có kiểu dữ liệu tường minh:

JavaScript là ngôn ngữ không kiểu, điều này có thể dẫn đến những lỗi không mong muốn do kiểu dữ liệu.

3.1.3. Ứng dụng

JavaScript được sử dụng rộng rãi trong nhiều ứng dụng khác nhau, đặc biệt là trong phát triển web. Dưới đây là một số ứng dụng phổ biến của JavaScript:

Phát triển trang web động:

JavaScript làm cho các trang web trở nên động, tương tác và linh hoạt hơn. Nó được sử dụng để thay đổi nội dung trang web mà không cần tải lại trang, cải thiện trải nghiệm người dùng.

Phát triển ứng dụng web đơn trang (SPA):

JavaScript thường được sử dụng trong các framework như React, Angular, và Vue.js để xây dựng các ứng dụng web đơn trang, giúp tăng cường hiệu suất và tương tác người dùng.

Xử lý sự kiện người dùng:

JavaScript được sử dụng để bắt lấy và xử lý sự kiện từ người dùng như nhấn nút, di chuyển chuột, hoặc gõ phím.

Ajax (Asynchronous Javascript and XML):

JavaScript, cùng với XMLHttpRequest hoặc Fetch API, được sử dụng để tạo các yêu cầu HTTP bất đồng bộ, giúp tải dữ liệu từ máy chủ mà không làm tải lại trang.

Validation form trên trình duyệt

JavaScript thường được sử dụng để kiểm tra dữ liệu người dùng ngay trên trình duyệt trước khi gửi dữ liệu lên máy chủ.

Animation và hiệu ứng trang web:

JavaScript giúp tạo ra các hiệu ứng đồ họa và animation trên trang web, cải thiện tính tương tác và thu hút người dùng.

Đối tượng JSON (Javascript Object Notation):

JavaScript thường được sử dụng để parse và tạo JSON, giúp truyền dữ liệu giữa máy chủ và trình duyệt một cách dễ dàng.

Phát triển ứng dụng di động:

Các framework như React Native cho phép sử dụng JavaScript để phát triển ứng dụng di động cho cả iOS và Android.

Game và đồ họa trực tuyến:

Một số thư viện như Phaser.js được sử dụng để phát triển game và đồ họa trực tuyến bằng JavaScript.

IoT (Internet of Things)

JavaScript cũng có thể được sử dụng trong phát triển ứng dụng cho Internet of Things (IoT), đặc biệt là khi kết hợp với Node.js để xây dựng các ứng dụng máy chủ IoT.

3.2. Tổng quan về NodeJS, Express

3.2.1. Định nghĩa

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

3.2.2. Ưu điểm, nhược điểm

***Ưu Điểm**

Bất đồng bộ và Non-blocking I/O:

Node.js sử dụng mô hình bất đồng bộ (asynchronous) và non-blocking I/O, giúp xử lý hàng loạt yêu cầu mà không phải chờ đợi, tăng hiệu suất và khả năng mở rộng.

Chung mã nguồn giữa client và server:

JavaScript có thể chạy cả trên máy chủ và trình duyệt, giúp đơn giản hóa quá trình phát triển và duy trì mã nguồn.

Community mạnh mẽ và sự hỗ trợ lớn:

Node.js có cộng đồng phát triển mạnh mẽ, với nhiều thư viện và framework có sẵn để giúp việc phát triển.

Hiệu suất cao:

Với mô hình không chặn, Node.js thường có hiệu suất cao đặc biệt trong việc xử lý các kết nối đồng thời.

Module hóa:

Node.js sử dụng hệ thống module giúp tổ chức mã nguồn thành các phần nhỏ, dễ duy trì và phát triển.

Có thể sử dụng cho ứng dụng thời gian thực:

Đối với các ứng dụng yêu cầu xử lý thời gian thực (real-time), như chat hoặc trò chơi trực tuyến, Node.js là một lựa chọn phổ biến.

Dễ tích hợp với các công nghệ khác:

Node.js dễ tích hợp với nhiều công nghệ khác nhau và có thể sử dụng trong các môi trường đa ngôn ngữ.

***Nhược Điểm**

Không phù hợp cho các tác vụ tính toán nặng:

Do single-threaded nature, Node.js không phù hợp cho các tác vụ tính toán nặng mà yêu cầu sự xử lý đa luồng.

Callback Hell:

Trong mô hình bất đồng bộ, có thể xảy ra tình trạng "Callback Hell" khi có quá nhiều callback lồng nhau, làm cho mã nguồn khó đọc và duy trì.

Chưa có hệ thống chuẩn quản lý dependency:

Trái với một số ngôn ngữ khác, Node.js không có một hệ thống quản lý dependency chính thức, điều này có thể tạo ra vấn đề khi quản lý các thư viện và phiên bản.

Thiếu một số tính năng tích hợp sẵn:

Node.js không có nhiều tính năng tích hợp sẵn như các framework khác, đôi khi bạn cần phải sử dụng các thư viện bên thứ ba để thực hiện các chức năng cụ thể.

Khó khăn khi xử lý lỗi:

Trong mô hình bất đồng bộ, việc xử lý lỗi có thể trở nên phức tạp hơn so với mô hình đồng bộ truyền thống.

3.2.3. Cài đặt NodeJS

Truy cập trang chính thức của Node.js:

Mở trình duyệt web và truy cập trang <https://nodejs.org/>.

Tải phiên bản LTS (Recommended for Most Users):

Trong trang chính, bạn nên tải bản LTS (Long Term Support) vì đây là phiên bản được khuyến nghị cho hầu hết người dùng.

Chạy trình cài đặt:

Mở tệp cài đặt sau khi đã tải xong và làm theo hướng dẫn trên màn hình để cài đặt Node.js và npm (Node Package Manager).

Kiểm tra cài đặt:

Mở Command Prompt hoặc PowerShell và gõ các lệnh sau để kiểm tra phiên bản đã cài đặt: `node -v` và `npm -v`

3.3. Tổng quan quan về MongoDB

3.3.1. Định nghĩa

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. [MongoDB](#) sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

3.3.2. Ưu điểm, nhược điểm

***Ưu Điểm**

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

***Nhược Điểm**

- Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
- Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

3.3.3. Cài đặt MongoDB

Truy cập trang chính thức của MongoDB

Mở trình duyệt web và truy cập trang [MongoDB Community Server Download](#).

Tải và chạy trình cài đặt

Tải phiên bản Community Server và chạy trình cài đặt. Làm theo hướng dẫn trên màn hình để cài đặt MongoDB.

Cài đặt dịch vụ MongoDB

MongoDB cung cấp một dịch vụ Windows để bạn có thể chạy MongoDB như một dịch vụ hệ thống. Bạn có thể cài đặt dịch vụ này trong quá trình cài đặt MongoDB.

Kiểm tra cài đặt

Mở Command Prompt và gõ lệnh sau để kiểm tra phiên bản đã cài đặt: `mongo --version`

3.4 Tổng quan về Vite + React

3.4.1 Định Nghĩa

- Vite: Vite là một công cụ xây dựng ứng dụng web tốt nhất cho việc phát triển front-end hiện đại. Nó sử dụng các đặc tính của ES modules để tạo ra

môi trường phát triển nhanh chóng và hiệu quả.

- **React:** React là một thư viện JavaScript phổ biến được phát triển bởi Facebook, được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web.

3.4.2 Ưu điểm và Nhược điểm

* Ưu điểm

- **Tốc độ phát triển nhanh:** Vite sử dụng ES modules để tạo ra một môi trường phát triển với thời gian khởi động nhanh, giúp tăng tốc quá trình phát triển.

- **Hot Module Replacement (HMR):** Vite hỗ trợ HMR, cho phép bạn cập nhật code và thấy kết quả ngay lập tức mà không cần làm mới trình duyệt.

- **Tích hợp tốt với React:** Vite được tối ưu cho việc sử dụng với React, với các plugin và cấu hình mặc định giúp tối ưu hóa trải nghiệm phát triển với React.

* Nhược điểm

- Chưa được ổn định hoàn toàn: Do Vite là một công cụ mới, có thể xuất hiện các vấn đề không đáng kể hoặc chưa được kiểm chứng trong môi trường sản xuất.

- Yêu cầu hiểu biết sâu rộng: Để tận dụng hết tiềm năng của Vite, bạn có thể cần phải hiểu biết sâu rộng về các khái niệm như ES modules, bundling, và cấu hình.

3.4.3 Cài đặt Vite + React

Bước 1: Tạo dự án React với Vite

Sử dụng Vite để tạo dự án React:

Mở terminal và chạy lệnh sau để tạo một dự án React mới với Vite:

“npx create-vite my-react-app --template react”

Trong đó, my-react-app là tên thư mục của dự án

Bước 2: Di chuyển vào thư mục dự án

“cd my-react-app”

Bước 3: Cài đặt các thư viện cần thiết

“npm install”

Bước 4: Chạy dự án

“npm run dev”

Sau khi chạy lệnh trên, Vite sẽ khởi động máy chủ phát triển và mở trình duyệt

Lưu ý: Nếu sử dụng Yarn, có thể thay thế npm bằng yarn trong các lệnh trên.

3.5 Tổng quan về Socket.io

3.5.1 Định nghĩa

Socket.IO là một thư viện JavaScript được sử dụng để thực hiện giao tiếp real-time hai chiều giữa client và server thông qua WebSockets. Nó hỗ trợ nhiều phương tiện giao tiếp như WebSockets, Polling, và một số kỹ thuật khác để tạo ra ứng dụng real-time có khả năng mở rộng. Socket.IO giúp giảm độ phức tạp của việc xây dựng ứng dụng real-time và làm cho nó trở nên dễ dàng hơn.

3.5.2 Ưu điểm và Nhược điểm

* Ưu điểm

- **Real-Time Communication:** Socket.IO cung cấp một cách dễ dàng để thiết lập kết nối real-time giữa client và server, giúp truyền thông dữ liệu ngay lập tức. Hỗ Trợ - -- **Nhiều Trình Duyệt và Thiết Bị:** Socket.IO tương thích với nhiều trình duyệt và thiết bị khác nhau, giúp đảm bảo rằng ứng dụng real-time của bạn có thể hoạt động trên nhiều nền tảng.

- **Tự Động Xác Định Kết Nối:** Socket.IO tự động xác định xem liệu client và server có thể sử dụng WebSockets hay không, và nếu không thì chuyển sang các kỹ thuật khác như Polling.

* Nhược điểm

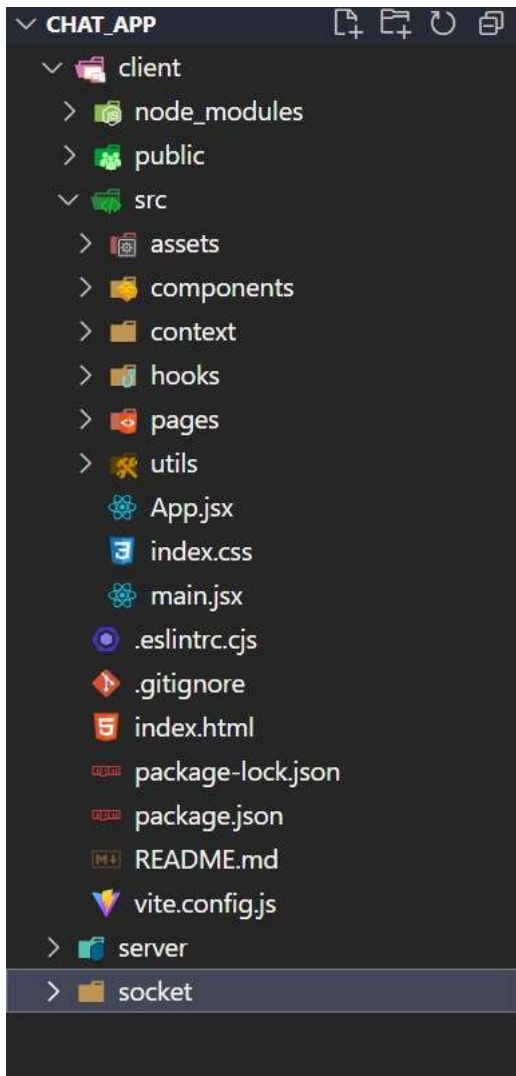
Tăng Tải Server: Nếu có nhiều kết nối đồng thời, server có thể phải xử lý một lượng lớn kết nối, điều này có thể dẫn đến tăng tải đột ngột và yêu cầu tài nguyên server cao.

Khả Năng Bảo Mật: Việc sử dụng WebSockets có thể gặp một số vấn đề liên quan đến bảo mật, và cần được cân nhắc cẩn thận để tránh các lỗ hổng bảo mật.

Khả Năng Nâng Cấp: Mặc dù Socket.IO hỗ trợ nhiều cách để giao tiếp real-time, nhưng sự phức tạp của nó có thể làm tăng độ khó khăn trong quá trình nâng cấp ứng dụng.

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1 Cấu trúc thư mục

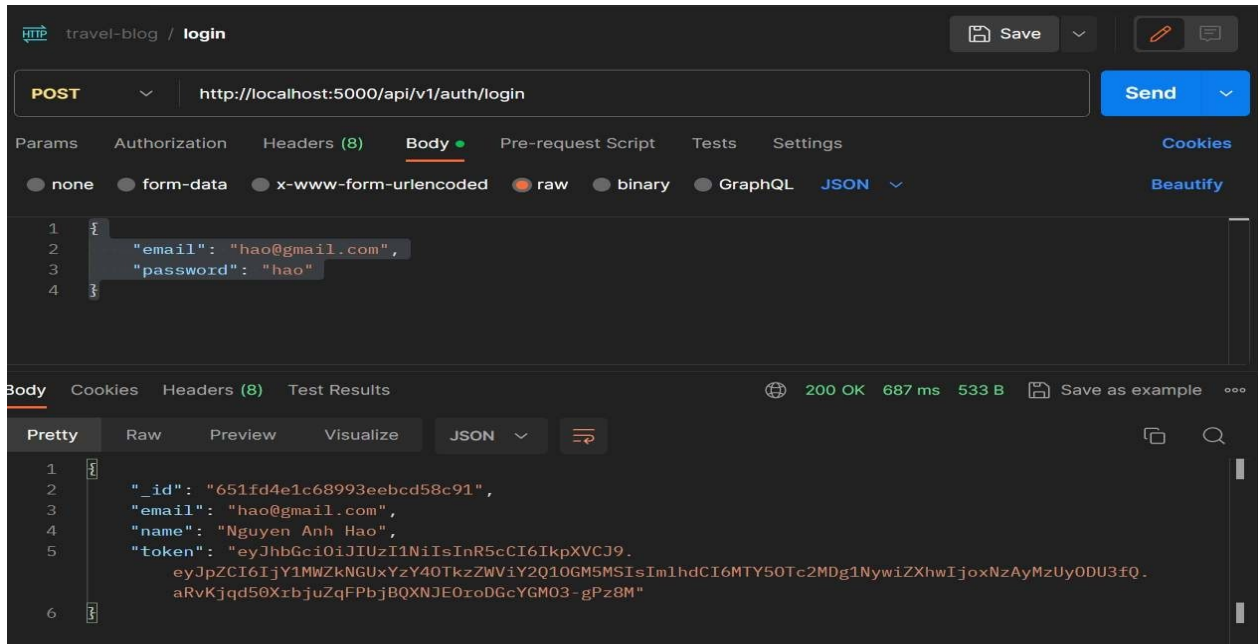


Hình 4.1: Cấu trúc thư mục

4.2 Các API

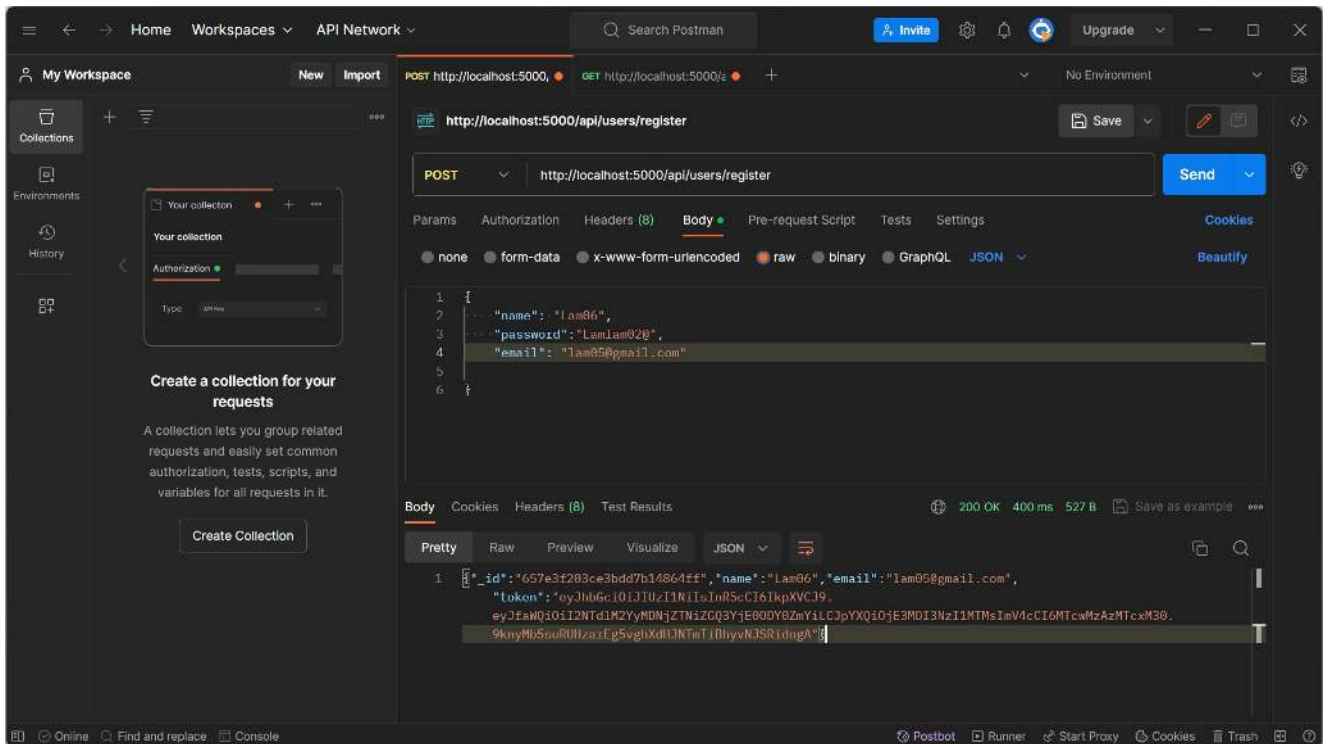
4.2.1 API Đăng nhập, Đăng ký

- Đăng nhập



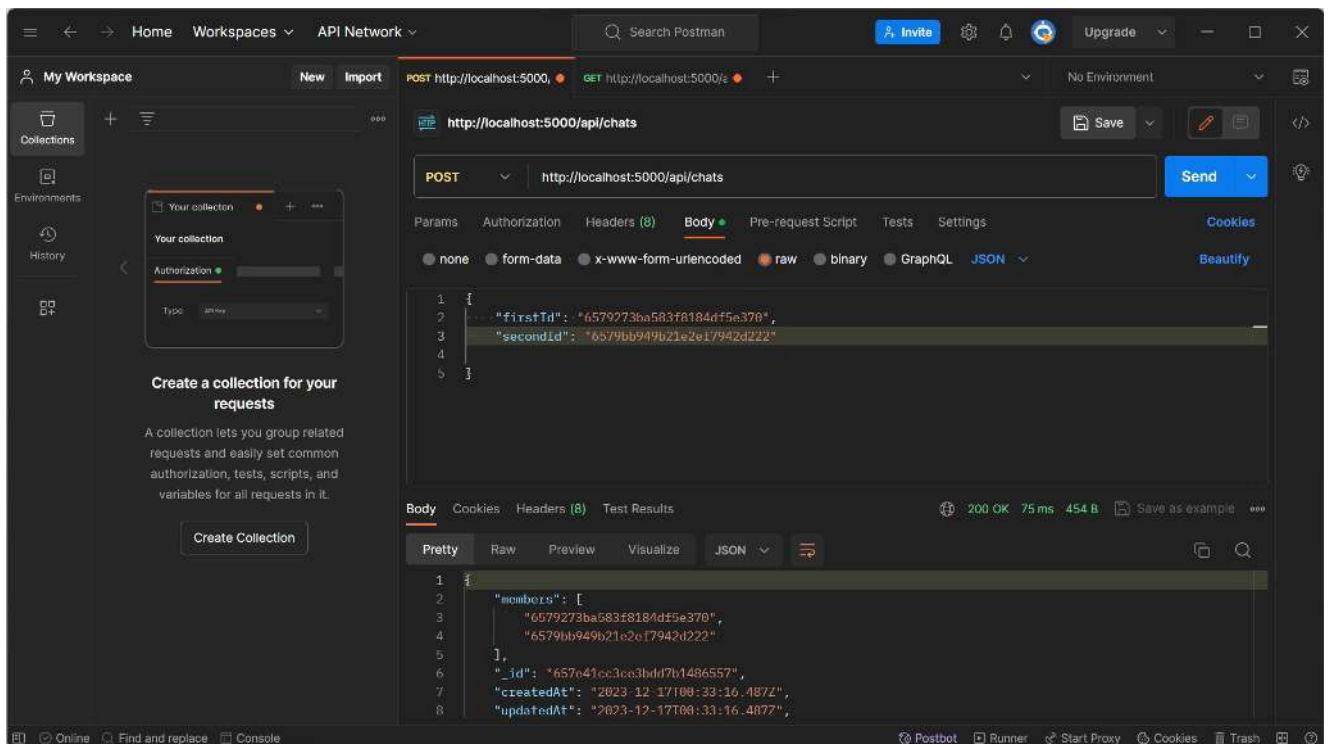
Hình 4.2: API Đăng nhập

- Đăng ký



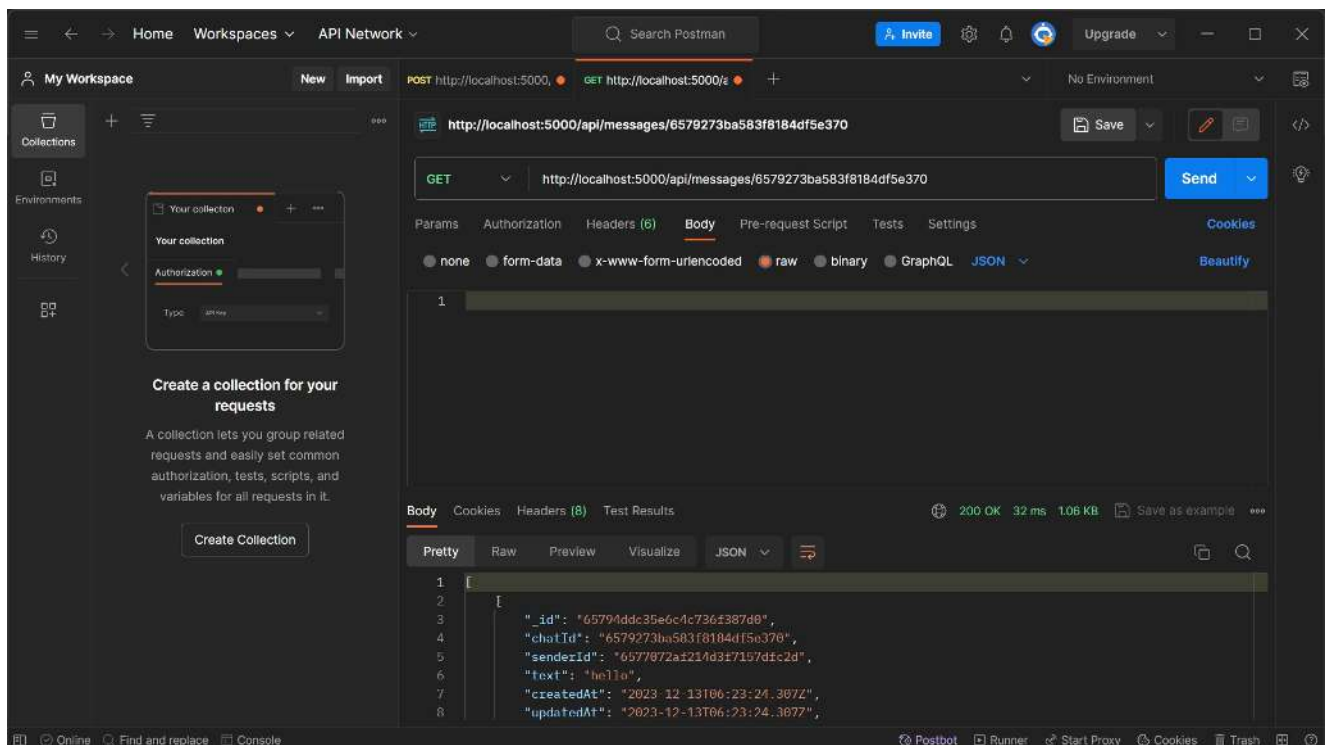
Hình 4.3: API Đăng ký

4.2.2 Chats API



Hình 4.4 Chats API

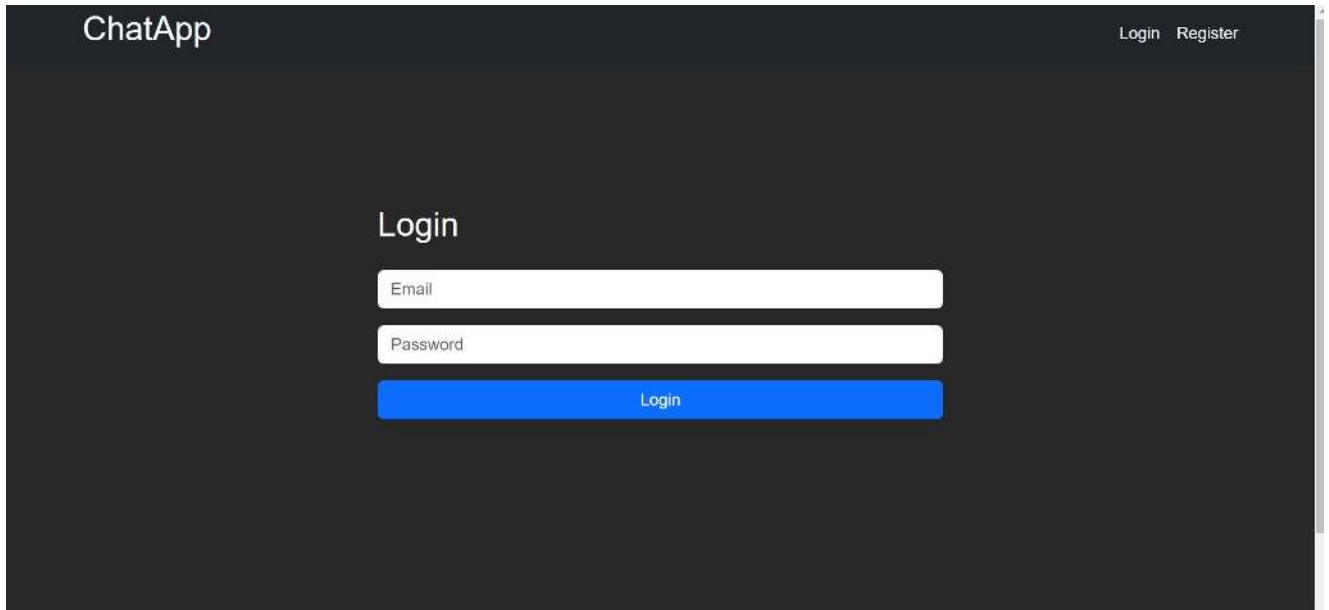
4.3.2 Message API



Hình 4.5 Message API

4.3 Giao diện Front-end

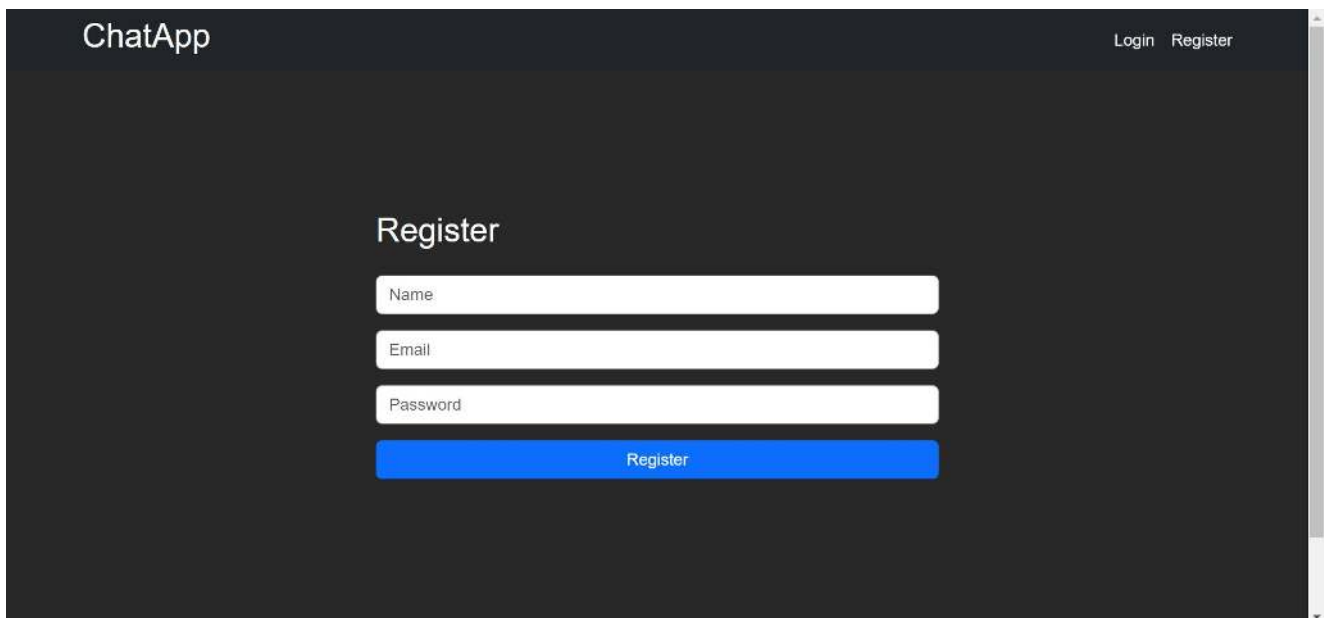
4.3.1 Giao diện đăng nhập



The screenshot shows the login interface of a web application named 'ChatApp'. The header contains the app name 'ChatApp' on the left and links for 'Login' and 'Register' on the right. The main content area is titled 'Login' and features two input fields: 'Email' and 'Password'. Below these fields is a blue button labeled 'Login'.

Hình 4.6: Form đăng nhập

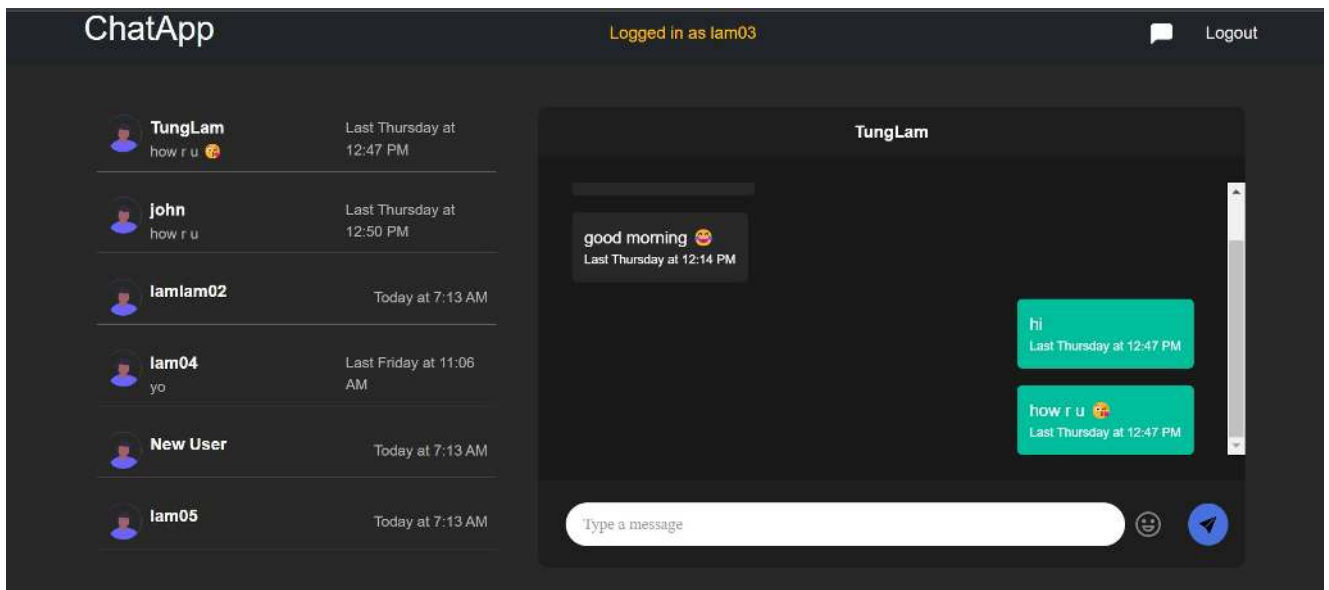
4.3.3 Giao diện đăng ký



The screenshot shows the registration interface of the 'ChatApp'. The header is identical to the login page, with 'ChatApp' on the left and 'Login' and 'Register' links on the right. The main content area is titled 'Register' and contains three input fields: 'Name', 'Email', and 'Password'. A blue button labeled 'Register' is positioned at the bottom of the form.

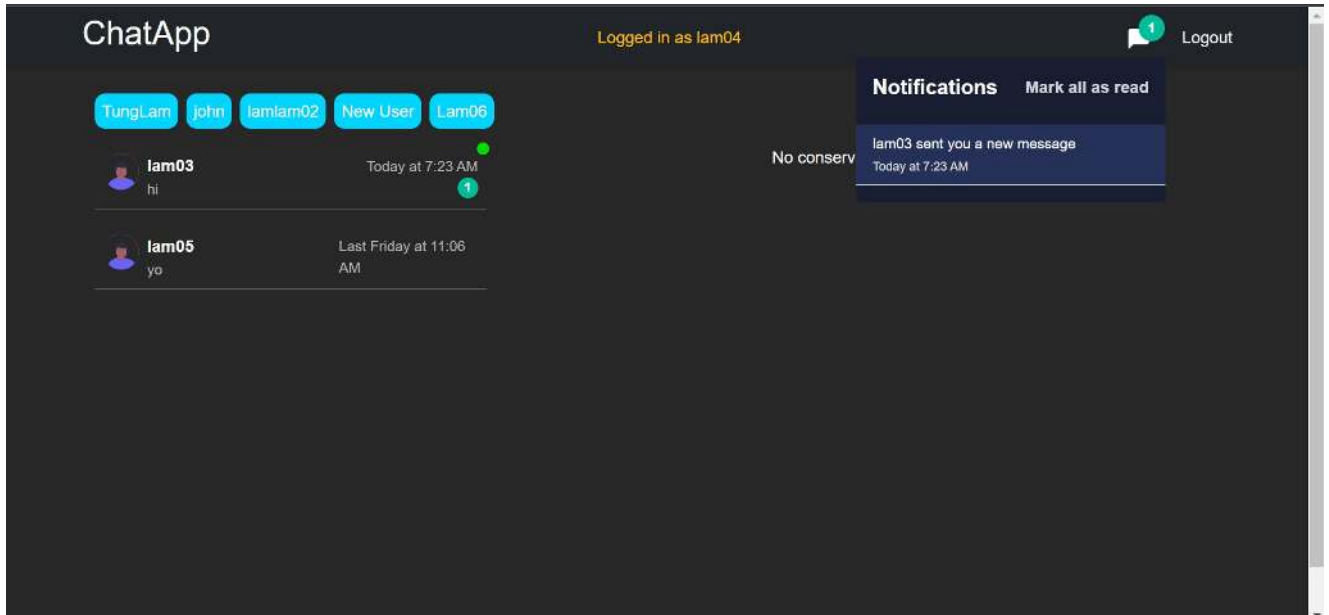
Hình 4.7: Form đăng ký

4.3.4 Giao diện trang chủ



Hình 3.11: Form trang chủ

4.3.5 Giao diện trang chủ và thông báo



KẾT LUẬN

Sau một thời gian tích cực làm việc, nghiên cứu của nhóm em với sự hướng dẫn, chỉ bảo nhiệt tình của thầy giáo giảng dạy bộ môn – thầy Đỗ Đức Cường, nhóm em đã hoàn thành bản báo cáo của mình với đề tài: “Xây dựng ChatBox”. Trong quá trình nghiên cứu và xây dựng lên chương trình này, nhóm em đã rất cố gắng nhưng đồ án này vẫn còn có nhiều thiếu sót.

Hạn chế

- Source chưa được deploy lên Internet
- Thiếu kinh nghiệm trong việc phát triển dự án nên gặp nhiều vấn đề về cách tổ chức dự án và thời gian thực hiện

Hướng phát triển

- Chỉnh sửa giao diện đẹp mắt, trực quan, tối ưu hóa hiển thị và độ mượt mà của các trang người dùng
- Nâng cấp thêm chức năng có thể group chat và gửi được hình ảnh
- Deploy lên Internet

Chúng em rất mong nhận được sự đóng góp, chỉ bảo của thầy, cô. Chúng em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] Tài liệu Javascript Amazon: <https://aws.amazon.com/vi/what-is/javascript/>
- [2] Tài liệu NodeJS: <https://viblo.asia/p/tong-quan-ve-node-js-924lJra0lPM>
- [3] Tài liệu Express: <https://topdev.vn/blog/express-js-la-gi/>
- [4] Tài liệu MongoDB: <https://topdev.vn/blog/mongodb-la-gi/>