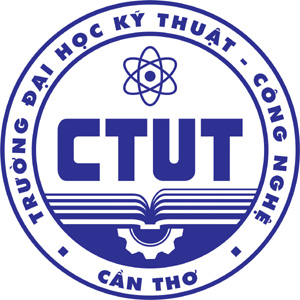
**TRƯỜNG ĐẠI HỌC - KỸ THUẬT CÔNG NGHỆ CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**

**☜☜☜†☞☞☞**

****

**ĐỒ ÁN**

**XÂY DỰNG WEBSITE MẠNG XÃ HỘI**

**GIẢNG VIÊN HƯỚNG DẪN SINH VIÊN THỰC HIỆN**

**Ths. Võ Thành Vinh** **Phạm Hữu Tiến - 2101074**

**Lê Hoàng Phúc - 2101091**

**Phạm Tuấn Vũ – 2100761**

**Phan Nguyễn Hoàng Trân - 2101538**

**Lớp: KTPM0121**

***Cần Thơ, tháng 12 năm 2024***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Họ và tên SVTH: Phạm Hữu Tiến MSSV: 210174

Lê Hoàng Phúc MSSV: 2101091

Phạm Tuấn Vũ MSSV: 2100761

Phan Nguyễn Hoàng Trân MSSV: 2101538

Tên đề tài: Xây dựng website mạng xã hội Nemo

Họ và tên GVHD: ThS. Võ Thành Vinh

Nội dung nhận xét:

Cần Thơ, ngày……tháng……năm 2024

Giảng viên hướng dẫn

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

Họ và tên SVTH: Phạm Hữu Tiến MSSV: 210174

Lê Hoàng Phúc MSSV: 2101091

Phạm Tuấn Vũ MSSV: 2100761

Phan Nguyễn Hoàng Trân MSSV: 2101538

Tên đề tài: Xây dựng website mạng xã hội Nemo

Họ và tên GVPB: ThS.Lâm Thanh Toản

Nội dung nhận xét:

Cần Thơ, ngày……tháng……năm 2024

Giảng viên phản biện

**NHẬN XÉT CỦA HỘI ĐỒNG ĐÁNH GIÁ ĐỒ ÁN**

Họ và tên SVTH: Phạm Hữu Tiến MSSV: 210174

Lê Hoàng Phúc MSSV: 2101091

Phạm Tuấn Vũ MSSV: 2100761

Phan Nguyễn Hoàng Trân MSSV: 2101538

Tên đề tài: Xây dựng website mạng xã hội Nemo

Họ và tên Hội đồng đánh giá Đồ Án :

1. Trưởng Ban: ThS. Lê Anh Tuấn

2. GVPB: ThS.Lâm Thanh Toản

3. Thư ký: ThS. Đặng Trung Tín

Nôi dung nhận xét:

Cần Thơ, ngày……tháng……năm 2024

Hội đồng đánh giá

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành và tri ân sâu sắc đối với các thầy cô của trường Đại học Kỹ thuật – Công nghệ Cần Thơ, đặc biệt là các thầy cô trong khoa Công nghệ thông tin đã tạo điều kiện để chúng em có thể thực hiện đồ án. Kế đến chúng em cũng xin được gửi lời cảm ơn chân thành nhất đến Ths. Võ Thành Vinh. Trong quá trình học tập nghiên cứu thực hiện Đồ Án với đề tài: "Xây dựng Website mạng xã hội Nemo" chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tận tình, tâm huyết từ thầy. Thầy đã giúp chúng em tích lũy thêm nhiều kiến thức để có thể hoàn thành một cách tốt nhất bài báo cáo này.

Tuy nhiên do giới hạn kiến thức chuyên ngành còn hạn chế và khả năng lý luận của bản thân còn nhiều thiết sót nên khó tránh khỏi những sai sót khi nghiên cứu và thực hiện đề tài kính mong sự chỉ dẫn và đóng góp của các Thầy, Cô để bài báo cáo đồ án của chúng em được hoàn thiện hơn.

Những ý kiến đóng góp của thầy cô sẽ giúp chúng em nhận ra những hạn chế và qua đó chúng em sẽ có thêm những nguồn tư liệu mới trên con đường học tập cũng như nghiên cứu sau này.

Em xin chân thành cảm ơn!

**LỜI CAM ĐOAN**

Nhóm em xin cam đoan bài báo cáo Đồ Án về đề tài “Xây dựng Website mạng xã hội Nemo ” là công trình nghiên cứu của nhóm em. Mọi sự giúp đỡ cho việc thực hiện bài báo cáo này đã được cảm ơn, các thông tin trích dẫn đã được chỉ dẫn nguồn gốc rõ ràng và được phép công bố. Nhóm em xin hoàn toàn chịu trách nhiệm về sự cam đoan này.

**TÓM TẮT ĐỒ ÁN**

Đề tài xây dựng Website Mạng xã hội Nemo dựa trên việc nghiên cứu công nghệ website đã thàng công trên thị trường, học tập, cải thiện và phát huy các chức năng đã được các website đấy hoàn thiện trước đó. Từ đó thực hiện đồ án và sáng tạo theo ý tưởng hình thành

Website gồm các chức năng chính :

* Đăng nhập, Đăng ký
* Trang cá nhân
* Đăng nội dung, quản lý nội dung đăng
* Tương tác nội dung đăng, bạn bè

Xây dựng một số chức năng phục vụ cho việc quản lý website như: thêm, sửa, xóa, tìm kiếm.

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc8816)

[LỜI CAM ĐOAN 5](#_Toc31387)

[TÓM TẮT ĐỒ ÁN 6](#_Toc27959)

[DANH MỤC TỪ VIẾT TẮT 9](#_Toc32573)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 13](#_Toc14863)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 15](#_Toc27543)

[2.1. Các công nghệ sử dụng 15](#_Toc21233)

[2.1.1. MongoDB 15](#_Toc834)

[2.1.2. MongoDB Compass 17](#_Toc28837)

[2.1.3. Visual Studio Code 18](#_Toc22823)

[2.1.4. NodeJS 20](#_Toc19089)

[2.1.5. NestJS 21](#_Toc32432)

[2.1.6. React 23](#_Toc28072)

[2.1.7. Cloudinary 24](#_Toc20850)

[2.1.8. Postman 25](#_Toc23536)

[2.1.9. WebSocket 26](#_Toc787)

[2.1.10. Netlify 27](#_Toc16712)

[2.2. Giới thiệu ngôn ngữ lập trình 28](#_Toc5500)

[2.2.1. Ngôn ngữ JavaScript 28](#_Toc12116)

[2.2.2. Ngôn ngữ TypeScript 30](#_Toc5537)

[CHƯƠNG 3. NỘI DUNG THỰC HIỆN 33](#_Toc8624)

[3.1. Chức năng và phi chức năng 33](#_Toc29013)

[3.1.1. Yêu cầu phi chức năng 33](#_Toc21724)

[3.1.2. Danh sách các chức năng 34](#_Toc26897)

[3.2. Phân tích chức năng yêu cầu 36](#_Toc3464)

[3.2.1. Đăng ký 36](#_Toc25745)

[3.2.2. Đăng xuất 36](#_Toc6942)

[3.2.3. Đăng nhập 37](#_Toc27056)

[3.2.4. Đổi mật khẩu 37](#_Toc30654)

[3.2.5. Xem NewFeed 38](#_Toc12023)

[3.2.6. Tìm kiếm 38](#_Toc14606)

[3.2.7. Xem thông tin người dùng 38](#_Toc20401)

[3.2.8. Sửa thông tin người dùng 39](#_Toc24710)

[3.2.9. Xem bài đăng cá nhân 39](#_Toc5502)

[3.2.10. Thêm bài đăng 39](#_Toc20258)

[3.2.11. Tương tác bài đăng 40](#_Toc11204)

[3.2.12. Lưu bài viết 40](#_Toc32671)

[3.2.13. Xem danh sách lưu bài 40](#_Toc29557)

[3.2.14. Xoá bài viết đã lưu 41](#_Toc947)

[3.2.15. Xem bình luận 41](#_Toc8683)

[3.2.16. Thêm bình luận 41](#_Toc25560)

[3.2.17. Xem danh sách bạn bè 42](#_Toc27649)

[3.2.18. Xóa bạn bè 42](#_Toc16039)

[3.2.19. Thêm yêu cầu bạn bè 42](#_Toc18057)

[3.2.20. Huỷ yêu cầu bạn bè 43](#_Toc11220)

[3.2.21. Phản hồi bình luận 43](#_Toc20428)

[3.2.22. Phản hồi yêu cầu kết bạn 43](#_Toc18509)

[3.2.23. Xem danh sách yêu cầu kết bạn 44](#_Toc20924)

[3.2.24. Tạo tin nhắn 44](#_Toc15370)

[3.2.25. Xoá tin nhắn 44](#_Toc2812)

[3.2.26. Xem danh sách hình ảnh đã nhắn 45](#_Toc20360)

[3.2.27. Tạo nhóm nhắn tin 45](#_Toc7405)

[3.2.28. Sửa nhóm nhắn tin 45](#_Toc5647)

[3.2.29. Xóa nhóm nhắn tin 46](#_Toc12130)

[3.2.30. Thêm thành viên vào nhóm nhắn tin 46](#_Toc6960)

[3.2.31. Xóa thành viên khỏi nhóm nhắn tin 46](#_Toc7846)

[3.3. Phân tích thiết kế giao diện người dùng 47](#_Toc13592)

[3.3.1. Phân tích thiết kế hệ thống 47](#_Toc28404)

[3.3.2. Thiết kế giao diện 77](#_Toc27317)

[3.3.3. Bảng tiến độ công việc 82](#_Toc22691)

[CHƯƠNG 4. KẾT QUẢ THỰC HIỆN 83](#_Toc5051)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 84](#_Toc6782)

[5.1. Kết quả đạt được 84](#_Toc18949)

[5.2. Hạn chế 84](#_Toc27691)

[5.3. Hướng phát triển 84](#_Toc16117)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Nguyên mẫu** | **Ý nghĩa** |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| JSON | JavaScript Object Notation | Một kiểu định dạng dữ liệu |
| BSON | Binary JavaScript Object Notation | Một kiểu định dạng dữ liệu |
| RAM | Random Access Memory | Một loại phần cứng liên quan đến bộ nhớ |
| SQL | Structured Query Language | Ngôn ngữ lập trình |
| GUI | Graphical User Interface | Giao điện đồ hoạ người dùng |
| CURD | Create Read Update Delete |  |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| OOP | Object Oriented Programming | Một mô hình lập trình |
| FP | Functional Programming | Một mô hình lập trình |
| FRP | Functional Reactive Programming | Một mô hình lập trình |
| UI | User Interface | Giao điện người dùng |
| XML | eXtensible Markup Language | Ngôn ngữ đánh dấu mở rộng |
| HTML | HyperText Markup Language | Ngôn ngữ lập trình |
| DOM | Document Object Model | Mô hình đối tượng tài liệu |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| URL | Uniform Resource Locator | Hệ thống định vị tài nguyên thống nhất |
| CSS | Cascading Style Sheets | Ngôn ngữ lập trình |
| JS | JavaScript | Ngôn ngữ lập trình |
| CSDL | Cơ sở dữ liệu |  |
| GB | Gigabyte | Đơn vị đo lường của dung lượng lưu trữ máy tính |
| GHz | Gigahertz | Đơn vị tần số do sự dao động |
| IDE | Integrated Development Environment | Phần mềm cung cấp môi trường tích hợp |
| TCP | Transmission Control Protocol | Một giao thức truyền dữ liệu |
| HTTP | Hyper Text Transfer Protocol | Một giao thức truyền dữ liệu |
| JDBC | Java Database Connectivity |  |
| CI | Continuous Integration | Tích hợp liên tục |
| CD | Continuous Delivery | Phân phối liên tục |
| DNS | Domain Name System | Hệ thống phân giải tên miền |
| CDN | Content Delivery Network | Một mạng lưới máy chủ phân tán |
| REST | Representational State Transfer | Một dạng chuyển đổi cấu trúc dữ liệu |

**DANH MỤC HÌNH ẢNH**

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

* 1. **Hiện trạng**

Hiện nay, mạng xã hội đã trở thành một phần thiết yếu trông cuộc sống hiện đại ngày nay, góp phần quan trọng trong công cuộc phát triển kinh tế và giải trí đa phương tiện. Các nên tảng như Facebook, X, Thread, Zalo … đã, đang, và sẽ phát triển rất mạnh trên thị trường.

* 1. **Mục tiêu đề tài**

Với đề tài" Xây dựng Website Mạng xã hội Nemo", Mục tiêu là học tập các thành công đang có trên thị trường, cải biến theo ý tưởng thiết kế và phát triển các điểm bất cập của các thành công đó thành điểm riêng của đồ án

* 1. **Phạm vi đề tài**
* Thiết kế xây đựng các chức năng cơ bản: đăng nhập, đăng ký, tạo bài biết, tương tác xã hội,…

- Website hoạt động trên giao diện của thiết bị Desktop, có thể hoạt động trên nền web mobile

- Sử dụng các công nghệ phổ biến cho Website hiện nay như : JavaScript, TypeScript (React), NestJS (NodeJS) và cơ sở dữ liệu MongoDB

- Không đi sâu vào các tính năng phức tạp như livestream, thương mại điện tử hay AI

* 1. **Phương pháp nghiên cứu**
* Tìm hiểu, nghiên cứu về hệ thống quản lý và các phần mềm liên quan
* Tham khảo một số trang website có sẵn trên mạng
  1. **Bố cục**

Đồ án gồm 5 chương:

*Chương 1***.** Tổng quan về đề tài: Trình bày hiện trạng, mục tiêu phạm vi và phương pháp nghiên cứu của đề tài.

*Chương 2***.** Cơ sở lý thuyết: Giới thiệu đôi nét những công cụ thực hiện và các ngôn ngữ được sử dụng.

*Chương 3***.** Nội dung thực hiện: Trình bày nội dụng phân tích đặc tả yêu cầu người dùng, thiết kế giao diện, phân tích thiết kế hệ thống qua hai sơ đồ phân rã chức năng và sơ đồ luồng dữ liệu

*Chương 4***.** Kết quả thực hiện: Trình bày giao diện chức năng đã thực hiện được và mô tả chức năng đó.

*Chương 5.* Kết luận và hướng phát triển

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Các công nghệ sử dụng

* + 1. **MongoDB**

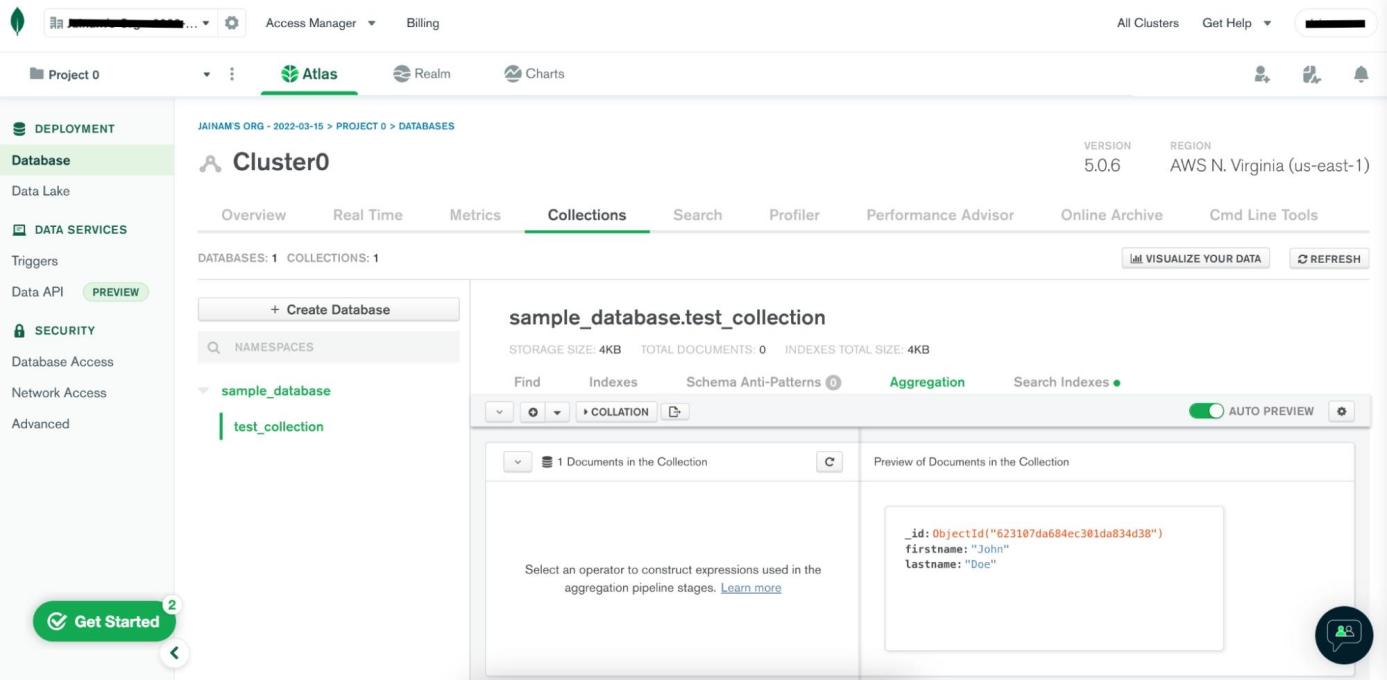
MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON.

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.

* **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.



**Hình 1 . MongoDB logo**



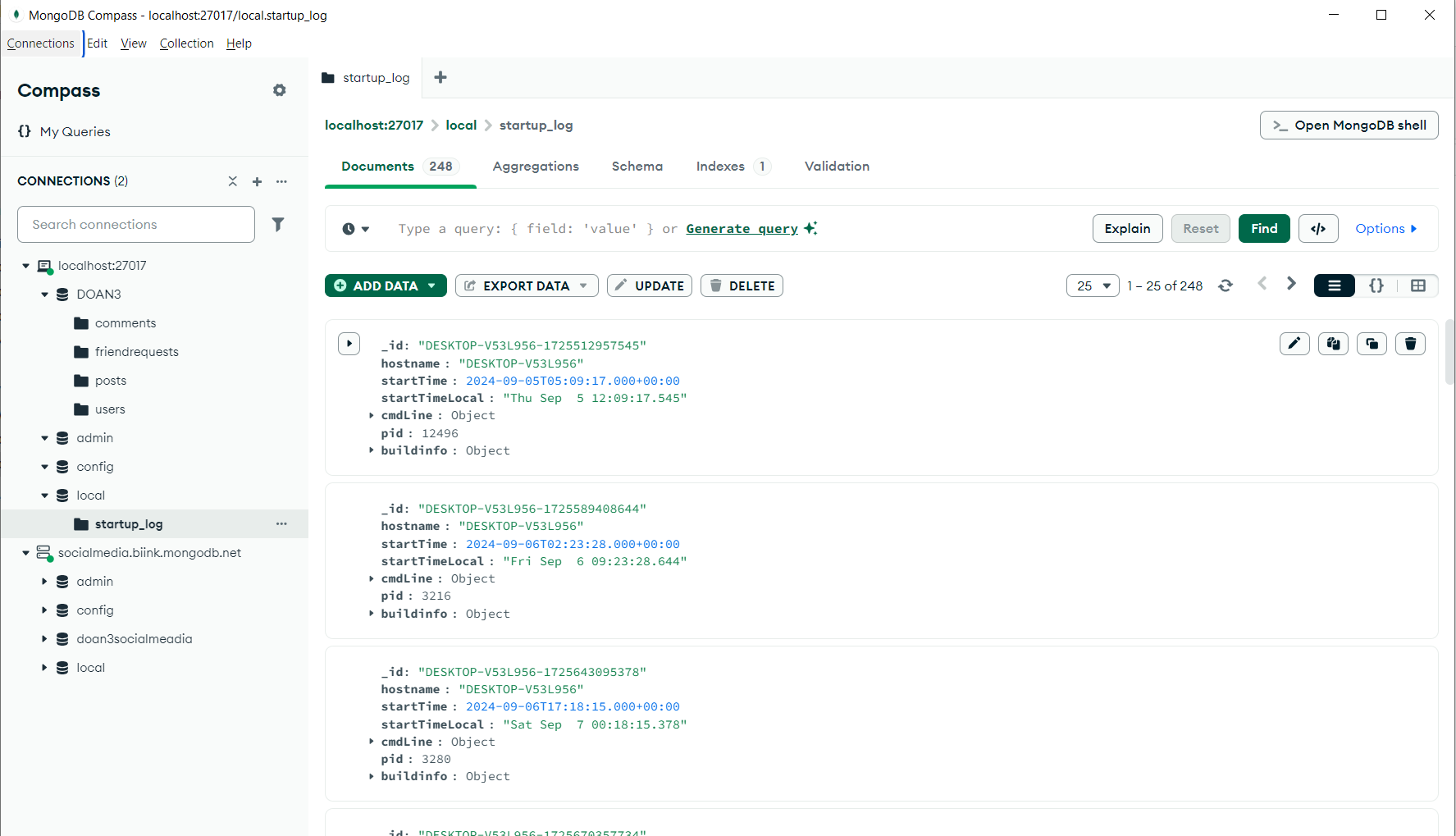
**Hình 2. MongoDB giao diện**

* + 1. **MongoDB Compass**

MongoDB Compass là một GUI cho MongoDB. Còn được gọi là MongoDB GUI. MongoDB cho phép người dùng phân tích nội dung dữ liệu được lưu trữ của họ mà không cần biết trước về cú pháp truy vấn MongoDB. Khi tìm hiểu khám phá dữ liệu trong môi trường trực quan có thể sử dụng Compass GUI để tối ưu hoá hiệu suất, quản lý chỉ mục và triển khai xác thực Document.

***Đặc điểm của MongoDB Compass***

* Người dùng dễ dàng khám phát toàn bộ dữ liệu trực quan.
* Cho phép xem hiệu suất truy vấn.
* Cung cấp nhanh thông tin chi tiết về hiệu năng truy vấn và tình trạng server.
* Không cần phải viết dòng lệnh.
* Tăng hiệu quả tiếp cận và tương tác tốt hơn với CRUD.
* Giúp người dùng xem xét và đưa ra quyết định index, xác thực tài liệu…



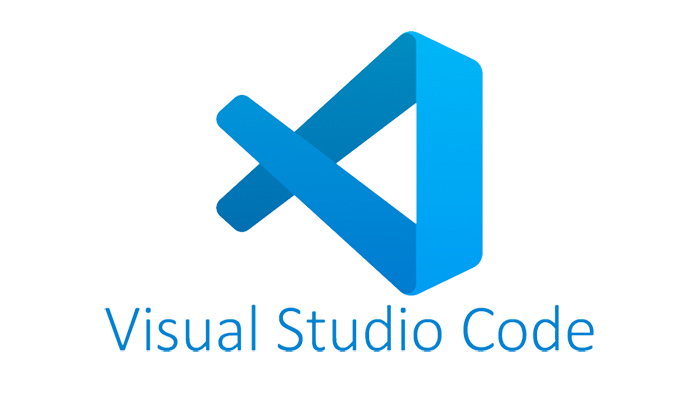
**Hình 3. MongoDB Compass giao diện**

* + 1. **Visual Studio Code**

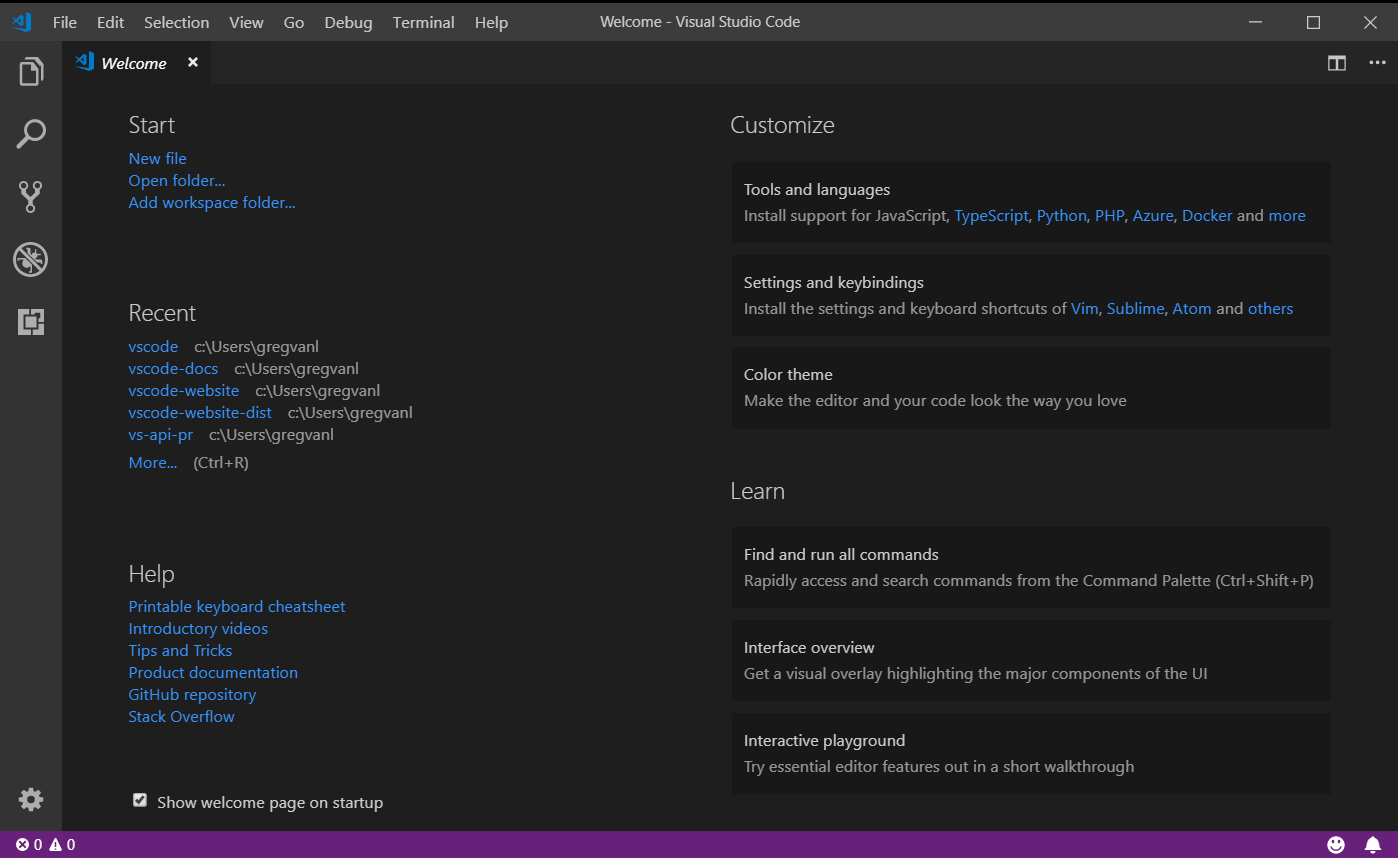
Visual Studio Code (VSCode) là một trình soạn thảo mã nguồn do Microsoft phát triển có thể chạy trên Windows, macOS và Linux. VSCode hoàn toàn miễn phí, mã nguồn mở và cung cấp hỗ trợ gỡ lỗi cũng như kiểm soát phiên bản Git tích hợp, đánh dấu cú pháp, đoạn mã,… Giao diện người dùng của VSCode có khả năng tùy biến cao, vì người dùng có thể chuyển sang các chủ đề, phím tắt và tùy chọn khác nhau.  
 VSCode ban đầu được công bố vào năm 2015 dưới dạng một dự án mã nguồn mở được lưu trữ trên GitHub trước khi phát hành lên web một năm sau đó. Kể từ đó,

trình soạn thảo mã của Microsoft đã trở nên phổ biến nhanh chóng.

* ***Ưu điểm:***
* Phát triển rất tích cực với sự hỗ trợ của Microsoft. Các tài liệu chính thức được duy trì tốt.
* Có sự hỗ trợ rất tích cực của cộng đồng với tất cả các plugin cần thiết. Nếu gửi một lỗi trên repo GitHub thường sẽ nhận được phản hồi trong vòng 4 ngày.
* Đối với một ứng dụng điện tử, tốc độ của VSCode rất cao, gần như tương đương với ST3 được xây dựng nguyên bản.
* ***Nhược điểm:***
* Tương tự như tất cả các ứng dụng điện tử khác, bộ nhớ và mức sử dụng pin của VSCode khá tệ.
* Không có git merge, điều mà nhiều người mong đợi, vì ST3 và Atom có khả năng làm điều đó.
* Các phím tắt mặc định không có ý nghĩa và người dùng phải cấu hình lại gần như tất cả chúng.



**Hình 4. VSCode logo**



**Hình 5. VSCode giao diện**

* + 1. **NodeJS**



**Hình 6. NodeJS logo**

Nodejs là một môi trường runtime chạy JavaScript đa nền tảng, open-source (mã nguồn mở), được sử dụng rộng rãi bởi lập trình viên cho nhiều dự án. Nodejs cung cấp nhiều thư viện khác nhau, giúp đơn giản hóa việc lập trình.

* ***Ưu điểm:***
* Hiệu suất cao: Nodejs chạy đơn luồng, sử dụng V8 Engine, giúp ứng dụng đảm bảo tốc độ khi có nhiều requests.
* Xử lý bất đồng bộ và I/O hướng sự kiện: Khả năng xử lý I/O bất đồng bộ, giúp Nodejs có thể xử lý nhiều tasks, mà không cần phải chờ kết quả của task trước đó.
* Phát triển ứng dụng: Có thể sử dụng để phát triển ứng dụng ở cả phía client và server.
* Module đa dạng: Nodejs sở hữu một cộng đồng duy trì, phát triển modules, thư viện giúp cho việc phát triển ứng dụng nhanh chóng.
* Stream và xử lý file lớn: Nodejs hỗ trợ streaming, cho phép xử lý các file có kích thước lớn không tốn nhiều tài nguyên.
* Phù hợp với ứng dụng real time: Do Nodejs xử lý bất đồng bộ, thích hợp với các ứng dụng real time như: chat applications, streaming services,..
* .***Nhược điểm:***
* Cần có kiến thức nền tảng về JavaScript.
* Khá phức tạp trong việc thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác, dễ dẫn đến tình trạng "callback hell".
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU core.
  + 1. **NestJS**

NestJS là một NodeJS framework dùng để phát triển server-side applications hiệu quả và có thể mở rộng. NestJS là sự kết hợp bởi OOP(Object Oriented Programming), FP(Functional Programming), FRP(Functional Reactive Programming).

NestJS sử dụng TypeScript để phát triển nhưng cũng hỗ trợ cả Javascript.

NestJS được lấy cảm hứng từ kiến trúc Agular (một Javascript framework do Google phát triển)

* ***Cấu trúc của NestJS:***
* Module (Các module): Module là một phần cơ bản trong cấu trúc NestJS. Mỗi ứng dụng NestJS bao gồm ít nhất một module gốc (root module) và có thể có nhiều module con. Module là nơi tổ chức các thành phần của ứng dụng như Controllers, Providers và các thành phần khác. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng.
* Controller (Bộ điều khiển): Controllers là thành phần chịu trách nhiệm xử lý các yêu cầu HTTP từ phía client và trả về kết quả tương ứng. Controllers là nơi xử lý các request và trả về các response. Các phương thức của controller được chú thích (decorated) bằng các decorator như `@Get()`, `@Post()`, `@Put()`, v.v., để chỉ định các route và phương thức HTTP tương ứng.
* Provider (Các nhà cung cấp): Providers là thành phần chịu trách nhiệm cung cấp các dịch vụ cho ứng dụng.
* Middleware (Trung gian): Middleware là các hàm xử lý mà NestJS sử dụng để xử lý các yêu cầu HTTP trước khi chúng đến các route xử lý chính. Middleware có thể được sử dụng để thực hiện các thao tác chung như xác thực, ghi log, xử lý lỗi, v.v.
* Filter (Bộ lọc): Filters được sử dụng để xử lý các exception (ngoại lệ) xảy ra trong ứng dụng. Filters cho phép xử lý và thay đổi response trước khi gửi về client khi có exception xảy ra.
* Guard (Bảo vệ): Guards được sử dụng để kiểm tra xem một yêu cầu có thể được xử lý hoặc không. Guards cho phép thực hiện các kiểm tra xác thực hoặc kiểm tra quyền trước khi xử lý một yêu cầu.
* Interceptor (Bộ chặn): Interceptors là các hàm xử lý mà NestJS sử dụng để chặn và thay đổi response trước khi được gửi về client. Interceptors có thể được sử dụng để thực hiện các thao tác chung trên response trước khi đi ra ngoài.
* Exception (Ngoại lệ): Exception handling (xử lý ngoại lệ) là một phần quan trọng của cấu trúc NestJS. Exception handling cho phép xử lý các exception xảy ra trong ứng dụng và trả về các thông báo lỗi thích hợp cho client.



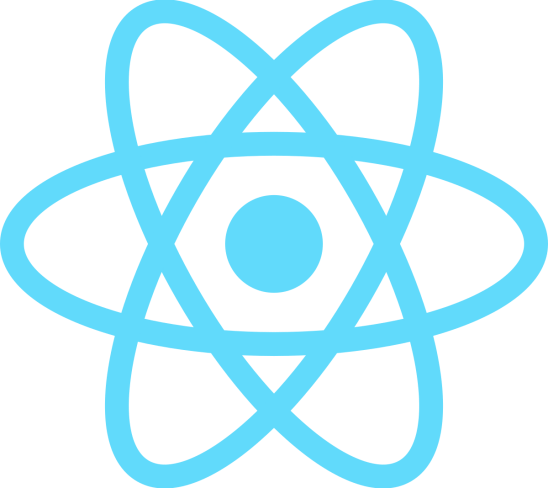
**Hình 7. NestJS logo**

* + 1. **React**

React là một thư viện JavaScript xây dựng giao diện người dùng

React giúp tạo các UI tương tác một cách dễ dàng. Thiết kế các khung nhìn đơn giản cho từng trạng thái trong ứng dụng, và React sẽ cập nhật và render đúng các thành phần phù hợp khi dữ liệu thay đổi.

* ***Các thành phần quan trọng trong React :***
* JSX (JavaScript XML) là một cú pháp mở rộng cho phép viết mã giống như HTML trong JavaScript. Trong các ngôn ngữ khác thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép kết hợp cả hai trong cùng một mã nguồn, giúp quản lý dễ dàng hơn, đặc biệt là trong các ứng dụng phức tạp.
* Virtual DOM (Document Object Model ảo) là một bản sao nhẹ hơn của DOM thật. DOM thật là cấu trúc cây chứa tất cả các thành phần HTML trong trang web. Khi người dùng tương tác với ứng dụng (ví dụ: nhập văn bản, nhấn nút), ứng dụng sẽ thay đổi nội dung và DOM thật phải được cập nhật.
* Component là các đơn vị cơ bản trong React, cung cấp cấu trúc cho giao diện người dùng. Mỗi component được khuyến khích càng nhỏ gọn và độc lập càng tốt để có thể tái sử dụng trong nhiều phần của ứng dụng hoặc thậm chí trong các ứng dụng khác.



**Hình 8. React logo**

* + 1. **Cloudinary**

Cloudinary là một cloud-based service, cung cấp một giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, tối ưu hóa và delivery.

Cloudinary khiến việc upload ảnh lên cloud trở nên dễ dàng, tự động thực thi các thao tác với ảnh một cách thông minh mà không cần phải cài đặt bất kì một phần mềm phức tạp nào khác. Cloudinary cung cấp các APIs toàn diện và màn hình quản lý giúp dễ dàng tích hợp vào các trang web và ứng dụng di động.

* ***Những tính năng chính :***

Cloudinary cung cấp một Ruby Gem để dễ dàng cho việc tương tác với các app viết bởi ngôn ngữ Ruby với một số framwork như là Rails hay Sinatra với các tương tác chính sau:

* Xây dựng các URL để chuyển đổi và thao tác với hình ảnh
* Rails view helper cho việc nhúng và thay đổi hình ảnh
* API wrappers: upload image, quản lý và nhiều thứ khác
* Upload image trực tiếp từ trình duyệt sử dụng một jQuery plugin.
* Tích hợp với Active Record
* CarrierWave plugin
* Hình ảnh tĩnh đồng bộ với CDN delivery
* Migration tool



**Hình 9. Cloudinary logo**

* + 1. **Postman**

Postman là công cụ được sử dụng trong thử nghiệm các API.

Thông thường, Postman sẽ được dùng cho API kiểu REST. Postman có thể gọi Rest API mà không cần viết dòng code nào.

* ***Các tính năng chính của Postman :***
* Postman’s Visualizer cho phép tăng cường việc trình bày dữ liệu phản hồi API bằng cách sử dụng HTML, CSS và JavaScript. Có thể tạo các hình ảnh tuỳ chỉnh như biểu đồ, bảng hoặc các phần tử khác để làm cho dữ liệu phản hồi dễ đọc và hiểu hơn.
* Postman cung cấp nhiều thư viện tích hợp sẵn bên ngoài mà có thể sử dụng trong các tab Pre-request Script và Test Script để thêm các chức năng bổ sung mà JavaScript không có sẵn.
* Postman cho phép kiểm soát thứ tự thực hiện các yêu cầu trong một collection bằng phương thức postman.setNextRequest. Bằng cách sử dụng logic điều kiện trong tab Pre-request Script hoặc Tests, người dùng có thể thiết lập yêu cầu tiếp theo được thực hiện dựa trên các điều kiện cụ thể.
* Postman cung cấp hỗ trợ mạnh mẽ cho GraphQL do tích hợp sẵn, giúp các nhà phát triển làm việc với các API GraphQL dễ dàng hơn.
* Postman cung cấp tính năng “Collections” để giúp tổ chức các yêu cầu API. Với collections, người dùng có thể nhóm các yêu cầu liên quan lại với nhau, giúp quản lý và thực hiện nhiều yêu cầu như một đơn vị đồng nhất.
* Postman bao gồm một khung kiểm thử mạnh mẽ, cho phép kiểm thử và xác nhận bằng các bài viết kiểm tra bằng JavaScript để xác thực các khía cạnh khác nhau của phản hồi API.

**Hình 10. Postman logo**

* + 1. **WebSocket**

WebSocket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

WebSockets mới xuất hiện trong HTML5, là một kỹ thuật Reverse Ajax. WebSockets cho phép các kênh giao tiếp song song hai chiều và hiện đã được hỗ trợ trong nhiều trình duyệt (Firefox, Google Chrome và Safari). Kết nối được mở thông qua một HTTP request (yêu cầu HTTP), được gọi là liên kết WebSockets với những header đặc biệt. Kết nối được duy trì để có thể viết và nhận dữ liệu bằng JavaScript như khi đang sử dụng một TCP socket đơn thuần.

* **Ưu điểm**
* WebSockets cung cấp khả năng giao tiếp hai chiều mạnh mẽ, có độ trễ thấp và dễ xử lý lỗi. Không cần phải có nhiều kết nối như phương pháp Comet long-polling và cũng không có những nhược điểm như Comet streaming.
* API cũng rất dễ sử dụng trực tiếp mà không cần bất kỳ các tầng bổ sung nào, so với Comet, thường đòi hỏi một thư viện tốt để xử lý kết nối lại, thời gian chờ timeout, các Ajax request (yêu cầu Ajax), các tin báo nhận và các dạng truyền tải tùy chọn khác nhau (Ajax long-polling và jsonp polling).
* **Nhược điểm**
* Do là một đặc tả mới của HTML5, nên vẫn chưa được tất cả các trình duyệt hỗ trợ.
* Không có phạm vi yêu cầu nào. Do WebSocket là một TCP socket chứ không phải là HTTP request, nên không dễ sử dụng các dịch vụ có phạm vi-yêu cầu, như SessionInViewFilter của Hibernate. Hibernate là một framework kinh điển cung cấp một bộ lọc xung quanh một HTTP request. Khi bắt đầu một request sẽ thiết lập một contest (chứa các transaction và liên kết JDBC) được ràng buộc với luồng request. Khi request đó kết thúc, bộ lọc hủy bỏ contest này.
  + 1. **Netlify**

Netlify là một nền tảng đám mây tích hợp đầy đủ các công cụ cho việc phát triển, triển khai và quản lý trang web. Netlify cung cấp một loạt các tính năng như hosting, tích hợp CI/CD (Continuous Integration/Continuous Deployment), địa chỉ DNS tùy chỉnh và nhiều tính năng khác để giúp nhà phát triển tập trung vào việc xây dựng ứng dụng mà không cần lo lắng về việc triển khai.

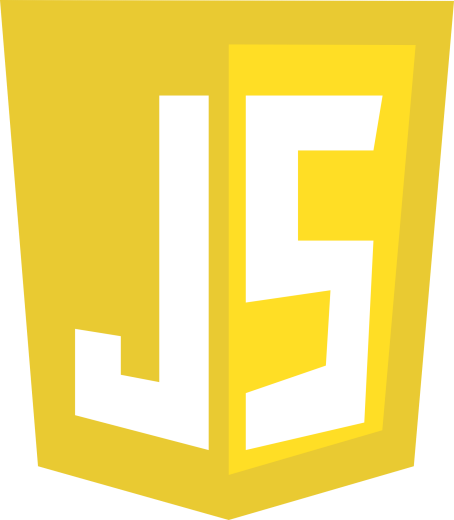
* **Tính năng của Netlify**
* Hosting tốc độ cao: Netlify cung cấp dịch vụ hosting với mạng lưu trữ toàn cầu (CDN), giúp tăng tốc độ tải trang và cung cấp trải nghiệm người dùng tốt hơn.
* Tích hợp CI/CD: Netlify tích hợp quy trình CI/CD vào quy trình làm việc, giúp tự động hóa quá trình kiểm thử và triển khai mỗi khi có sự thay đổi trong mã nguồn.
* Tích hợp dễ dàng với GitHub và GitLab: Netlify kết nối trực tiếp với kho lưu trữ GitHub và GitLab, cho phép bạn kích hoạt triển khai tự động mỗi khi có sự thay đổi trong mã nguồn.
* Form và Functions: Netlify cung cấp các tính năng như xử lý form và functions, giúp bạn xây dựng ứng dụng động và tương tác mạnh mẽ.
* Địa chỉ DNS tùy chỉnh: Người dùng có thể dễ dàng quản lý và cấu hình tên miền trực tiếp từ bảng điều khiển Netlify.



**Hình 11. Netlify logo**

* 1. **Giới thiệu ngôn ngữ lập trình**

### Ngôn ngữ JavaScript

**Hình 12. JavaScript logo**

**JavaScript**là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua.

JavaScript có thể [học nhanh](https://www.bitdegree.org/learn/javascript-basics/" \t "_blank) và dễ dàng áp dụng cho nhiều mục đích khác nhau, từ việc cải thiện tính năng của website đến việc chạy game và tạo phần mềm nền web. Hơn nữa, có hàng ngàn mẫu template JavaScript và ứng dụng ngoài kia, nhờ vào sự cống hiến của cộng đồng, đặc biệt là Github.

JavaScript là ngôn ngữ lập trình mang đến sự sinh động của website. Khác với HTML (thường chuyên cho nội dung) và CSS (thường chuyên dùng cho phong cách), và khác hẳn với PHP (chạy trên server chứ không chạy dưới máy client).

Là ngôn ngữ dễ học, được phát triển bởi Netscape, và đang được dùng trên 92% webstie. JS có thể được gắn vào một element của trang web hoặc sự kiện của trang web như cú click chuột. Hoạt động trên đa trình duyệt và đa thiết bị. Nhanh và nhẹ hơn các ngôn ngữ lập trình khác. Có thể ít an toàn hơn vì độ phổ biến của JS. Người dùng có thể thêm JavaScript trực tiếp vào HTML hoặc có thể lưu trên files riêng biệt và gọi lên khi cần.

* ***Lịch sử phát triển***

JavaScript được tạo trong mười ngày bởi Brandan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, sau đó được đổi thành Mona rồi LiveScript trước khi thật sự trở thành JavaScript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc quyền bởi Netscape và chỉ có các tính năng hạn chế, nhưng JS tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với JS.

Trong năm 1996, JavaScript được chính thức đặt tên là ECMAScript. ECMAScript 2 phát hành năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999, liên tục phát triển thành JavaScript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính bàn.

JavaScript liên tục phát triển kể từ đó, có lục đạt đến 92% website đang sử dụng JavaScript vào năm 2016. Chỉ trong 20 năm, từ một ngôn ngữ lập trình riêng trở thành công cụ quan trọng nhất trên bộ công cụ của các chuyên viên lập trình web.

* ***Ưu điểm:***

JavaScript có rất nhiều ưu điểm khiến JS vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

* Không cần một compiler vì web browser có thể biên dịch bằng HTML
* Dễ học hơn, nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.
* Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn
* Có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới
* JS hoạt động trên nhiều trình duyệt, nền tảng
* Có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database
* Giúp website tương tác tốt hơn với khách truy cập
* ***Nhược điểm:***

Mọi ngôn ngữ lập trình đều có các khuyết điểm Một phần là vì ngôn ngữ đó khi phát triển đến một mức độ như JavaScript cũng sẽ thu hút lượng lớn hacker, scammer tận dụng những lỗ hổng và các lỗi bảo mật để lợi dụng. Một số khuyết điểm có thể kể đến là:

* Dễ bị khai thác
* Có thể được dùng để thực thi mã độc trên máy tính của người dùng
* Nhiều khi không được hỗ trợ trên mọi trình duyệt
* JS code snippets lớn
* Có thể bị triển khai khác nhau tuỳ từng thiết bị dẫn đến việc không đồng nhất.

### Ngôn ngữ TypeScript



**Hình 13. TypeScript logo**

TypeScript (TS) là một ngôn ngữ lập trình mở rộng từ JavaScript, bổ sung hệ thống kiểu tĩnh và hỗ trợ các tính năng như interfaces và generics. Typescript giúp phát hiện lỗi ngay ở giai đoạn biên dịch (compile-time) thay vì đợi đến lúc chạy chương trình (runtime), từ đó tăng độ an toàn và giảm thiểu lỗi cho lập trình viên.

* ***Lịch sử phát triển***

TypeScript được công bố lần đầu tiên vào tháng 10 năm 2012 (ở phiên bản 0.8), sau hai năm phát triển nội bộ tại Microsoft. Ngay sau khi thông báo, TypeScript bị chỉ trích về việc thiếu hỗ trợ IDE trưởng thành ngoài Microsoft Visual Studio, vốn chưa có trên Linux và OS X vào thời điểm đó. Ngày nay ,TypeScript đã có hỗ trợ trong các IDE khác, đặc biệt là trong Eclipse, thông qua một trình cắm thêm. Các trình soạn thảo văn bản khác nhau, bao gồm Emacs, Vim, Webstorm, Atom và Visual Studio Code của riêng Microsoft cũng hỗ trợ TypeScript.

TypeScript 0.9, được phát hành vào năm 2013, đã thêm hỗ trợ cho generic.

TypeScript 1.0 được phát hành tại hội nghị nhà phát triển Build của Microsoft vào năm 2014.Visual Studio 2013 Update 2 cung cấp hỗ trợ tích hợp cho TypeScript.

Tháng 7 năm 2014, nhóm phát triển đã công bố một trình biên dịch TypeScript mới, đạt hiệu suất gấp 5 lần. Đồng thời, mã nguồn, ban đầu được lưu trữ trên CodePlex, đã được chuyển sang GitHub.

Ngày 22 tháng 9 năm 2016, TypeScript 2.0 đã được phát hành, giới thiệu một số tính năng, bao gồm khả năng cho các lập trình viên tuỳ ý ngăn các biến được gán giá trị null

TypeScript 3.0 được phát hành vào ngày 30 tháng 7 năm 2018 mang đến nhiều bổ sung ngôn ngữ như tuple (bộ giá trị) trong rest parameter (tham số phần còn lại) và spread expression (biểu thức trải rộng), rest parameter với các kiểu tuple, generic rest parameter, v.v

TypeScript 4.0 phát hành vào ngày 20 tháng 8 năm 2020. Mặc dù 4.0 không giới thiệu bất kỳ thay đổi đột phá nào, nhưng TS đã bổ sung các tính năng ngôn ngữ như Hệ thống JSX tuỳ chỉnh và các kiểu Variadic Tuple (bộ giá trị đa dạng)

* ***Ưu điểm***

1. **Thuận tiện và hoàn toàn miễn phí:** TypeScript hỗ trợ quá trình tìm kiếm giúp tiết kiệm thời gian kiểm tra lại code, không cần thông qua bất cứ ai để có thể tìm thông tin dữ liệu, ngoài ra TypeScript làm giảm phần trăm va chạm lỗi trong thời gian vận hành. Ngoài ra, TypeScript cũng là một trong những phần mềm nổi trội được Microsoft hỗ trợ hoàn toàn miễn phí
2. **Thao tác nhanh chóng và đơn giản hơn**: TypeScript có thao tác khá đơn giản, tiết kiệm thời gian hơn nhưng lại đem đến kết quả tốt đến bất ngờ, TS khắc phục tình trạng xuất hiện lỗi và dễ đọc hơn.
3. **Khả năng tái cấu trúc tốt** : TypeScript giúp bước chỉnh sửa code trở nên dễ dàng hơn nhờ hiệu quả của lệnh Rename Symbol/Find All Occurrences.
4. **Giảm tỷ lệ mắc lỗi trong hệ thống**: Tỉ lệ mắc lỗi trong hệ thống khi sử dụng TypeScript khá thấp nhờ vào việc cảnh báo lỗi ngay khi viết code.
5. **Hợp nhất mã đơn giản:** TypeScript có thể hợp nhất mã một cách đơn giản để có thể dễ dàng kiểm tra đánh giá đoạn mã vừa mới cho ra đời kia bằng cách sử dụng Typedef (kiểm tra biên dịch).
6. **Hỗ trợ tối ưu hoá quy trình làm việc:** TypeScript sẽ không khuyến khích người dùng nhảy bước, thực hiện sai thao tác. TypeScript khuyến khích người dùng đưa ra quyết định về kiểu dữ liệu khi sử dụng ngôn ngữ kiểu tĩnh trước khi thực hiện thao tác, các bước tiếp theo

* ***Nhược điểm***

1. ****Bắt buộc dùng biên dịch** : Để có thể vận hành một tệp TypeScript với đuôi .js trên nền tảng Node.js bắt buộc phải dùng trình biên dịch để có thể sử dụng.**
2. ****Bước thiết lập cồng kềnh** : Trước khi có thể sử dụng được TypeScript cần đảm bảo rằng máy chủ Node.js, trình thử nghiệm và webpack đều có thể hoạt động với TypeScript, nếu không thì sẽ không sử dụng được. Bên cạnh đó, mỗi khi apply thêm bất kỳ library nào như Redux, React và Styled-Component thì cũng phải thêm Typedef vào.**
3. ****Chỉ là phần ngôn ngữ mở rộng hỗ trợ**: Chức năng của TypeScript cũng chỉ là để biên dịch về JavaScript, không phải là một ngôn ngữ có thể vận hành độc lập và cũng đồng thời không thể thay thế được vai trò của JavaScript. Chức năng của TypeScript bị giới hạn bởi chức năng của JavaScript, TypeScript chỉ là được nâng cấp từ điểm yếu của JavaScript.**

**CHƯƠNG 3. NỘI DUNG THỰC HIỆN**

## Chức năng và phi chức năng

### Yêu cầu phi chức năng

#### Yêu cầu thực thi

Website hoạt động ổn định trên nền web và có thể là web mobile, chính xác đáp ứng được nhu cầu của người sử dụng. Cần hạn chế lỗi phát sinh xuống mức thấp nhất có thể. Khi có lỗi, cần được phát hiện và xử lý nhanh chóng, đảm bảo website hoạt động bình thường và liên tục.

#### Yêu cầu về giao diện

Yêu cầu về trang chủ hiển thị trang chủ tổng quan. Yêu cầu về giao diện đăng nhập, đăng ký đơn giản, dễ nhìn với các trường thông tin. Yêu cầu tìm kiếm và lọc cung cấp ô tìm kiếm để nhanh chóng. Thông tin người dùng hiển thị rõ ràng, tính năng liên quan đến bài đăng phải dễ dàng tiếp cận.

#### Yêu cầu về hiệu suất

Thời gian trả về kết quả tìm kiếm dưới 5 giây. Tốc độ xử lý các chức năng liên quan đến bài đăng và người dùng không quá 10 giây

#### Các đặc điểm chất lượng phầm mềm

1. Dễ sửa lỗi: Cập nhật nội dung, kiểm soát các hoạt động của web, theo dõi xác định và xử lý lỗi nhanh, nâng cấp và phát triển web
2. Dễ sử dụng: Phần mềm dễ sử dụng, giao diện thân thiện với người dùng. Nội dung được tối ưu hóa.
3. Tính chính xác: Dữ liệu trong hệ thống và dữ liệu xuất ra cần phải chính xác.
4. Tính thao tác giữa các thành phần: Thao tác giữa các thành phần trong cơ sở dữ liệu cần phải được tương thích với nhau.
5. Tính có thể kiểm thử: Dễ dàng kiểm tra hoạt động của phần mềm bằng các CSDL mẫu để phát hiện lỗi của phần mềm.

#### Các quy tắc nghiệp vụ

Website được vận hành phải tuân theo các nguyên tắc cụ thể do người lập trình đặt.

#### Các yêu cầu khác

Yêu cầu về phần cứng:

* Bộ xử lý 32 bit (x86) hoặc 64 bit (x64) có tốc độ từ 1.5GHz trở lên
* Ram 2GB
* Đĩa cứng có dung lượng trống từ 10GB trở lên
* Hệ điều hành windows 7 trở lên
* Hệ quản trị cơ sở dữ liệu MongoDB
* Website phải có dung lượng không quá lớn

### Danh sách các chức năng

* *Các chức năng chung:*
* Đăng nhập
* Đăng ký
* Đăng xuất
* Xem Newfeed
* Tìm kiếm
* Đổi mật khẩu
* *Các chức năng dành cho người dùng*
* Quản lý thông tin cá nhân :
* Xem thông tin cá nhân
* Sửa thông tin cá nhân
* Quản lý bài đăng:
* Xem bài đăng
* Thêm bài đăng
* Tương tác bài đăng
* Quản lý bài viết đã lưu:
* Xem danh sách lưu
* Lưu bài
* Xoá bài đã lưu
* Quản lý bình luận:
* Xem bình luận trong bài đăng
* Thêm bình luận bài đăng
* Phản hồi bình luận

- Quản lý bạn bè

* Xem danh sách bạn bè
* Xóa bạn bè
* Xem danh sách yêu cầu kết bạn
* Thêm yêu cầu kết bạn
* Huỷ yêu cầu kết bạn
* Phản hồi kết bạn

- Quản lý tin nhắn

* Tạo Tin nhắn
* Xoá Tin nhắn
* Xem danh sách hình ảnh đã nhắn

- Quản lý nhóm nhắn tin

* Tạo nhóm nhắn tin
* Sửa nhóm nhắn tin
* Xoá nhóm nhắn tin
* Thêm thành viên vào nhóm nhắn tin
* Xoá thành viên khỏi nhóm nhắn tin

## Phân tích chức năng yêu cầu

* + 1. **Đăng ký**

Chức năng đăng ký cho phép người đăng ký tài khoản mới, sử dụng cho việc đăng nhập để thực hiện các chức năng liên quan như đăng bài, bình luận,…

* Chức năng dùng để đăng ký tài khoản truy cập vào hệ thống.
* Tác nhân: người dùng
* Mô tả chức năng nhỏ: Thoát: Khi người dùng muốn huỷ việc đăng ký tài khoản.

Yêu cầu chức năng: Người dùng phải điền đầy đủ các thông tin (có thể bỏ trống một số trường không bắt buột) sau đó nhấn Đăng ký để đăng ký tài khoản. Người dùng không được bỏ trống hoặc nhập thiếu thông tin bắt buộc.

* + 1. **Đăng xuất**

Chức năng đăng xuất cho phép người dùng đăng xuất khỏi hệ thống khi người dùng muốn bảo mật tài khoản, hoặc đăng nhập bằng tài khoản khác vào hệ thống.

* Chức năng dùng để đăng xuất khỏi hệ thống
* Tác nhân: Người dùng
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ việc đăng xuất.

Yêu cầu chức năng: Khi muốn đăng xuất khỏi hệ thống thì hiện tại đã đăng nhập vào hệ thống rồi và người dùng phải xác nhận việc đồng ý đăng xuất khỏi hệ thống.

* + 1. **Đăng nhập**

Chức năng đăng nhập cho phép người dùng đăng nhập vào hệ thống.

* Chức năng dùng để đăng nhập vào hệ thống.
* Tác nhân: Người dùng
* Mô tả chức năng nhỏ:

+ Quên mật khẩu: Khi người dùng không nhớ mật khẩu thì chức năng Quên mật khẩu cho phép người dùng có thể đặt lại mật khẩu mới bằng cách xác nhận thông tin.

+ Thoát: Chức năng Thoát được sử dụng khi người dùng muốn huỷ việc đăng nhập và trở về trang chủ.

Yêu cầu chức năng: Khi muốn đăng nhập vào hệ thống thì người dùng phải nhập đủ và chính xác SDT đăng nhập, Mật khẩu sau đó nhấn Đăng nhập để đăng nhập vào hệ thống. Người dùng không được bỏ trống hoặc nhập thiếu thông tin

* + 1. **Đổi mật khẩu**

Chức năng thay đổi mật khẩu cho phép người dùng đổi mật khẩu khi người dùng muốn bảo mật tài khoản và thay đổi mật khẩu định kỳ sau một khoảng thời gian.

* Chức năng dùng để thay đổi mật khẩu cho tài khoản.
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ việc đổi mật khẩu.

Yêu cầu chức năng: người dùng phải nhập đủ 2 trường: Mật khẩu cũ, Mật khẩu mới sau đó nhấn Đổi mật khẩu để tiến hành việc đổi mật khẩu. Mật khẩu cũ phải chính xác. Mật khẩu cũ, Mật khẩu mới, không được bỏ trống. Mật khẩu mới không được trùng với Mật khẩu cũ

* + 1. **Xem NewFeed**

Chức năng xem NewFeed cho phép người dùng xem được những bài đăng của người khác hiện lên ở trang NewFeed ngay sau khi đăng nhập

* Chức năng dùng để hiển thị các bài đăng của người khác hiện ở trang NewFeed
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: không có

Yêu cầu chức năng: Tại giao diện trang NewFeed, người dùng có thể xem các các bài đăng của người khác và hiện được các tương tác như like, bình luận, xem thời gian đăng bài,…

* + 1. **Tìm kiếm**

Chức năng tìm kiếm cho phép người dùng tìm kiếm người dùng khác hoặc bài đăng của người khác

* Chức năng dùng để tìm kiếm theo thông tin nhập
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: không có

Yêu cầu chức năng: Người dùng nhập liệu vào trường tìm kiếm, hiển thị kết quả tìm kiếm ở trang kết quả tìm kiếm. Khi người dùng không được để trống trường tìm kiếm

* + 1. **Xem thông tin người dùng**

Chức năng xem thông tin người dùng cho phép người dùng xem lại thông tin cá nhân sau khi đăng nhập

* Chức năng dùng để xem thông tin cá nhân
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: không có

Yêu cầu chức năng: Người dùng có thể xem lại thông tin của bản thân trong trang cá nhân sau khi đăng nhập

* + 1. **Sửa thông tin người dùng**

Chức năng sửa thông tin người dùng cho phép người dùng sửa lại thông tin cá nhân đã đăng ký trước đó

* Chức năng dùng để sửa thông tin cá nhân
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ việc sửa thông tin

Yêu cầu chức năng: Người dùng có thể sửa lại thông tin của bản thân đã đăng ký trong trang cá nhân sau khi đăng nhập

* + 1. **Xem bài đăng cá nhân**

Chức năng xem bài đăng cá nhân cho cho phép người dùng xem những bài đăng do tài khoản đăng nhập đăng và chia sẻ

* Chức năng dùng để xem bài viết đã đăng
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng có thể xem lại những bài biết tài khoản đăng nhập đã đăng, hiển thị thời gian, số tương tác của bài đăng đó

* + 1. **Thêm bài đăng**

Chức năng thêm bài đăng cá nhân cho cho phép người dùng thêm những bài đăng do tài khoản đăng nhập đăng

* Chức năng dùng để thêm bài đăng
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ việc thêm bài đăng

Yêu cầu chức năng: Người dùng sẽ đăng bài, nhập đầy đủ các trường trong giao diện đăng bài, chọn chế độ đăng bài, có thể đăng bài kèm file định dạng hình ảnh.

* + 1. **Tương tác bài đăng**

Chức năng thêm bài đăng cá nhân cho cho phép người dùng thêm những bài đăng do tài khoản đăng nhập đăng

* Chức năng dùng để thêm bài đăng
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ việc thêm bài đăng

Yêu cầu chức năng: Người dùng sẽ đăng bài, nhập đầy đủ các trường trong giao diện đăng bài, chọn chế độ đăng bài, có thể đăng bài kèm file định dạng hình ảnh.

* + 1. **Lưu bài viết**

Chức năng lưu bài đăng cho cho phép người dùng lưu những bài đăng do tài khoản đăng nhập đăng hoặc bài đăng của người khác

* Chức năng dùng để lưu bài đăng
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng có thể lưu bài đăng của tài khoản đăng hoặc của người khác, sắp xếp theo trình tự lưu.

* + 1. **Xem danh sách lưu bài**

Chức năng xem danh sách bài đăng đã lưu cho cho phép người dùng xem lại những bài đăng đã lưu trước đó

* Chức năng dùng để lưu bài đăng
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng có thể xem lại những bài viết đã lưu trước đó, hiển thị thời gian đăng và tương tác xã hội của bài đăng đó

* + 1. **Xoá bài viết đã lưu**

Chức năng xoá bài đăng đã lưu cho cho phép người dùng xoá bài đăng đã lưu trước đó ra khỏi danh sách đã lưu

* Chức năng dùng để xoá bài đăng đã lưu
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ xoả bài đăng đã lưu

Yêu cầu chức năng: Người dùng có thể loại bỏ bài đăng đã lưu ra khỏi danh sách bài đã lưu

* + 1. **Xem bình luận**

Chức năng xem bình luận cho phép người dùng xem bình luận của một bài đăng bất kì

* Chức năng dùng để xem bình luận
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng xem bình luận của một bài viết bất kì, hiển thị thời gian bình luận, tương tác bình luận và có thể hiển thị bình luận file định dạng hình ảnh

* + 1. **Thêm bình luận**

Chức năng Thêm bình luận cho phép người dùng thêm bình luận của một bài đăng bất kì

* Chức năng dùng để thêm bình luận vào bài viết
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ thêm bình luận

Yêu cầu chức năng: Người dùng thêm bình luận của một bài viết bất kì, sau khi thêm, hiển thị bình luận lên danh sách bình luận và có thể bình luận bằng file định dạng hình ảnh

* + 1. **Xem danh sách bạn bè**

Chức năng xem danh sách bạn bè cho phép người dùng xem danh sách bạn bè của tài khoản đăng nhập.

* Chức năng dùng để xem danh sách bạn bè
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng xem danh sách bạn bè của tài khoản đăng nhập

* + 1. **Xóa bạn bè**

Chức năng xoá bạn bè cho phép người dùng xoá bạn bè ra khỏi dang sách bạn của tài khoản đăng nhập.

* Chức năng dùng để xoá bạn bè
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+Thoát: Khi người dùng muốn huỷ xoá kết bạn

Yêu cầu chức năng: Người dùng xoá bạn bè ra khỏi danh sách, bạn bè sau khi xoá có thể thêm lại thông qua lời mời kết bạn

* + 1. **Thêm yêu cầu bạn bè**

Chức năng thêm yêu cầu bạn bè cho phép người dùng gửi yêu cầu kết bạn cho người dùng khác

* Chức năng dùng để thêm yêu cầu kết bạn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:Không có

Yêu cầu chức năng: Người dùng gửi yêu cầu kết bạn cho người khác

* + 1. **Huỷ yêu cầu bạn bè**

Chức năng huỷ yêu cầu bạn bè cho phép người dùng huỷ yêu cầu kết bạn cho người dùng khác đã gửi từ trước

* Chức năng dùng để huỷ yêu cầu kết bạn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng huỷ gửi yêu cầu kết bạn cho người khác đã gửi từ trước

* + 1. **Phản hồi bình luận**

Chức năng phản hồi bình luận cho phép người dùng bình luận thêm con cho một bình luận đã được thêm từ trước của tài khoản đăng nhập hoặc của người khác

* Chức năng dùng để phản hồi bình luận
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ phản hồi bình luận

Yêu cầu chức năng: Người dùng có thể bình luận phản hồi một bình luận đã tồn tại trước đó, bình luận con đó có tất cả những yếu tố của một bình luận thông thường

* + 1. **Phản hồi yêu cầu kết bạn**

Chức năng phản hồi yêu cầu kết bạn cho phép người dùng phản hồi yêu cầu kết bạn của người khác gửi đến tài khoản đăng nhập

* Chức năng dùng để phản hồi yêu cầu kết bạn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát: Khi người dùng muốn huỷ phản hồi yêu cầu

Yêu cầu chức năng: Người dùng có 2 phương thức phản hồi: Chấp nhận và Từ chối. Khi Chấp nhận, loại bỏ yêu cầu và thêm người gửi yêu cầu vào danh sách bạn bè. Khi Từ Chối, loại bỏ yêu cầu ra khỏi danh sách yêu cầu kết bạn

* + 1. **Xem danh sách yêu cầu kết bạn**

Chức năng xem danh sách yêu cầu kết bạn cho phép người dùng xem danh sách các yêu cầu kết bạn từ người khác gửi đến tài khoản đăng nhập

* Chức năng dùng để xem danh sách yêu cầu kết bạn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:Không có

Yêu cầu chức năng: Người dùng có thể xem danh sách yêu cầu kết bạn từ người khách, xếp theo trình tự thời gian gửi, có thể xem thông tin người khác khi nhấn xem trang cá nhân của tài khoản đó

* + 1. **Tạo tin nhắn**

Chức năng tạo tin nhắn cho phép người dùng cùng một người khác trao đổi với nhau thông qua Room Chat

* Chức năng dùng để tạo tin nhắn trao đổi
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:Không có

Yêu cầu chức năng: Người dùng có thể tạo tin nhắn, có thể là văn bản hay file định dạng hình ảnh, cho một người khác, có thể là bạn bè hoặc người lạ, thông qua Room Chat

* + 1. **Xoá tin nhắn**

Chức năng xoá tin nhắn cho phép người dùng xoá tin nhắn đã gửi trong Room Chat

* Chức năng dùng để xoá tin nhắn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ xoá tin nhắn

Yêu cầu chức năng: Người dùng có thể xoá tin nhắn đã gửi cho trong Room Chat

* + 1. **Xem danh sách hình ảnh đã nhắn**

Chức năng xem danh sách hình ảnh đã nhắn cho phép người dùng xem danh sách hình ảnh đã gửi trong Room Chat

* Chức năng dùng để xem danh sách hình ảnh đã gửi
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ: Không có

Yêu cầu chức năng: Người dùng có thể xem danh sách hình đã gửi trong Room Chat,

* + 1. **Tạo nhóm nhắn tin**

Chức năng tạo nhóm tin nhắn cho phép người dùng cùng nhiều người khác trao đổi với nhau thông qua Group Chat

* Chức năng dùng để tạo nhóm tin nhắn trao đổi
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:Không có

Yêu cầu chức năng: Người dùng có thể tạo nhóm tin nhắn, Kết quả tạo được sẽ là Group Chat. Group Chat có các tín năng của Room Chat thông thường và có thể mời bạn bè hoặc người lạ vào Group Chat.

* + 1. **Sửa nhóm nhắn tin**

Chức năng sửa nhóm tin nhắn cho phép người dùng sửa thông tin nhóm tin nhắn do tài khoản đăng nhập tạo từ trước

* Chức năng dùng để sửa nhóm tin nhắn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ sửa nhóm tin nhắn

Yêu cầu chức năng: Người dùng có thể sửa thông tin nhóm tin nhắn do tài khoản đăng nhập tạo từ trước.

* + 1. **Xóa nhóm nhắn tin**

Chức năng xóa nhóm tin nhắn cho phép người dùng xóa nhóm tin nhắn do tài khoản đăng nhập tạo từ trước

* Chức năng dùng để xóa nhóm tin nhắn
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ xóa nhóm tin nhắn

Yêu cầu chức năng: Người dùng xóa nhóm tin nhắn do tài khoản đăng nhập tạo từ trước, tất cả tin nhắn và hình ảnh đã gửi sẽ không tồn tại

* + 1. **Thêm thành viên vào nhóm nhắn tin**

Chức năng thêm thành viên vào nhóm nhắn tin cho phép người dùng đang sở Group Chat có thể thêm người khác vào Group Chat

* Chức năng dùng để thêm thành viên vào nhóm nhắn tin
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ thêm thành viên vào nhóm tin nhắn

Yêu cầu chức năng: Người dùng có thể thêm thành viên vào nhóm tin nhắn, có thể là bạn bè hoặc người lạ.

* + 1. **Xóa thành viên khỏi nhóm nhắn tin**

Chức năng xóa thành viên khỏi nhóm nhắn tin cho phép người dùng đang sở Group Chat có thể xóa người khác khỏi Group Chat

* Chức năng dùng để xóa thành viên khỏi nhóm nhắn tin
* Tác nhân: Người dùng.
* Mô tả chức năng nhỏ:

+ Thoát : Khi người dùng muốn huỷ xóa thành viên khỏi nhóm tin nhắn

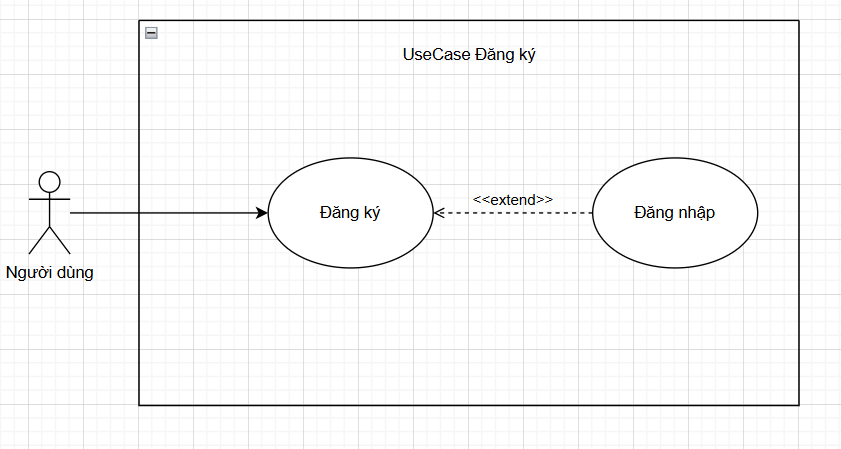
Yêu cầu chức năng: Người dùng có thể xóa thành viên khỏi nhóm tin nhắn,tin nhắn và hình ảnh người đã xóa sẽ còn tồn tại ở Group Chat, người bị xóa sẽ không thể xem được những tin nhắn đã nhắn trong Group Chat

## Phân tích thiết kế giao diện người dùng

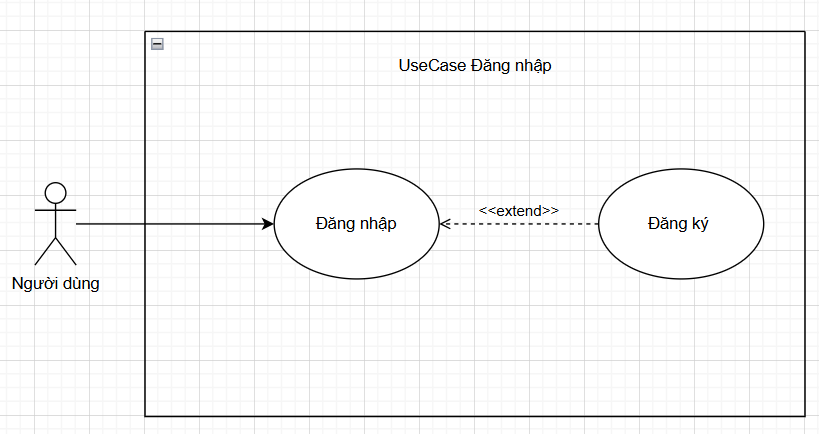
* + 1. **Phân tích thiết kế hệ thống** 
       1. **Sơ Đồ UseCase**

**Mô tả Actor của hệ thống:**

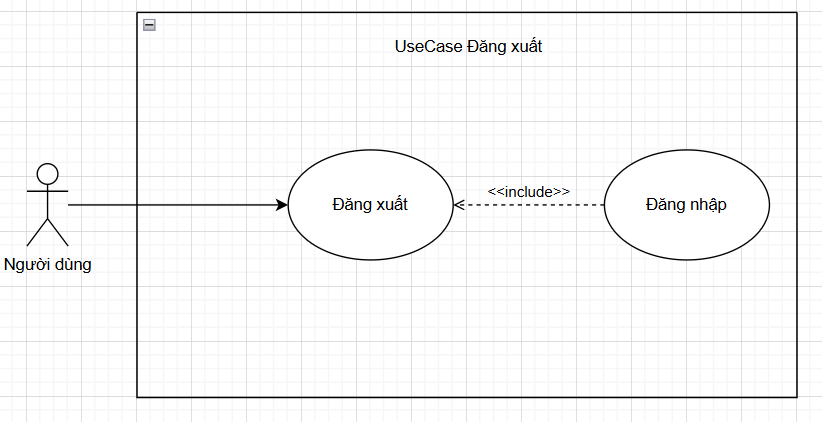
|  |  |
| --- | --- |
| Tên Actor | Tên Use Case |
| Người dùng | * Đăng nhập * Đăng ký * Đăng xuất * Xem Newfeed * Tìm kiếm * Đổi mật khẩu * Quản lý thông tin cá nhân : * Xem thông tin cá nhân * Sửa thông tin cá nhân * Quản lý bài đăng: * Xem bài đăng * Thêm bài đăng * Tương tác bài đăng * Quản lý bài viết đã lưu: * Xem danh sách lưu * Lưu bài * Xoá bài đã lưu * Quản lý bình luận: * Xem bình luận trong bài đăng * Thêm bình luận bài đăng * Phản hồi bình luận   - Quản lý bạn bè   * Xem danh sách bạn bè * Xóa bạn bè * Xem danh sách yêu cầu kết bạn * Thêm yêu cầu kết bạn * Huỷ yêu cầu kết bạn * Phản hồi kết bạn   - Quản lý tin nhắn   * Tạo Tin nhắn * Xoá Tin nhắn * Xem danh sách hình ảnh đã nhắn   - Quản lý nhóm nhắn tin   * Tạo nhóm nhắn tin * Sửa nhóm nhắn tin * Xoá nhóm nhắn tin * Thêm thành viên vào nhóm nhắn tin * Xoá thành viên khỏi nhóm nhắn tin |

* + - * 1. **Đăng ký** 

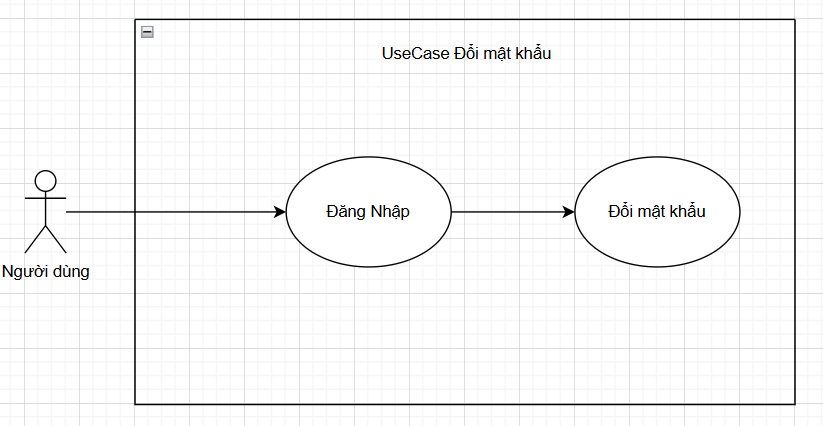
|  |  |
| --- | --- |
| **Use Case Name** | **Đăng ký** |
| **Description** | Người dùng muốn đăng ký vào web để sử dụng mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn đăng ký vào web |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập |
| **Post-Condition(s):** | Người dùng đăng ký tài khoản web thành công  Hệ thống ghi nhận thông tin người dùng |
| **Basic Flow** | 1. Người dùng truy cập vào web 2. Người dùng chọn chức năng đăng ký 3. Người dùng nhập thông tin đăng ký 4. Hệ thống xác nhận đăng ký thành công 5. Hệ thống lưu thông tin người dùng vào cơ sở dữ liệu 6. Hệ thống chuyển sang giao diện đăng nhập |
| **Alternative Flow** | Không có |
| **Exception Flow** | 4a. Hệ thống xác nhận thông tin đăng ký không hợp lệ và thông báo không thành công  4a1. Người dùng chọn hủy đăng ký  Use case dừng lại |

* + - * 1. **Đăng nhập** 

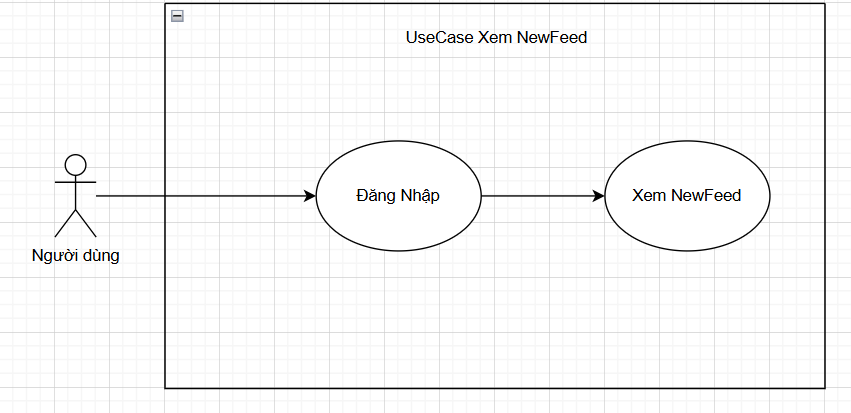
|  |  |
| --- | --- |
| **Use Case Name** | **Đăng nhập** |
| **Description** | Người dùng muốn đăng nhập vào web để sử dụng mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn đăng nhập vào web |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã sở hưu tài khoản từ trước |
| **Post-Condition(s):** | Người dùng đăng ký tài khoản web thành công  Hệ thống ghi nhận thông tin người dùng |
| **Basic Flow** | 1. Người dùng truy cập vào web 2. Người dùng nhập thông tin đăng nhập 3. Hệ thống xác nhận đăng ký thành công 4. Hệ thống xác nhận thông tin đăng nhập 5. Hệ thống chuyển sang giao diện trang NewFeed |
| **Alternative Flow** | Không có |
| **Exception Flow** | 4a. Hệ thống xác nhận thông tin đăng nhập thông báo không thành công  Use case dừng lại |

* + - * 1. **Đăng xuất** 

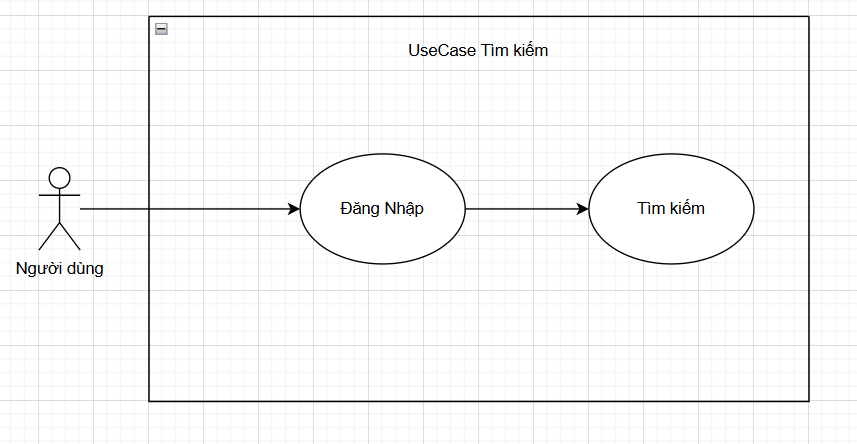
|  |  |
| --- | --- |
| **Use Case Name** | **Đăng xuất** |
| **Description** | Người dùng muốn đăng xuất khỏi web mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn đăng xuất khỏi web |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng đăng xuất web thành công  Hệ thống kết lúc phiên đó của tài khoản |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Đăng Xuất 3. Người dùng xác nhận đăng xuất 4. Hệ thống xác nhận kết thúc phiên đăng nhập |
| **Alternative Flow** | Không có |
| **Exception Flow** | 3a. Người dùng xác nhận Hủy đăng xuất  4.a Hệ thống tiếp tục phiên đăng nhập đó  Use case dừng lại |

* + - * 1. **Đổi mật khẩu** 

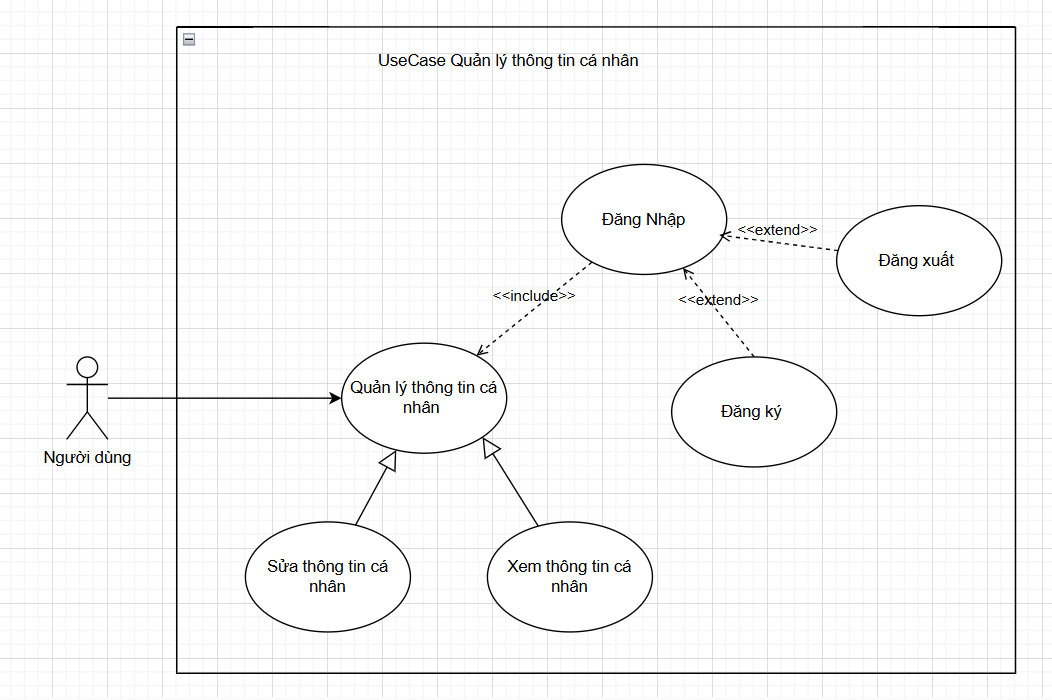
|  |  |
| --- | --- |
| **Use Case Name** | **Đổi mật khẩu** |
| **Description** | Người dùng muốn đổi mật khẩu tài khoản web mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn đổi mật khẩu tài khoả |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng đổi mật khẩu thành công  Hệ thống ghi nhận kết quả đổi mật khẩu |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Đổi Mật khẩu 3. Người dùng nhập các trường đổi mật khẩu 4. Người dùng xác nhận đổi mật khẩu 5. Hệ thống ghi nhận thông tin đổi mật khẩu |
| **Alternative Flow** | Không có |
| **Exception Flow** | 4a. Người dùng hủy phiên đổi mật khẩu  5b. Hệ thống ghi nhận đổi mật khẩu không thành công  Use case dừng lại |

* + - * 1. **Xem NewFeed** 

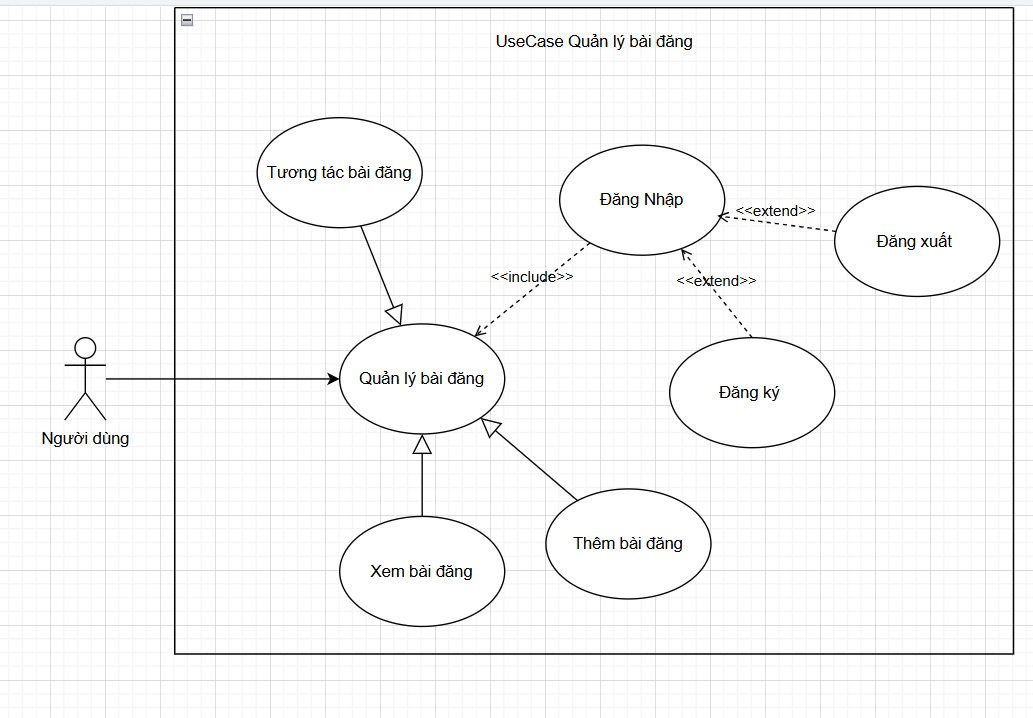
|  |  |
| --- | --- |
| **Use Case Name** | **Xem NewFeed** |
| **Description** | Người dùng muốn trang NewFeed của web mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn trang NewFeed |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập |
| **Post-Condition(s):** | Người dùng Xem được trang NewFeed |
| **Basic Flow** | 1. Người dùng truy cập vào web 2. Người dùng nhập thông tin đăng nhập 3. Hệ thống xác nhận đăng ký thành công 4. Hệ thống xác nhận thông tin đăng nhập 5. Hệ thống chuyển sang giao diện trang NewFeed |
| **Alternative Flow** | Không có |
| **Exception Flow** | 4a. Hệ thống xác nhận thông tin đăng nhập thông báo không thành công  Use case dừng lại |

* + - * 1. **Tìm kiếm** 

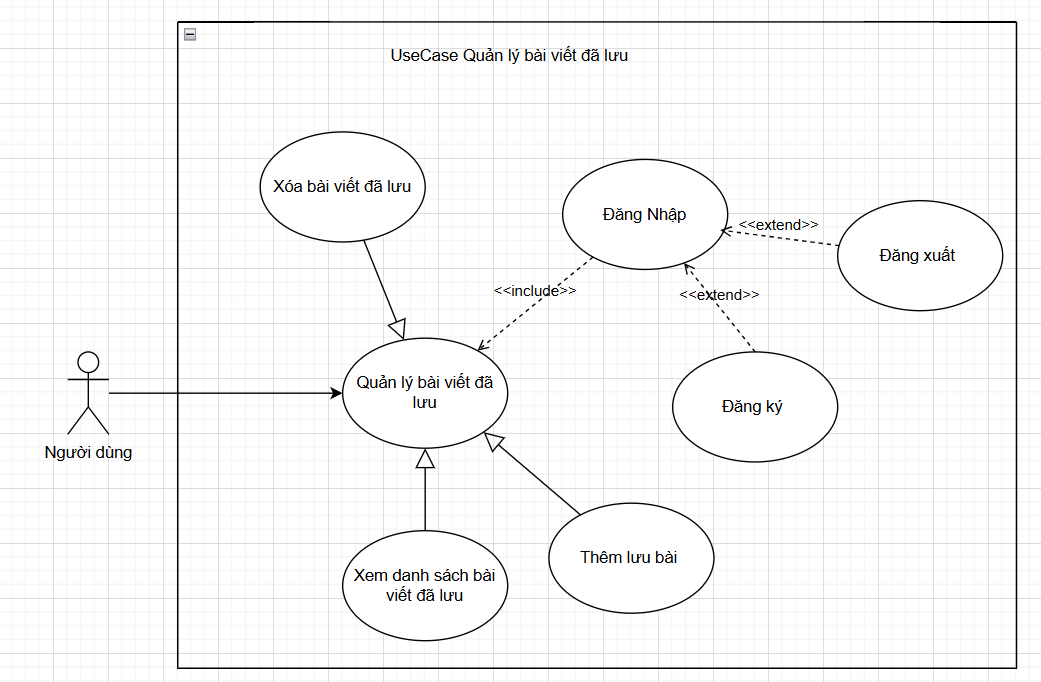
|  |  |
| --- | --- |
| **Use Case Name** | **Tìm kiếm** |
| **Description** | Người dùng muốn tìm kiếm nội dung trên web mạng xã hội Nemo |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn tìm kiếm nội dung |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận tìm kiếm  Hệ thống xuất kết quả tìm kiếm |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng nhập thông tin tìm kiếm lên thanh tìm kiếm 3. Người dùng xác nhận tìm kiếm 4. Hệ thống xuất kết quả tìm kiếm |
| **Alternative Flow** | Không có |
| **Exception Flow** | 3a. Người dùng không xác nhận tìm kiếm  Use case dừng lại |

* + - * 1. **Quản lý thông tin cá nhân** 

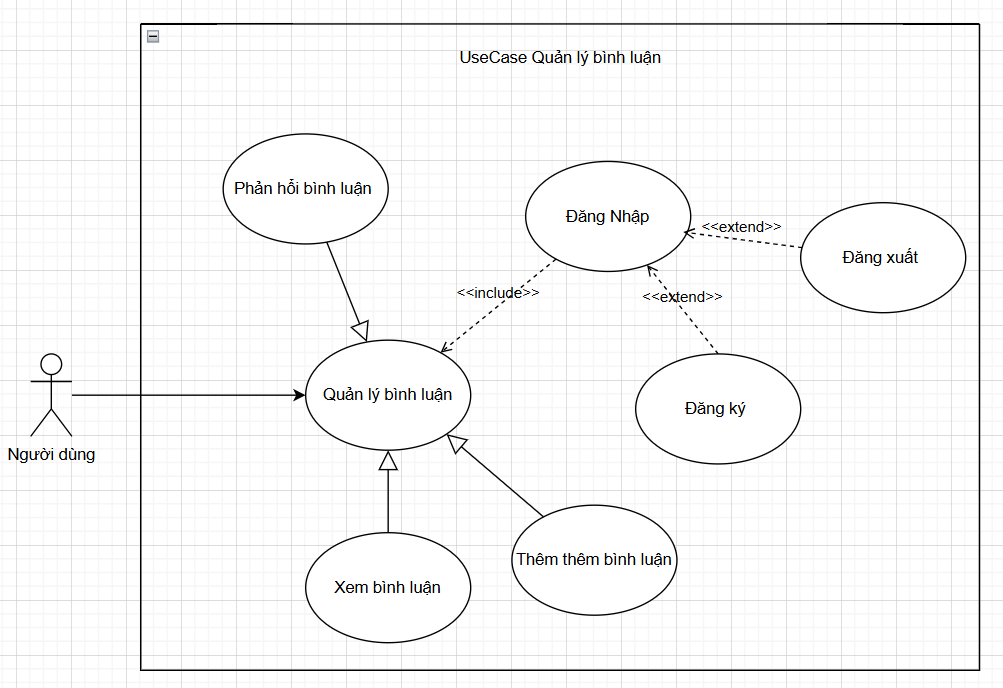
|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý thông tin cá nhân** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý thông tin cá nhân |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý thông tin cá nhân |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý thông tin cá nhân |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Trang cá nhân 3. Hệ thống hiện thi giao diện trang cá nhân 4. Người dùng chọn xem thông tin cá nhân 5. Hệ thống hiển thị trang thông tin cá nhân |
| **Alternative Flow** | 4a. Người dùng chọn chức năng sửa thông tin cá nhân  5a. Người dùng nhập các trường sửa thông tin cá nhân  6a. Người dùng xác nhận sửa thông tin cá nhân  7a. Hệ thống xác nhận sửa thông tin cá nhân |
| **Exception Flow** | 6a1. Người dùng xác nhận hủy sửa thông tin cá nhân  Use case dừng lại |

* + - * 1. **Quản lý bài đăng** 

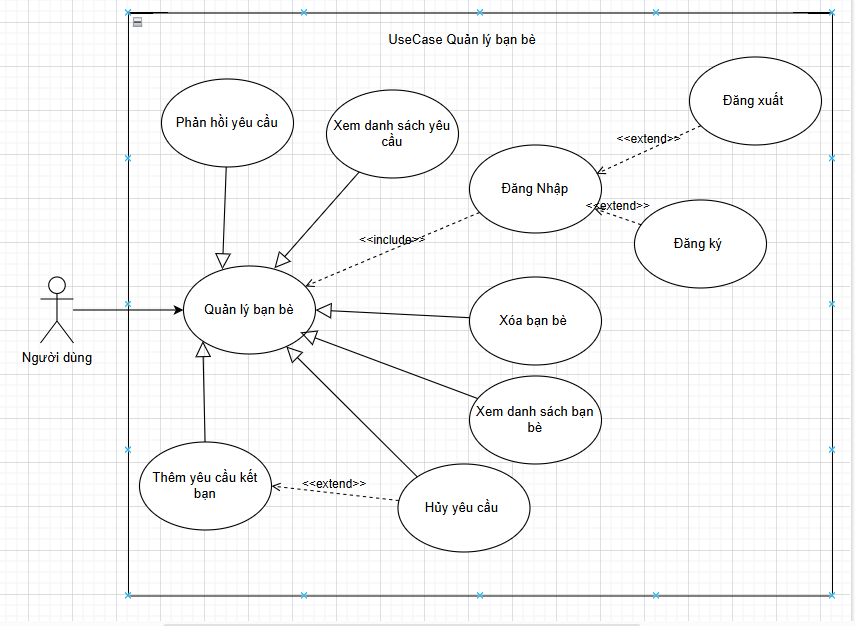
|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý bài đăng** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý bài đăng |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý bài đăng |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý bài đăng |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Trang cá nhân 3. Hệ thống hiện thi giao diện trang cá nhân và hiển thị danh sách bài viết |
| **Alternative Flow** | 2a. Người dùng nhập trường nội dung cho trường thêm bài đăng  3a Người dùng xác nhận thêm bài đăng  4a. Hệ thông xác nhận thông tin thêm bài đăng  2b. Người dùng chọn bài đăng bất kỳ trên NewFeed  3b. Người dùng nhấn Thích  4b. Hệ thống xác nhận kết quả Thích bài đăng  3b1. Người dùng nhấn Không thích  4b1. Hê thống ghi nhận kết quả không thích bài đăng |
| **Exception Flow** | 5a1. Người dùng xác nhận hủy sửa thông tin cá nhân  Use case dừng lại |

* + - * 1. **Quản lý bài viết đã lưu** 

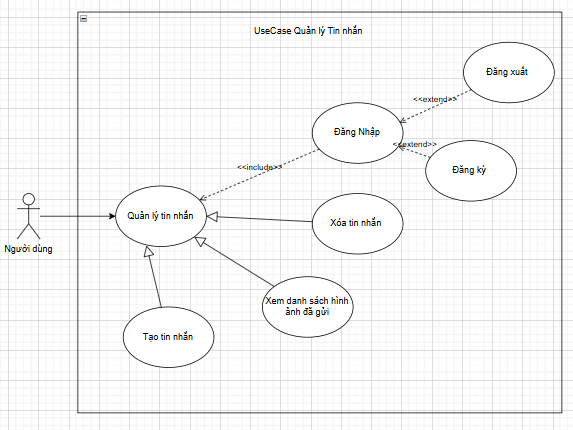
|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý bài viết đã lưu** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý bài viết đã lưu |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý bài viết đã lưu |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý bài viết đã lưu |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Đã lưu 3. Hệ thống hiện thi giao diện trang danh sách bài viết đã lưu 4. Chọn bài bất kì trên danh sách bài viết đã lưu 5. Người dùng chọn Bỏ lưu 6. Hệ thống xác nhận bỏ lưu bài viết |
| **Alternative Flow** | 2a. Người dùng chọn bài viết bất kì trên NewFeed  3a Người dùng chọn Lưu bài  4a. Hệ thông xác nhận lưu bài |
| **Exception Flow** | * 1. Người dùng xác nhận hủy bỏ lưu   3a.1 Người dùng xác nhận hủy lưu  Use case dừng lại |

* + - * 1. **Quản lý bình luận** 

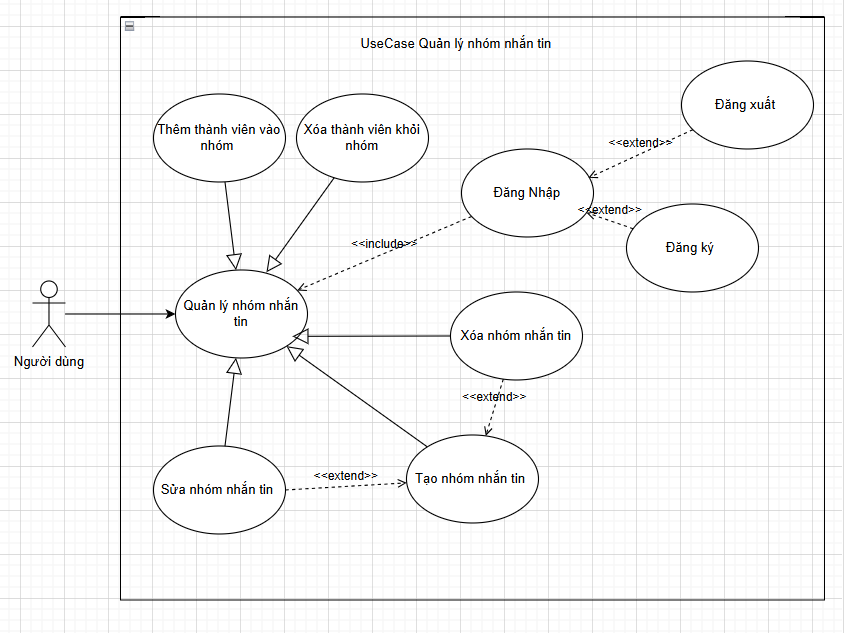
|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý bài bình luận** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý bình luận |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý bình luận |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý bình luận |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng chọn bài viết bất kì trên NewFeed 3. Người dùng chọn Bình luận 4. Hệ thống hiển thị danh sách bình luận 5. Người dùng nhập nội dung thêm bình luận 6. Người dùng xác nhận thêm bình luận 7. Hệ thống xác nhận thêm bình luận |
| **Alternative Flow** | 5a. Người dùng chọn bình luận bất kì trong danh sách bình luận  6a. Người dùng chọn Phản hồi  7a. Người dùng nhập nội dung phản hồi bình luận  8a. Người dùng xác nhận phản hồi bình luận  9a. Hệ thống xác nhận phản hồi bình luận |
| **Exception Flow** | 6.1 Người dùng xác nhận hủy thêm bình luận  8a.1 Người dùng xác nhận hủy phản hồi bình luận  Use case dừng lại |

* + - * 1. **Quản lý bạn bè** 

|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý bạn bè** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý bạn bè |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý bạn bè |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý bạn bè |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng tìm Bạn Bè 3. Hệ thống hiện thi giao diện trang danh sách bạn bè 4. Người dùng chọn người bạn bất kì trên danh sách 5. Người dùng chọn Xóa bạn bè 6. Người dùng xác nhận xóa 7. Hệ thống xác nhận xóa bạn bè |
| **Alternative Flow** | 3a. Người dùng tìm Yêu Cầu Kết Bạn  4a. Hệ thống hiện danh sách yêu cầu kết bạn  5a. Người dùng chọn yêu cầu bất kì trên danh sách  6a. Người dùng xác nhận phản hồi yêu cầu  7a. Hệ thống xác nhận phải hồi yêu cầu kết bạn  2b. Người dùng chọn người đăng bài bất kì trên NewFeed  3b. Người dùng chọn gửi yêu cầu kết bạn  4b. Hệ thống xác nhận gửi yêu cầu kết bạn  2c. Người dùng chọn người đăng bài mà đã gửi yêu cầu kết bạn cho họ từ trước  3c. Người dùng chọn Hủy yêu cầu kết bạn  4c . Hệ thống ghi nhận hủy yêu cầu kết bạn |
| **Exception Flow** | 3.1. Hệ thống hiện thị danh sách rỗng  6.1. Người dùng xác nhận hủy xóa bạn bè  4a.1. Hệ thống hiện danh sách rỗng  Use case dừng lại |

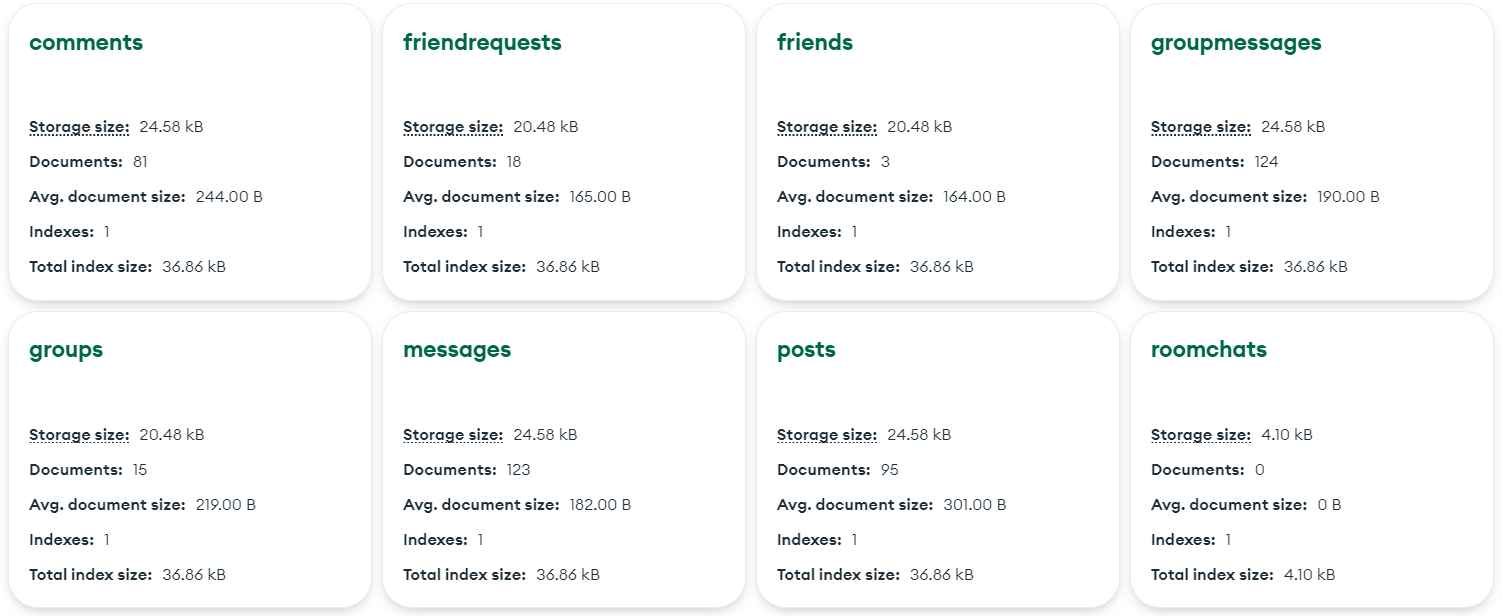
* + - * 1. **Quản lý tin nhắn** 

|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý tin nhắn** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý tin nhắn |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý tin nhắn |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý tin nhắn |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng chọn người đăng bài bất kì trên NewFeed 3. Người dùng chọn Nhắn tin 4. Hệ thống hiện thị giao diện RoomChat 5. Người dùng nhập nội dung tin nhắn 6. Người dùng xác nhận gửi 7. Hệ thống xác nhận tin nhắn gửi |
| **Alternative Flow** | 5a. Người dùng chọn tin nhắn đã gửi trong giao diện RoomChat  6a. Người dùng chọn Xóa tin nhắn  7a. Người dùng xác nhận Xóa  8a. Hệ Thống xác nhận tin nhắn xóa  5b. Người dùng xem danh sách hình ảnh đã gửi hiển thị trên RoomChat |
| **Exception Flow** | 6.1. Người dùng xác nhận hủy tin nhắn  7a.1. Người dùng xác nhận hủy xóa tin nhắn  Use case dừng lại |

* + - * 1. **Quản lý nhóm tin nhắn** 

|  |  |
| --- | --- |
| **Use Case Name** | **Quản lý nhóm tin nhắn** |
| **Description** | Người dùng muốn thực hiện các chức năng quản lý nhóm tin nhắn |
| **Actor(s)** | Người dùng |
| **Trigger** | Người dùng muốn quản lý nhóm tin nhắn |
| **Pre-Condition(s):** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập  Người dùng đã đăng nhập vào web trước đó |
| **Post-Condition(s):** | Người dùng xác nhận thực hiện các chức năng quản lý nhóm tin nhắn |
| **Basic Flow** | 1. Người dùng truy cập trang web đã đăng nhập 2. Người dùng chọn Nhắn tin 3. Hệ thống hiện thị giao diện RoomChat và danh sách người đã nhắn tin 4. Người dùng chọn tạo GroupChat 5. Người dùng nhập các trường tạo GroupChat 6. Người dùng xác nhận tạo 7. Hệ thống xác nhận tạo nhóm nhắn tin |
| **Alternative Flow** | 4a. Người dùng chọn GroupChat  5a. Hệ thống hiện thị danh sách nhóm nhắn tin đã có trước  6a. Người dùng chọn nhóm nhắn nhắn tin do tài khoản đăng nhập tạo  7a. Người dùng chọn Sửa GroupChat  8a. Người dùng nhập các trường sửa GroupChat  9a.Người dùng xác nhận sửa nhóm nhắn tin  10a. Hệ thống xác nhận sửa nhóm nhắn tin  4b. Người dùng chọn GroupChat  5b. Hệ thống hiện thị danh sách nhóm nhắn tin đã có trước  6b. Người dùng chọn nhóm nhắn nhắn tin do tài khoản đăng nhập tạo  7b. Người dùng chọn Xóa GroupChat  8b. Người dùng xác nhận xóa nhóm nhắn tin  9b. Hệ thống xác nhận xóa nhóm nhắn tin  4c. Người dùng chọn GroupChat  5c. Hệ thống hiện thị danh sách nhóm nhắn tin đã có trước  6c. Người dùng chọn nhóm nhắn nhắn tin do tài khoản đăng nhập tạo  7c. Người dùng chọn Thêm thành viên  8c. Hệ thống hiển thị danh sách thành viên có thể thêm  9c. Người dùng chọn thành viên bất kì để thêm  10c. Hệ thống xác nhận thêm thành viên  4c. Người dùng chọn GroupChat  5c. Hệ thống hiện thị danh sách nhóm nhắn tin đã có trước  6c. Người dùng chọn nhóm nhắn nhắn tin do tài khoản đăng nhập tạo  7c. Người dùng chọn Xóa thành viên  8c. Hệ thống hiển thị danh sách thành viên có thể xóa  9c. Người dùng chọn thành viên bất kì để xóa  10c. Hệ thống xác nhận xóa thành viên |
| **Exception Flow** | 6.1. Người dùng xác nhận hủy tin nhắn  7a.1. Người dùng xác nhận hủy xóa tin nhắn  Use case dừng lại |

* + - 1. **Cơ Sở Dữ Liệu**





**Giải thích cơ sở dữ liệu:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng comments** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã bình luận | ObjectId |  |
| 2 | content | Nội dung | String |  |
| 3 | author | Người bình luận | String | users |
| 4 | post | Bài đăng được bình luận | String | posts |
| 5 | isActive | Có được hoạt động không? | Boolean |  |
| 6 | replyTo | Phản hồi | Array | comments |
| 7 | likes | Lượt thích | Array | users |
| 8 | img | Hình ảnh | Array |  |
| 9 | createdAt | Ngày tạo | date |  |
| 10 | updatedAt | Ngày cập nhật | date |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng friendrequests** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã yêu cầu | ObjectId |  |
| 2 | sender | Người gửi | String | users |
| 3 | receiver | Người nhận | String | users |
| 4 | status | Trạng thái | String | Chỉ chứa 3 giá trị: pending ,accepted, declined |
| 5 | createdAt | Ngày tạo | date |  |
| 6 | updatedAt | Ngày cập nhật | date |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng friends** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã yêu cầu | ObjectId |  |
| 2 | sender | Người gửi | String | users |
| 3 | receiver | Người nhận | String | users |
| 4 | status | Trạng thái | String | Mặc định giá trị: friend |
| 5 | createdAt | Ngày tạo | date |  |
| 6 | updatedAt | Ngày cập nhật | date |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng groupmessages** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã nhóm tin nhắn | ObjectId |  |
| 2 | group | Mã nhóm | String | groups |
| 3 | sender | Người gửi | ObjectId | users |
| 4 | content | Nội dung | String |  |
| 5 | mediaURL | Link hình ảnh | Array |  |
| 6 | reading | Người đọc | Array | users |
| 7 | createdAt | Ngày tạo | date |  |
| 8 | updatedAt | Ngày cập nhật | date |  |

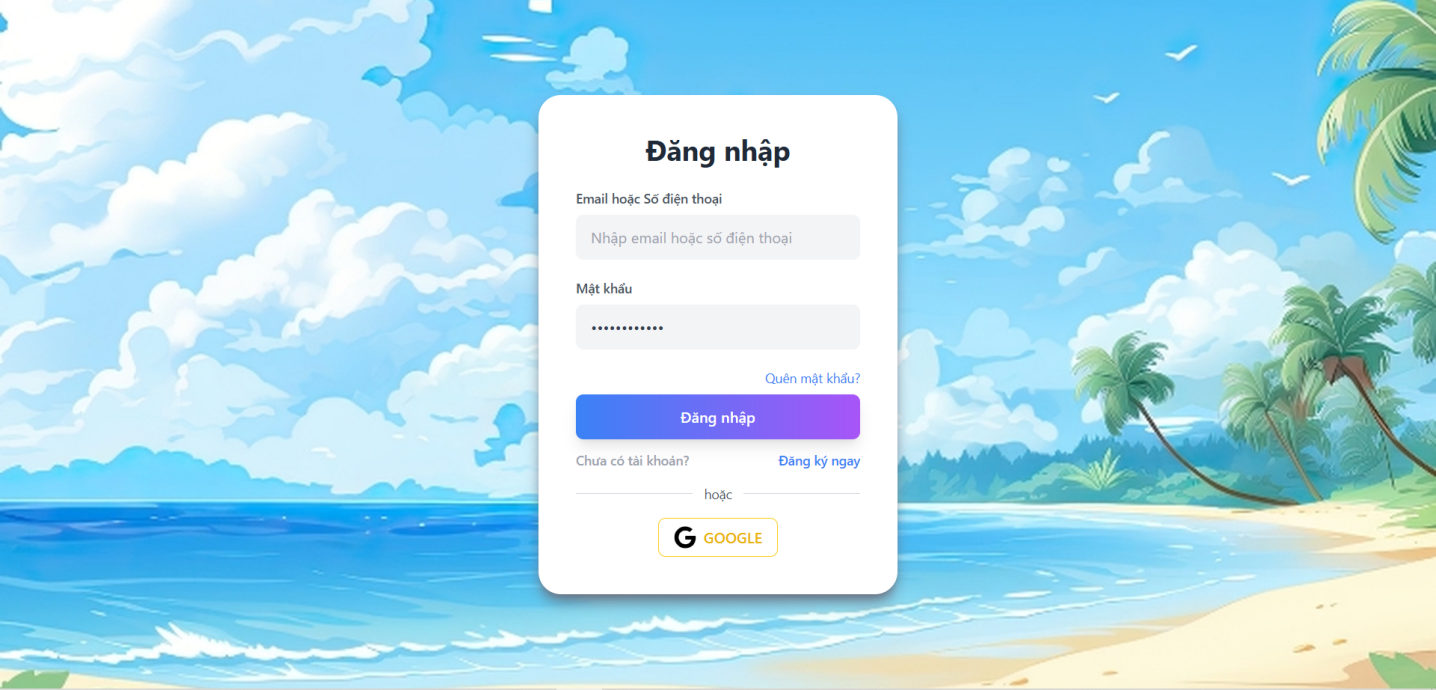
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng groups** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã nhóm | ObjectId |  |
| 2 | name | Tên nhóm | String |  |
| 3 | avatarGroup | Hình đại diện nhóm | Array |  |
| 4 | owner | Chủ nhóm | ObjectId | users |
| 5 | participants | Người tham gia | Array | users |
| 6 | numberMember | Số lượng thành viên | Number |  |
| 7 | createdAt | Ngày tạo | date |  |
| 8 | updatedAt | Ngày cập nhật | date |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng messages** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã tin nhắn | ObjectId |  |
| 2 | sender | Người gửi | ObjectId | users |
| 3 | receiver | Người nhận | ObjectId | users |
| 4 | content | Nội dung | String |  |
| 5 | mediaURL | Link hình ảnh | Array |  |
| 6 | status | Trạng thái | String | Chỉ nhận 3 giá trị:  sent, received, seen |
| 7 | isLive | Tồn tại hay không | boolean |  |
| 8 | createdAt | Ngày tạo | date |  |
| 9 | updatedAt | Ngày cập nhật | date |  |

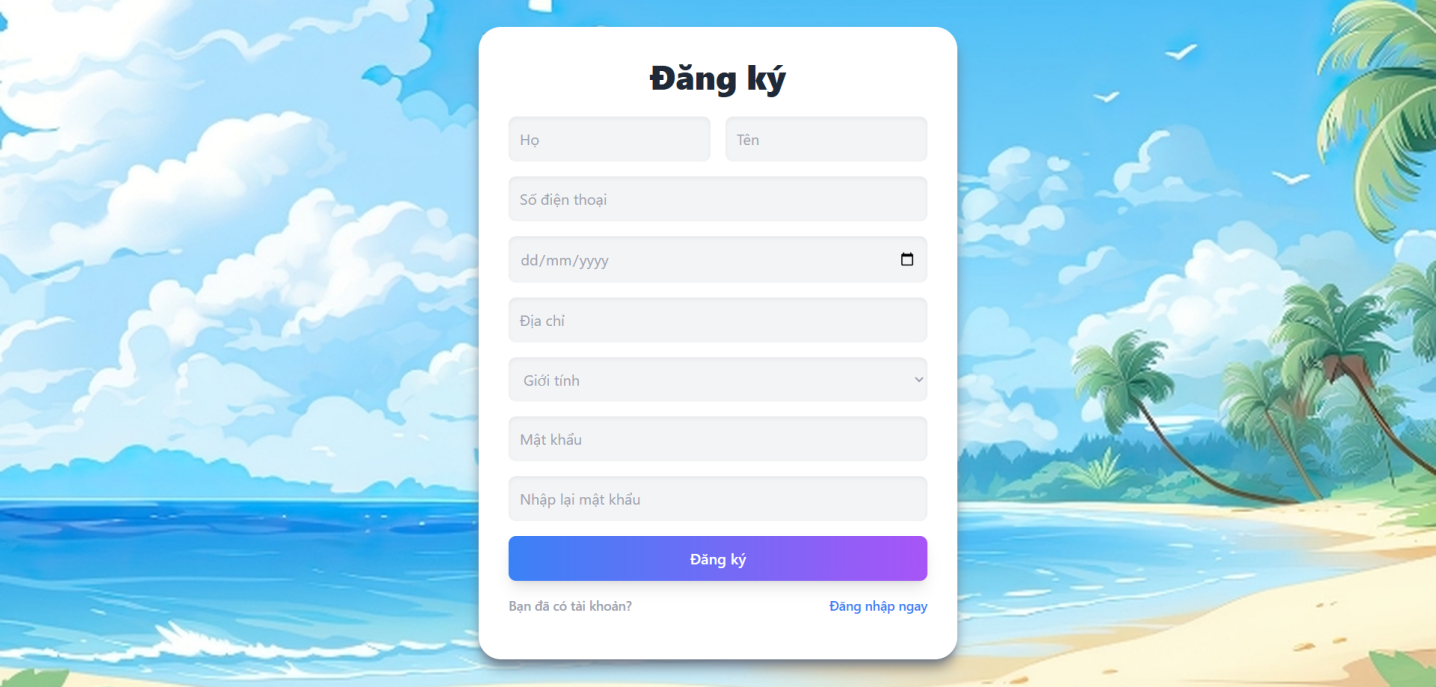
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng posts** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã bài đăng | ObjectId |  |
| 2 | content | Nội dung | String |  |
| 3 | author | Người đăng | String | users |
| 4 | likes | Lượt thích | Array | users |
| 5 | dislikes | Lượt không thích | Array | users |
| 6 | comments | Bình luận | Array | comments |
| 7 | likesCount | Lượt thích | Number |  |
| 8 | commentCount | Lượt bình luận | Number |  |
| 9 | img | Hình ảnh | Array |  |
| 10 | privacy | Chế độ | String | Chỉ nhận 4 giá trị  : public, friend, private, specific |
| 11 | allowedUsers | Người đã theo dõi | Array |  |
| 12 | createdAt | Ngày tạo | date |  |
| 13 | updatedAt | Ngày cập nhật | date |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bảng users** | | | | |
| **STT** | **Tên cột** | **Diễn giải** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | \_id | Mã người dùng | ObjectId |  |
| 2 | numberPhone | Số điện thoại | String | Mỗi số điện thoại chỉ riêng cho một người dùng |
| 3 | password | Mật khẩu | String |  |
| 4 | firstName | Tên | String |  |
| 5 | lastName | Họ | String |  |
| 6 | address | Địa chỉ | String |  |
| 7 | gender | Giới tính | Boolean |  |
| 8 | birthday | Sinh nhật | String |  |
| 9 | bookmark | Đánh dấu đã lưu | Array |  |
| 10 | role | Vai trò | Boolean |  |
| 11 | isActive | Có hoạt động không ? | Boolean |  |
| 12 | createdAt | Ngày tạo | date |  |
| 13 | updatedAt | Ngày cập nhật | date |  |
| 14 | refreshToken | Làm mới Token | String |  |
| 15 | avatar | Ảnh đại diện | String |  |
| 16 | coverImage | Ảnh bìa | String |  |
| 18 | email | Thư điện tử | String |  |
| 19 | otp | Mật khẩu dùng một lần | String |  |
| 20 | otpEpirationTime | Thời gian tồn tại của OTP | date |  |
| 21 | friends | Bạn bè | Array |  |

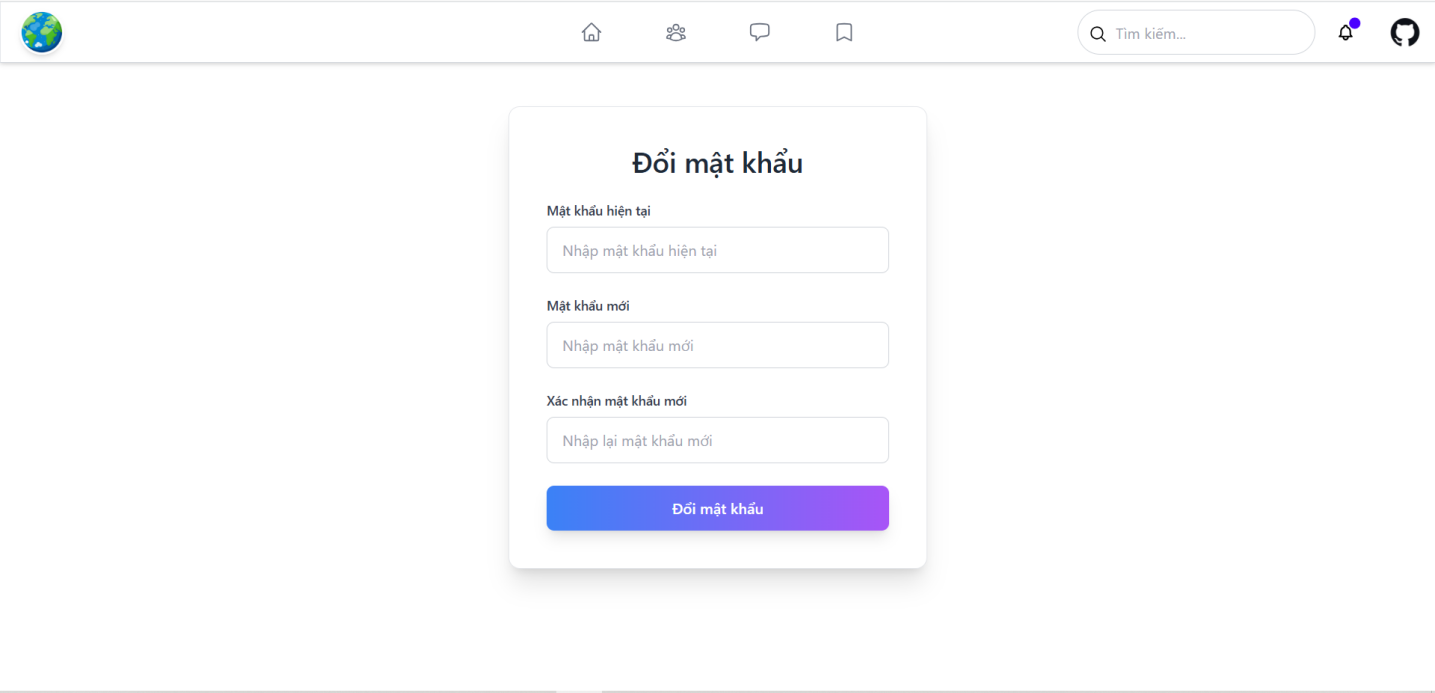
* + 1. **Thiết kế giao diện**
* **Giao diện đăng nhập**



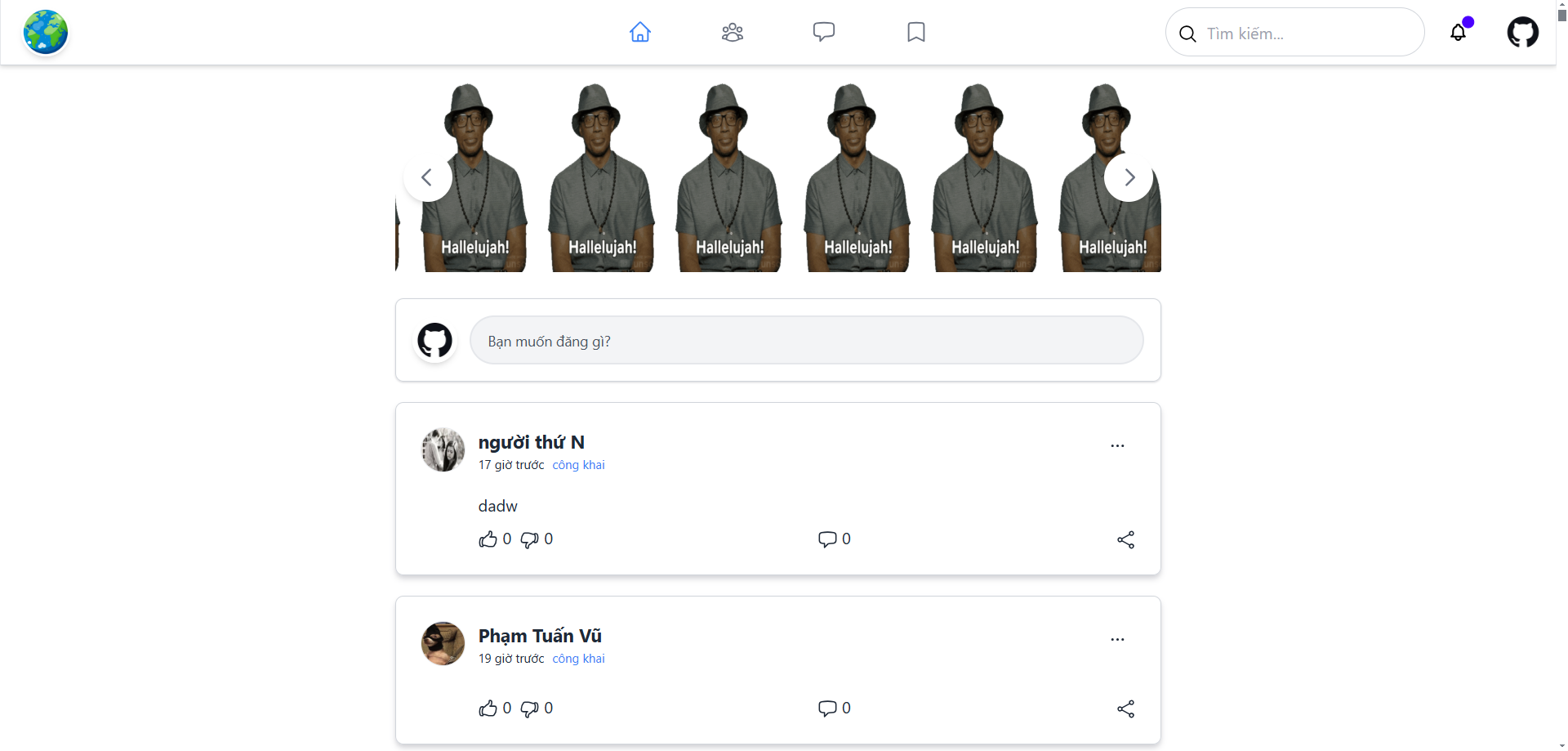
* **Giao diện đăng ký**



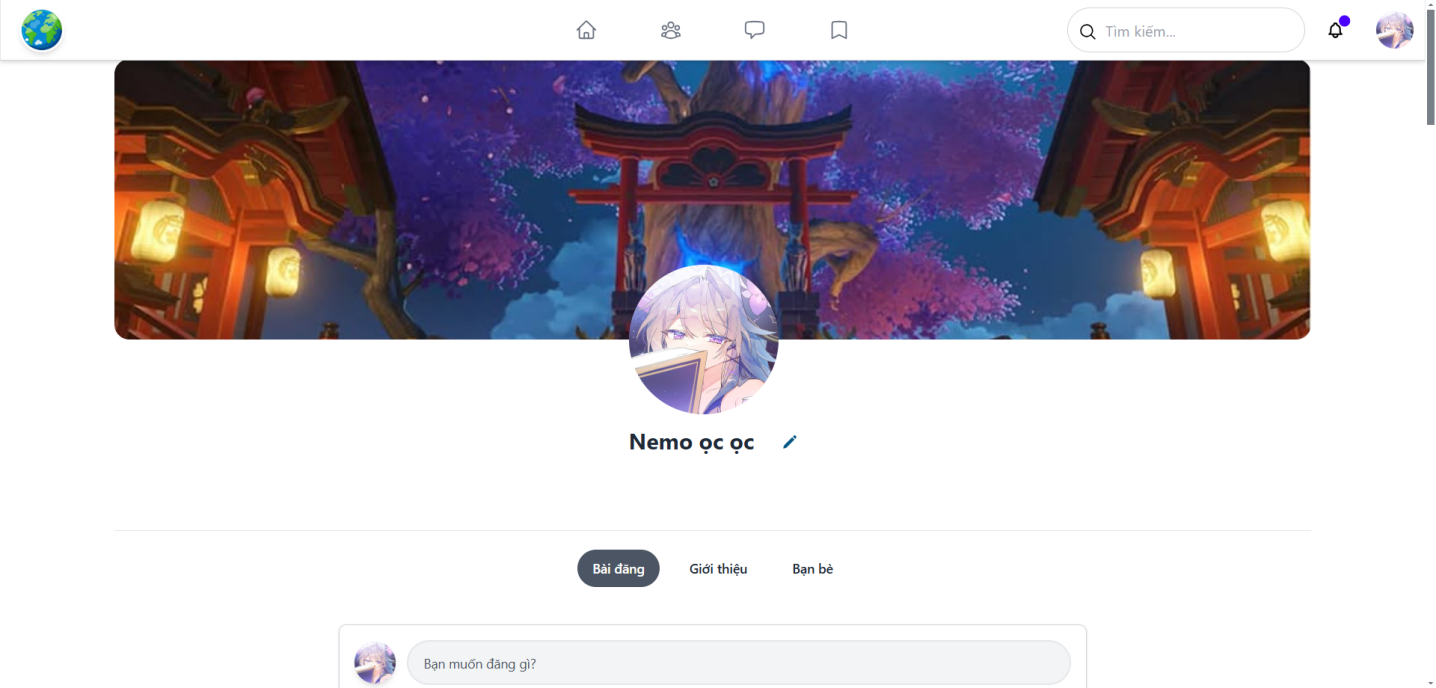
* **Giao diện đổi mật khẩu**



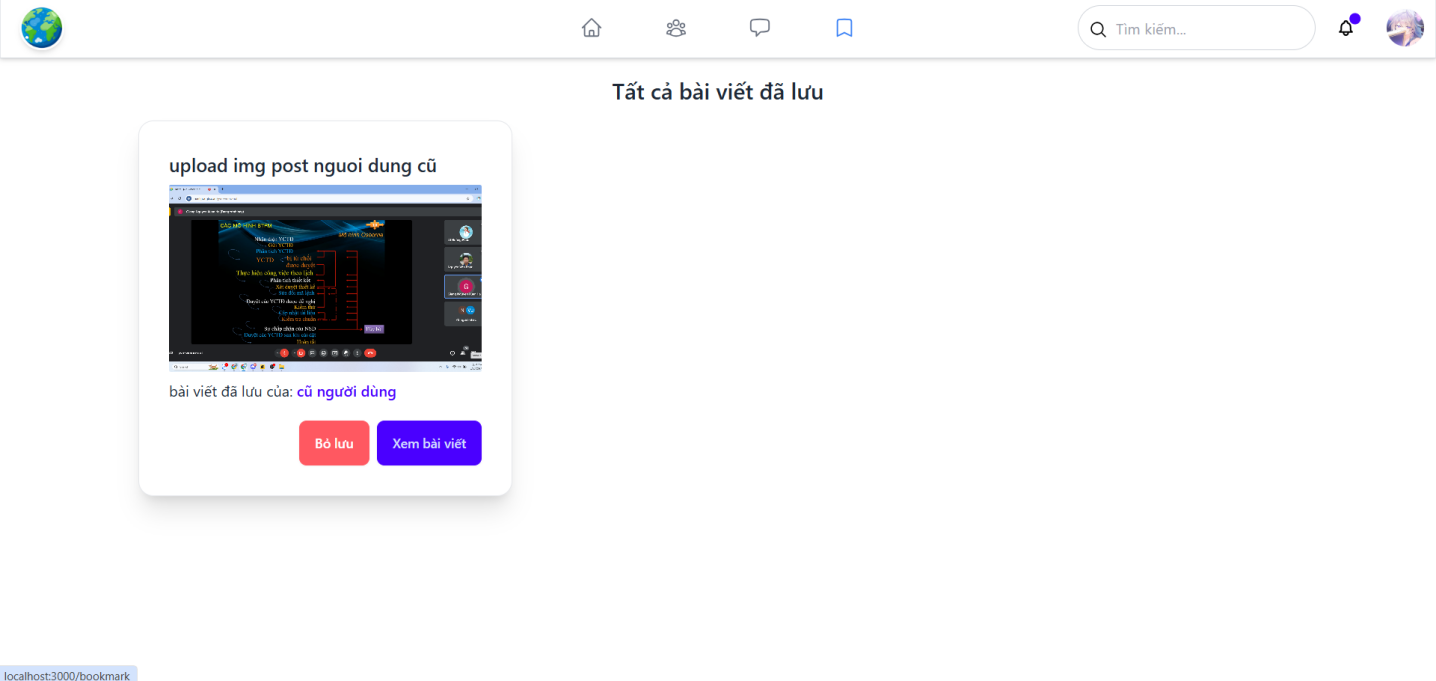
* **Giao diện Trang NewFeed**



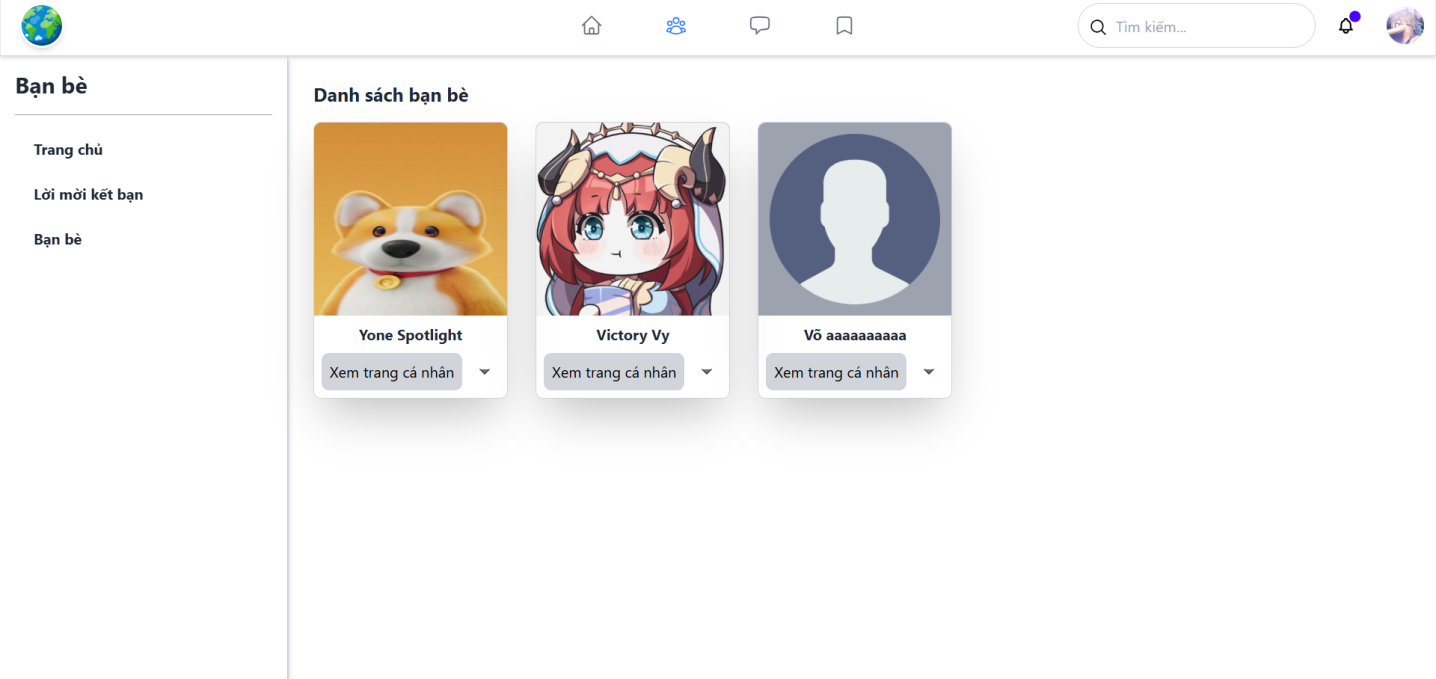
* **Giao diện Trang Cá nhân**



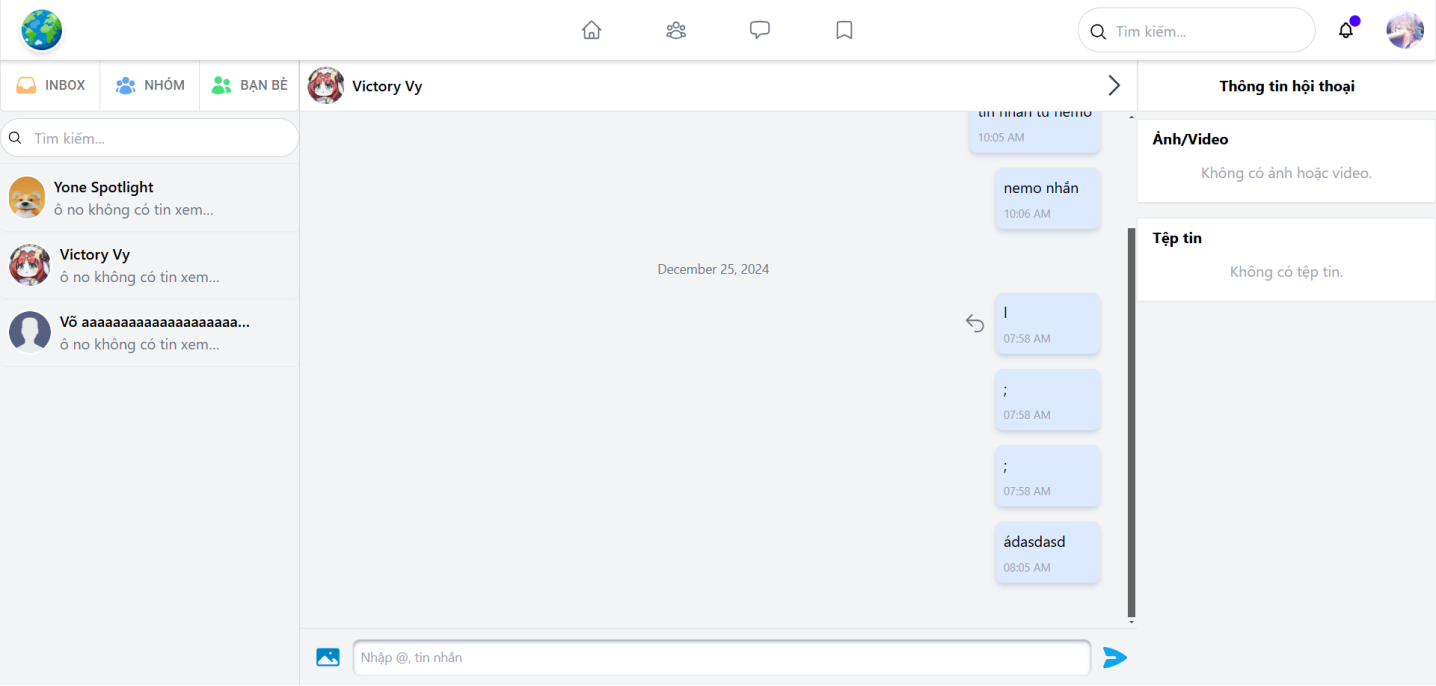
* **Giao diện Trang Đã lưu**

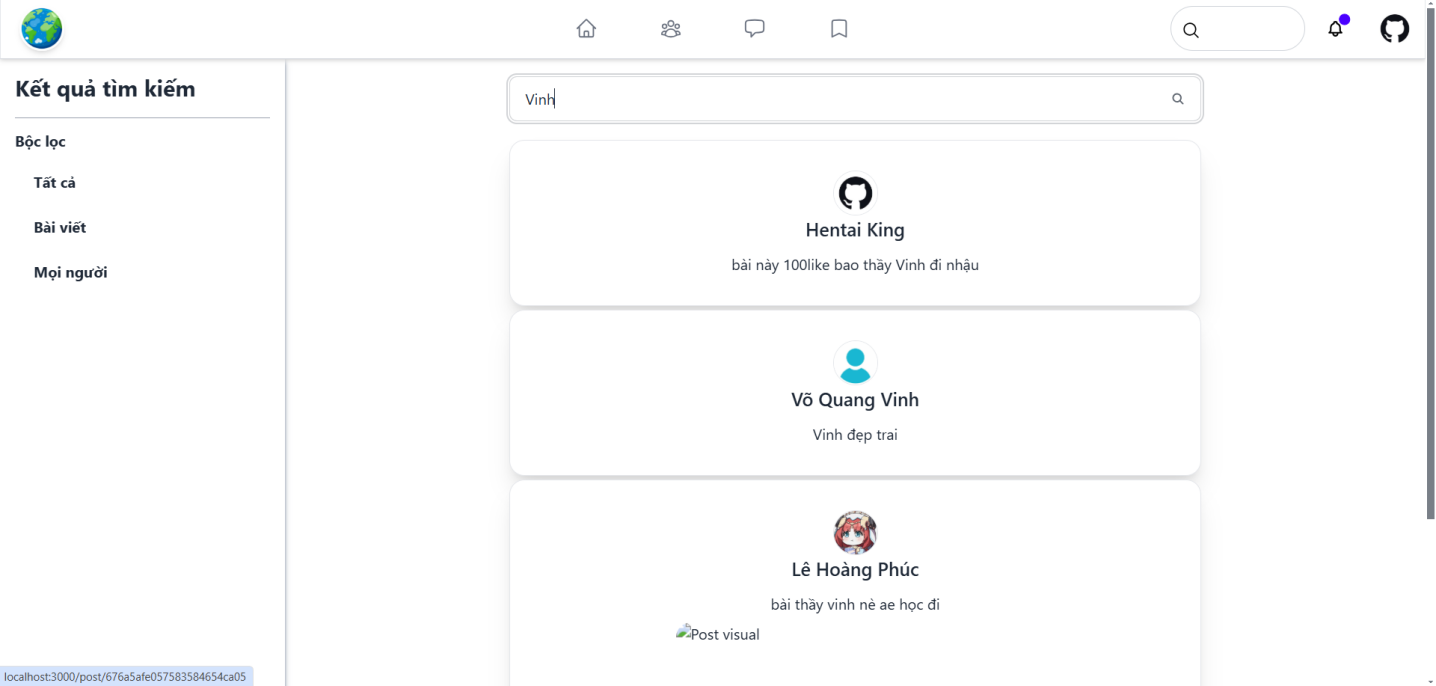


* **Giao diện Trang Bạn Bè**



* **Giao diện RoomChat và nhắn tin**



* **Giao diện Trang kết quả tìm kiếm**
* 
  + 1. **Bảng tiến độ công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên công việc** | **Thời gian** | **Người**  **thực hiện** | **Tiến độ** |
| 1 | Phân tích thiết kế. | 2 tuần | Phạm Hữu Tiến | Hoàn thành |
| 2 | Tạo cơ sở dữ liệu và nhập thông tin vào cơ sở dữ liệu. | 1/4 tuần | Phạm Hữu Tiến  Phan Nguyễn Hoàng Trân  Lê Hoàng Phúc  Phạm Tuấn Vũ | Hoàn thành |
| 3 | Cài đặt các công cụ hỗ trợ và kết nối đến CSDL | 1/4 tuần | Phạm Hữu Tiến  Phan Nguyễn Hoàng Trân  Lê Hoàng Phúc  Phạm Tuấn Vũ | Hoàn thành |
| 4 | Nghiên cứu và tìm hiểu cơ sở lý thuyết. | 1 tuần | Phạm Hữu Tiến  Phan Nguyễn Hoàng Trân  Lê Hoàng Phúc  Phạm Tuấn Vũ | Hoàn thành |
| 5 | Xây dựng cơ sở backend website | 15 tuần | Phạm Hữu Tiến  Phan Nguyễn Hoàng Trân | Hoàn thành |
| 6 | Xây dựng giao diện frontend website | 15 tuần | Lê Hoàng Phúc  Phạm Tuấn Vũ | Hoàn thành |
| 7 | Viết tài liệu báo cáo | 1 tuần | Phan Nguyễn Hoàng Trân | Hoàn thành |

**CHƯƠNG 4. KẾT QUẢ THỰC HIỆN**

**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## Kết quả đạt được

* Xây dựng được hoàn chỉnh trang web
* Web có thể hoạt động được trên nền web mobile
* Hoàn thành được hầu hết các chức năng được yêu cầu trong đặc tả

## Hạn chế

* Giao diện chưa được tối ưu
* Có thể gặp lỗi trong quá trình sử dụng

## Hướng phát triển

* Hoàn thiện các chức năng còn thiếu.
* Phát triển thêm các chức năng đã có
* Nghiên cứu và xây dựng thêm đăng nhập bằng tài khoản facebook hay bằng google.

**TÀI LIỆU THAM KHẢO**

**WEBSITE**

<https://www.w3schools.com/>

<https://docs.nestjs.com/>

<https://react.dev/learn>