

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA KỸ THUẬT – CÔNG NGHỆ



Sinh viên thực hiện
NGUYỄN TRUNG TIẾN
MSSV: 191783
LỚP: DH19TIN03

TÊN ĐỀ TÀI: WEBSITE CHAT REALTIME

BÁO CÁO THỰC TẬP TỐT NGHIỆP
Ngành: Công Nghệ Thông Tin
Mã số ngành: 7480201

Tháng 6 Năm 2023

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA KỸ THUẬT – CÔNG NGHỆ

Sinh viên thực hiện
NGUYỄN TRUNG TIẾN

TÊN ĐỀ TÀI: WEBSITE CHAT REALTIME

BÁO CÁO THỰC TẬP TỐT NGHIỆP

Ngành: Công Nghệ Thông Tin

Mã số ngành: 7480201

CÁN BỘ HƯỚNG DẪN
TRẦN THANH NAM

Tháng 6 Năm 2023

LỜI CẢM TẠ

Em xin chân thành cảm ơn quý Thầy Cô trong khoa Công nghệ thông tin đã trang bị những kiến thức quý báu cho em trong suốt quá trình học tập và làm đề tài tại Trường Đại học Nam Cần Thơ. Nhờ công lao giảng dạy, chỉ bảo tận tình của quý Thầy Cô mà em đã có được những kiến thức chuyên ngành về công nghệ thông tin để thực hiện đề tài này.

Em xin chân thành cảm ơn!

Cần thơ, ngày ... tháng ... năm 2023

Sinh viên thực hiện

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

Giảng viên hướng dẫn

[illegible]

Giảng viên phản biện

MỤC LỤC

| | |
|---|------|
| DANH MỤC HÌNH | vii |
| DANH MỤC BẢNG..... | viii |
| DANH MỤC TỪ VIẾT TẮT..... | ix |
| TỔNG QUAN CÔNG TY THỰC TẬP..... | 1 |
| CHƯƠNG 1: GIỚI THIỆU | 2 |
| 1.1 Lý do chọn đề tài | 2 |
| 1.2 Hướng tiếp cận đề tài..... | 2 |
| 1.3 Ưu và nhược điểm của đề tài | 3 |
| 1.3.1 Ưu điểm | 3 |
| 1.3.2 Nhược điểm | 4 |
| CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU | 5 |
| 2.1 Cơ sở lý luận và ý nghĩa thực tiễn của đề tài..... | 5 |
| 2.1.1 Cơ sở lý luận..... | 5 |
| 2.1.2 Ý nghĩa thực tiễn của đề tài..... | 5 |
| 2.2 Phương pháp nghiên cứu | 6 |
| 2.2.1 Giới thiệu về công nghệ được áp dụng | 6 |
| CHƯƠNG 3: GIỚI THIỆU TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU | 13 |
| 3.1 Tổng quan về hệ thống | 13 |
| 3.1.1 Mục tiêu..... | 13 |
| 3.1.2 Phạm vi ứng dụng..... | 14 |
| 3.1.3 KHẢ NĂNG PHÁT TRIỂN: | 14 |
| 3.2 Phân khảo sát hiện trạng | 15 |
| 3.2.1 Hiện trạng | 15 |
| 3.2.2 Phân tích yêu cầu..... | 16 |
| 3.2.3 Môi trường triển khai | 19 |
| CHƯƠNG 4: THIẾT KẾ CƠ SỞ DỮ LIỆU..... | 20 |
| 4.1 Các loại thực thể | 20 |
| 4.2 Mô hình quan hệ (ERD) | 22 |
| 4.3 Mô hình luồng dữ liệu phân cấp (DFD) | 22 |
| 4.4 Chuẩn hóa dữ liệu quan hệ | 24 |
| 4.5 Mô hình tổng thể kiến trúc..... | 31 |

| | |
|---|----|
| 4.5.1 Danh sách các Component/Package..... | 31 |
| 4.5.2 Giải thích tương tác, giao tiếp giữa các components | 31 |
| CHƯƠNG 5: ĐẶC TẢ GIAO DIỆN | 33 |
| 5.1 phần thiết kế giao diện cho user | 33 |
| 5.1.1 Thiết kế màn hình chính..... | 33 |
| 5.1.2 Thiết kế giao diện đăng ký | 33 |
| 5.1.3 Thiết kế giao diện đăng nhập admin với user | 34 |
| 5.1.4 Giao diện quên mật khẩu..... | 34 |
| 5.1.5 Giao diện gợi ý kết bạn | 35 |
| 5.1.6 Giao diện nhắn tin | 35 |
| 5.1.7 Giao diện quản lý thông tin tài khoản | 36 |
| 5.1.8 Giao diện thông báo | 36 |
| 5.2 Giao diện cho quản lý admin | 37 |
| 5.2.1 Giao diện quản lý user..... | 37 |
| CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN..... | 38 |
| TÀI LIỆU THAM KHẢO..... | 39 |

DANH MỤC HÌNH

| | |
|--|----|
| Hình 4.1 Mô hình quan hệ (ERD) | 22 |
| Hình 4.2 Mô hình DFD Mức 0..... | 22 |
| Hình 4.3 Mô hình DFD Mức 1..... | 23 |
| Hình 4.4 Mô hình DFD Mức 2.0..... | 23 |
| Hình 4.5 Mô hình DFD Mức 2.1..... | 24 |
| Hình 4.6 Mô hình DFD Mức 2.2..... | 24 |
| Hình 4.7 Mô hình tổng thể kiến trúc | 31 |
| Hình 4.8 Danh sách các Component..... | 31 |
| Hình 5.1 Giao diện màn hình chính | 33 |
| Hình 5.2 Giao diện đăng ký | 33 |
| Hình 5.3 Giao diện đăng nhập admin và user | 34 |
| Hình 5.4 Giao diện quên mật khẩu..... | 34 |
| Hình 5.5 Giao diện gợi ý kết bạn | 35 |
| Hình 5.6 Giao diện nhắn tin | 35 |
| Hình 5.7 Giao diện quản lý thông tin tài khoản | 36 |
| Hình 5.8 Giao diện thông báo | 36 |
| Hình 5.9 Giao diện quản lý user..... | 37 |

DANH MỤC BẢNG

| | |
|-------------------------------------|----|
| Bảng 4.1 Các loại thực thể | 20 |
| Bảng 4.2 AppMessage | 24 |
| Bảng 4.3 AppConversation | 25 |
| Bảng 4.4 AppInfoConversation | 26 |
| Bảng 4.5 AppColorConversation | 26 |
| Bảng 4.6 AppNickname | 27 |
| Bảng 4.7 MstStatusRequest | 27 |
| Bảng 4.8 AppFriendRequest | 27 |
| Bảng 4.9 AppUser | 28 |
| Bảng 4.10 AppRole | 29 |
| Bảng 4.11 AppVerifyCode..... | 29 |
| Bảng 4.12 AppFriendShip..... | 30 |

DANH MỤC TỪ VIẾT TẮT

| Từ viết tắt | Giải thích |
|-------------|-----------------------------|
| CSDL | Cơ sở dữ liệu |
| SQL | Structured Query Language |
| CPU | Central Processing Unit |
| RAM | Random Access Memory |
| HDD | Hard Disk Drive |
| ERD | Entity Relationship Diagram |
| DFD | Data Flow Diagram |

TỔNG QUAN CÔNG TY THỰC TẬP

TodayClass là một công ty giáo dục trực tuyến hàng đầu tại Việt Nam, cam kết mang đến cho mọi người một môi trường học tập chất lượng và tiện ích từ những khóa học đa dạng và đội ngũ giảng viên giàu kinh nghiệm. Với sứ mệnh truyền cảm hứng và mở rộng cánh cửa tri thức, TodayClass đã xây dựng nên một nền tảng giáo dục đa dạng và linh hoạt để phục vụ cộng đồng học sinh, sinh viên và những người muốn nâng cao kiến thức.

Trên trang web của mình (<https://www.todayclass.edu.vn/>), TodayClass cung cấp một loạt các khóa học đa dạng trong nhiều lĩnh vực khác nhau như Toán học, Ngôn ngữ, Khoa học, Xã hội, Âm nhạc và nhiều hơn nữa. Đội ngũ giảng viên của TodayClass được chọn lọc kỹ càng, bao gồm những chuyên gia, giáo viên có kinh nghiệm và nhiệt huyết trong việc chia sẻ kiến thức.

TodayClass tập trung vào việc cung cấp trải nghiệm học tập trực tuyến tốt nhất cho học viên. Trang web của họ có giao diện thân thiện, dễ sử dụng và tương thích trên nhiều thiết bị, cho phép học viên tiếp cận nội dung học tập mọi lúc, mọi nơi và theo tốc độ riêng của mình. Học viên có thể truy cập vào các bài giảng video, tài liệu học tập, bài kiểm tra và nhận phản hồi từ giảng viên để nâng cao hiệu suất học tập.

Một điểm đặc biệt của TodayClass là cộng đồng học viên đông đảo và sôi động. Học viên có thể tham gia vào các cuộc thảo luận, trao đổi ý kiến và chia sẻ kiến thức với nhau thông qua hệ thống phòng chat và diễn đàn trực tuyến.

TodayClass cũng cam kết cung cấp chất lượng dịch vụ và hỗ trợ tận tâm đến từng học viên. Họ luôn sẵn lòng giải đáp mọi thắc mắc, đồng hành và hỗ trợ học viên trong quá trình học tập, từ việc chọn khóa học phù hợp cho đến việc giải đáp các vấn đề liên quan đến nội dung học.

Với tầm nhìn "Truyền cảm hứng, Mở cánh tầm vươn tri thức", TodayClass đã và đang xây dựng một cộng đồng học tập trực tuyến chất lượng và đáng tin cậy, mang lại những cơ hội phát triển vượt bậc cho mọi người trong hành trình chinh phục tri thức.

CHƯƠNG 1: GIỚI THIỆU

1.1 Lý do chọn đề tài

Việc thiết kế một website chat realtime là một đề tài hấp dẫn và quan trọng trong lĩnh vực công nghệ thông tin hiện đại. Chúng tôi đã lựa chọn đề tài này vì một số lý do chính.

Thứ nhất, việc thiết kế một website chat realtime mang lại nhiều tiện ích và lợi ích cho cả người dùng và doanh nghiệp. Với tính năng chat realtime, người dùng có thể trò chuyện và giao tiếp trực tiếp với nhau một cách nhanh chóng và thuận tiện. Điều này tạo ra một trải nghiệm gần gũi và tương tác giữa các thành viên trong một cộng đồng hoặc tổ chức. Đồng thời, doanh nghiệp cũng có thể tận dụng tính năng này để tăng cường hỗ trợ khách hàng, tạo ra một kênh giao tiếp trực tuyến hiệu quả và tăng cường tương tác với khách hàng.

Thứ hai, trong thời đại số hóa phát triển nhanh chóng, việc sở hữu một website chat realtime đóng vai trò quan trọng trong việc xây dựng sự hiện diện trực tuyến của một tổ chức hoặc công ty. Với sự phổ biến của internet, mọi người ngày càng có xu hướng tìm kiếm thông tin, sản phẩm và dịch vụ trực tuyến. Do đó, có một kênh giao tiếp trực tuyến nhanh chóng và tiện lợi sẽ giúp cải thiện sự tương tác giữa tổ chức và khách hàng. Thiết kế một website chat realtime sẽ giúp tạo dựng niềm tin và sự tương tác tích cực giữa hai bên, từ đó tạo ra lợi ích lâu dài cho cả doanh nghiệp và khách hàng.

Tóm lại, việc lựa chọn đề tài thiết kế website chat realtime là một quyết định có ý nghĩa vì những tiện ích và lợi ích mà nó mang lại. Chúng tôi tin rằng việc thực hiện dự án này sẽ đóng góp vào việc nâng cao chất lượng giao tiếp và tương tác trực tuyến, mang lại lợi ích thiết thực cho cả người dùng và doanh nghiệp.

1.2 Hướng tiếp cận đề tài

Visual Studio là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) phổ biến và mạnh mẽ được phát triển bởi Microsoft. Với giao diện đồ họa trực quan và tích hợp đa nền tảng, Visual Studio hỗ trợ việc phát triển ứng dụng trên nhiều ngôn ngữ lập trình, bao gồm C#, C++, Java, Python và nhiều ngôn ngữ khác. Visual Studio cung cấp một loạt các công cụ và tính năng mạnh mẽ để giúp lập trình viên tạo ra các ứng dụng chất lượng cao. IDE này bao gồm trình

biên dịch, trình gỡ lỗi, trình soạn thảo mã, hỗ trợ quản lý phiên bản, và khả năng xây dựng và triển khai ứng dụng trên nhiều nền tảng. Ngoài ra, Visual Studio cũng cung cấp các công cụ và tiện ích hỗ trợ phát triển ứng dụng di động, web và máy tính cá nhân. Điểm mạnh của Visual Studio là tích hợp tốt với các công cụ và dịch vụ của Microsoft như Azure, SQL Server, và Team Foundation Server, giúp lập trình viên tối ưu quy trình phát triển và tăng cường hiệu suất làm việc.

Visual Studio Code (thường được gọi là VS Code) là một trình soạn thảo mã nguồn mở, nhẹ nhàng và mạnh mẽ, phát triển bởi Microsoft. Mặc dù không phải là một IDE đầy đủ như Visual Studio, VS Code cung cấp một môi trường phát triển linh hoạt và được ưa chuộng bởi rất nhiều lập trình viên. VS Code hỗ trợ đa ngôn ngữ và đa nền tảng, cho phép lập trình viên làm việc trên các dự án phổ biến như JavaScript, TypeScript, Python, C++, Java và nhiều ngôn ngữ khác. Nó đi kèm với một loạt các tính năng hữu ích như gợi ý mã, gỡ lỗi, tìm kiếm và thay thế thông minh, tích hợp điều khiển phiên bản, và khả năng mở rộng thông qua các tiện ích và phần mở rộng của cộng đồng. Visual Studio Code cũng cho phép người dùng tùy chỉnh giao diện và cấu hình theo ý muốn, đồng thời hỗ trợ tích hợp với công cụ quản lý phiên bản và dịch vụ như Git và Azure. Với hiệu suất cao và khả năng mở rộng, VS Code là một công cụ lập trình đáng tin cậy cho việc phát triển ứng dụng trên nhiều nền tảng và ngôn ngữ.

1.3 Ưu và nhược điểm của đề tài

1.3.1 Ưu điểm

Giao tiếp realtime: Hệ thống chat realtime cho phép người dùng gửi và nhận tin nhắn một cách ngay lập tức, tạo ra trải nghiệm giao tiếp trực quan và thú vị. Điều này rất hữu ích trong việc xây dựng ứng dụng nhóm, trang web thương mại điện tử hoặc các ứng dụng yêu cầu giao tiếp trực tuyến nhanh chóng.

Tương tác thời gian thực: Hệ thống chat realtime cho phép người dùng xem và phản hồi tin nhắn ngay lập tức, mà không cần tải lại trang. Điều này tạo ra một trải nghiệm trực quan và liên tục, tương tác trực tiếp giữa người dùng và nền tảng.

Cập nhật dữ liệu nhanh chóng: Với sử dụng giao thức WebSocket, hệ thống chat realtime có khả năng cập nhật dữ liệu nhanh chóng và hiệu quả. Thông tin mới như tin nhắn mới, sự tham gia của người dùng khác, hoặc các sự kiện khác có thể được hiển thị ngay lập tức.

Tính tương thích đa nền tảng: Sử dụng .NET 6 Web API làm phần server và React TypeScript làm phần front-end, hệ thống chat có tính tương thích đa nền tảng.

Điều này cho phép người dùng truy cập và sử dụng hệ thống trên nhiều thiết bị và trình duyệt khác nhau.

1.3.2 Nhược điểm

Phức tạp trong triển khai: Xây dựng một hệ thống chat realtime yêu cầu sự hiểu biết sâu về các công nghệ liên quan như WebSocket, .NET 6 Web API và React TypeScript. Điều này đòi hỏi kiến thức chuyên môn và kỹ năng phát triển phần mềm tốt.

Độ trễ mạng: Mặc dù giao tiếp realtime là một ưu điểm, nhưng đôi khi việc truyền dữ liệu qua mạng có thể gặp độ trễ. Điều này có thể ảnh hưởng đến trải nghiệm người dùng, đặc biệt trong các kết nối mạng không ổn định hoặc chậm.

Quản lý dữ liệu và bảo mật: Hệ thống chat realtime đòi hỏi việc quản lý dữ liệu tin nhắn, thông tin người dùng và bảo mật dữ liệu một cách nghiêm ngặt. Điều này đòi hỏi sự chú trọng và kỹ năng trong việc xử lý và bảo vệ dữ liệu cá nhân và thông tin nhạy cảm.

Tính mở rộng: Để xây dựng hệ thống chat realtime linh hoạt và có khả năng mở rộng, việc thiết kế kiến trúc phần mềm phù hợp là rất quan trọng. Điều này đòi hỏi kiến thức về kiến trúc phần mềm và khả năng ứng phó với lưu lượng truy cập và số lượng người dùng lớn.

CHƯƠNG 2: CƠ SỞ LÝ LUẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1 Cơ sở lý luận và ý nghĩa thực tiễn của đề tài

2.1.1 Cơ sở lý luận

Hệ thống chat realtime là một đề tài mang tính cần thiết và lý thú trong thế giới công nghệ ngày nay. Trong môi trường kết nối không ngừng phát triển và sự gia tăng của giao tiếp trực tuyến, việc thiết kế một hệ thống chat realtime đáp ứng nhu cầu giao tiếp ngay lập tức là rất quan trọng.

Cơ sở lý luận của đề tài này dựa trên sự phát triển của công nghệ WebSocket. Giao thức này cho phép thiết lập kết nối liên tục và song song giữa máy chủ và trình duyệt, giúp truyền dữ liệu trong thời gian thực mà không cần phải tải lại trang. Điều này mở ra những cơ hội mới cho việc tạo ra các ứng dụng giao tiếp trực tuyến chất lượng cao và hiệu quả.

2.1.2 Ý nghĩa thực tiễn của đề tài

Cải thiện trải nghiệm người dùng: Hệ thống chat realtime cho phép người dùng gửi và nhận tin nhắn một cách ngay lập tức, tạo ra một trải nghiệm giao tiếp trực quan, liền mạch và thuận tiện. Điều này đặc biệt quan trọng đối với các ứng dụng nhóm, trang web thương mại điện tử hoặc các ứng dụng yêu cầu giao tiếp trực tuyến nhanh chóng.

Tích hợp dễ dàng với các ứng dụng khác: Hệ thống chat realtime có khả năng tích hợp với các ứng dụng và dịch vụ khác, như email, hội thảo trực tuyến, hoặc các nền tảng quản lý công việc. Điều này tạo ra sự liên kết giữa các phương tiện giao tiếp và giúp người dùng tập trung và tổ chức công việc hiệu quả hơn.

Tích hợp dễ dàng với các ứng dụng khác: Hệ thống chat realtime có khả năng tích hợp với các ứng dụng và dịch vụ khác, như email, hội thảo trực tuyến, hoặc các nền tảng quản lý công việc. Điều này tạo ra sự liên kết giữa các phương tiện giao tiếp và giúp người dùng tập trung và tổ chức công việc hiệu quả hơn.

Ứng dụng trong lĩnh vực kinh doanh: Hệ thống chat realtime có thể được áp dụng trong lĩnh vực kinh doanh, như trang web thương mại điện tử, chăm sóc khách hàng trực tuyến và hỗ trợ bán hàng. Việc cung cấp giao tiếp trực tuyến ngay lập tức giữa khách hàng và doanh nghiệp giúp tạo ra sự tương tác tốt hơn, tăng cường sự tin tưởng và cải thiện trải nghiệm mua sắm.

2.2 Phương pháp nghiên cứu

2.2.1 Giới thiệu về công nghệ được áp dụng

Để phát triển và hoàn thiện được đề tài, bằng cách tiếp cận các công nghệ mới hiện nay. Chúng em đã áp dụng các công nghệ được vào đề tài này như: ASP NET CORE Web Api, SignalR, ReactJS, Redux Toolkit, Typescript, Material UI, nền tảng HTML CSS JavaScript, một số các phương thức API.

a) Asp.net core web api

ASP.NET Core Web API là một framework phát triển ứng dụng web của Microsoft, được thiết kế để xây dựng và triển khai các dịch vụ web mạnh mẽ, linh hoạt và dễ dàng mở rộng. Với việc tập trung vào hiệu suất cao, khả năng mở rộng và tính bảo mật, ASP.NET Core Web API là một lựa chọn lý tưởng để xây dựng các ứng dụng web chất lượng cao và hỗ trợ các dịch vụ RESTful.

ASP.NET Core Web API cung cấp một cơ chế mạnh mẽ cho việc xây dựng các API đa nền tảng, cho phép chúng ta tương tác với các ứng dụng và thiết bị khác nhau. Nó hỗ trợ các định dạng dữ liệu phổ biến như JSON và XML, và cung cấp khả năng tùy chỉnh linh hoạt để xử lý yêu cầu và trả về dữ liệu theo các chuẩn RESTful.

ASP.NET Core Web API cũng có thể tích hợp tốt với các công nghệ khác trong hệ sinh thái của Microsoft như Entity Framework Core cho việc truy cập cơ sở dữ liệu, Identity Framework cho việc quản lý xác thực và phân quyền, và Azure để triển khai ứng dụng vào môi trường đám mây.

Với khả năng tùy chỉnh cao, ASP.NET Core Web API cho phép chúng ta xây dựng các ứng dụng web đáng tin cậy và hiệu suất cao, phục vụ các yêu cầu của môi trường phát triển ứng dụng hiện đại. Sự kết hợp giữa linh hoạt, hiệu suất và tính bảo mật của ASP.NET Core Web API làm nó trở thành một lựa chọn hàng đầu cho việc xây dựng và triển khai các dịch vụ web mạnh mẽ và đáng tin cậy.

b) ReactJS

React là một thư viện JavaScript phổ biến và mạnh mẽ, được phát triển bởi Facebook. Với thiết kế tập trung vào việc xây dựng giao diện người dùng hiệu quả và linh hoạt, React đã trở thành một công cụ không thể thiếu trong việc phát triển ứng dụng web hiện đại.

React sử dụng cơ chế Virtual DOM để tối ưu hóa việc cập nhật giao diện người dùng. Thay vì cập nhật toàn bộ DOM, React chỉ cập nhật những phần thay đổi, giúp cải thiện hiệu suất và tăng tốc độ render của ứng dụng. Bên cạnh đó, React

cũng cho phép tái sử dụng các thành phần (components), giúp việc phát triển và bảo trì ứng dụng trở nên dễ dàng và hiệu quả.

React cung cấp một hệ thống quản lý trạng thái (state) mạnh mẽ thông qua React Hooks. Hooks giúp quản lý trạng thái của các thành phần một cách đơn giản và dễ hiểu, giúp tăng khả năng tái sử dụng code và giảm thiểu sự phức tạp của ứng dụng.

Ngoài ra, cộng đồng phát triển xung quanh React là rất lớn và đầy năng động, với hàng ngàn thư viện và công cụ hỗ trợ. Điều này mang lại lợi ích lớn cho việc tìm kiếm giải pháp và chia sẻ kiến thức trong quá trình phát triển ứng dụng.

Với khả năng linh hoạt, hiệu suất và khả năng tái sử dụng code, React là lựa chọn hàng đầu cho việc xây dựng giao diện người dùng đáng tin cậy, phức tạp và tương tác cao. Việc sử dụng React trong phát triển ứng dụng web giúp tăng cường trải nghiệm người dùng và giảm thiểu công sức phát triển, đồng thời mang lại sự linh hoạt và khả năng mở rộng cho dự án của bạn.

c) Redux toolkit

Redux Toolkit là một bộ công cụ phát triển ứng dụng Redux cung cấp các công cụ và hàm giúp giảm bớt độ phức tạp và công việc lặp lại trong quá trình quản lý trạng thái của ứng dụng. Với mục tiêu đơn giản hóa việc sử dụng Redux, Redux Toolkit là một lựa chọn lý tưởng cho việc phát triển ứng dụng web phức tạp.

Redux Toolkit giới thiệu một khái niệm mới gọi là "slices" (phần), cho phép chúng ta chia nhỏ store của Redux thành các phần nhỏ, mỗi phần đảm nhận việc quản lý trạng thái của một phần cụ thể trong ứng dụng. Điều này giúp tăng tính tổ chức và dễ bảo trì của mã nguồn, đồng thời giảm thiểu sự trùng lặp và đảm bảo tính nhất quán trong quản lý trạng thái.

Redux Toolkit cung cấp một API giản đơn và mạnh mẽ cho việc xác định actions và reducers. Thay vì phải viết nhiều code nhàm chán để xác định action types và viết reducers, Redux Toolkit cho phép chúng ta sử dụng syntax ngắn gọn và tự động tạo ra các action types và reducers tương ứng. Điều này giúp giảm thiểu công việc lặp lại và tăng cường hiệu suất phát triển.

Redux Toolkit cũng tích hợp sẵn các middleware phổ biến như Redux Thunk và Redux Saga để xử lý các tác vụ bất đồng bộ, như gọi API hoặc xử lý side effects. Điều này giúp quản lý các tác vụ không đồng bộ trở nên dễ dàng và tiện lợi trong quá trình phát triển ứng dụng.

Với những tiện ích và khả năng giảm bớt độ phức tạp, Redux Toolkit là một công cụ hữu ích trong việc phát triển ứng dụng Redux. Sự đơn giản và hiệu quả của

Redux Toolkit giúp giảm thiểu công sức và thời gian phát triển, đồng thời tăng cường tính ổn định và khả năng mở rộng của ứng dụng.

d) HTML, CSS, Javascript

HTML cung cấp cấu trúc cơ bản của các trang web, được cải tiến và sửa đổi bởi các công nghệ khác như CSS và JavaScript.

CSS được sử dụng để kiểm soát trình bày, định dạng và bố cục.

JavaScript được sử dụng để kiểm soát hành vi của các yếu tố khác nhau.

e) HTML

Là cốt lõi của mọi trang web, bất kể sự phức tạp của một trang web hoặc số lượng công nghệ liên quan. Đó là một kỹ năng thiết yếu cho bất kỳ chuyên gia lập trình web nào. Đó là điểm khởi đầu cho bất cứ ai học cách tạo nội dung cho web. Và may mắn thay cho chúng ta, nó thật dễ dàng để học hỏi. HTML là viết tắt của HyperText Markup Language- Ngôn ngữ định dạng văn bản. “Markup Language” có nghĩa là, thay vì sử dụng ngôn ngữ lập trình để thực hiện các chức năng, HTML sử dụng các thẻ để xác định các loại nội dung khác nhau và mục đích phục vụ cho trang web. Các ngôn ngữ đánh dấu với cách thức hoạt động theo cách giống như bạn đã làm khi bạn gắn nhãn các loại nội dung đó, ngoại trừ chúng sử dụng code để làm điều đó – cụ thể, chúng sử dụng các thẻ HTML, còn được gọi là “element”. Các thẻ này có tên khá trực quan: thẻ header, thẻ paragraph, thẻ image, v.v.

Mỗi trang web được tạo thành từ một loạt các thẻ HTML này biểu thị từng loại nội dung trên trang. Mỗi loại nội dung trên trang được “wrapped”, tức là được bao quanh bởi các thẻ HTML. Sử dụng HTML, bạn có thể thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển, tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết, xây dựng bảng, điều khiển một số kiểu, và nhiều hơn nữa.

f) CSS

CSS là viết tắt của Cascading Style Sheets. Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn. Các ngôn ngữ này được giữ riêng biệt để đảm bảo các trang web được xây dựng chính xác trước khi chúng được định dạng lại.

Trong khi HTML là cấu trúc cơ bản của trang web của bạn. CSS mang lại cho trang web của bạn phong cách mà bạn muốn. Những màu sắc đặc trưng, font chữ phù hợp, và hình ảnh nền của website? Tất cả là nhờ CSS. CSS gần như tạo nên

bộ mặt của một website. Và nếu trang web của bạn ưa nhìn thì nó sẽ hấp dẫn và lôi cuốn được người dùng phải không.

Để có một trang web đẹp bạn không chỉ dựa vào CSS. Mà các ngôn ngữ phải được thực hiện dựa trên sự sáng tạo và các bản thiết kế bạn tạo ra trước đó. Ngoài ra, bạn cần phải kết hợp nó với nhiều hơn các ngôn ngữ khác nữa.

g) Javascript

JavaScript là ngôn ngữ phức tạp hơn HTML hay CSS. Và nó không được phát hành ở dạng beta cho đến năm 1995. Ngày nay, JavaScript được hỗ trợ bởi tất cả các trình duyệt web hiện đại. Và được sử dụng trên hầu hết mọi trang web trên web để có chức năng mạnh mẽ và phức tạp hơn.

JavaScript là ngôn ngữ lập trình dựa trên logic. Nó có thể được sử dụng để sửa đổi nội dung trang web. Và khiến nó hoạt động theo nhiều cách khác nhau để đáp ứng với hành động của người dùng. Các cách sử dụng phổ biến cho JavaScript bao gồm các hộp xác nhận, kêu gọi hành động và thêm các danh tính mới vào thông tin hiện có.

JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác. Hầu hết các hành vi động mà bạn sẽ thấy trên một trang web là nhờ JavaScript. Nó giúp tăng cường các hành vi và kiểm soát mặc định của trình duyệt. Ngôn ngữ lập trình này có thể làm rất nhiều thứ trên trang web của bạn. Nó làm cho mọi thứ trở nên tiện nghi hơn bao giờ hết. Tuy nhiên để học được ngôn ngữ này bạn cần phải rất kiên trì. Bởi đây là ngôn ngữ khá phức tạp và khó để học.

Với những chia sẻ về vai trò của lập trình HTML, CSS và JavaScript trong việc xây dựng website. Mong rằng các bạn sẽ có những cái nhìn tổng quan hơn về ngành lập trình này. Chúc bạn thành công. Nếu yêu thích, bạn hãy bắt tay ngay vào tìm hiểu sâu hơn về HTML, CSS và JavaScript.

h) Các giao thức api

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng. Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API. API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức

kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành. API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++. API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện dễ sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập, dễ hiểu hơn. Web API hiện đại dùng cho các đối tượng cụ thể, chẳng hạn như mobile developer với document, version khác nhau. API key: Đây là loại code (string) được truyền tải bởi các chương trình máy tính gọi là API để xác định chương trình, nhà phát triển hoặc người dùng nó tới trang web. Các API key được sử dụng với mục đích nhằm giới hạn, kiểm soát sử dụng API. Chẳng hạn như ngăn chặn sự việc lạm dụng API.

Trong đồ án này chúng em sử dụng các phương thức API như (Get, Post, Ajax).

i) SignalR

ASP.NET SignalR là một thư viện cho các lập trình viên Asp.Net đơn giản hóa quá trình thêm chức năng web real-time trong phát triển ứng dụng. Real-time web functionality là gì ? Đó là khả năng server đẩy những nội dung tới client đã được kết nối một cách tức thì. Nó khác với giao thức HTTP thông thường: server đợi những yêu cầu từ client và trả về nội dung tương ứng.

SignalR có thể sử dụng trong bất kỳ chức năng web real-time nào. Trong đó ứng dụng chat trên web là một ví dụ điển hình. Ngoài ra, các ứng dụng cho dashboards, monitoring, collaborative là những gợi ý cho việc sử dụng SignalR.

SignalR cung cấp một API đơn giản cho việc tạo server-to-client remote procedure call (RPC) để gọi những hàm javascript trong trình duyệt (và những nền tảng khác) từ code .Net của server-side. SignalR cũng bao gồm API cho việc quản lý kết nối (connect và disconnect events) và những kết nối nhóm.

SignalR xử lý quản lý kết nối một cách tự động, và cho bạn truyền đi thông điệp tới tất cả các client đã được kết nối một cách đồng loạt, giống như một chat room. Bạn cũng có thể gửi những thông điệp tới những client được xác định. Kết nối giữa client và server là liên tục, không giống như kết nối HTTP cổ điển, cái mà sẽ thành lập lại kết nối cho mỗi lần giao tiếp.

SignalR hỗ trợ chức năng "server push", trong server code có thể gọi tới client code trong trình duyệt bởi "Remote Procedure Calls" (RPC), hơn là sử dụng Service Bus, SQL Server hay Redis.

j) Design pattern

Ở Đây dự án này tôi áp dụng 3 loại Design Pattern là Unit Of Work, Dependency Injection, Repository.

Design Pattern (hay còn gọi là mẫu thiết kế) là một giải pháp chung để giải quyết các vấn đề phổ biến khi thiết kế phần mềm trong lập trình hướng đối tượng OOP.

Với sự đúc kết và công nhận từ nhiều nhà nghiên cứu, Design Pattern là mẫu chuẩn tối ưu nhất, có thể áp dụng để giải quyết không chỉ một vấn đề mà nhiều vấn đề có tính chất tương tự nhau, lặp đi lặp lại nhiều lần trong lập trình.

Design Pattern không dành riêng cho một ngôn ngữ lập trình cụ thể nào. Nó có thể được áp dụng trong hầu hết các ngôn ngữ lập trình OOP như: PHP, C#, Java, Python và nhiều ngôn ngữ khác.

k) Typescript

TypeScript là một ngôn ngữ lập trình phổ biến, phát triển bởi Microsoft, và là phiên bản mở rộng của JavaScript. Với mục tiêu giúp xây dựng các ứng dụng JavaScript lớn và phức tạp hơn một cách dễ dàng và tin cậy hơn, TypeScript đã trở thành một công cụ quan trọng trong cộng đồng phát triển ứng dụng web hiện đại.

TypeScript đặc biệt nổi bật với khả năng tường minh kiểu dữ liệu (static typing), cho phép xác định rõ ràng kiểu dữ liệu của biến, tham số và giá trị trả về. Điều này giúp phát hiện và sửa lỗi nhanh chóng, tăng tính ổn định và đáng tin cậy của mã nguồn, đồng thời cung cấp trợ giúp trong quá trình phát triển và tự động hoàn thiện mã (code completion).

TypeScript cũng hỗ trợ các tính năng hiện đại của ECMAScript như classes, modules, và async/await. Bằng việc sử dụng TypeScript, chúng ta có thể sử dụng các tính năng mới nhất của JavaScript và đảm bảo tương thích ngược với các phiên bản JavaScript trước đó.

Một ưu điểm quan trọng khác của TypeScript là khả năng tích hợp dễ dàng vào các dự án JavaScript hiện có. TypeScript được biên dịch thành JavaScript chuẩn, cho phép chúng ta sử dụng mã nguồn TypeScript trong các dự án JavaScript mà không cần thay đổi đáng kể cấu trúc và logic hiện có.

Với việc kết hợp giữa tính tường minh kiểu dữ liệu, khả năng sử dụng các tính năng hiện đại của JavaScript và tích hợp dễ dàng vào các dự án JavaScript, TypeScript là một lựa chọn lý tưởng cho việc phát triển ứng dụng web. Sự mạnh

mẽ, tin cậy và dễ sử dụng của TypeScript giúp giảm thiểu lỗi phát sinh, tăng cường khả năng bảo trì và cải thiện hiệu suất của ứng dụng.

1) Material UI

Material-UI là một thư viện giao diện người dùng (UI) phổ biến và mạnh mẽ được xây dựng trên nền tảng React. Với sự tập trung vào thiết kế theo phong cách Material Design của Google, Material-UI cung cấp các thành phần và công cụ giúp xây dựng giao diện người dùng đẹp mắt, tương tác và dễ sử dụng trong các ứng dụng web.

Material-UI cung cấp một bộ sưu tập rộng các thành phần UI đã được thiết kế sẵn như nút, ô đầu chọn, thanh trượt, menu và nhiều hơn nữa. Các thành phần này đã được tối ưu hóa để hoạt động tốt trên các trình duyệt và thiết bị khác nhau, đồng thời cung cấp các tính năng tùy chỉnh để phù hợp với nhu cầu của từng dự án.

Material-UI cũng hỗ trợ một hệ thống định dạng linh hoạt (flexbox grid) giúp xây dựng giao diện linh hoạt và đáp ứng (responsive), tự động điều chỉnh kích thước và bố trí của các thành phần. Điều này giúp tạo ra trải nghiệm người dùng nhất quán trên nhiều kích thước màn hình và thiết bị khác nhau.

Với Material-UI, việc tạo giao diện người dùng tương tác và hấp dẫn trở nên dễ dàng hơn bao giờ hết. Thư viện cung cấp các tính năng như animations, transitions và tương tác tái sử dụng giúp tạo ra các hiệu ứng mượt mà và hấp dẫn. Ngoài ra, Material-UI hỗ trợ cả chủ đề tùy chỉnh, cho phép tùy chỉnh các màu sắc, kiểu chữ và các yếu tố giao diện khác để phù hợp với phong cách và thương hiệu của dự án.

Với sự kết hợp giữa phong cách thiết kế Material Design, các thành phần UI đa dạng và tính linh hoạt, Material-UI là một lựa chọn lý tưởng cho việc xây dựng giao diện người dùng chất lượng cao trong các ứng dụng React. Thư viện giúp tăng cường trải nghiệm người dùng, giảm thiểu công sức phát triển và đảm bảo tính nhất quán và một hình ảnh chuyên nghiệp cho ứng dụng của bạn.

CHƯƠNG 3: GIỚI THIỆU TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

3.1 Tổng quan về hệ thống

3.1.1 Mục tiêu

a) Giao tiếp thời gian thực

Một trong những mục tiêu quan trọng nhất của web chat realtime là cung cấp khả năng giao tiếp thời gian thực giữa người dùng. Người dùng có thể gửi và nhận tin nhắn ngay lập tức mà không cần làm mới trang web. Điều này tạo ra trải nghiệm tương tác tức thì, giúp người dùng cảm thấy gần gũi hơn và thuận tiện trong việc trao đổi thông tin.

b) Đa nền tảng và đa thiết bị

Mục tiêu của web chat realtime là tương thích trên nhiều nền tảng và thiết bị khác nhau. Điều này cho phép người dùng truy cập vào chat từ các trình duyệt web trên máy tính, điện thoại thông minh và các thiết bị di động khác. Đa nền tảng giúp đảm bảo rằng mọi người dùng đều có thể truy cập và sử dụng chat một cách dễ dàng.

c) Bảo mật và riêng tư

Một yếu tố quan trọng trong web chat realtime là bảo mật và riêng tư. Người dùng cần được đảm bảo rằng thông tin và dữ liệu cá nhân của họ được bảo vệ và không bị rò rỉ. Mục tiêu là triển khai các biện pháp bảo mật như mã hóa dữ liệu và xác thực người dùng để đảm bảo tính an toàn và riêng tư trong quá trình trò chuyện.

d) Tích hợp chức năng mở rộng

Một trong những mục tiêu quan trọng khác của web chat realtime là khả năng mở rộng và tích hợp với các chức năng khác. Ví dụ, có thể tích hợp cơ chế gửi tệp đính kèm, chia sẻ hình ảnh hoặc video, tạo nhóm chat và gọi video để tăng cường trải nghiệm và tính năng của ứng dụng.

e) Quản lý trạng thái và lịch sử chat

Mục tiêu cuối cùng của web chat realtime là quản lý trạng thái và lịch sử chat. Người dùng có thể xem lại tin nhắn trước đó, theo dõi các hoạt động và biết ai đang online. Điều này giúp tăng tính nhất quán và khả năng theo dõi trong quá trình trò chuyện.

3.1.2 Phạm vi ứng dụng

Giao tiếp người dùng: Ứng dụng web chat realtime cho phép người dùng giao tiếp và trao đổi thông tin với nhau thông qua tin nhắn. Người dùng có thể bắt đầu cuộc trò chuyện, gửi tin nhắn, nhận tin nhắn và tham gia vào các cuộc trò chuyện nhóm.

Tính năng tương tác: Ứng dụng web chat realtime có thể cung cấp các tính năng tương tác như gửi tin nhắn tức thì, gửi hình ảnh, tệp đính kèm, biểu tượng cảm xúc và đánh dấu tin nhắn quan trọng. Người dùng có thể tương tác và chia sẻ thông tin một cách linh hoạt và đa dạng.

Quản lý người dùng: Ứng dụng web chat realtime thường có chức năng quản lý người dùng như đăng ký, đăng nhập và quản lý tài khoản. Người dùng có thể tạo tài khoản, thay đổi thông tin cá nhân và quản lý danh sách liên lạc.

Bảo mật và quyền riêng tư: Một yếu tố quan trọng trong ứng dụng web chat realtime là đảm bảo tính bảo mật và quyền riêng tư của người dùng. Các biện pháp bảo mật như mã hóa dữ liệu và xác thực người dùng có thể được áp dụng để đảm bảo rằng thông tin và dữ liệu cá nhân không bị lộ ra bên ngoài.

Quản lý lịch sử chat: Ứng dụng web chat realtime có thể cung cấp khả năng lưu trữ và quản lý lịch sử chat. Điều này cho phép người dùng xem lại tin nhắn trước đó, tìm kiếm thông tin và theo dõi các cuộc trò chuyện đã diễn ra.

Đa nền tảng và đa thiết bị: Phạm vi ứng dụng web chat realtime có thể bao gồm việc hỗ trợ trên nhiều nền tảng và thiết bị khác nhau. Người dùng có thể truy cập và sử dụng ứng dụng từ các trình duyệt web trên máy tính, điện thoại di động và các thiết bị di động khác.

3.1.3 KHẢ NĂNG PHÁT TRIỂN:

Linh hoạt và mở rộng: Một ứng dụng web chat realtime nên được thiết kế để dễ dàng mở rộng và linh hoạt trong việc thêm tính năng mới. Kiến trúc phần mềm của ứng dụng cần phải cho phép việc thêm, sửa đổi và loại bỏ các thành phần một cách dễ dàng, đảm bảo tính mở rộng và khả năng mở rộng theo yêu cầu của dự án.

Tích hợp dễ dàng: Khả năng tích hợp với các dịch vụ và công nghệ khác là một yếu tố quan trọng. Ứng dụng web chat realtime nên cung cấp các giao diện lập trình ứng dụng (APIs) để cho phép tích hợp với các hệ thống khác như cơ sở dữ liệu, hệ thống xác thực, dịch vụ bên thứ ba và các công nghệ khác.

Dễ bảo trì và kiểm thử: Ứng dụng web chat realtime cần được xây dựng và tổ chức một cách có cấu trúc, dễ bảo trì và kiểm thử. Mã nguồn nên tuân thủ các nguyên tắc tốt nhất và chuẩn mã hóa, giúp giảm thiểu lỗi và tăng tính ổn định của ứng dụng.

Cộng đồng hỗ trợ: Sự phát triển của một ứng dụng web chat realtime cũng phụ thuộc vào sự tồn tại và sự hỗ trợ của một cộng đồng lớn và sôi động. Có một cộng đồng hỗ trợ sẽ giúp đáp ứng các câu hỏi, chia sẻ kiến thức và tài liệu, đồng thời cung cấp các bản vá lỗi và cập nhật mới nhất.

Tính mở và tương thích: Khả năng phát triển của ứng dụng web chat realtime cũng được củng cố bởi tính mở và tương thích. Nếu ứng dụng sử dụng các công nghệ phổ biến và chuẩn, nó sẽ có thể dễ dàng tích hợp và tương thích với các thành phần, công cụ và dịch vụ khác.

3.2 Phân khảo sát hiện trạng

3.2.1 Hiện trạng

a) Tính năng

Đánh giá hiện trạng của ứng dụng web chat realtime bao gồm các tính năng có sẵn và khả năng hoạt động của chúng. Các tính năng chính bao gồm gửi và nhận tin nhắn, tạo cuộc trò chuyện nhóm, gửi tệp đính kèm, biểu tượng cảm xúc và nhiều hơn nữa. Ngoài ra, việc xem lại lịch sử chat, quản lý người dùng và bảo mật cũng là những yếu tố quan trọng cần được xem xét.

b) Hiệu suất

Đánh giá hiệu suất của ứng dụng web chat realtime liên quan đến khả năng xử lý và phản hồi nhanh chóng. Nó bao gồm thời gian phản hồi của hệ thống khi người dùng gửi tin nhắn, khả năng xử lý đồng thời của ứng dụng và khả năng mở rộng để xử lý tải cao khi số lượng người dùng tăng lên.

c) Giao diện người dùng

Hiện trạng của giao diện người dùng ảnh hưởng đến trải nghiệm người dùng. Giao diện người dùng cần được thiết kế đơn giản, dễ sử dụng và thân thiện. Điều này bao gồm cách tổ chức cuộc trò chuyện, hiển thị tin nhắn, biểu tượng cảm xúc và các chức năng khác.

d) Bảo mật

Đánh giá hiện trạng bảo mật của ứng dụng web chat realtime là cực kỳ quan trọng. Nó bao gồm cách mã hóa thông tin người dùng, xác thực người dùng và đảm bảo quyền riêng tư. Một ứng dụng tốt cần áp dụng các biện pháp bảo mật phù hợp để đảm bảo rằng thông tin người dùng không bị lộ ra ngoài và không bị tấn công từ phía bên ngoài.

e) Tính ổn định và đáng tin cậy

Hiện trạng của ứng dụng web chat realtime nên được đánh giá dựa trên tính ổn định và đáng tin cậy của nó. Điều này bao gồm khả năng xử lý lỗi, khôi phục hệ thống khi có sự cố và đảm bảo rằng ứng dụng hoạt động một cách liên tục và không bị gián đoạn.

f) Hỗ trợ và cập nhật

Hiện trạng của ứng dụng web chat realtime nên bao gồm cách hỗ trợ từ nhà phát triển và cập nhật thường xuyên để cải thiện và nâng cao ứng dụng. Cập nhật bản vá lỗi, tính năng mới và tài liệu hỗ trợ giúp đảm bảo rằng ứng dụng luôn hoạt động tốt và đáp ứng nhu cầu của người dùng.

3.2.2 Phân tích yêu cầu

a) Mô tả

Website chat realtime là một nền tảng tương tác trực tuyến mà người dùng có thể sử dụng để gửi và nhận tin nhắn một cách tức thì. Điều đặc biệt về website này là khả năng truyền tin nhắn ngay lập tức giữa các người dùng, cho phép họ trò chuyện và giao tiếp một cách dễ dàng và tiện lợi.

Với tính năng chủ đạo là trò chuyện realtime, website này cho phép người dùng tạo ra các cuộc trò chuyện cá nhân hoặc nhóm, gửi và nhận tin nhắn văn bản, gửi tệp đính kèm và biểu tượng cảm xúc. Người dùng có thể xem lịch sử chat và quản lý danh sách người dùng trong hệ thống.

Với việc sử dụng công nghệ web và giao diện thân thiện, website chat realtime mang đến trải nghiệm người dùng dễ sử dụng và giao tiếp trực quan. Giao diện được thiết kế đơn giản và trực quan, cho phép người dùng tương tác một cách dễ dàng mà không cần sử dụng quá nhiều công cụ phức tạp.

Đối với hiệu suất, website chat realtime được tối ưu hóa để đảm bảo việc truyền và xử lý tin nhắn nhanh chóng và chính xác. Hệ thống được thiết kế để đáp ứng tải cao và có khả năng mở rộng, giúp đảm bảo rằng người dùng có thể truy cập và sử dụng website một cách mượt mà ngay cả khi có số lượng người dùng lớn.

Đồng thời, bảo mật là một yếu tố quan trọng của website chat realtime. Các biện pháp bảo mật được triển khai để đảm bảo an toàn cho dữ liệu và thông tin cá nhân của người dùng. Các biện pháp bảo mật bao gồm mã hóa dữ liệu, xác thực người dùng và quản lý quyền truy cập, nhằm đảm bảo rằng chỉ người dùng hợp lệ mới có thể truy cập và sử dụng website.

Tóm lại, website chat realtime cung cấp một giao diện tương tác trực tuyến cho phép người dùng gửi và nhận tin nhắn một cách tức thì. Với tính năng, hiệu suất, giao diện người dùng và bảo mật tốt, nó mang đến trải nghiệm trò chuyện dễ dàng và an toàn cho người dùng.

b) Yêu cầu

Yêu cầu chung

Tích hợp đa nền tảng: Website chat realtime nên hỗ trợ đa nền tảng, cho phép người dùng truy cập và sử dụng từ các thiết bị khác nhau như máy tính, điện thoại di động hoặc máy tính bảng. Điều này đảm bảo rằng người dùng có thể truy cập và trò chuyện một cách thuận tiện từ bất kỳ thiết bị nào.

Tích hợp các tính năng mở rộng: Ngoài tính năng trò chuyện cơ bản, website chat realtime có thể tích hợp các tính năng mở rộng như chia sẻ hình ảnh, gọi video, tạo nhóm, gửi file âm thanh, tích hợp các biểu tượng cảm xúc đa dạng và nhiều tính năng tương tác khác. Điều này tạo ra một trải nghiệm trò chuyện đa dạng và phong phú cho người dùng.

Tích hợp tính năng thông báo: Để đảm bảo rằng người dùng không bỏ lỡ bất kỳ tin nhắn nào, website chat realtime có thể tích hợp tính năng thông báo. Điều này cho phép người dùng nhận thông báo trực tiếp trên thiết bị của họ khi có tin nhắn mới hoặc hoạt động liên quan đến cuộc trò chuyện.

Tích hợp tính năng tìm kiếm và lọc: Với số lượng tin nhắn và cuộc trò chuyện ngày càng tăng, website chat realtime có thể tích hợp tính năng tìm kiếm và lọc để người dùng có thể dễ dàng tìm kiếm và quản lý thông tin trong cuộc trò chuyện.

Các chức năng chính của website

Đăng nhập và đăng ký: Người dùng có thể tạo tài khoản mới hoặc đăng nhập vào hệ thống để sử dụng các tính năng của website chat.

Tạo và quản lý cuộc trò chuyện: Người dùng có thể tạo ra các cuộc trò chuyện cá nhân, đặt tên và mô tả cho cuộc trò chuyện.

Gửi lời mời kết bạn: Người dùng có thể gửi lời mời kết bạn cho người khác để và khi họ chấp nhận thì hai người sẽ có thể nhắn tin với nhau.

Gửi và nhận tin nhắn: Người dùng có thể gửi và nhận tin nhắn văn bản trong cuộc trò chuyện, với khả năng xem lịch sử tin nhắn và gửi tin nhắn mới.

Gửi file đính kèm: Người dùng có thể gửi và nhận các tệp hình ảnh trong cuộc trò chuyện.

Biểu tượng cảm xúc: Website chat realtime có thể cung cấp một bộ biểu tượng cảm xúc đa dạng để người dùng có thể thể hiện cảm xúc và tương tác một cách thú vị.

Thông báo realtime: Hệ thống có khả năng gửi thông báo trực tiếp đến người dùng khi có tin nhắn mới hoặc các hoạt động liên quan đến cuộc trò chuyện.

Bảo mật và quyền riêng tư: Website cần có các biện pháp bảo mật để bảo vệ thông tin người dùng và đảm bảo quyền riêng tư. Điều này bao gồm mã hóa dữ liệu, xác thực người dùng, quản lý quyền truy cập và các biện pháp bảo mật khác.

Giao diện thân thiện và tương tác: Website cần có giao diện người dùng đơn giản, thân thiện và dễ sử dụng, cho phép người dùng tương tác một cách dễ dàng và thoải mái.

Tính linh hoạt và mở rộng: Hệ thống cần được thiết kế để có khả năng mở rộng và mở rộng tính năng trong tương lai, để đáp ứng nhu cầu người dùng và thích ứng với sự phát triển công nghệ.

Yêu cầu phần cứng

Cấu hình vừa đủ dùng:

-RAM: 4GB DDR4 Bus 2400 trở lên.

-CPU: Intel Core i3 7100 trở lên.

-Card đồ họa: Intel HD Graphic 630 trở lên

-HDD: 1 ổ cứng SSD tối thiểu 120G chuyên dùng để cài win và 1 ổ cứng HDD ít nhất là 500G để lưu trữ dữ liệu người dùng.

Yêu cầu phần mềm

Hệ điều hành: Microsoft Windows 7 trở lên;

Hệ quản trị CSDL: Microsoft SQL Server 2012 R2 - Microsoft SQL Server-2019, tính năng vượt trội Microsoft Access về lập trình quản lý CSDL, khả năng lưu trữ lớn, chạy trên môi trường mạng, tính bảo mật lại cao, hỗ trợ tốt phong chữ Unicode.

Công cụ lập trình, thiết kế giao diện: Visual studio 2022 - là ngôn ngữ lập trình khá đơn giản, hỗ trợ thiết kế giao diện tương đối đầy đủ, có khả năng kết nối mạnh với các hệ CSDL khác nhau, có khả năng biên dịch chương trình thực thi độc lập và tạo bộ cài đặt chuẩn, có khả năng tích hợp các ActiveX Control hỗ trợ tốt phong chữ Unicode cả giao diện người dùng lẫn nhập.

Yêu cầu về tính bảo mật

Mỗi người dùng sẽ được cấp một tài khoản (ID User) và mật khẩu (Password) để đăng nhập vào Admin. Mật khẩu này được mã hóa theo dạng MD5 để tăng tính bảo mật cho tài khoản để dễ dàng quản lý Website và tránh các đợt tấn công của Hacker.

3.2.3 Môi trường triển khai

a) Hệ điều hành

Hệ điều hành: Microsoft Windows 10 trở lên.

b) Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2019 bởi vì:

Phù hợp lưu trữ dữ liệu vừa và không quá lớn.

Khả năng quản lý tốt và mang tính phổ thông.

Sử dụng đơn giản, dễ dàng tiếp cận, phù hợp CSDL dùng chung.

c) Công cụ phát triển

Backend: Visual Studio 2012

Frontend: Visual Studio Code

d) Ngôn ngữ

Tiếng Việt Nam, dùng phong Unicode vì phù hợp với môi trường mạng, thói quen sử dụng và quy định Nhà Nước.

CHƯƠNG 4: THIẾT KẾ CƠ SỞ DỮ LIỆU

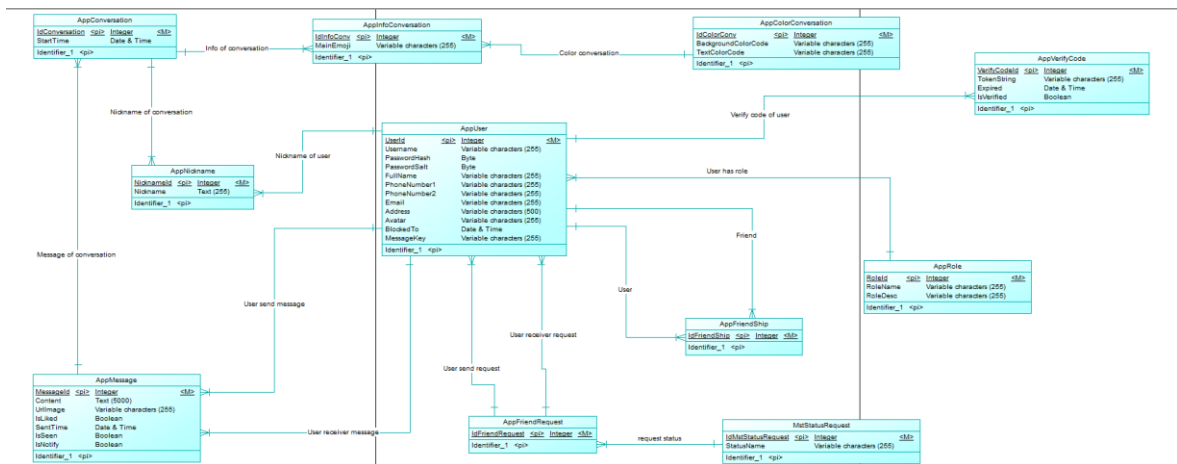
4.1 Các loại thực thể

Bảng 4.1 Các loại thực thể

| STT | Thực Thể | Chi tiết | Ghi chú |
|-----|----------------------|--|---------|
| 1 | AppMessage | + Id của tin nhắn (Id) + Id cuộc trò chuyện (ConversationId) + Id người gửi (SenderId) + Id người nhận (ReceiverId) + Hình ảnh (UrlImage) + Like tin nhắn (IsLiked) + Thời gian gửi tin nhắn (SentTime) + Tin nhắn đã xem hay chưa (IsSeen) + Thời gian tạo tin nhắn (Created At) + Tin nhắn này có dạng thông báo (IsNotify) | |
| 2 | AppConversation | + Id cuộc trò chuyện (Id) + Id của user (UserId1) + Id của friend (UserId2) | |
| 3 | AppColorConversation | + Id màu sắc (Id) + Mã màu nền (BackgroundColorCode) + Mã màu chữ (TextColorCode) | |
| 4 | AppInfoConversation | + Id thông tin cuộc trò chuyện (Id) + Id cuộc trò chuyện (ConversationId) + Màu sắc cuộc trò chuyện (ColorId) + Biểu cảm chính (MainEmoji) | |
| 5 | AppNickname | + Id biệt danh (Id) + Id user (UserId) + Tên biệt danh (Nickname) | |
| 6 | AppFriendRequest | + Id gửi yêu cầu kết bạn (Id) + Id người gửi (SenderId) + Id người nhận (ReceiverId) + Id trạng thái yêu cầu (StatusId) | |
| 7 | MstStatusRequest | + Id trạng thái (Id) + Tên trạng thái (StatusName) | |
| 8 | User | + Tên tài khoản (Username) + Họ tên người dùng (Fullname) | |

| | | | |
|----|---------------|---|--|
| | | <ul style="list-style-type: none"> + Số điện thoại (PhoneNumber1) + Số điện thoại 2 (PhoneNumber2) + Địa chỉ Email (Email) + Mật khẩu mã hóa (PasswordHash) + Mật khẩu băm (PasswordSalt) + Địa chỉ (Address) + Ảnh đại diện (Avatar) + Ngày khóa (BlockedTo) + Trạng thái hoạt động (IsOnline) + Id vai trò (RoleId) + Mã tin nhắn (MessageKey) | |
| 9 | AppRole | <ul style="list-style-type: none"> + Id vai trò (Id) + Tên vai trò (Name) + Mô tả vai trò (Desc) | |
| 10 | AppVerifyCode | <ul style="list-style-type: none"> + Id mã xác thực (Id) + Mã xác thực (TokenString) + Thời gian hết hạn mã (Expired) + Mã đã xác thực hay chưa (IsVerified) + Id người dùng (UserId) | |
| 11 | AppFriendShip | <ul style="list-style-type: none"> + Id bạn bè (Id) + Id của mình (UserId1) + Id của bạn bè (UserId2) | |

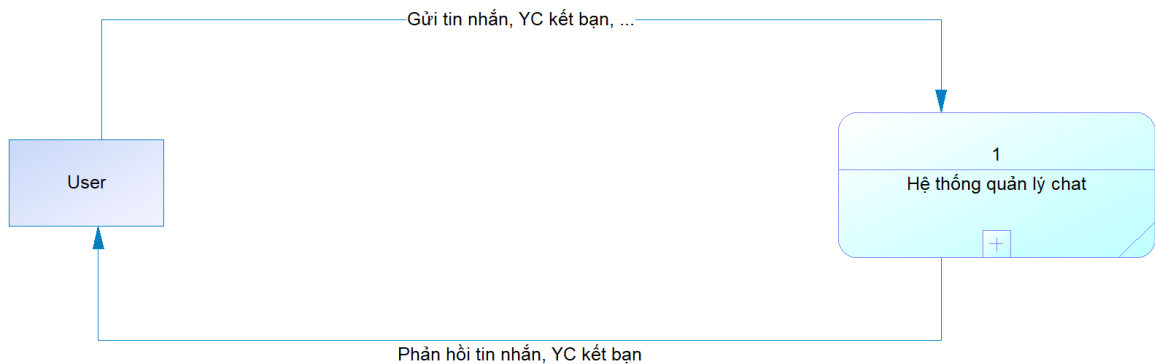
4.2 Mô hình quan hệ (ERD)



Hình 4.1 Mô hình quan hệ (ERD)

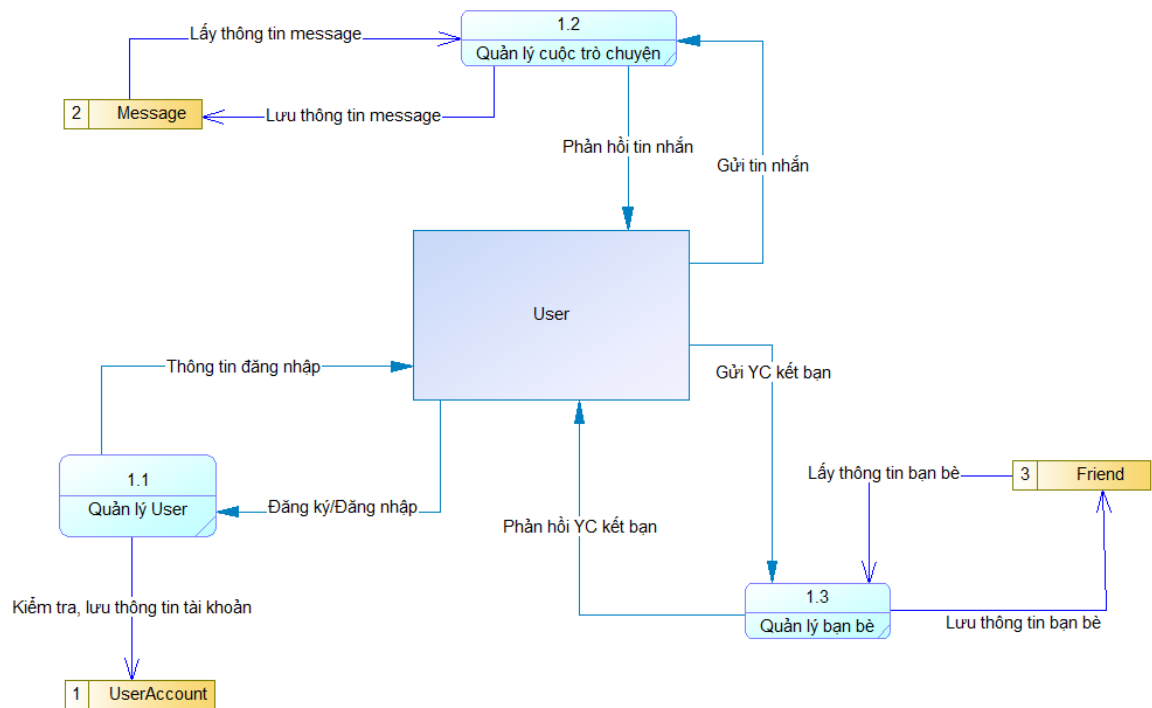
4.3 Mô hình luồng dữ liệu phân cấp (DFD)

Mức 0



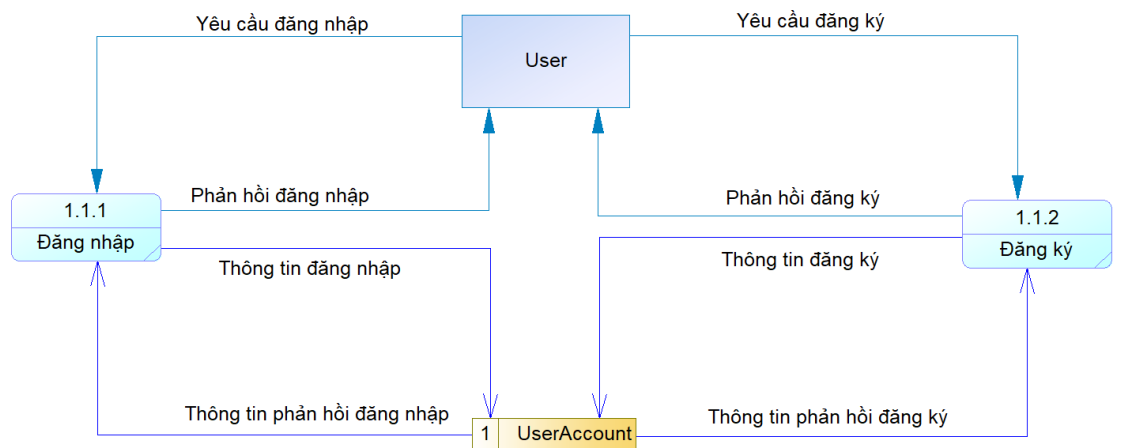
Hình 4.2 Mô hình DFD Mức 0

Mức 1



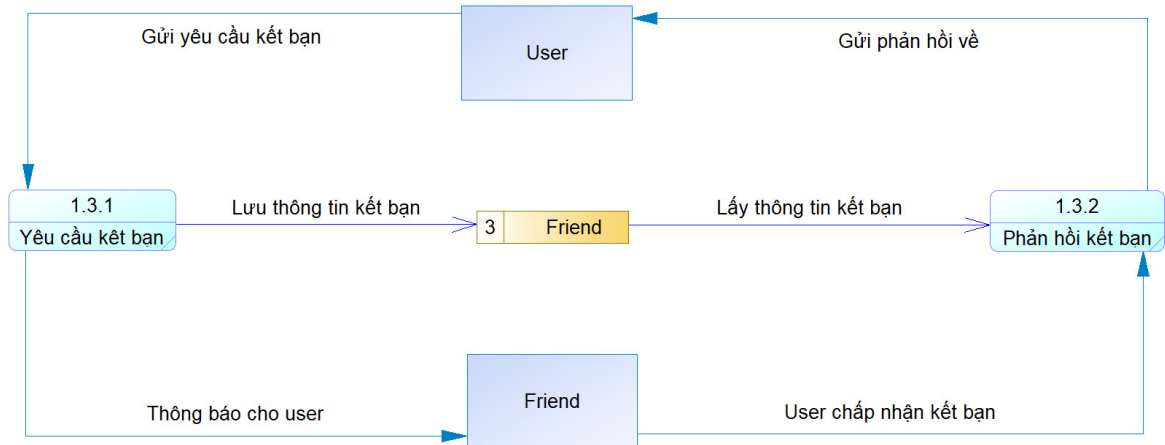
Hình 4.3 Mô hình DFD Mức 1

Mức 2.0: Quản lý user



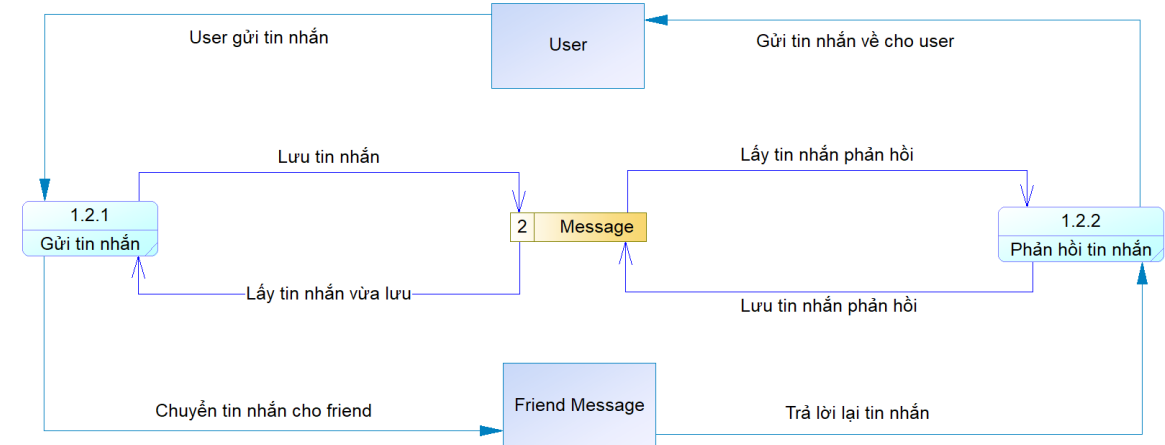
Hình 4.4 Mô hình DFD Mức 2.0

Mức 2.1: Quản lý bạn bè



Hình 4.5 Mô hình DFD Mức 2.1

Mức 2.2: Quản lý tin nhắn



Hình 4.6 Mô hình DFD Mức 2.2

4.4 Chuẩn hóa dữ liệu quan hệ

Bảng 4.2 AppMessage

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|----------------|--------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | ConversationId | Bigint | | | | | | x | x | Id cuộc trò chuyện |
| 3 | SenderId | Bigint | | | | | | | x | Id người gửi |

| | | | | | | | | | |
|----|--------------|-----------|------|--|--|--|---|---|------------------------------|
| 4 | ReceiverId | Bigint | | | | | | x | Id người nhận |
| 5 | Content | Nvarchar | 5000 | | | | | | Nội dung tin nhắn |
| 6 | Image | Nvarchar | 100 | | | | | | Hình ảnh tin nhắn |
| 7 | IsLiked | Boolean | | | | | | | Thích tin nhắn |
| 8 | SentTime | DateTime | | | | | | | Ngày giờ gửi |
| 9 | Created At | Datetime | | | | | x | | Thời gian tạo bài viết |
| 10 | Updated At | Datetime | | | | | | | Thời gian cập nhật bài viết |
| 11 | Published At | Datetime | | | | | | | Thời gian công khai bài viết |
| 12 | IsSeen | Boolean | | | | | | | Đã xem tin nhắn |
| 13 | IsNotify | Boolean | | | | | x | | Tin nhắn dạng thông báo |
| 14 | DeleteDate | Datetime2 | 7 | | | | | | Ngày xóa bài viết |

Bảng AppMessage dùng để lưu trữ tin nhắn của người dùng với nội dung đã được mã hóa.

Bảng 4.3 AppConversation

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|-----------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | UserId1 | Bigint | | | | | | x | x | Id của user |
| 3 | UserId2 | Bigint | | | | | | x | x | Id của bạn bè |
| 4 | StartTime | DateTime | | | | | | | | Ngày bắt đầu nhắn |

Bảng AppConversation để cho biết các message thuộc cuộc trò chuyện nào, từ đó sẽ dễ dàng truy xuất dữ liệu và gửi cho người dùng.

Bảng 4.4 AppInfoConversation

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|----------------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | ConversationId | Bigint | | | | | | x | x | Id cuộc trò chuyện |
| 3 | ColorId | Bigint | | | | | | | x | Id màu sắc |
| 4 | MainEmoji | Nvarchar | 200 | | | | | | | Biểu cảm chính |

Bảng AppInfoConversation dùng để lưu các thông tin của cuộc trò chuyện như biểu cảm, màu sắc,...

Bảng 4.5 AppColorConversation

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|---------------------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 5 | BackgroundColorCode | Nvarchar | 200 | | | | | | | Mã màu nền |
| 6 | TextColorCode | Nvarchar | 200 | | | | | | | Mã màu chữ |

Bảng AppColorConversation dùng để lưu các thông tin màu sắc để người dùng có thể chọn lựa màu sắc để thay đổi cho cuộc trò chuyện.

Bảng 4.6 AppNickname

| TT | Tên | Kiểu | Kích thước | M IN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|----------------|----------|------------|------|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | ConversationId | Bigint | | | | | | x | x | Id cuộc trò chuyện |
| 3 | UserId | Bigint | | | | | | x | x | Id người dùng |
| 4 | Nickname | Nvarchar | 200 | | | | | | | Biệt danh |

Bảng AppNickname dùng để lưu biệt danh của người dùng với nhau.

Bảng 4.7 MstStatusRequest

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|------------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | StatusName | Nvarchar | 200 | | | | | | | Tên trạng thái |

Bảng MstStatusRequest dùng để chỉ trạng thái của yêu cầu kết bạn là đang gửi, đã chấp nhận hoặc đã từ chối.

Bảng 4.8 AppFriendRequest

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|-----|------|------------|-----|-----|------------|----------|----------|-----------------|-----------|
|----|-----|------|------------|-----|-----|------------|----------|----------|-----------------|-----------|

| | | | | | | | | | | |
|---|------------|--------|--|--|--|---|--|---|---|---|
| 1 | Id | Bigint | | | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | SenderId | Bigint | | | | | | x | x | Id người gửi |
| 3 | ReceiverId | Bigint | | | | | | x | x | Id người nhận |
| 4 | StatusId | Bigint | | | | | | x | x | Id trạng thái |

Bảng AppFriendRequest dùng để lưu thông báo của những người gửi kết bạn cho bạn bè.

Bảng 4.9 AppUser

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|--------------|----------|---------------|-----|-----|---------------|-------------|-------------|-----------------------|---|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | FullName | Nvarchar | 200 | | | | | x | | Họ tên người dùng |
| 5 | PhoneNumber1 | Nvarchar | 20 | | | | | | | Số điện thoại 1 |
| 6 | PhoneNumber2 | Nvarchar | 20 | | | | | | | Số điện thoại 2 |
| 7 | Email | Nvarchar | 100 | | | | x | x | | Địa chỉ Email |
| 8 | PasswordHash | Nvarchar | 200 | | | | | x | | Mật khẩu mã hóa |
| 9 | PasswordSalt | Nvarchar | 200 | | | | | x | | Mật khẩu mã hóa |
| 10 | Address | Nvarchar | 500 | | | | | | | Địa chỉ |
| 11 | Avatar | Nvarchar | 200 | | | | | | | Ảnh đại diện |
| 12 | BlockedTo | DateTime | | | | | | | | Khóa đến ngày |

| | | | | | | | | | | |
|----|------------|----------|-----|--|--|--|--|--|---|-------------|
| 13 | AppRoleId | Bigint | | | | | | | x | Id vai trò |
| 14 | MessageKey | Nvarchar | 200 | | | | | | | Mã tin nhắn |

Bảng User dùng để lưu thông tin tài khoản của người dùng khi muốn truy cập vào để nhắn tin.

Bảng 4.10 AppRole

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Bigint | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | Name | Nvarchar | 200 | | | | | | | Tên vai trò |
| 3 | Desc | Nvarchar | 500 | | | | | | | Mô tả |

Bảng AppRole dùng để phân quyền cho tài khoản.

Bảng 4.11 AppVerifyCode

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|-------------|----------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Int | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | CreateDate | Datetime | | | | | | | | Thời gian tạo |
| 3 | TokenString | Nvarchar | 200 | | | | | | | Mã xác thực |
| 4 | Expired | DateTime | | | | | | | | Ngày hết hạn |
| 5 | IsVerified | Boolean | | | | | | | | Mã đã xác thực hay chưa |

| | | | | | | | | | | |
|---|--------|--------|--|--|--|--|--|--|--|-------------------|
| 6 | UserId | Bigint | | | | | | | | Id người xác thực |
|---|--------|--------|--|--|--|--|--|--|--|-------------------|

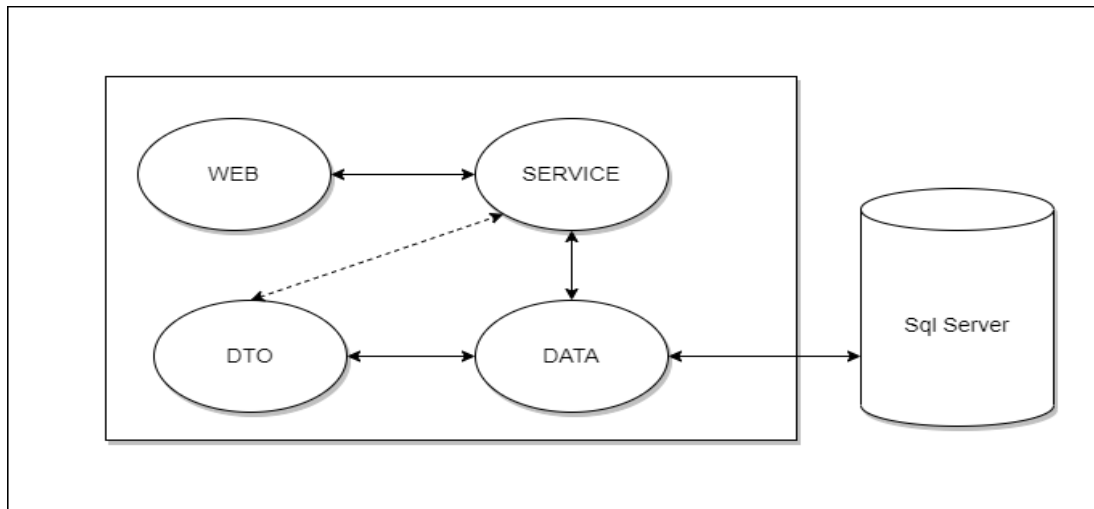
Bảng AppVerifyCode dùng để lưu mã code khi người dùng quên mật khẩu và sẽ gửi mã này qua mail của người dùng đó để xác nhận.

Bảng 4.12 AppFriendShip

| TT | Tên | Kiểu | Kích thước | MIN | MAX | Khóa chính | Duy nhất | NOT NULL | RBTV Khóa ngoại | Diễn giải |
|----|---------|--------|------------|-----|-----|------------|----------|----------|-----------------|-------------------------------------|
| 1 | Id | Int | | 1 | | x | | x | | Kiểu số tự tăng dùng làm khóa chính |
| 2 | UserId1 | Bigint | | | | | | x | x | Id người dùng |
| 3 | UserId2 | Bigint | | | | | | x | x | Id bạn bè |

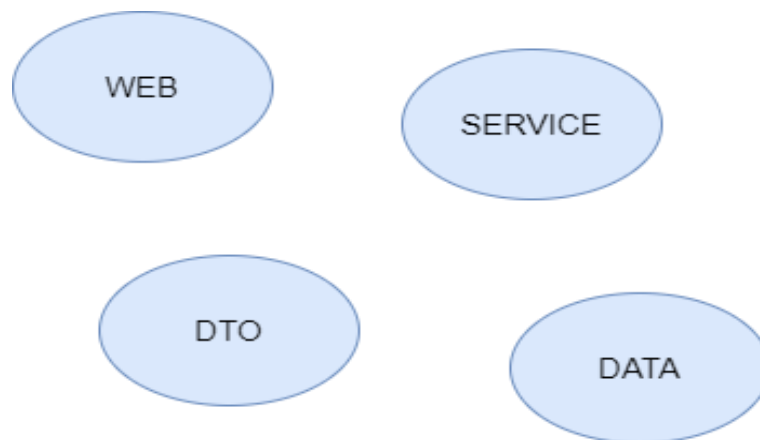
Bảng AppFriendShip dùng để lưu thông tin của những người đã là bạn bè với nhau.

4.5 Mô hình tổng thể kiến trúc



Hình 4.7 Mô hình tổng thể kiến trúc

4.5.1 Danh sách các Component/Package



Hình 4.8 Danh sách các Component

4.5.2 Giải thích tương tác, giao tiếp giữa các components

WEB: Có nhiệm vụ giao tiếp với người dùng. Nó bao gồm các thành phần giao diện và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp BUS

SERVICE: Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của WEB layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống DATA và lưu xuống hệ quản trị CSDL.

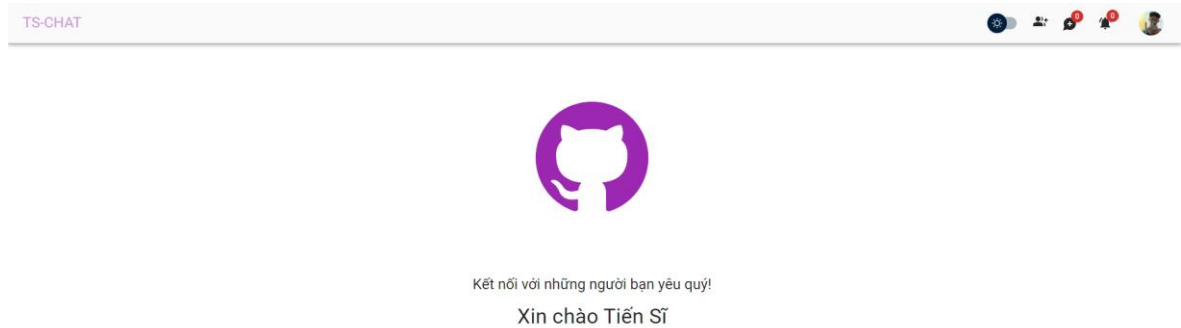
DATA: Giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu

DTO: Đây là một gói dữ liệu được trao đổi giữa các lớp, được xây dựng dưới dạng lớp đối tượng

CHƯƠNG 5: ĐẶC TẢ GIAO DIỆN

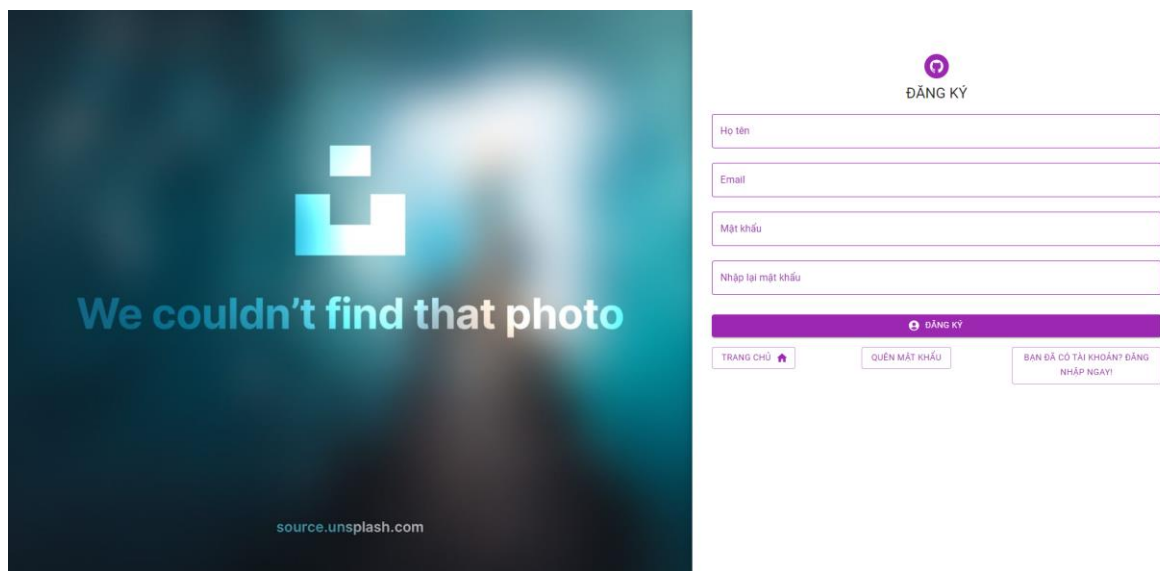
5.1 phần thiết kế giao diện cho user

5.1.1 Thiết kế màn hình chính



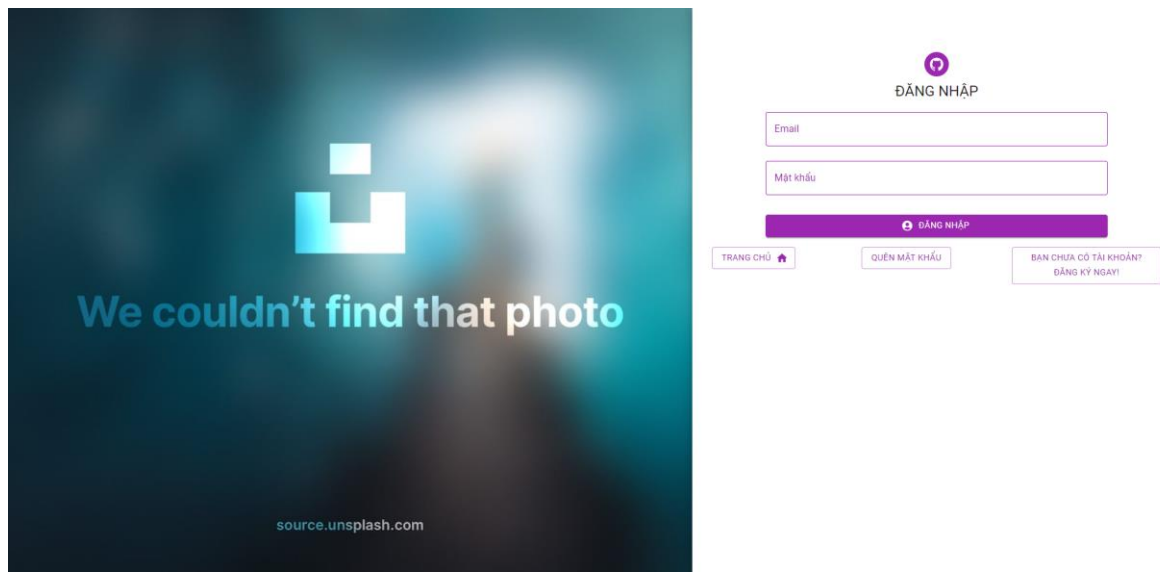
Hình 5.1 Giao diện màn hình chính

5.1.2 Thiết kế giao diện đăng ký



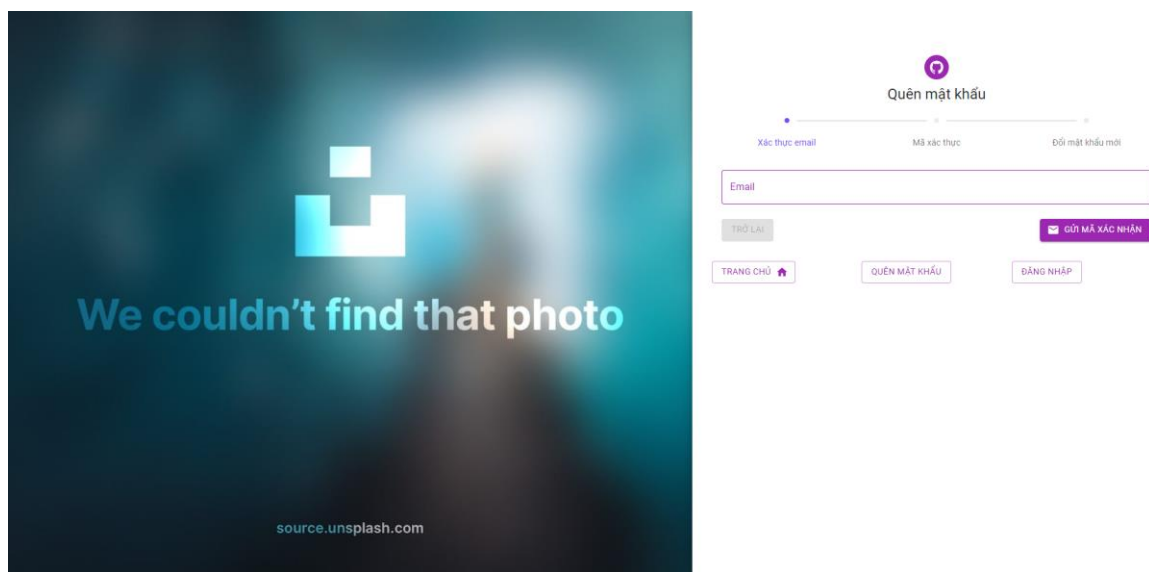
Hình 5.2 Giao diện đăng ký

5.1.3 Thiết kế giao diện đăng nhập admin với user



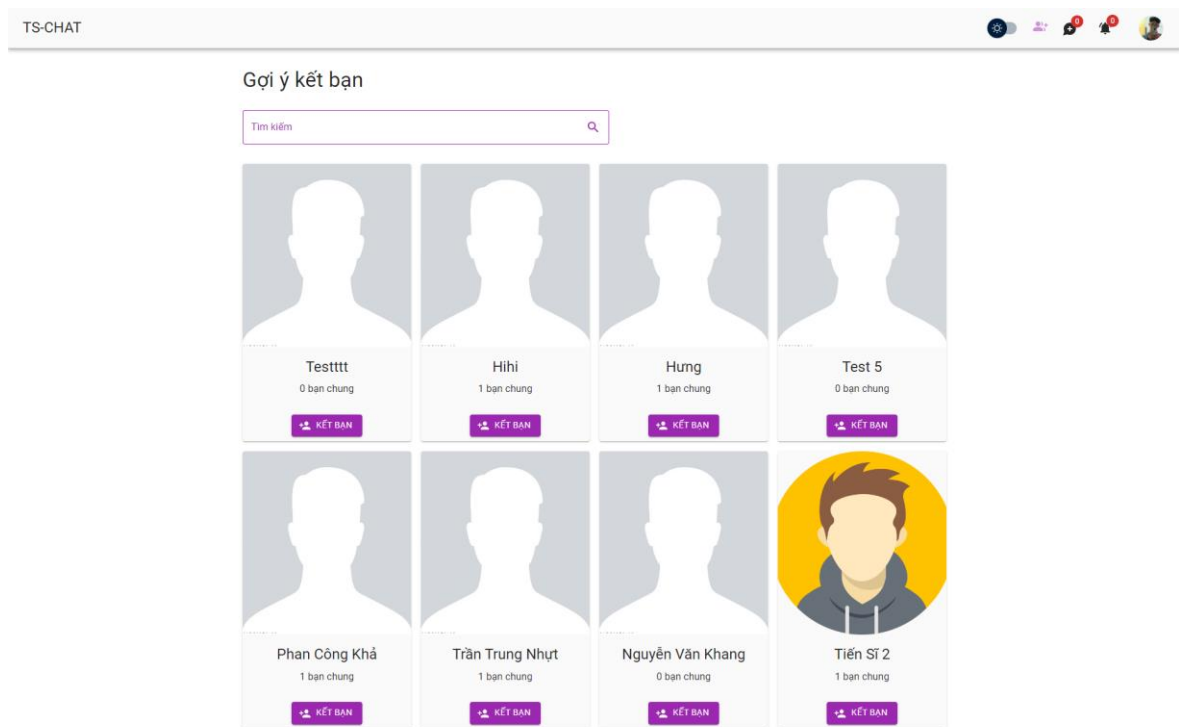
Hình 5.3 Giao diện đăng nhập admin và user

5.1.4 Giao diện quên mật khẩu



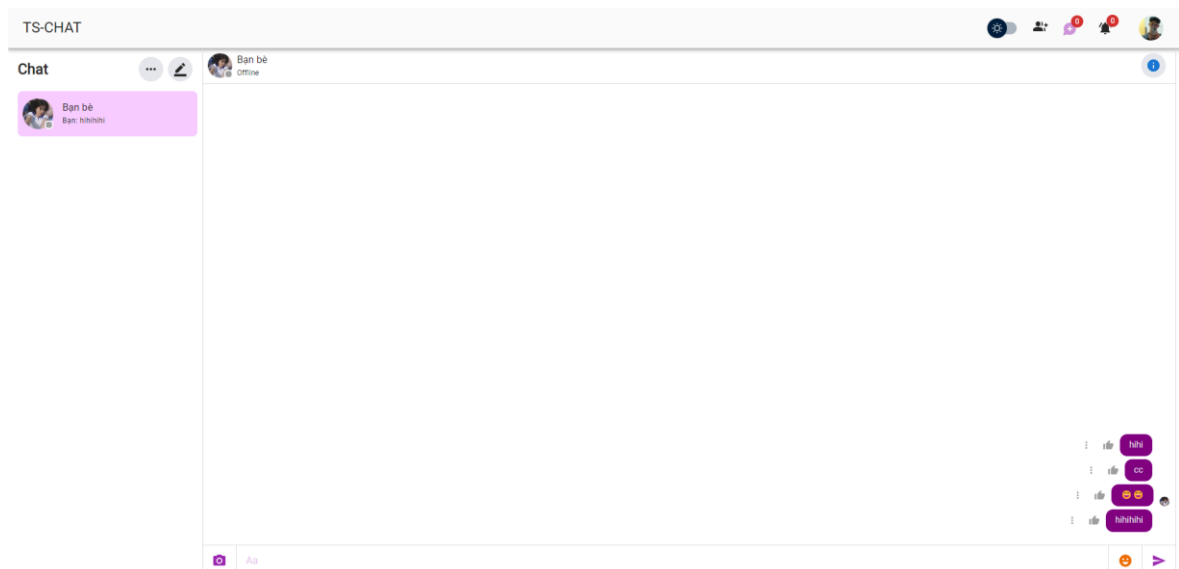
Hình 5.4 Giao diện quên mật khẩu

5.1.5 Giao diện gợi ý kết bạn



Hình 5.5 Giao diện gợi ý kết bạn


5.1.6 Giao diện nhắn tin



Hình 5.6 Giao diện nhắn tin

5.1.7 Giao diện quản lý thông tin tài khoản

TS-CHAT



Tiến Sĩ
Mật khẩu: *****

THÔNG TIN CÁ NHÂN

BẠN BÈ

Họ tên

Tiến Sĩ

Số điện thoại

0928666158

Địa chỉ






Thành phố Cần Thơ


LƯU

HỦY


Hình 5.7 Giao diện quản lý thông tin tài khoản

5.1.8 Giao diện thông báo







Test 5 đã chấp nhận lời mời kết bạn của bạn.
0 bạn chung




Bích Ngọc đã chấp nhận lời mời kết bạn của bạn.
1 bạn chung




Nguyễn Văn Khang đã chấp nhận lời mời kết bạn của bạn.
0 bạn chung




Phan Công Khả đã chấp nhận lời mời kết bạn của bạn.
1 bạn chung



Bích Ngọc đã chấp nhận lời mời kết bạn của bạn.
1 bạn chung



Bích Ngọc đã chấp nhận lời mời kết bạn của bạn.
1 bạn chung

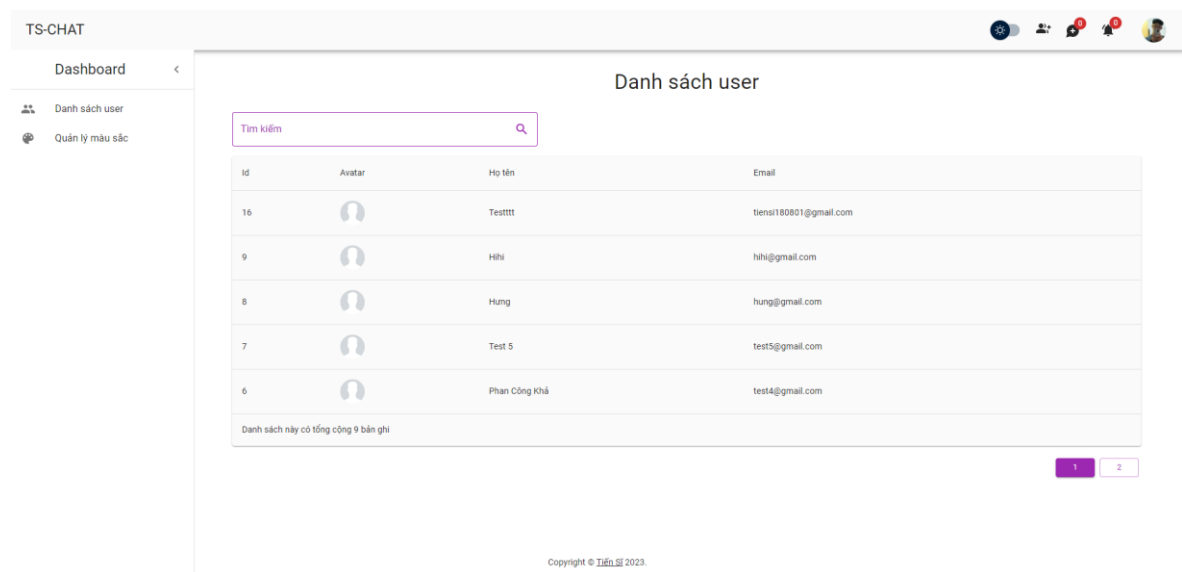


Bích Ngọc đã chấp nhận lời mời kết bạn của bạn.
1 bạn chung

Hình 5.8 Giao diện thông báo

5.2 Giao diện cho quản lý admin

5.2.1 Giao diện quản lý user



Hình 5.9 Giao diện quản lý user

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Nhận xét: Sau khi hoàn thiện thì website có thể đáp ứng được nhu cầu sử dụng của người dùng tuy nhiên đây là lần đầu tiên tôi xây dựng nên phần mềm vẫn còn nhiều khuyết điểm và chưa hoàn hảo như những website chat khác trên thị trường và vẫn còn thiếu nhiều chức năng.

6.2. Ưu điểm:

- Giao diện thân thiện với người dùng.
- Không có nhiều thao tác phức tạp giúp người dùng dễ dàng tiếp cận học hỏi nhanh và dễ dàng sử dụng.
- Giải quyết việc nhắn tin nhanh chóng thông qua website, tiết kiệm nhiều thời gian.

6.3. Nhược điểm: Chức năng, giao diện vẫn chưa chuyên nghiệp so với các website đã góp mặt lâu trên thị trường.

6.4. Hướng phát triển: Từ giao diện chương trình hoạt động trên nền tảng Web có thể phát triển thêm khả năng tương thích về giao diện khi chạy trên nền tảng di động (Android, Ios,..) để người dùng có thể tương tác nhiều hơn trên website. Vì website làm ra vẫn còn thiếu các chức năng như gọi điện thoại, video call, gửi nhiều ảnh, thay đổi màu sắc cuộc trò chuyện để tăng trải nghiệm người dùng và phần quản lý vẫn chưa có nhiều chức năng nên sẽ hoàn thiện các chức năng này trong tương lai gần.

TÀI LIỆU THAM KHẢO

- [1]. Dương Anh Đức, Giáo trình phân tích và thiết kế hướng đối tượng bằng UML.
- [2]. Huỳnh Anh Đức, Kỹ thuật và ứng dụng UML, NXB lao động.
- [3]. Nguyễn Tiến – Ngô Quốc Việt, Kỹ thuật và ứng dụng bằng UML, NXB thống kê.
- [4]. Phạm Hữu Khang ,C# 2005: Lập trình Windows Form (Tập 2).
- [5]. Phạm Hữu Khang, C# 2005: Lập trình hướng đối tượng (Tập 3).
- [6]. Phạm Hữu Khang, C# 2005: Lập trình cơ sở dữ liệu(Tập 4).
- [7]. Đề cương bài giảng: Lý thuyết cơ sở dữ liệu, giáo trình trường ĐHSP KT Hưng Yên.
- [8]. Đề cương bài giảng: Lập trình hướng đối tượng, giáo trình trường ĐHSP KT Hưng Yên.