ỦY BAN NHÂN DÂN TP.HCM

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**CHUYÊN ĐỀ PHÁT TRIỂN WEB 1**

**NĂM HỌC 2024-2025**

**Tên đề tài: Phát Triển Website chat mini**

**Nhóm: M**

**Nguyễn Văn Lê Tri Tổng-21211TT2036 (NT)**

**Ngô Minh Tấn-21211TT1966 (NP)**

**Nguyễn Phương Tri-21211TT1539**

**Huỳnh Tấn Huy-2121TT1369**

**Hồ Đào Việt Thắng-21211TT0883**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Chủ nhiệm đề tài: Phan Thanh Nhuần**

**TP. Hồ Chí Minh, năm 2024**

ỦY BAN NHÂN DÂN TP.HCM

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**CHUYÊN ĐỀ PHÁT TRIỂN WEB 1**

**NĂM HỌC 2024 – 2025**

**Tên đề tài: Phát Triển Website chat mini**

**Nhóm: M**

**Nguyễn Văn Lê Tri Tổng-21211TT2036 (NT)**

**Ngô Minh Tấn-21211TT1966 (NP)**

**Nguyễn Phương Tri-21211TT1539**

**Huỳnh Tấn Huy-2121TT1369**

**Hồ Đào Việt Thắng-21211TT0883**

**Chủ nhiệm đề tài: Phan Thanh Nhuần**

**TP. Hồ Chí Minh, năm 2024**

**Mục lục**

[**PHẦN 1: MỞ ĐẦU** 1](#_heading=h.gjdgxs)

[**I: KẾ HOẠCH THỰC HIỆN** 1](#_heading=h.30j0zll)

[**1. Phân chia công việc** 1](#_heading=h.1fob9te)

[**2. Lịch họp nhóm** 4](#_heading=h.3znysh7)

[**PHẦN 2: GIỚI THIỆU ĐỀ TÀI** 5](#_heading=h.2et92p0)

[**I: Website chat mini** 5](#_heading=h.tyjcwt)

[**1. Giới thiệu MINI CHAT:** 5](#_heading=h.3dy6vkm)

[**II: MÔ TẢ CÔNG NGHỆ SỬ DỤNG:** 5](#_heading=h.1t3h5sf)

[**1. Laravel** 5](#_heading=h.4d34og8)

[**2. Vue.js** 6](#_heading=h.2s8eyo1)

[**3. Docker** 7](#_heading=h.17dp8vu)

[**PHẦN 3: LƯỢC ĐỒ QUAN HỆ** 8](#_heading=h.3rdcrjn)

[**I.**](#_heading=h.26in1rg) **Sơ đồ ERD** 8

[**II. Mô tả database** 9](#_heading=h.35nkun2)

[**1. Table oauth\_clients** 9](#_heading=h.1ksv4uv)

[**2. Table friends** 10](#_heading=h.44sinio)

[**3. Table failed\_jobs** 11](#_heading=h.2jxsxqh)

[**4. Table Friend\_requests** 13](#_heading=h.z337ya)

[**5. Table Oauth\_auth\_code** 14](#_heading=h.3j2qqm3)

[**6. Table Oauth\_access\_tokens** 15](#_heading=h.1y810tw)

[**7. Table Users** 17](#_heading=h.4i7ojhp)

[**8. Oauth\_personal\_access\_clients** 19](#_heading=h.2xcytpi)

[**9. oauth\_refresh\_tokens** 20](#_heading=h.1ci93xb)

[**10. Password\_Reset Token** 21](#_heading=h.3whwml4)

[**11. Messages** 21](#_heading=h.2bn6wsx)

[**12.Personal\_access\_tokens** 23](#_heading=h.qsh70q)

[**13. Websockets\_statistícs\_entries** 25](#_heading=h.3as4poj)

[**14. group\_messages** 26](#_heading=h.1pxezwc)

[**15. group\_message\_statuses** 27](#_heading=h.49x2ik5)

[**16. groups** 28](#_heading=h.2p2csry)

[**17. group\_users** 29](#_heading=h.147n2zr)

[**PHẦN 4: HÌNH ẢNH THIẾT KẾ CỦA DỰ ÁN** 30](#_heading=h.3o7alnk)

[**I: Trang trò chuyện cùng với bạn bè** 30](#_heading=h.23ckvvd)

[**II. Trang danh sách bạn bè** 31](#_heading=h.32hioqz)

[**III. Trang cài đặt người dùng** 32](#_heading=h.41mghml)

[**IV. Trang thông tin tài khoản bạn bè** 33](#_heading=h.vx1227)

[**V. Trang xóa bạn bè** 34](#_heading=h.1v1yuxt)

[**VI. Trang tìm kiếm thêm bạn** 36](#_heading=h.2u6wntf)

[**VII. Trang thêm bạn hiện thông tin khi nhấn tìm kiếm** 37](#_heading=h.3tbugp1)

[**VIII. Trang tìm kiếm người dùng** 38](#_heading=h.nmf14n)

[**IX. Trang hủy lời mời kết bạn** 39](#_heading=h.1mrcu09)

[**X. Trang lời mời kết bạn** 40](#_heading=h.2lwamvv)

[**XI: Trang cài đặt ngôn ngữ** 42](#_heading=h.3l18frh)

[**XII: Trang đăng xuất** 43](#_heading=h.4k668n3)

[**PHẦN 5: TÀI LIỆU THAM KHẢO** 44](#_heading=h.1egqt2p)

[**I. Docker:** https://www.docker.com/ 44](#_heading=h.3ygebqi)

[**II. Laravel framwork:** https://laravel.com/ 44](#_heading=h.2dlolyb)

[**III. Chat GPT**: https://www.chatgpt.com/ 44](#_heading=h.sqyw64)

[**IV. Vuejs:** https://vuejs.org/ 44](#_heading=h.3cqmetx)

[**V. Stackoverflow:** https://stackoverflow.com/ 44](#_heading=h.1rvwp1q)

**Danh mục hình ảnh**

[Hình 1: Sơ đồ ERD 8](#_heading=h.lnxbz9)

[Hình 2:Trò chuyện 30](#_heading=h.ihv636)

[Hình 3: Danh sách bạn bè 31](#_heading=h.1hmsyys)

[Hình 4: Cài đặt người dùng 32](#_heading=h.2grqrue)

[Hình 5: Thông tin tài khoản bạn bè 33](#_heading=h.3fwokq0)

[Hình 6: Xóa bạn bè 34](#_heading=h.4f1mdlm)

[Hình 7: Tìm kiếm thêm bạn 36](#_heading=h.19c6y18)

[Hình 8: Thêm bạn hiện thông tin khi nhấn tìm kiếm 37](#_heading=h.28h4qwu)

[Hình 9: Tìm kiếm người dùng 38](#_heading=h.37m2jsg)

[Hình 10: Hủy lời mời kết bạn 39](#_heading=h.46r0co2)

[Hình 11: Lời mời kết bạn 40](#_heading=h.111kx3o)

[Hình 12: Cài đặt ngôn ngữ 42](#_heading=h.206ipza)

[Hình 13 : Đăng xuất 43](#_heading=h.2zbgiuw)

# **PHẦN 1: MỞ ĐẦU**

## **I: KẾ HOẠCH THỰC HIỆN**

### **1. Phân chia công việc**

| **#** | Thành viên phụ trách | Công việc được giao | Thời gian thức hiện | Trạng thái | Mức độ | Đánh giá |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Nguyễn Văn Lê Tri Tổng | Đưa ra ý tưởng đề tài và hướng dẫn các thành viên thực hiện |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế database của dự án |  |  | **100%** | **Hoàn thành** |
|  |  | Định nghĩa chức năng hoạt động của các trang của dự án |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang danh sách bạn bè |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu docker |  |  | **100%** | **Hoàn thành** |
| 2 | Ngô Minh Tấn | Tìm hiểu về laravel |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế database |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang trò chuyện |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang thêm bạn |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu docker |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu Vuejs |  |  | **100%** | **Hoàn thành** |
| 3 | Nguyễn Phương Tri | Thiết kế UX/UI trang lời mời kết bạn |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang login |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang register |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang chủ |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu docker |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế database |  |  | **100%** | **Hoàn thành** |
| 4 | Huỳnh Tấn Huy | Tìm hiểu về Vuejs |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang lời mời kết bạn |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang tìm kiếm bạn bè |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang thông tin tài khoản bạn bè |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế database |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu docker |  |  | **100%** | **Hoàn thành** |
| 5 | Hồ Đào Việt Thắng | Tìm hiểu về Vuejs |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang cài đặt ngôn ngữ |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang cá nhân |  |  | **100%** | **Hoàn thành** |
|  |  | Tìm hiểu docker |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế UX/UI trang trò chuyện |  |  | **100%** | **Hoàn thành** |
|  |  | Thiết kế database |  |  | **100%** | **Hoàn thành** |

### **2. Lịch họp nhóm**

| # | Ngày tháng | Nội dung trao đổi | Ghi chú |
| --- | --- | --- | --- |
| 1 | 21/09/2024  14h-17h | Nói sơ lược về đề tài nhóm thực hiện nghiên cứu |  |
| 2 | 23/09/2024  19h-21h | Phân chia công việc hợp lý cho từng thành viên phụ trách |  |
| 3 | 27/09/2024  19h30-23h | Từng thành viên báo cáo mức độ hoành thành được giao |  |
| 4 | 30/09/2024  14h-17h30 | Thành viên trong nhóm hỏi và giải đáp thắc mắc cho nhau về khó khăn hay thuận lợi của từng công việc được giao. |  |
| 5 | 05/10/2024  18h30-22h | Đóng góp ý kiến cho báo cáo của nhóm và thảo luận cách trình bày báo cáo |  |

# **PHẦN 2: GIỚI THIỆU ĐỀ TÀI**

## **I: Website chat mini**

### **1. Giới thiệu MINI CHAT:**

* Trong thời đại công nghệ hiện nay, việc giao tiếp trực tuyến đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Ứng dụng trò chuyện trực tuyến không chỉ giúp kết nối mọi người một cách nhanh chóng và tiện lợi mà còn hỗ trợ trong công việc, học tập và giải trí. Với mục tiêu tạo ra một nền tảng giao tiếp đơn giản nhưng hiệu quả, đề tài “Phát triển trang web chat mini” đã được thực hiện nhằm cung cấp một giải pháp trò chuyện trực tuyến dễ sử dụng và thân thiện với người dùng.
* **Trang web trò chuyện mini được phát triển với các tính năng cơ bản như gửi và nhận tin nhắn, thông báo tin nhắn, hình ảnh mới và quản lý danh sách bạn bè. Đề tài này không chỉ tập trung vào việc xây dựng một hệ thống hoạt động mượt mà mà chú ý đến trải nghiệm của người dùng, bảo mật thông tin và khả năng mở rộng trong tương lai. Hơn nữa, việc tích hợp các công nghệ hiện đại** như WebSocket và mã hóa dữ liệu giúp bảo mật và hiệu suất của hệ thống. Qua đó, người dùng có thể yên tâm sử dụng mà không lo lắng về vấn đề bảo mật thông tin cá nhân.
* Ngày càng có nhiều người tiêu dùng bỏ qua các kênh dịch vụ khách hàng truyền thống như điện thoại và email và thay vào đó chọn sử dụng các kênh hiện đại hơn như Facebook, Twitter và Live Web Chat.
* Tất cả các công nghệ luôn có ưu và nhược điểm. Và mặc dù thực tế là nó sẽ hoạt động hoàn hảo đối với một số người, nhưng công nghệ tương tự có thể không dành cho những người khác.
* Trò chuyện trực tiếp trên web cũng giống như Facebook và Twitter, đó là một cam kết rất lớn đối với một công ty và cần có thời gian và nỗ lực để thực hiện đúng.

13h58, ngày 07/03/2025

Thành phần gồm có: 45/50

Nguồn: URL

Điều 16, khoản HSSV: copy nhận 0 => Yes

Nguyễn Văn A: N0 => YES: nhận điểm 0 cho môn học

Môn nào: ngày , .. d

**II: MÔ TẢ CÔNG NGHỆ SỬ DỤNG:**

### **1. Laravel**

* Laravel là một framework phát triển ứng dụng web mã nguồn mở được viết bằng PHP. Nó được thiết kế với mục đích giúp các nhà phát triển xây dựng các ứng dụng web một cách nhanh chóng, hiệu quả và dễ dàng bảo trì. Laravel được tạo ra bởi Taylor Otwell và được phát hành lần đầu vào năm 2011. Tính đến thời điểm hiện tại, Laravel là một trong những framework phát triển web PHP phổ biến nhất trên thế giới.
* Các đặc điểm nổi bật của laravel
* Kiến trúc MVC (Model-View-Controller):
* Model: Đại diện cho dữ liệu và logic liên quan đến dữ liệu. Trong Laravel, các model thường tương tác với cơ sở dữ liệu.
* View: Đại diện cho giao diện người dùng, nơi hiển thị thông tin cho người dùng cuối.
* Controller: Điều khiển luồng của ứng dụng, xử lý các yêu cầu từ người dùng và gửi dữ liệu tới view hoặc model.

Eloquent ORM: Eloquent là một ORM (Object-Relational Mapping) mạnh mẽ trong Laravel, cho phép bạn tương tác với cơ sở dữ liệu thông qua các đối tượng và mô hình dữ liệu.

* Nó cung cấp cách dễ dàng để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên các bảng cơ sở dữ liệu mà không cần phải viết các truy vấn SQL trực tiếp.
* Routing mạnh mẽ: Laravel cung cấp một cú pháp rõ ràng và linh hoạt để xác định các tuyến đường (routes) cho ứng dụng của bạn.
* Điều này giúp quản lý các yêu cầu **HTTP** và điều hướng chúng tới các controller tương ứng một cách dễ dàng.
* Blade Templating Engine: Blade là một templating engine mạnh mẽ trong Laravel, cho phép bạn tạo và quản lý giao diện người dùng một cách dễ dàng.
* Blade cung cấp các tính năng như kế thừa (inheritance), phân chia layout (layout inheritance) và các hàm trợ giúp, giúp làm cho việc phát triển giao diện trở nên linh hoạt và hiệu quả.
* Bảo mật: Laravel cung cấp các tính năng bảo mật tích hợp sẵn như middleware, CSRF (Cross-Site Request Forgery) protection, hashing mật khẩu, và hệ thống xác thực.
* Điều này giúp đảm bảo rằng ứng dụng của bạn được bảo vệ chắc chắn khỏi các lỗ hổng bảo mật phổ biến.
* Cộng đồng mạnh mẽ: Laravel có một cộng đồng lớn và năng động, với nhiều tài liệu, package mở rộng và hỗ trợ từ cộng đồng.
* Cộng đồng này không chỉ cung cấp sự hỗ trợ cho việc giải quyết vấn đề mà còn chia sẻ các kiến thức và kinh nghiệm, giúp việc học và phát triển ứng dụng Laravel trở nên dễ dàng hơn.

### **2. Vue.js**

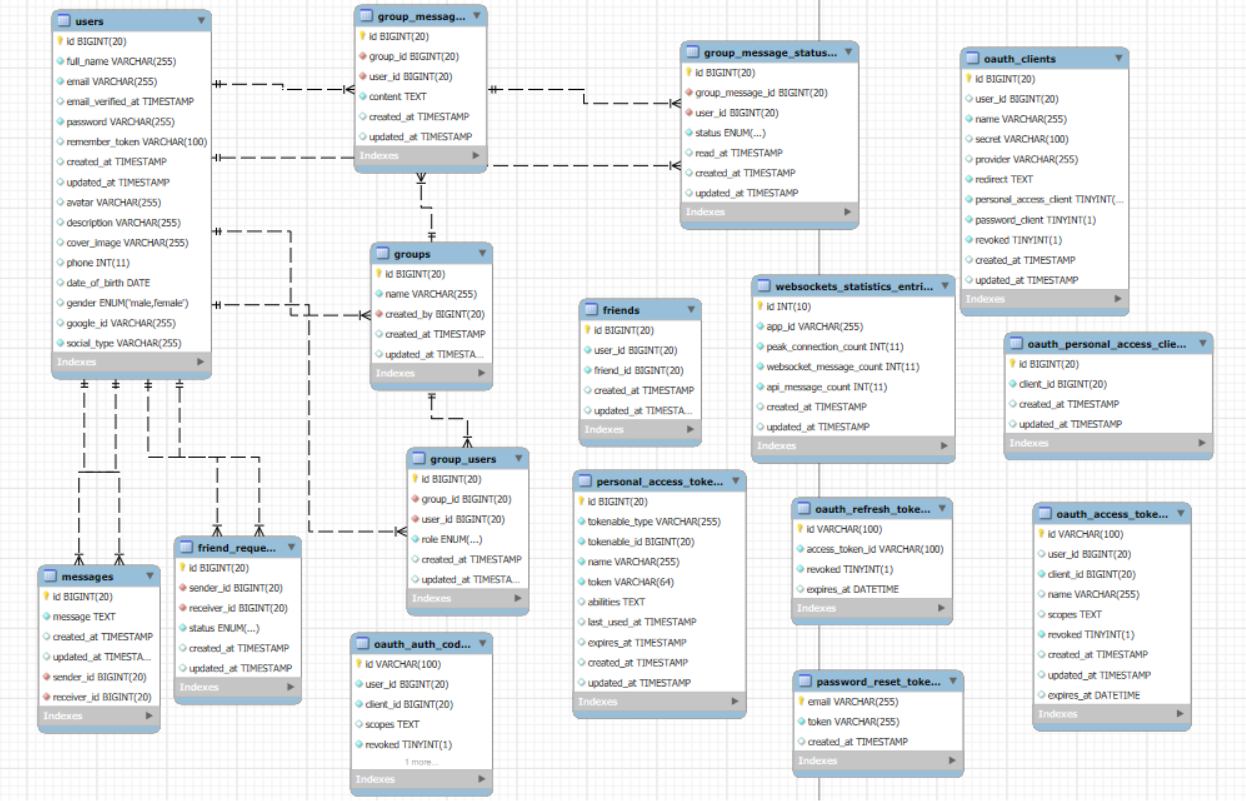
* Vue.js là một framework JavaScript mã nguồn mở, được sử dụng để xây dựng các giao diện người dùng (UI) động và tương tác. Được phát triển bởi Evan You và ra mắt lần đầu vào năm 2014, Vue.js đã nhanh chóng trở thành một trong những framework phổ biến nhất trong cộng đồng lập trình web nhờ vào tính linh hoạt và dễ sử dụng của nó.
* Vue.js được thiết kế theo hướng tiệm tiến (progressive), cho phép các nhà phát triển sử dụng từng phần của framework mà không cần phải áp dụng toàn bộ. Điều này giúp Vue.js dễ dàng tích hợp vào các dự án hiện có hoặc sử dụng cho các ứng dụng mới từ đầu. Một trong những điểm mạnh của Vue.js là khả năng tạo ra các thành phần (components) tái sử dụng, giúp quản lý mã nguồn hiệu quả và tăng cường khả năng bảo trì.
* Framework này sử dụng mô hình MVVM (Model-View-ViewModel), cho phép liên kết dữ liệu hai chiều (two-way data binding) giữa model và view. Điều này có nghĩa là khi dữ liệu trong model thay đổi, giao diện người dùng sẽ tự động cập nhật và ngược lại. Vue.js cũng hỗ trợ tốt cho việc quản lý trạng thái ứng dụng, giúp các nhà phát triển dễ dàng kiểm soát và theo dõi các thay đổi trong ứng dụng.
* Một số tính năng nổi bật của Vue bao gồm: Reactivity: Hệ thống phản ứng của Vue.js cho phép cập nhật giao diện người dùng một cách tự động khi dữ liệu thay đổi.
* Component-based Architecture: Kiến trúc dựa trên thành phần giúp tái sử dụng mã nguồn và quản lý dự án dễ dàng hơn.
* Single File Components: Cho phép viết HTML, CSS và JavaScript trong cùng một file .vue, giúp tổ chức mã nguồn gọn gàng và dễ quản lý.
* Ecosystem: Vue.js có một hệ sinh thái phong phú với nhiều thư viện và công cụ hỗ trợ như Vue Router, Vuex, và Vue CLI, giúp tăng cường khả năng phát triển và triển khai ứng dụng.
* Với những ưu điểm trên, Vue.js đã trở thành lựa chọn hàng đầu cho nhiều dự án phát triển web hiện đại, từ các ứng dụng nhỏ đến các hệ thống phức tạp.

### **3. Docker**

* Docker là một nền tảng container hóa mã nguồn mở, được phát triển để đóng gói, vận chuyển và chạy ứng dụng một cách dễ dàng và đồng nhất trên nhiều môi trường khác nhau. Docker cho phép các nhà phát triển tạo ra các container, trong đó chứa tất cả các phần mềm và thư viện cần thiết để chạy ứng dụng, cùng với các tài nguyên hệ thống được cô lập từ môi trường máy chủ.
* Một trong những ưu điểm lớn nhất của Docker là khả năng đảm bảo tính nhất quán giữa các môi trường phát triển, kiểm thử và sản xuất. Điều này giúp giảm thiểu các vấn đề liên quan đến sự khác biệt về cấu hình môi trường, từ đó tăng cường hiệu quả làm việc và giảm thiểu lỗi phát sinh.
* Các tính năng nổi bật của Docker bao gồm: Containerization: Docker sử dụng công nghệ container để đóng gói ứng dụng và các phụ thuộc của nó vào một đơn vị duy nhất. Điều này giúp ứng dụng có thể chạy trên bất kỳ môi trường nào mà không cần phải lo lắng về sự khác biệt về cấu hình.
* Portability: Với Docker, bạn có thể dễ dàng di chuyển ứng dụng từ máy phát triển sang máy chủ sản xuất mà không gặp phải các vấn đề về tương thích.
* Isolation: Docker cung cấp một môi trường cô lập cho mỗi container, giúp đảm bảo rằng các ứng dụng không ảnh hưởng lẫn nhau và tăng cường bảo mật.
* Scalability: Docker cho phép bạn dễ dàng mở rộng ứng dụng bằng cách thêm hoặc bớt các container một cách linh hoạt. Điều này rất hữu ích trong các hệ thống microservices, nơi mà mỗi dịch vụ có thể chạy trong một container riêng biệt.
* Efficiency: Docker container rất nhẹ và nhanh, giúp tiết kiệm tài nguyên hệ thống và tăng tốc độ khởi động ứng dụng.
* Docker đã trở thành một công cụ quan trọng trong quy trình DevOps, giúp tự động hóa việc triển khai và quản lý ứng dụng. Với Docker, các nhà phát triển có thể tập trung vào việc viết mã mà không cần phải lo lắng về việc cấu hình môi trường, từ đó tăng cường hiệu quả và chất lượng sản phẩm.

# **PHẦN 3: LƯỢC ĐỒ QUAN HỆ**

## **Sơ đồ ERD**



*Hình 1: Sơ đồ ERD*

**II. Mô tả database**

### **1. Table oauth\_clients**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint unsigned | Primary key |
| user\_id | bigint unsigned |  |
| name | varchar |  |
| secret | varchar |  |
| provider | varchar |  |
| redirect | text |  |
| personal\_access\_client | tinyint |  |
| password\_client | tinyint |  |
| revoked | tinyint |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table* ***oauth\_clients***

* **id (bigint unsigned):**
* Khóa chính của bảng, là mã định danh duy nhất cho từng bản ghi client trong hệ thống.
* **user\_id (bigint unsigned):**
* ID người dùng sở hữu client này, dùng để liên kết với bảng người dùng.
* **name (varchar):**
* Tên của client, giúp phân biệt các ứng dụng khác nhau mà người dùng sử dụng.
* **secret (varchar):**
* Mã bảo mật của client, sử dụng trong quá trình xác thực client khi người dùng kết nối.
* **provider (varchar):**
* Nhà cung cấp dịch vụ xác thực mà client đang sử dụng, ví dụ như Google, Facebook, v.v.
* **redirect (text):**
* URL chuyển hướng sau khi người dùng được xác thực thành công.
* **personal\_access\_client (tinyint):**
* Cờ đánh dấu xem client này có phải là client truy cập cá nhân không (1 là đúng, 0 là không).
* **password\_client (tinyint):**
* Cờ đánh dấu xem client này sử dụng xác thực qua mật khẩu hay không (1 là có, 0 là không).
* **revoked (tinyint):**
* Trạng thái bị thu hồi của client, nếu giá trị là 1, client này đã bị hủy quyền truy cập.
* **created\_at (timestamp):**
* Thời gian tạo client.
* **updated\_at (timestamp):**
* Thời gian cập nhật gần nhất của thông tin client.

### **2. Table friends**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint unsigned | Primary key |
| user\_id | bigint unsigned |  |
| friend\_id | bigint unsigned |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa* ***Table friends***

* **id (bigint unsigned, Primary key):**
* Đây là cột chứa khóa chính của bảng, được sử dụng để định danh duy nhất cho mỗi bản ghi trong bảng.
* Kiểu dữ liệu là bigint unsigned, nghĩa là nó chứa số nguyên lớn không âm. Nó có thể được tự động tăng dần (AUTO\_INCREMENT), giúp tạo giá trị duy nhất cho mỗi bản ghi mới.
* **user\_id (bigint unsigned):**
* Cột này lưu trữ ID của người dùng. Đây là một khóa ngoại để liên kết với bảng người dùng (nếu có).
* bigint unsigned nghĩa là nó chứa giá trị ID của người dùng, không âm và có thể lưu trữ số lớn.
* **friend\_id (bigint unsigned):**
* Cột này lưu trữ ID của bạn bè mà người dùng đang kết nối hoặc tương tác.
* bigint unsigned cũng lưu trữ giá trị lớn không âm và thường được liên kết với bảng người dùng để thể hiện quan hệ bạn bè giữa các người dùng trong hệ thống.
* **created\_at (timestamp):**
* Đây là dấu thời gian khi bản ghi được tạo. timestamp lưu trữ thời gian và ngày tháng chính xác khi hành động kết bạn (hoặc bất kỳ hành động nào) diễn ra.
* Cột này thường được tự động thiết lập thời gian hiện tại khi bản ghi được tạo.
* **updated\_at (timestamp):**
* Đây là dấu thời gian khi bản ghi được cập nhật lần cuối. timestamp lưu trữ thời gian khi bản ghi được thay đổi.
* Thường được tự động cập nhật mỗi khi có sự thay đổi dữ liệu trong bản ghi.

### **3. Table failed\_jobs**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint unsigned | Primary Key |
| uuid | varchar |  |
| connection | text |  |
| queue | text |  |
| payload | longtext |  |
| exception | longtext |  |
| failed\_at | timestamp |  |

*Bảng định nghĩa table Failed\_jobs*

* **id (bigint unsigned):**
* Cột này là khóa chính của bảng, có chức năng định danh duy nhất cho mỗi bản ghsi.
* Kiểu dữ liệu là bigint unsigned, nghĩa là nó lưu trữ các giá trị số nguyên không âm lớn. Có thể sử dụng AUTO\_INCREMENT để tự động tăng giá trị cho mỗi bản ghi mới.
* **uuid (varchar):**
* UUID (Universally Unique Identifier) là một chuỗi ký tự dạng varchar, được sử dụng để tạo ra các giá trị duy nhất trên toàn hệ thống.
* Được sử dụng để định danh duy nhất cho các mục đích khác, không chỉ dựa vào khóa chính id, giúp tăng tính bảo mật và quản lý các yêu cầu riêng biệt.
* **connection (text):**
* Cột này lưu trữ tên hoặc thông tin của kết nối liên quan đến bản ghi, chẳng hạn như kết nối tới hệ thống hoặc dịch vụ bên ngoài.
* Kiểu dữ liệu là text, cho phép lưu trữ lượng lớn ký tự.
* **queue (text):**
* Cột này lưu trữ thông tin về hàng đợi (queue) mà bản ghi đang liên quan. Trong hệ thống xử lý tác vụ, hàng đợi thường được sử dụng để lưu các tác vụ hoặc nhiệm vụ cần xử lý.
* text cho phép lưu trữ các chuỗi ký tự với độ dài không xác định.
* **payload (longtext):**
* Payload là phần dữ liệu chính của bản ghi, chứa thông tin cần thiết để xử lý tác vụ.
* Kiểu dữ liệu longtext cho phép lưu trữ lượng lớn dữ liệu, thường được sử dụng khi cần lưu trữ dữ liệu có kích thước lớn hơn text thông thường.
* **exception (longtext):**
* Cột này lưu trữ thông tin về ngoại lệ (exception) hoặc lỗi xảy ra trong quá trình xử lý nhiệm vụ. Nếu một tác vụ gặp lỗi, thông tin lỗi sẽ được lưu trong cột này để tiện cho việc kiểm tra và khắc phục.
* longtext giúp lưu trữ dữ liệu ngoại lệ có thể rất lớn, bao gồm các thông báo lỗi và stack trace.
* **failed\_at (timestamp):**
* Dấu thời gian ghi nhận thời điểm khi tác vụ gặp thất bại.
* timestamp lưu trữ thời gian chính xác để có thể theo dõi khi lỗi xảy ra và hỗ trợ xử lý lại tác vụ (nếu cần).

### **4. Table Friend\_requests**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint unsigned | Primary Key |
| sender\_id | bigint unsigned |  |
| receiver\_id | bigint unsigned |  |
| status | enum |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table* ***Friend\_requests***

* **id (bigint unsigned):**
* Đây là khóa chính của bảng, được dùng để định danh duy nhất mỗi bản ghi.
* Kiểu dữ liệu bigint unsigned nghĩa là nó chứa số nguyên lớn không âm, thường được tự động tăng dần (AUTO\_INCREMENT) để đảm bảo giá trị duy nhất cho mỗi bản ghi mới.
* **sender\_id (bigint unsigned):**
* Cột này lưu trữ ID của người gửi trong mối quan hệ giữa hai người dùng.
* bigint unsigned chứa giá trị ID của người dùng gửi, thường là khóa ngoại để tham chiếu tới bảng người dùng (users).
* **receiver\_id (bigint unsigned):**
* Cột này lưu trữ ID của người nhận trong mối quan hệ.
* bigint unsigned chứa giá trị ID của người dùng nhận, cũng có thể là khóa ngoại tham chiếu đến bảng người dùng.
* **status (enum):**
* Cột này lưu trữ trạng thái của mối quan hệ hoặc trạng thái giao tiếp giữa người gửi và người nhận.
* Kiểu dữ liệu enum cho phép xác định một tập giá trị cố định. Ví dụ, giá trị có thể là:
  + "pending": Yêu cầu đang chờ được chấp nhận.
  + "accepted": Yêu cầu đã được chấp nhận.
  + "rejected": Yêu cầu bị từ chối.
  + "blocked": Người dùng đã bị chặn.
* Giúp theo dõi tình trạng của mối quan hệ hoặc yêu cầu kết nối.
* **created\_at (timestamp):**
* Cột này ghi lại thời điểm bản ghi được tạo.
* timestamp lưu trữ thời gian và ngày tháng chính xác khi yêu cầu hoặc mối quan hệ được tạo ra.
* **updated\_at (timestamp):**
* Cột này ghi lại thời điểm bản ghi được cập nhật lần cuối.
* timestamp cho phép theo dõi những thay đổi hoặc cập nhật trạng thái của mối quan hệ (ví dụ: khi một yêu cầu được chấp nhận hoặc bị từ chối).

### **5. Table Oauth\_auth\_code**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | varchar | Primary Key |
| user\_id | varchar |  |
| client\_id | varchar |  |
| scopes | varchar |  |
| revoked | text |  |
| expires\_at | double |  |

*Bảng định nghĩa table* ***Oauth\_auth\_code***

* **id (varchar):**
* Đây là khóa chính của bảng, dùng để định danh duy nhất cho mỗi bản ghi.
* Kiểu dữ liệu varchar chứa chuỗi ký tự, có thể là mã token hoặc giá trị định danh duy nhất của một phiên làm việc hoặc một cấp quyền.
* **user\_id (varchar):**
* Cột này lưu trữ ID của người dùng liên quan đến bản ghi.
* varchar chứa chuỗi ký tự, đại diện cho ID của người dùng trong hệ thống, thường là khóa ngoại để liên kết với bảng người dùng (users).
* **client\_id (varchar):**
* Cột này lưu trữ ID của client (ứng dụng hoặc dịch vụ sử dụng hệ thống cấp quyền).
* varchar chứa chuỗi ký tự, đại diện cho ID của client, giúp hệ thống biết được ứng dụng nào đang yêu cầu hoặc sử dụng dữ liệu người dùng.
* **scopes (varchar):**
* Cột này lưu trữ phạm vi quyền truy cập (scopes) mà client được phép thực hiện đối với tài khoản người dùng.
* varchar chứa chuỗi các phạm vi quyền truy cập, ví dụ như read, write, hoặc các phạm vi cụ thể khác mà client có thể thực hiện trên tài nguyên của người dùng.
* **revoked (text):**
* Cột này lưu trữ trạng thái của quyền truy cập, cho biết liệu quyền truy cập đã bị thu hồi hay chưa.
* text có thể chứa các giá trị như "true" hoặc "false" để chỉ định quyền truy cập còn hiệu lực hay đã bị thu hồi.
* **expires\_at (double):**
* Cột này lưu trữ thời điểm hết hạn của quyền truy cập hoặc token.
* Kiểu dữ liệu double được sử dụng để lưu trữ dấu thời gian (timestamp) dưới dạng số thập phân, có thể đại diện cho thời gian hết hạn token hoặc quyền truy cập, thường là dưới dạng số giây kể từ một mốc thời gian (Unix timestamp).

### **6. Table Oauth\_access\_tokens**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | varchar | Primary Key |
| user\_id | bigint unsigned |  |
| client\_id | bigint unsigned |  |
| name | varchar |  |
| scopes | text |  |
| revoked | tinyint |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |
| expires\_at | datetime |  |

*Bảng định nghĩa table* ***Oauth\_access\_tokens***

* **id (varchar):**
* Đây là khóa chính của bảng, dùng để định danh duy nhất cho mỗi bản ghi.
* varchar chứa chuỗi ký tự, có thể là mã định danh hoặc token để nhận diện quyền truy cập hoặc phiên làm việc.
* **user\_id (bigint unsigned):**
* Cột này lưu trữ ID của người dùng liên quan đến bản ghi.
* bigint unsigned chứa giá trị số nguyên không âm, đại diện cho ID của người dùng trong hệ thống và thường là khóa ngoại liên kết với bảng người dùng (users).
* **client\_id (bigint unsigned):**
* Cột này lưu trữ ID của client (ứng dụng hoặc dịch vụ sử dụng hệ thống cấp quyền).
* bigint unsigned chứa giá trị số nguyên không âm, đại diện cho ID của client, dùng để xác định ứng dụng nào đang yêu cầu quyền truy cập.
* **name (varchar):**
* Cột này lưu trữ tên của phiên làm việc hoặc token, có thể là mô tả ngắn gọn về phiên hoặc quyền truy cập.
* varchar chứa chuỗi ký tự, có thể chứa tên tùy ý do người dùng hoặc hệ thống đặt.
* **scopes (text):**
* Cột này lưu trữ phạm vi quyền truy cập (scopes), tức là các quyền cụ thể mà token hoặc phiên làm việc được cấp phép để truy cập.
* text chứa chuỗi các phạm vi quyền truy cập như read, write, hoặc các quyền khác.
* **revoked (tinyint):**
* Cột này lưu trữ trạng thái quyền truy cập có bị thu hồi hay không.
* tinyint thường lưu giá trị 0 hoặc 1, với 0 nghĩa là quyền truy cập vẫn còn hiệu lực, và 1 nghĩa là quyền đã bị thu hồi (revoked).
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo của bản ghi hoặc phiên làm việc.
* timestamp lưu dấu thời gian khi token hoặc phiên làm việc được tạo.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời gian cập nhật lần cuối của bản ghi.
* timestamp lưu dấu thời gian khi token hoặc phiên làm việc được cập nhật gần nhất, ví dụ như khi thay đổi phạm vi quyền truy cập hoặc khi bị thu hồi.
* **expires\_at (datetime):**
* Cột này lưu trữ thời gian hết hạn của token hoặc phiên làm việc.
* datetime chứa thời gian hết hạn dưới dạng ngày giờ cụ thể, cho biết khi nào quyền truy cập hoặc phiên làm việc không còn hợp lệ.

### **7. Table Users**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint unsigned | Primary Key |
| full\_name | varchar |  |
| email | varchar |  |
| email\_verified\_at | timestamp |  |
| password | varchar |  |
| remember\_token | varchar |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |
| avatar | varchar |  |
| description | varchar |  |
| cover\_image | varchar |  |
| phone | int |  |
| date\_of\_birth | date |  |
| gender | enum |  |
| google\_id | varchar |  |
| social\_type | varchar |  |

*Bảng định nghĩa table Users*

* **id (bigint unsigned):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi người dùng.
* bigint unsigned lưu trữ số nguyên không âm, thường được tự động tăng (AUTO\_INCREMENT).
* **full\_name (varchar):**
* Cột này lưu trữ tên đầy đủ của người dùng.
* varchar chứa chuỗi ký tự đại diện cho tên người dùng.
* **email (varchar):**
* Cột này lưu trữ địa chỉ email của người dùng.
* varchar chứa chuỗi ký tự cho email của người dùng, thường là duy nhất để xác thực và liên lạc.
* **email\_verified\_at (timestamp):**
* Cột này lưu trữ thời điểm xác thực email của người dùng.
* timestamp chứa dấu thời gian khi email được xác nhận là hợp lệ.
* **password (varchar):**
* Cột này lưu trữ mật khẩu của người dùng (đã được mã hóa).
* varchar chứa chuỗi ký tự đại diện cho mật khẩu đã mã hóa.
* **remember\_token (varchar):**
* Cột này lưu trữ token ghi nhớ đăng nhập của người dùng.
* varchar chứa một chuỗi ký tự được sử dụng để duy trì phiên làm việc của người dùng mà không cần đăng nhập lại.
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo tài khoản của người dùng.
* timestamp chứa dấu thời gian khi người dùng đăng ký.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời gian cập nhật lần cuối của thông tin tài khoản người dùng.
* timestamp lưu dấu thời gian khi thông tin tài khoản được chỉnh sửa.
* **avatar (varchar):**
* Cột này lưu trữ URL hoặc đường dẫn đến ảnh đại diện của người dùng.
* varchar chứa chuỗi ký tự mô tả đường dẫn tới hình ảnh của người dùng.
* **description (varchar):**
* Cột này lưu trữ mô tả ngắn về người dùng (chẳng hạn như tiểu sử).
* varchar chứa chuỗi ký tự cho phần mô tả cá nhân.
* **cover\_image (varchar):**
* Cột này lưu trữ URL hoặc đường dẫn tới ảnh bìa của người dùng.
* varchar chứa chuỗi ký tự đại diện cho đường dẫn tới ảnh bìa.
* **phone (int):**
* Cột này lưu trữ số điện thoại của người dùng.
* int chứa giá trị số nguyên đại diện cho số điện thoại.
* **date\_of\_birth (date):**
* Cột này lưu trữ ngày sinh của người dùng.
* date chứa giá trị ngày tháng đại diện cho ngày sinh.
* **gender (enum):**
* Cột này lưu trữ giới tính của người dùng.
* enum chứa một danh sách các giá trị có thể chọn như "male", "female", "other", hoặc "prefer\_not\_to\_say".
* **google\_id (varchar):**
* Cột này lưu trữ ID của Google nếu người dùng đăng nhập bằng tài khoản Google.
* varchar chứa chuỗi ký tự đại diện cho ID người dùng trong Google.
* **social\_type (varchar):**
* Cột này lưu trữ loại mạng xã hội mà người dùng sử dụng để đăng nhập, chẳng hạn như "google", "facebook", "twitter", v.v.
* varchar chứa chuỗi ký tự mô tả mạng xã hội được dùng để đăng nhập.

### **8. Oauth\_personal\_access\_clients**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| client\_id | bigint |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table oauth\_personal\_access\_clients*

* **id (bigint):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi bản ghi.
* bigint chứa giá trị số nguyên lớn, thường được sử dụng với AUTO\_INCREMENT để tạo các giá trị duy nhất cho mỗi bản ghi.
* **client\_id (bigint):**
* Cột này lưu trữ ID của client liên quan đến bản ghi.
* bigint chứa giá trị số nguyên lớn, đại diện cho client (ứng dụng hoặc dịch vụ) sử dụng hệ thống.
* **created\_at (timestamp):**
* Cột này lưu trữ thời điểm tạo của bản ghi.
* timestamp chứa dấu thời gian khi bản ghi được tạo.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời điểm cập nhật lần cuối của bản ghi.
* timestamp chứa dấu thời gian khi bản ghi được chỉnh sửa hoặc cập nhật lần cuối.

### **9. oauth\_refresh\_tokens**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| Id | varchar | Primary Key |
| access\_token\_id | varchar |  |
| Revoked | tinyint |  |
| expires\_at | datetime |  |

*Bảng định nghĩa table oauth\_refresh\_tokens*

* **id (varchar):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi bản ghi.
* varchar chứa chuỗi ký tự, có thể là một mã định danh hoặc token duy nhất cho mỗi phiên hoặc quyền truy cập.
* **access\_token\_id (varchar):**
* Cột này lưu trữ ID của access token liên quan đến bản ghi.
* varchar chứa chuỗi ký tự đại diện cho ID của access token, thường dùng để liên kết với bảng quản lý token hoặc phiên truy cập.
* **revoked (tinyint):**
* Cột này lưu trữ trạng thái thu hồi của token.
* tinyint chứa giá trị 0 hoặc 1, trong đó 0 nghĩa là token vẫn hợp lệ, và 1 nghĩa là token đã bị thu hồi (revoked).
* **expires\_at (datetime):**
* Cột này lưu trữ thời gian hết hạn của token.
* datetime chứa ngày và giờ cụ thể, chỉ định thời điểm token hết hiệu lực.

### **10. Password\_Reset Token**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| email | varchar | Primary Key |
| token | timestamp |  |
| created\_at | timestamp |  |

*Bảng định nghĩa table Password reset Token*

* **email (varchar):**
* Cột này lưu trữ địa chỉ email của người dùng.
* varchar chứa chuỗi ký tự đại diện cho email, thường được sử dụng để xác thực và liên lạc với người dùng.
* **token (timestamp):**
* Cột này lưu trữ token xác thực (hoặc mã) liên quan đến một hành động cụ thể, chẳng hạn như xác thực địa chỉ email hoặc đặt lại mật khẩu.
* Có thể cần làm rõ về kiểu dữ liệu: nếu đây là token, thường sẽ là kiểu varchar thay vì timestamp. Nếu ý định là lưu thời gian token được tạo, thì cần đổi lại tên cột cho phù hợp.
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo của bản ghi hoặc token.
* timestamp chứa dấu thời gian khi bản ghi được tạo, giúp theo dõi thời điểm bắt đầu hiệu lực của token.

### **11. Messages**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| message | text |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |
| sender\_id | bigint |  |
| receiver\_id | bigint |  |

*Bảng định nghĩa table messages*

* **id (bigint):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi tin nhắn.
* bigint chứa giá trị số nguyên lớn, thường được sử dụng với AUTO\_INCREMENT để tạo ra các giá trị duy nhất cho mỗi bản ghi.
* **message (text):**
* Cột này lưu trữ nội dung của tin nhắn.
* text chứa chuỗi ký tự dài, cho phép lưu trữ tin nhắn có độ dài biến đổi, từ ngắn đến dài.
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo của tin nhắn.
* timestamp chứa dấu thời gian khi tin nhắn được gửi, cho phép theo dõi thời điểm gửi tin nhắn.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời gian cập nhật lần cuối của tin nhắn.
* timestamp chứa dấu thời gian khi tin nhắn được chỉnh sửa (nếu có), giúp theo dõi lịch sử sửa đổi của tin nhắn.
* **sender\_id (bigint):**
* Cột này lưu trữ ID của người gửi tin nhắn.
* bigint chứa giá trị số nguyên lớn, thường là khóa ngoại liên kết với bảng người dùng, cho phép xác định ai đã gửi tin nhắn.
* **receiver\_id (bigint):**
* Cột này lưu trữ ID của người nhận tin nhắn.
* bigint chứa giá trị số nguyên lớn, thường là khóa ngoại liên kết với bảng người dùng, cho phép xác định ai là người nhận tin nhắn.

### **12.Personal\_access\_tokens**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| tokenable\_type | varchar |  |
| tokenable\_id | bigint |  |
| name | varchar |  |
| token | varchar |  |
| abilities | text |  |
| last\_used\_at | timestamp |  |
| expires\_at | timestamp |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table Personal\_Access\_Tokens*

* **id (bigint):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi bản ghi.
* bigint chứa giá trị số nguyên lớn, thường được sử dụng với AUTO\_INCREMENT để tạo ra các giá trị duy nhất cho mỗi bản ghi.
* **tokenable\_type (varchar):**
* Cột này lưu trữ kiểu đối tượng mà token này thuộc về.
* varchar chứa chuỗi ký tự mô tả loại đối tượng, có thể là tên của mô hình (model) trong hệ thống như "User", "Admin", v.v.
* **tokenable\_id (bigint):**
* Cột này lưu trữ ID của đối tượng mà token này thuộc về.
* bigint chứa giá trị số nguyên lớn, thường là khóa ngoại liên kết với bảng tương ứng, cho phép xác định đối tượng cụ thể mà token đại diện.
* **name (varchar):**
* Cột này lưu trữ tên của token.
* varchar chứa chuỗi ký tự đại diện cho tên hoặc mô tả ngắn gọn về token, có thể hữu ích cho việc quản lý và phân loại các token khác nhau.
* **token (varchar):**
* Cột này lưu trữ giá trị của token.
* varchar chứa chuỗi ký tự, thường là một mã duy nhất được sử dụng để xác thực người dùng hoặc đối tượng trong hệ thống.
* **abilities (text):**
* Cột này lưu trữ các quyền truy cập mà token này có.
* text chứa chuỗi mô tả các quyền hoặc hành động mà token được phép thực hiện, có thể là các phạm vi quyền (scopes) hoặc các quyền truy cập khác.
* **last\_used\_at (timestamp):**
* Cột này lưu trữ thời gian sử dụng cuối cùng của token.
* timestamp chứa dấu thời gian khi token được sử dụng lần cuối, giúp theo dõi mức độ sử dụng của token.
* **expires\_at (timestamp):**
* Cột này lưu trữ thời gian hết hạn của token.
* timestamp chứa dấu thời gian cho biết khi nào token sẽ không còn hiệu lực.
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo của token.
* timestamp chứa dấu thời gian khi token được phát hành.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời gian cập nhật lần cuối của token.
* timestamp chứa dấu thời gian khi thông tin của token được chỉnh sửa.

### **13. Websockets\_statistícs\_entries**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | int | Primary Key |
| app\_id | varchar |  |
| peak\_connection\_count | int |  |
| websocket\_message\_count | int |  |
| api\_message\_count | int |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table Websockets statistics entries*

* **id (int):**
* Đây là khóa chính của bảng, định danh duy nhất cho mỗi bản ghi.
* int chứa giá trị số nguyên, thường được sử dụng với AUTO\_INCREMENT để tạo ra các giá trị duy nhất cho mỗi bản ghi.
* **app\_id (varchar):**
* Cột này lưu trữ ID của ứng dụng liên quan đến bản ghi.
* varchar chứa chuỗi ký tự đại diện cho ID của ứng dụng, có thể được sử dụng để xác định ứng dụng cụ thể trong hệ thống.
* **peak\_connection\_count (int):**
* Cột này lưu trữ số lượng kết nối tối đa mà ứng dụng đạt được tại một thời điểm nhất định.
* int chứa giá trị số nguyên, cho phép theo dõi hiệu suất của ứng dụng trong việc quản lý kết nối.
* **websocket\_message\_count (int):**
* Cột này lưu trữ số lượng tin nhắn được gửi qua WebSocket trong khoảng thời gian nhất định.
* int chứa giá trị số nguyên, cho phép đo lường mức độ tương tác của người dùng với ứng dụng thông qua giao thức WebSocket.
* **api\_message\_count (int):**
* Cột này lưu trữ số lượng tin nhắn được gửi qua API trong khoảng thời gian nhất định.
* int chứa giá trị số nguyên, giúp theo dõi mức độ sử dụng API của ứng dụng.
* **created\_at (timestamp):**
* Cột này lưu trữ thời gian tạo của bản ghi.
* timestamp chứa dấu thời gian khi bản ghi được tạo, giúp theo dõi thời điểm mà thông tin được lưu trữ.
* **updated\_at (timestamp):**
* Cột này lưu trữ thời gian cập nhật lần cuối của bản ghi.
* timestamp chứa dấu thời gian khi bản ghi được chỉnh sửa hoặc cập nhật lần cuối.

### **14. group\_messages**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| group\_id | bigint |  |
| user\_id | bigint |  |
| content | text |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table group\_messages*

* **id (bigint):**
* Đây là khóa chính của bảng, chứa giá trị duy nhất cho mỗi bản ghi. Sử dụng kiểu bigint để lưu trữ số nguyên lớn và unsigned để không chứa số âm, giúp tối ưu hóa phạm vi lưu trữ.
* **group\_id (bigint):**
* Cột này liên kết với một nhóm (có thể là khóa ngoại tham chiếu đến bảng nhóm), xác định thông điệp thuộc về nhóm nào. Kiểu bigint unsigned cho phép lưu trữ các giá trị nhóm với kích thước lớn.
* **user\_id (bigint):**
* Xác định người dùng đã gửi tin nhắn trong nhóm, cũng có thể là khóa ngoại tham chiếu đến bảng người dùng. Giống như group\_id, nó sử dụng bigint unsigned để hỗ trợ các ID người dùng có kích thước lớn.
* **content (text):**
* Chứa nội dung tin nhắn. Sử dụng kiểu text để lưu trữ các chuỗi ký tự dài, phù hợp với các tin nhắn văn bản có kích thước lớn.
* **created\_at (timestamp):**
* Lưu trữ thời điểm tin nhắn được tạo ra (thời gian gửi tin nhắn). Giá trị này thường tự động được điền khi tạo bản ghi mới.
* **updated\_at (timestamp):**
* Lưu trữ thời điểm lần cuối cùng tin nhắn được cập nhật. Thông thường, giá trị này được cập nhật tự động mỗi khi có thay đổi đối với bản ghi.

### **15. group\_message\_statuses**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| group\_message\_id | bigint |  |
| user\_id | bigint |  |
| status | enum |  |
| read\_at | timestamp |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table group\_message\_statuses*

* **id (bigint):**
* Khóa chính của bảng, chứa giá trị duy nhất cho mỗi bản ghi. Được sử dụng để định danh mỗi trạng thái của tin nhắn trong nhóm. Kiểu bigint với unsigned giúp lưu trữ các giá trị lớn và không âm.
* **group\_message\_id (bigint):**
* Khóa ngoại liên kết với bảng group\_messages, xác định trạng thái của một tin nhắn cụ thể trong nhóm. Dùng kiểu bigint unsigned để lưu trữ ID của tin nhắn nhóm.
* **user\_id (bigint):**
* Xác định người dùng liên quan đến trạng thái đọc/không đọc của tin nhắn. Dùng kiểu bigint unsigned để lưu trữ ID người dùng và thường liên kết với bảng người dùng.
* **status (enum):**
* Trạng thái của tin nhắn, chỉ định liệu người dùng đã đọc tin nhắn hay chưa. Sử dụng kiểu dữ liệu enum để giới hạn các giá trị hợp lệ là 'read' (đã đọc) hoặc 'unread' (chưa đọc).
* **read\_at (timestamp):**
* Lưu trữ thời điểm người dùng đọc tin nhắn. Nếu tin nhắn chưa được đọc, trường này sẽ có thể trống hoặc null.
* **created\_at (timestamp):**
* Thời gian bản ghi trạng thái được tạo. Giá trị này thường được điền tự động khi bản ghi mới được tạo.
* **updated\_at (timestamp):**
* Thời gian trạng thái được cập nhật lần cuối. Giá trị này tự động được điền mỗi khi có thay đổi đối với bản ghi.

### **16. groups**

| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| name | bigint |  |
| created\_by | bigint |  |
| created\_at | enum |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table group*

* **id (bigint):**
* Đây là khóa chính của bảng, dùng để định danh duy nhất mỗi nhóm. Được khai báo với kiểu bigint unsigned để lưu trữ các giá trị lớn và không âm.
* **name (varchar):**
* Tên của nhóm. Kiểu varchar(255) giới hạn độ dài của tên nhóm tối đa là 255 ký tự.
* **created\_by (bigint):**
* Lưu trữ ID của người đã tạo ra nhóm. Đây có thể là khóa ngoại liên kết đến bảng người dùng, xác định ai là người tạo nhóm.
* **created\_at (timestamp):**
* Thời gian nhóm được tạo. Giá trị này thường được điền tự động khi bản ghi nhóm mới được tạo.
* **updated\_at (timestamp):**
* Thời gian lần cuối cùng thông tin nhóm được cập nhật. Tự động cập nhật khi có bất kỳ thay đổi nào đối với bản ghi nhóm.

### **17. group\_users**

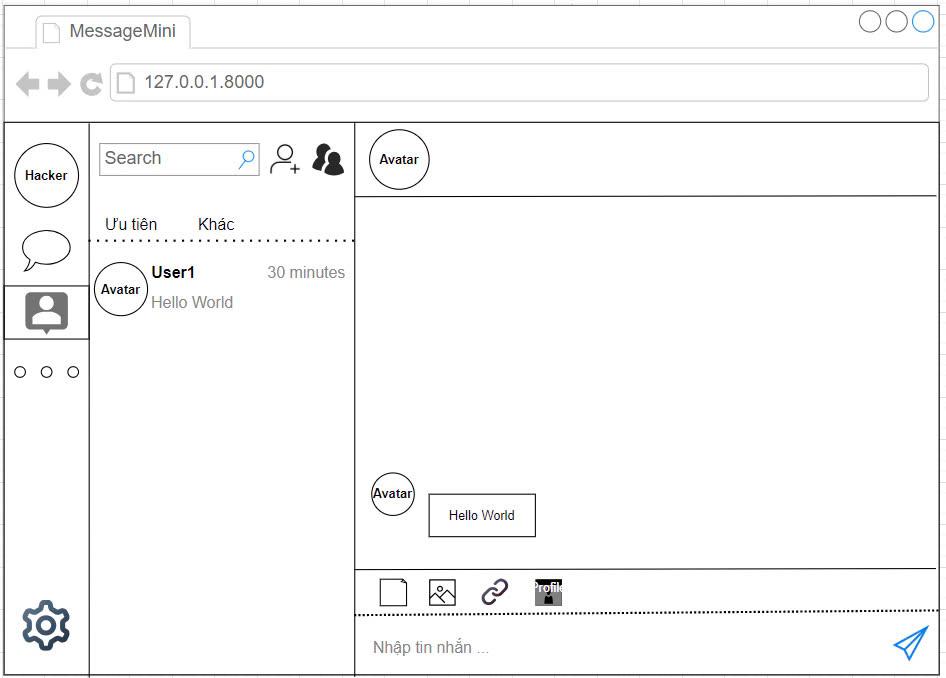
| **Field Name** | **Type** | **Constraint** |
| --- | --- | --- |
| id | bigint | Primary Key |
| group\_id | bigint |  |
| user\_id | bigint |  |
| role | enum |  |
| created\_at | timestamp |  |
| updated\_at | timestamp |  |

*Bảng định nghĩa table group\_users*

* **Id (bigint):**
* Trường này là khóa chính của bảng, giúp phân biệt các bản ghi khác nhau.
* **group\_id (bigint):**
* Trường này liên kết người dùng với một nhóm cụ thể, cho phép xác định nhóm mà người dùng đang tham gia.
* **user\_id (bigint):**
* Trường này xác định người dùng cụ thể trong hệ thống.
* **role (enum):**
* Trường này xác định vai trò của người dùng trong nhóm, có thể là:
* gold: Vai trò cao nhất với quyền hạn đặc biệt.
* silver: Vai trò trung bình với quyền hạn nhất định.
* member: Vai trò cơ bản với quyền hạn hạn chế.
* **created\_at (timestamp):**
* Trường này lưu trữ thời điểm bản ghi được tạo, giúp theo dõi lịch sử.
* **updated\_at (timestamp):**
* Trường này lưu trữ thời điểm bản ghi được cập nhật lần cuối, có thể dùng để theo dõi sự thay đổi trong mối quan hệ giữa người dùng và nhóm.

# **PHẦN 4: HÌNH ẢNH THIẾT KẾ CỦA DỰ ÁN**

## **I: Trang trò chuyện cùng với bạn bè**



*Hình 2:Trò chuyện*

Mô tả:

- Cho phép người dùng có thể trò chuyện và tương tác với nhau, người dùng còn có thể gửi file, hình ảnh hoặc tệp đính kèm, tin nhắn được gửi sẽ hiển thị ngay lập tức ở đây, thời gian nhận được tin nhắn cũng hiện ở đây.

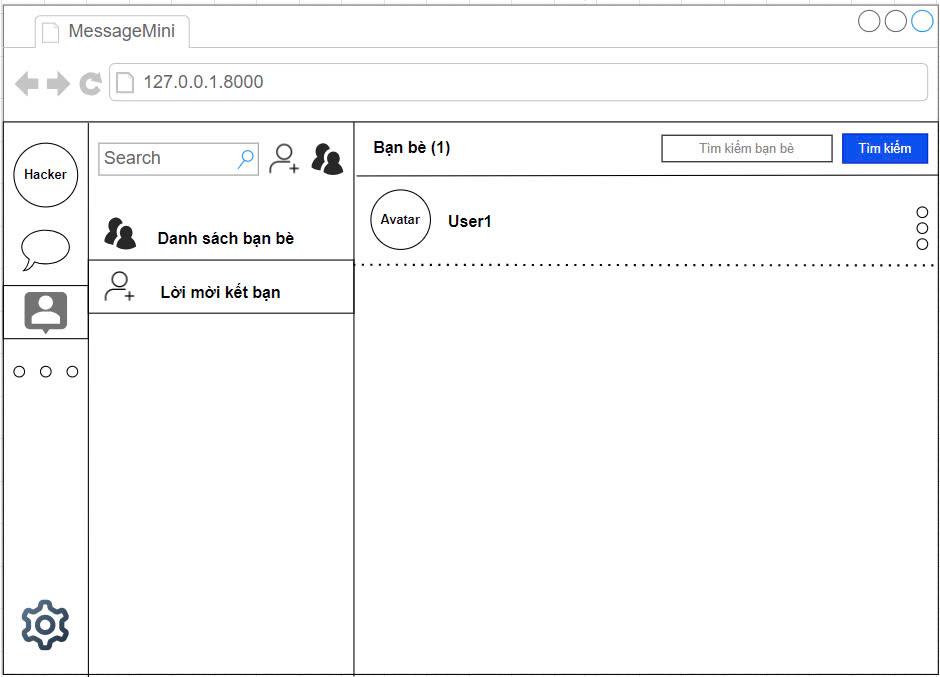
- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút thêm bạn: Click vào để kết bạn  
- Hacker: Ảnh đại diện

- Nút gửi file: Bấm vào để chọn file để gửi  
- Nút gửi ảnh: Bấm vào để chọn ảnh để gửi

- Thanh nhập tin nhắn: Bấm vào để soạn tin nhắn văn bản vào

- Hell World: tin nhắn mà người dùng gửi gần nhất

## **II. Trang danh sách bạn bè**

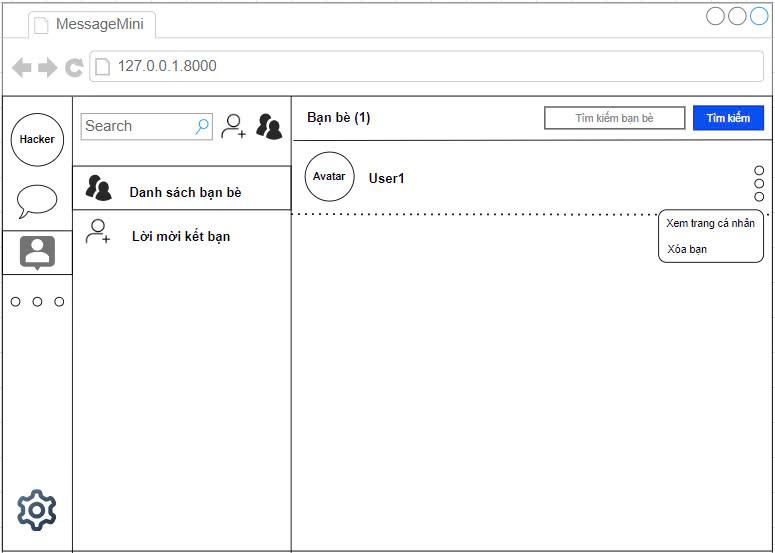


*Hình 3: Danh sách bạn bè*

Mô tả:

- Ở đây sẽ hiển thị danh sách các bạn bè có thể nhắn tin  
- Tìm kiếm bạn bè theo tên  
- Xem trang cá nhân của họ  
- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút thêm bạn: Click vào để kết bạn  
- Hacker: Ảnh đại diện   
- Lời mời kết bạn: Bấm vào để xem lời mời kết bạn đã nhận

## **III. Trang cài đặt người dùng**



*Hình 4: Cài đặt người dùng*

Mô tả:

- Khi nhấn tùy chọn sẽ hiện 2 mục cho người dùng chọn.

- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút xem trang cá nhân: bấm vào để xem trang cá nhân của người dùng

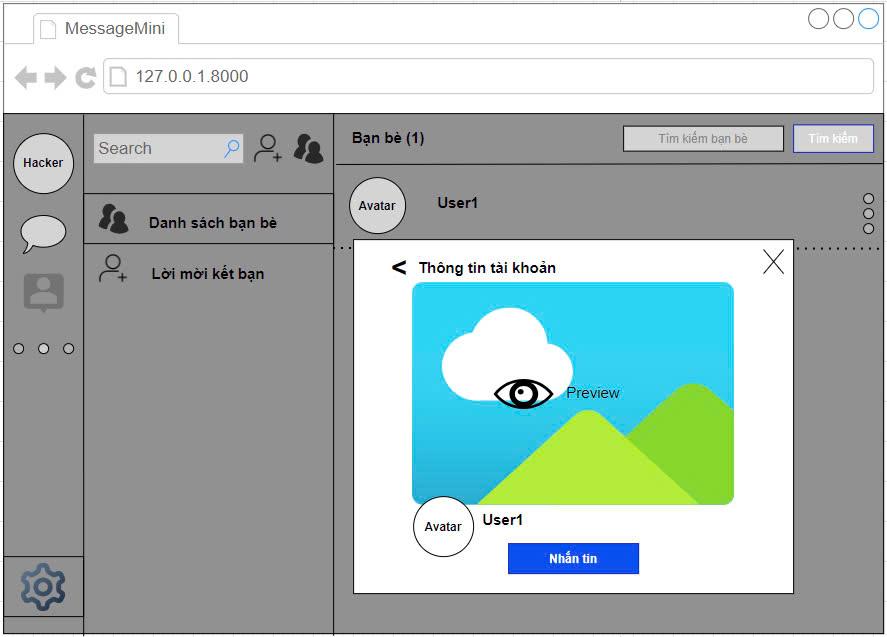
- Nút xóa bạn: Bấm để xóa người dùng khỏi danh sách bạn bè

- Hacker: Ảnh đại diện

- Thanh tìm kiếm bạn bè: Nhập tên bạn bè mà muốn tìm

- Nút tìm kiếm: Sau khi đã nhập rồi, bấm tìm kiếm

## **IV. Trang thông tin tài khoản bạn bè**



*Hình 5: Thông tin tài khoản bạn bè*

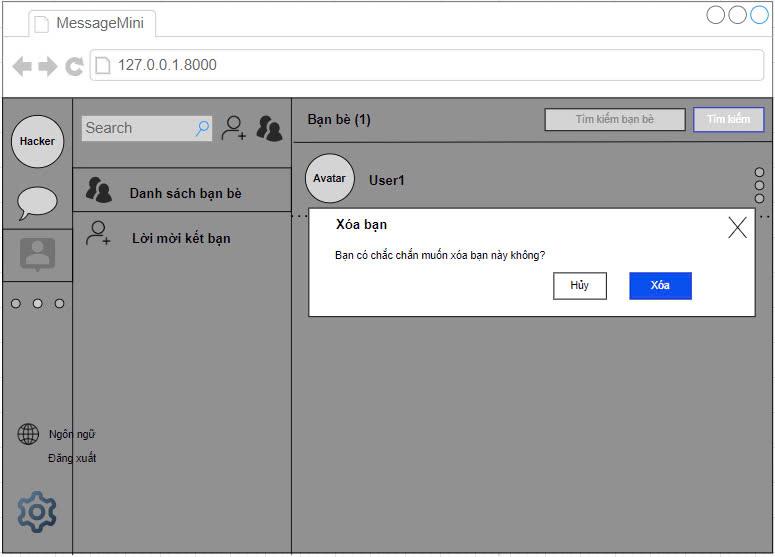
Mô tả:

- Khi người dùng click vào xem trang cá nhân sẽ hiện tên, ảnh đại diện, ảnh bìa và nút nhắn tin.

- Hình ảnh: Ảnh bìa của người dùng

- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút nhắn tin: Bấm vào để bắt đầu cuộc trò chuyện  
- Nút thêm bạn: Click vào để kết bạn  
- Hacker: Ảnh đại diện

## **V. Trang xóa bạn bè**



*Hình 6: Xóa bạn bè*

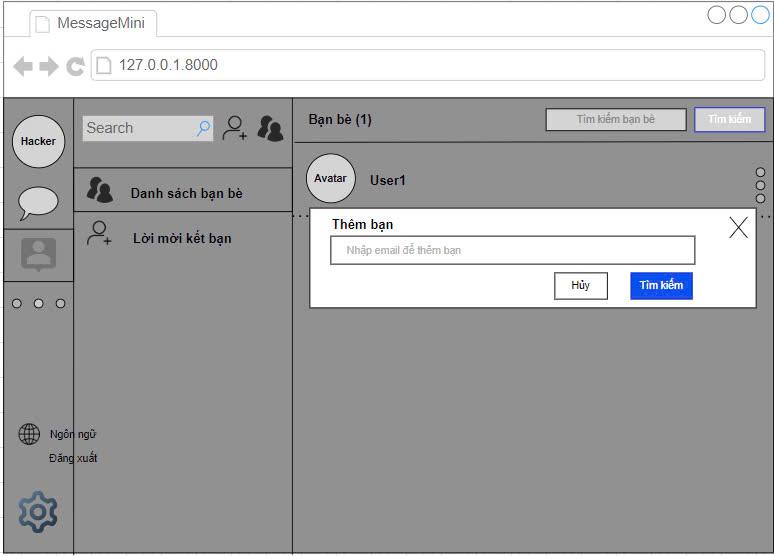
Mô tả:

- Khi bạn click vào nút xóa bạn sẽ hiện thông báo. Nếu click vào xóa sẽ không còn trong danh sách bạn bè còn hủy thì sẽ dừng hành động xóa.

- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Hacker: Ảnh đại diện

- Nút hủy: Bấm vào để hủy và không xóa bạn nữa  
- Xóa: Bấm vào để xóa bạn

## **VI. Trang tìm kiếm thêm bạn**



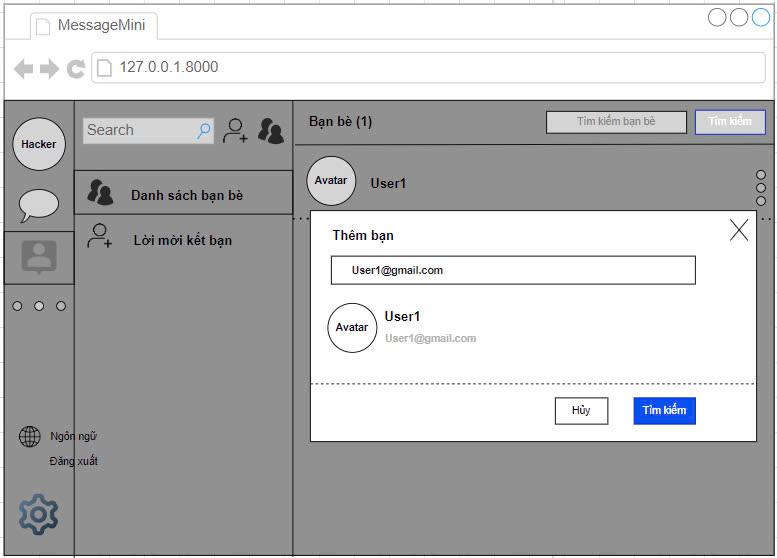
*Hình 7: Tìm kiếm thêm bạn*

Mô tả:

- Cho phép người dùng tìm kiếm bạn bè theo email đã đăng ký tài khoản  
- Bấm vào nút hủy sẽ thoát ra giao diện  
- Avatar: Hiển thị ảnh đại diện của người dùng

- Hacker: Ảnh đại diện   
- Nút hủy: Bấm vào để thoát khỏi thanh tìm kiếm  
- Nút tìm kiếm: Sau khi đã nhập thông tin, bấm để tìm

## **VII. Trang thêm bạn hiện thông tin khi nhấn tìm kiếm**



*Hình 8: Thêm bạn hiện thông tin khi nhấn tìm kiếm*

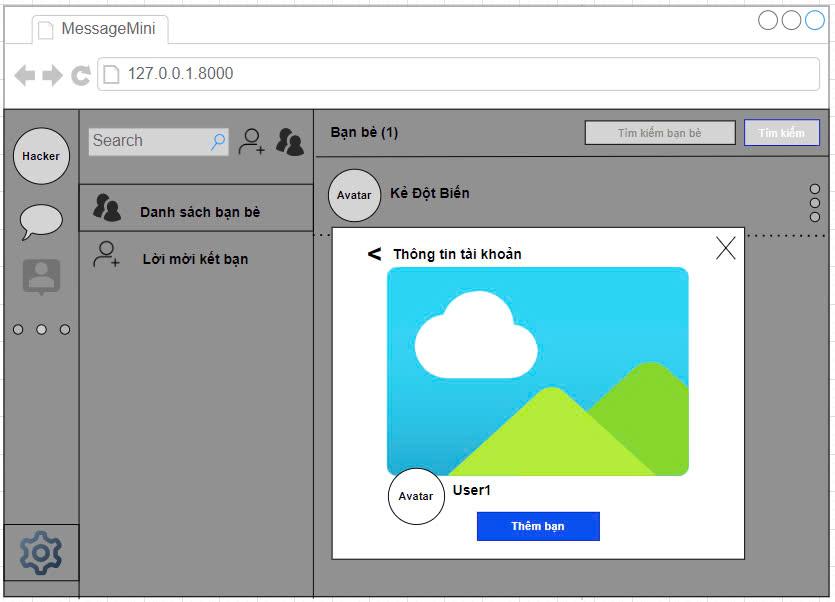
Mô tả:

- Sau khi nhập email và tìm kiếm sẽ hiện người dùng đăng ký bằng email đó.  
- Avatar: Hiển thị ảnh đại diện của người dùng

- Hủy: Bấm vào nút hủy sẽ thoát ra giao diện chính

- User: Hiển thị tên của người dùng  
- [User1@gmail.com](mailto:User1@gmail.com): hiển thị email của người dùng

## **VIII. Trang tìm kiếm người dùng**



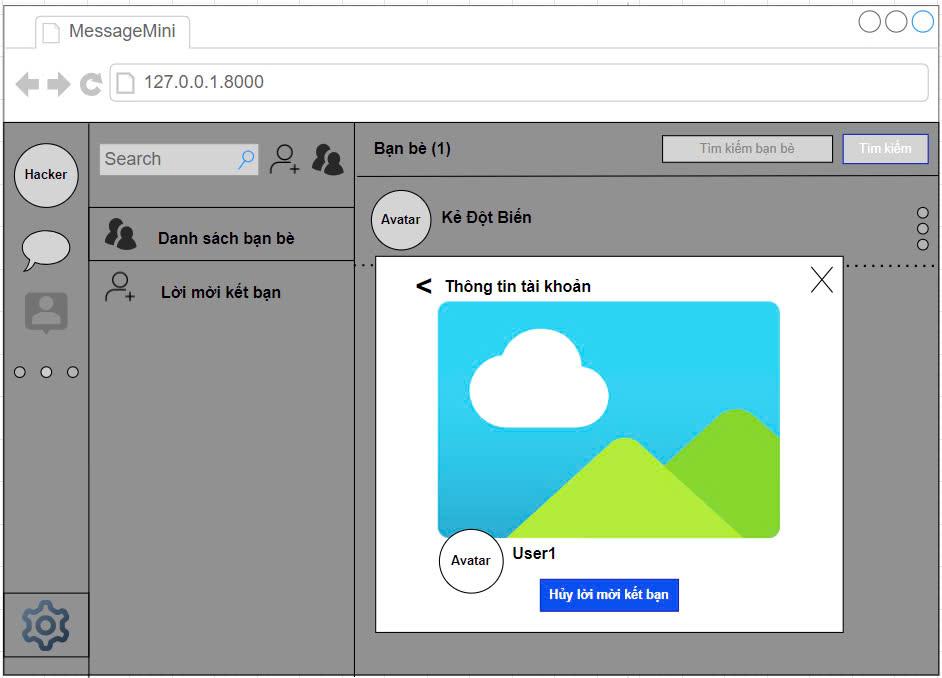
*Hình 9: Tìm kiếm người dùng*

Mô tả :

- Sau khi tìm kiếm bằng email sẽ hiện thông tin tài khoản cho phép bạn gửi lời mời kết bạn.

- Hình ảnh: Ảnh bìa của người dùng.  
- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút thêm bạn: Click vào để kết bạn  
- Hacker: Ảnh đại diện

## **IX. Trang hủy lời mời kết bạn**



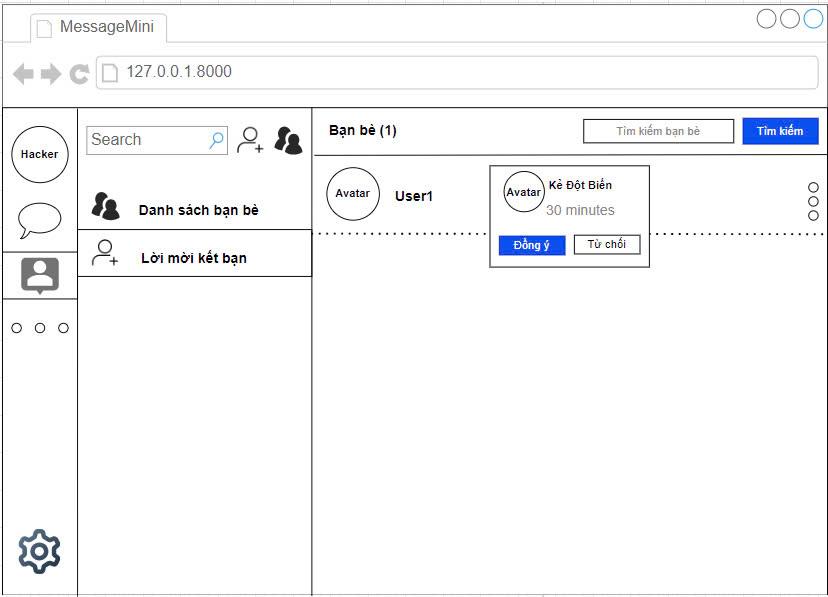
*Hình 10: Hủy lời mời kết bạn*

Mô tả:

- Sau khi nhấn thêm bạn người dùng có thể thu hồi lời mời kết bạn đã gửi trước đó, khi đối phương chưa đồng ý lời mời.

- Hình ảnh: Ảnh bìa của người dùng.  
- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Nút hủy kết bạn: Click vào để hủy lời mời  
- Hacker: Ảnh đại diện

## **X. Trang lời mời kết bạn**



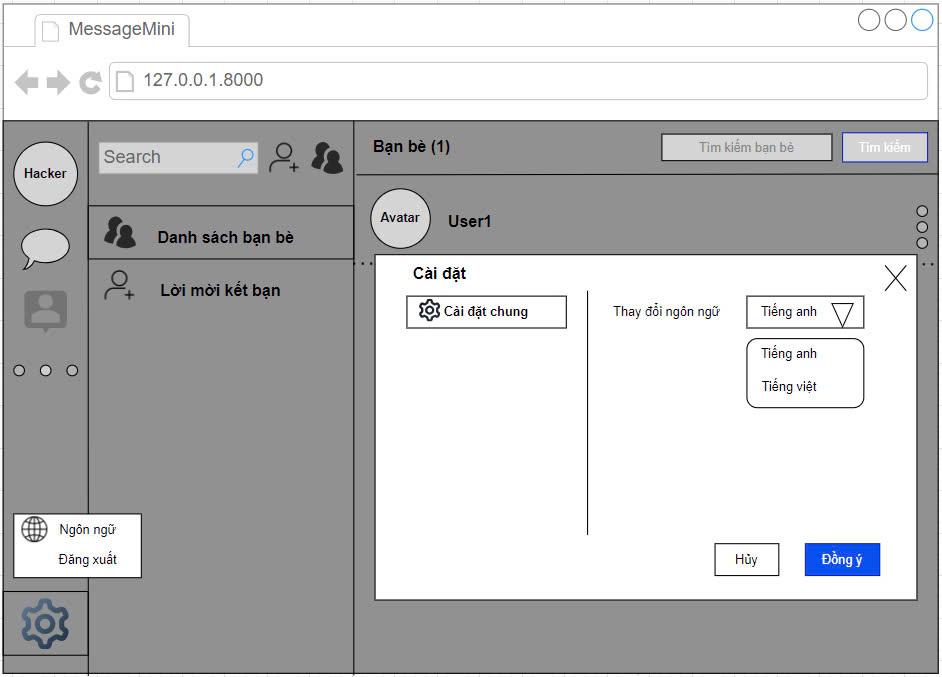
*Hình 11: Lời mời kết bạn*

Mô tả:

- Khi có người gửi lời mời kết bạn cho bạn sẽ hiện lên thông báo gửi lời mời kết bạn. Khi bạn click vào đồng ý người đó sẽ vào danh sách bạn bè, nếu bạn click vào từ chối thì thông báo sẽ biến mất.  
-

- Avatar: Ảnh đại diện của người đùng.  
- Kẻ đột biến: Tên của người dùng  
- Hacker: Ảnh đại diện   
- Nút đồng ý: Bấm vào sẽ đồng ý kết bạn  
- Nút từ chối: Bấm vào sẽ không đồng ý kết bạn  
- 30 minutes: Thời gian mà lời mời kết bạn đã gửi tới

## **XI: Trang cài đặt ngôn ngữ**



*Hình 12: Cài đặt ngôn ngữ*

Mô tả:

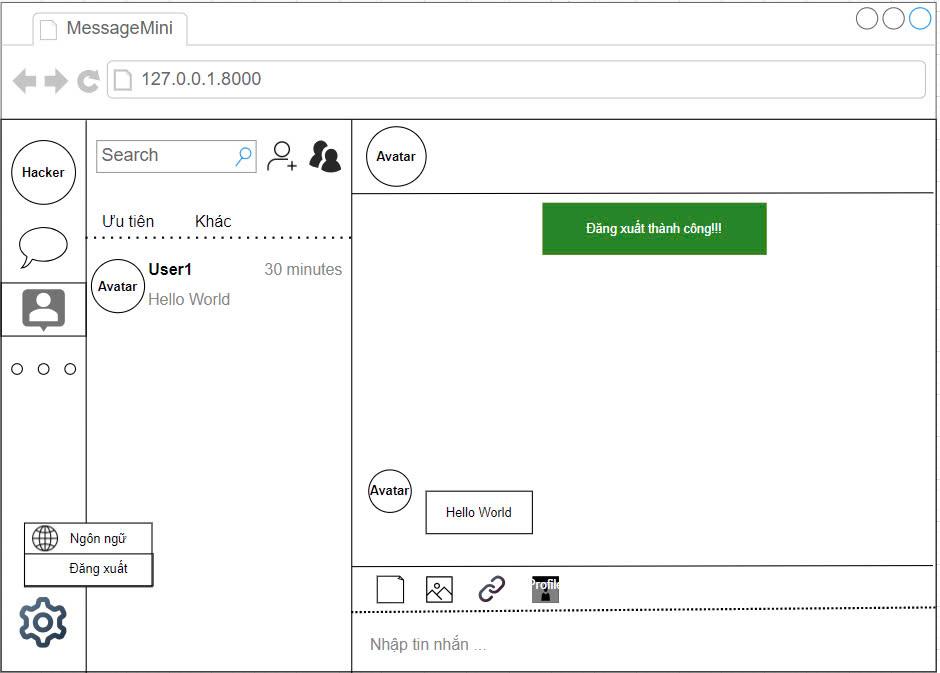
- Khi người dùng nhấn vào cài đặt => Nhấn vào ngôn ngữ sẽ hiện bảng lựa chọn ngôn ngữ cho phép người dùng đổi sang ngôn ngữ khác.

- Nút đăng xuất: Bấm vào để đăng xuất khỏi tài khoản  
- Nút ngôn ngữ: Bấm vào để thay đổi các ngôn ngữ

- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Hacker: Ảnh đại diện  
- Thay đổi ngôn ngữ: Có thể chọn ngôn ngữ tiếng anh hoặc tiếng Việt  
- Nút hủy: Bấm để thoát khỏi giao diện cài đặt

- Nút đồng ý: Khi đã chọn ngôn ngữ phù hợp bạn có thể bấm vào lưu thay đổi

## **XII: Trang đăng xuất**



*Hình 13 : Đăng xuất*

Mô tả :

Khi người dùng nhấn vào cài đặt => đăng xuất sẽ hiện thông báo và trở về trang đăng nhập lại.  
- Avatar: Ảnh đại diện của người đùng.  
- User1: Tên của người dùng  
- Hacker: Ảnh đại diện

- Nút đăng xuất:

# **PHẦN 5: TÀI LIỆU THAM KHẢO**

## **I. Docker:** <https://www.docker.com/>

## **II. Laravel framwork:** <https://laravel.com/>

## **III. Chat GPT**: <https://www.chatgpt.com/>

## **IV. Vuejs:** <https://vuejs.org/>

## **V. Stackoverflow:** <https://stackoverflow.com/>