

# TRƯỜNG ĐẠI HỌC SÀI GÒN

## KHOA CÔNG NGHỆ THÔNG TIN



### HỌC PHẦN: XÂY DỰNG PHẦN MỀM THEO MÔ HÌNH PHÂN LỚP

Dự án Clone Instagram

Nhóm sinh viên thực hiện:

Họ và tên	MSSV
Văn Tuấn Kiệt	3122410202
Mai Phúc Lâm	3122410207
Nguyễn Đức Duy Lâm	3122410208
Nguyễn Hữu Lộc	3122410213
Hồ Hưng Lộc	3122410219
Vinh	3122410213

Giáo viên hướng dẫn: Từ Lãng Phiêu

TP.HCM, 2025

## **MỤC LỤC**

# Mục lục

<b>1</b>	<b>Lời cảm ơn</b>	<b>4</b>
<b>2</b>	<b>Giới thiệu</b>	<b>5</b>
<b>3</b>	<b>Lý do chọn đề tài</b>	<b>6</b>
<b>4</b>	<b>Yêu cầu chức năng</b>	<b>7</b>
<b>5</b>	<b>Yêu cầu phi chức năng</b>	<b>8</b>
<b>6</b>	<b>Cấu trúc mã nguồn</b>	<b>9</b>
6.1	Mô hình ứng dụng . . . . .	9
6.2	Công cụ . . . . .	14
6.3	Kiến trúc phần mềm . . . . .	16
<b>7</b>	<b>Các tính năng được xây dựng</b>	<b>19</b>
<b>8</b>	<b>Sơ đồ ERD</b>	<b>21</b>
8.1	Entity User . . . . .	21
8.2	Entity Conversation . . . . .	22
8.3	Entity ConversationMember . . . . .	22
8.4	Entity Post . . . . .	23
8.5	Entity Message . . . . .	23
8.6	Entity Comment . . . . .	24
8.7	Entity Notification . . . . .	24
8.8	Entity Friend . . . . .	25
8.9	Entity PostEmotion . . . . .	25
<b>9</b>	<b>Kết quả đạt được</b>	<b>26</b>
9.1	Đăng ký . . . . .	26
9.2	Đăng nhập . . . . .	27

9.3	Chức năng đăng bài viết . . . . .	28
9.3.1	Hình ảnh . . . . .	28
9.3.2	Video . . . . .	30
9.4	Chức năng tương tác . . . . .	30
9.4.1	Like . . . . .	30
9.4.2	Comment . . . . .	32
9.5	Chức năng quản lí bạn bè . . . . .	33
9.5.1	Thêm bạn . . . . .	33
9.5.2	Huỷ kết bạn . . . . .	34
9.5.3	Danh sách bạn bè . . . . .	35
9.6	Chức năng quản lí hồ sơ cá nhân . . . . .	36
9.6.1	Cập nhật thông tin người dùng . . . . .	36
9.6.2	Cập nhật ảnh đại diện . . . . .	38
9.7	Trang Admin . . . . .	38
9.7.1	quản lí user . . . . .	38
9.7.2	quản lí post . . . . .	39
9.8	Nhắn tin real time . . . . .	39
9.9	Gọi real time . . . . .	41
9.10	Thông báo . . . . .	43
<b>10</b>	<b>Hướng phát triển</b>	<b>44</b>
10.1	Hạn chế . . . . .	44
10.2	Hướng phát triển trong tương lai . . . . .	44

# **1 Lời cảm ơn**

Trước hết, chúng em xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn, người đã tận tình giảng dạy, hỗ trợ và định hướng giúp chúng em hoàn thành dự án này. Những kiến thức mà thầy/cô truyền đạt không chỉ giúp chúng em trong quá trình thực hiện đồ án mà còn là nền tảng quan trọng cho việc phát triển bản thân trong tương lai.

Bên cạnh đó, chúng em xin gửi lời cảm ơn đến gia đình và bạn bè, những người luôn động viên và tạo điều kiện thuận lợi để chúng em có thể tập trung hoàn thành dự án một cách tốt nhất. Sự ủng hộ của mọi người là động lực lớn giúp chúng em vượt qua những khó khăn trong quá trình nghiên cứu và triển khai.

Cuối cùng, chúng em xin cảm ơn các tài liệu tham khảo, cộng đồng lập trình viên, và các diễn đàn công nghệ đã cung cấp nhiều nguồn kiến thức quý giá, giúp chúng em giải quyết những vấn đề gặp phải trong quá trình thực hiện dự án.

## 2 Giới thiệu

Mạng xã hội ngày nay đóng vai trò quan trọng trong cuộc sống con người, giúp kết nối mọi người trên toàn thế giới, chia sẻ thông tin và tạo ra nhiều giá trị về mặt cá nhân lẫn kinh doanh. Trong số đó, Instagram là một trong những nền tảng nổi bật nhất với hàng triệu người dùng hoạt động mỗi ngày, cung cấp trải nghiệm chia sẻ hình ảnh, video và tương tác trực quan.

Dự án **Clone Instagram** được xây dựng với mục tiêu mô phỏng lại các tính năng quan trọng của Instagram, bao gồm:

- **Đăng ký, đăng nhập:** Người dùng có thể tạo tài khoản, đăng nhập và chỉnh sửa thông tin cá nhân.
- **Đăng bài:** Cho phép người dùng đăng tải hình ảnh, viết chú thích và sử dụng hashtag.
- **Thả tim, bình luận:** Người dùng có thể tương tác với bài đăng bằng cách thả tim và để lại bình luận.
- **Nhắn tin:** Hỗ trợ tính năng nhắn tin riêng tư giữa các người dùng.
- **Quản lý hồ sơ cá nhân:** Người dùng có thể xem, chỉnh sửa hồ sơ và theo dõi các hoạt động của mình.

Dự án sử dụng **ReactJS** làm thư viện chính để xây dựng giao diện, kết hợp với **Firebase** hoặc **Node.js** để quản lý dữ liệu. Ngoài ra, các thư viện hỗ trợ như **Redux**, **React Router**, **Ant Design**, **Tailwind CSS** cũng được sử dụng để tối ưu trải nghiệm người dùng và nâng cao hiệu suất ứng dụng.

### 3 Lý do chọn đề tài

Mạng xã hội ngày càng phát triển mạnh mẽ, trở thành công cụ không thể thiếu trong đời sống hiện đại. Trong đó, Instagram nổi bật nhờ giao diện tối giản, thân thiện và tích hợp nhiều tính năng hấp dẫn như chia sẻ ảnh, video ngắn, stories, nhẫn tin. Với nhu cầu ngày càng cao trong việc phát triển ứng dụng mạng xã hội, việc tìm hiểu và xây dựng một dự án mô phỏng Instagram là cơ hội tốt để sinh viên:

- **Nâng cao kỹ năng lập trình:** Dự án sử dụng ReactJS giúp sinh viên hiểu rõ hơn về cách phát triển giao diện động, tối ưu hiệu suất và quản lý dữ liệu.
- **Làm quen với kiến trúc ứng dụng hiện đại:** Clone Instagram giúp áp dụng nhiều công nghệ phổ biến như Redux, REST API.
- **Phát triển tư duy UI/UX:** Clone lại Instagram giúp sinh viên học hỏi cách thiết kế giao diện trực quan, thân thiện với người dùng.
- **Ứng dụng thực tế cao:** Sản phẩm có thể mở rộng thêm nhiều tính năng mới, phục vụ học tập hoặc phát triển thành một dự án hoàn chỉnh.

## 4 Yêu cầu chức năng

### Quản lý bài post

- **Thêm bài post:** Đăng tải bài viết kèm hình ảnh, video, văn bản.
- **Chia sẻ bài post:** Thả cảm xúc, hiển thị danh sách bài viết.
- **Bình luận bài post:** Văn bản, emoji, hình ảnh.

### Quản lý bình luận

- **Thêm comment:** Thả cảm xúc, trả lời comment.
- **Lọc comment:** Theo nội dung, thời gian, người bình luận.

### Quản lý tin nhắn

- **Gửi tin nhắn:** Văn bản, gọi thoại/video.
- **Hiển thị trạng thái:** Tin nhắn, trạng thái hoạt động.

### Quản lý hồ sơ cá nhân

- **Chỉnh sửa thông tin:** Cập nhật avatar.
- **Xem danh sách bài post:** Chính sửa bài viết.

### Quản lý danh sách bạn bè

- **Gửi, hủy, xác nhận:** Lời mời kết bạn.
- **Chặn bạn:** Xem danh sách bạn bè, tìm kiếm bạn bè.

## 5 Yêu cầu phi chức năng

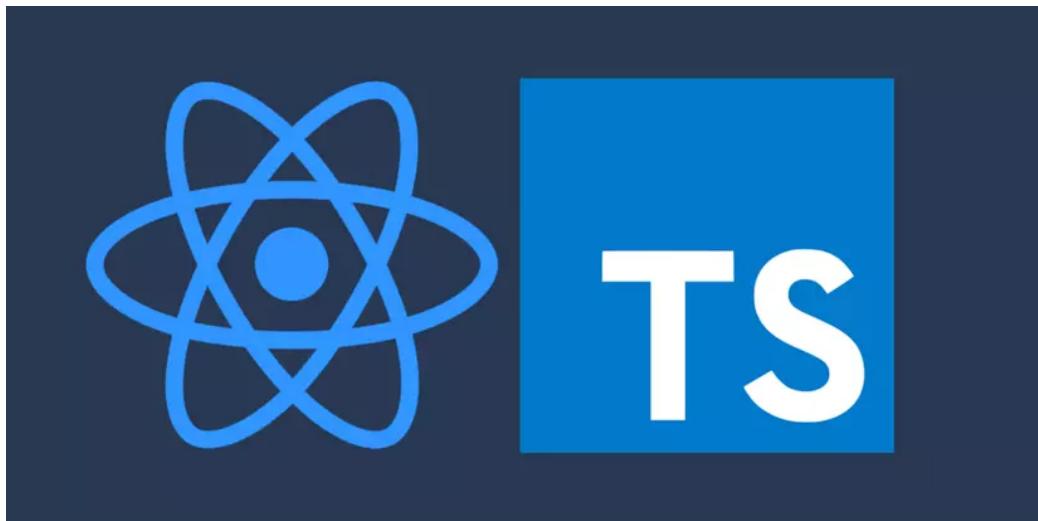
- **Hiệu suất:** Tải trang nhanh, tìm kiếm và hiển thị dữ liệu tối ưu.
- **Bảo mật:** Mã hóa dữ liệu, bảo vệ quyền riêng tư, chống tấn công.
- **Trải nghiệm người dùng:** Giao diện thân thiện, điều hướng dễ dàng.
- **Khả năng mở rộng:** Hỗ trợ nhiều người dùng, dữ liệu lớn.
- **Độ tin cậy:** Ổn định, uptime  $\geq 99.9\%$ .

# 6 Cấu trúc mã nguồn

## 6.1 Mô hình ứng dụng

Frontend:

- **ReactJS:** React là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook vào năm 2013, nhằm xây dựng giao diện người dùng cho các ứng dụng web. React cho phép các lập trình viên phát triển các thành phần giao diện (components) một cách hiệu quả, dễ bảo trì và tái sử dụng. Với React, người dùng có thể trải nghiệm giao diện mượt mà và dễ dàng tương tác với ứng dụng.



Backend:

- **Spring Boot** là một framework mã nguồn mở dựa trên nền tảng Spring Framework, được thiết kế để đơn giản hóa việc phát triển các ứng dụng Java, đặc biệt là các ứng dụng web và microservices. Spring Boot được Pivotal Software ra mắt lần đầu tiên vào năm 2014, với mục tiêu giảm thiểu cấu hình thủ công bằng cách cung cấp các tính năng tự động cấu

hình (auto-configuration), các thư viện tích hợp sẵn, và khả năng chạy độc lập (embedded server). Framework này hỗ trợ nhiều tính năng như RESTful API, WebSocket, quản lý cơ sở dữ liệu (through qua Spring Data JPA), và tích hợp với các dịch vụ bên thứ ba như RabbitMQ, Cloudinary. Spring Boot được sử dụng rộng rãi trong các dự án phát triển phần mềm doanh nghiệp, và tương thích với nhiều nền tảng như Windows, macOS, và Linux.



#### Hệ quản trị cơ sở dữ liệu:

- **MySQL:** là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS), thuộc quyền sở hữu của Oracle, được sử dụng để quản lý và lưu trữ dữ liệu. Nó sử dụng SQL (Structured Query Language) làm ngôn ngữ chính để truy vấn và thao tác với cơ sở dữ liệu. MySQL phổ biến trong các ứng dụng web, đặc biệt là các ứng dụng sử dụng kiến trúc LAMP (Linux, Apache, MySQL, PHP/Python/Perl). Các ứng dụng web lớn nhất như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Nó đã tương thích với nhiều hệ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.



### Giao thức truyền thông:

- **WebSocket:** là một giao thức truyền thông giúp cho việc thiết lập kênh truyền thông hai chiều giữa máy chủ và máy khách. WebSocket hoạt động bằng cách thiết lập kết nối HTTP liên tục với máy chủ và sau đó nâng cấp nó lên kết nối websocket hai chiều bằng cách gửi Upgrade header. WebSocket được hỗ trợ trong hầu hết các trình duyệt web hiện đại và cho các trình duyệt không hỗ trợ, chúng tôi có các thư viện cung cấp dự phòng cho các kỹ thuật khác như Comet và HTTP Long Polling.



### Dịch vụ bên thứ ba:

- **Cloudinary:** Cloudinary là một nền tảng quản lý hình ảnh và video dựa trên đám mây (cloud-based media management platform), được thành lập vào năm 2011 tại Israel, nhằm hỗ trợ các nhà phát triển xử lý và phân phối nội dung truyền thông một cách hiệu quả. Nền tảng này cho phép tải lên, lưu trữ, tối ưu hóa và phân phối hình ảnh hoặc video thông qua mạng phân phối nội dung (CDN), giúp tăng tốc độ tải và cải thiện trải nghiệm người dùng. Cloudinary tự động điều chỉnh kích thước, nén, và chuyển đổi định dạng hình ảnh (như WebP, AVIF) để phù hợp với thiết bị, đồng thời cung cấp các tính năng như cắt, thêm hiệu ứng, hoặc chèn watermark. Được sử dụng rộng rãi trong các ứng dụng web và di động, Cloudinary là lựa chọn phổ biến cho các nền tảng mạng xã hội, thương mại điện tử như Shopify, và các công ty lớn như TED hay Bleacher Report. Nền tảng này tích hợp dễ dàng với nhiều công nghệ, bao gồm AWS S3, và hỗ trợ các SDK cho các ngôn ngữ lập trình như JavaScript, Python, và Java.



- **AWS S3:** AWS S3 (Amazon Simple Storage Service) là một dịch vụ

lưu trữ đối tượng (object storage) dựa trên đám mây, được Amazon Web Services (AWS) ra mắt vào năm 2006, nhằm cung cấp giải pháp lưu trữ dữ liệu an toàn, bền bỉ và có khả năng mở rộng cao. S3 cho phép người dùng lưu trữ bất kỳ loại dữ liệu nào, từ hình ảnh, video, đến tài liệu, dưới dạng các đối tượng (objects) trong các "bucket" (thùng chứa), với dung lượng tối đa lên đến 5TB cho mỗi đối tượng. Dịch vụ này đảm bảo độ bền 99.999999999



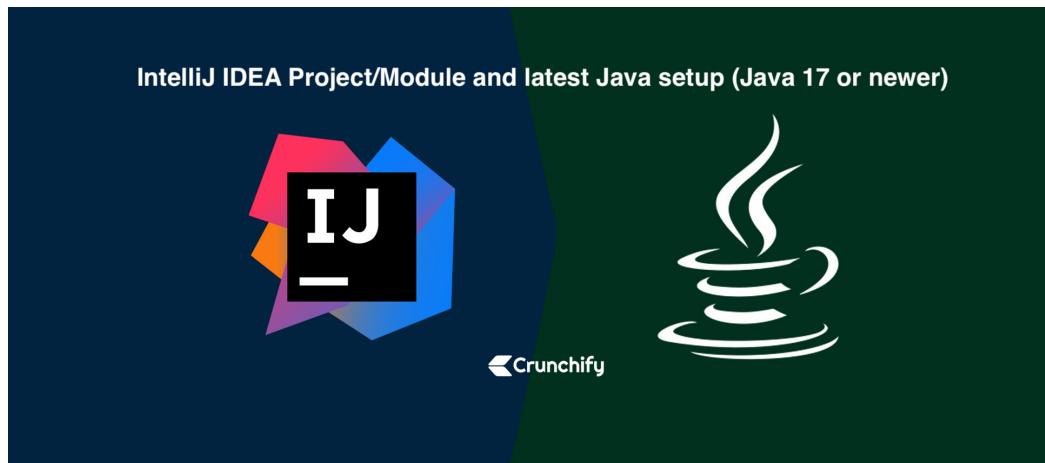
- **RabbitMQ:** RabbitMQ là một phần mềm mã nguồn mở hoạt động như một trình môi giới tin nhắn (message broker), được phát triển bởi Pivotal Software vào năm 2007, nhằm hỗ trợ giao tiếp bất đồng bộ giữa các ứng dụng hoặc dịch vụ. RabbitMQ triển khai giao thức AMQP (Advanced Message Queuing Protocol), cho phép nhận tin nhắn từ một ứng dụng (producer), lưu trữ chúng trong hàng đợi (queue), và chuyển tiếp đến ứng dụng khác (consumer) một cách đáng tin cậy. Phần mềm này đảm bảo tin nhắn không bị mất bằng cách lưu trữ chúng cho đến khi được xử lý, đồng thời hỗ trợ các giao thức khác như MQTT, STOMP, và WebSocket để phù hợp với nhiều loại ứng dụng. RabbitMQ được sử dụng rộng rãi trong các hệ thống phân tán, như xử lý thông báo, tác vụ nền, hoặc giao tiếp giữa các microservices, với các khách hàng lớn như Reddit, 9GAG, và Trello. Nó tương thích với nhiều nền tảng như Linux, Windows, và macOS, và có thể được triển khai trên các dịch vụ đám mây như Amazon MQ hoặc CloudAMQP.



## 6.2 Công cụ

IntelliJ IDEA:

- IntelliJ IDEA là môi trường phát triển tích hợp (IDE) đầu tiên do JetBrains phát triển, được thiết kế chủ yếu cho các công nghệ dựa trên JVM như Java, Groovy, Kotlin và Scala, đồng thời hỗ trợ Java EE và phát triển web. Phiên bản đầu tiên của IntelliJ IDEA được phát hành vào tháng 1 năm 2001. Trong dự án Instagram clone, công cụ này được sử dụng để viết mã và phát triển API backend bằng Spring Boot, giúp tăng hiệu suất lập trình với các tính năng như gợi ý mã thông minh, kiểm tra lỗi, và tích hợp công cụ quản lý dự án.



### Visual Studio Code (VS Code):

- Visual Studio Code là một trình soạn thảo mã nguồn nhẹ nhưng mạnh mẽ, hỗ trợ đa nền tảng như Windows, macOS và Linux. Với các tính năng như hoàn thành mã thông minh, tích hợp Git, và hỗ trợ nhiều tiện ích mở rộng, VS Code được sử dụng để phát triển frontend của dự án Instagram clone với ReactJS. Công cụ này giúp lập trình viên dễ dàng viết mã, gỡ lỗi, và quản lý các thư viện như Ant Design, Redux, và TailwindCSS một cách hiệu quả.

### Postman:

- Postman là một công cụ phổ biến được sử dụng để kiểm thử và làm việc với các API, đặc biệt là API kiểu REST. API đóng vai trò quan trọng trong việc kết nối các thành phần của ứng dụng, và Postman giúp đơn giản hóa quá trình gọi và kiểm tra API mà không cần viết mã. Công cụ này hỗ trợ tất cả các phương thức HTTP như GET, POST, PUT, PATCH, DELETE, và cho phép lưu lại lịch sử các yêu cầu (request) để sử dụng lại khi cần. Trong dự án Instagram clone, Postman được sử dụng để kiểm tra các API như đăng bài post (POST /api/post), lấy danh sách bài post (GET /api/post), và gửi tin nhắn.

### MySQL Workbench:

- MySQL Workbench là một công cụ quản lý cơ sở dữ liệu MySQL, cung cấp giao diện đồ họa để phát triển, thiết kế và quản lý cơ sở dữ liệu một cách trực quan. Công cụ này cho phép dễ dàng tạo, chỉnh sửa cơ sở dữ liệu, thực hiện các thao tác như đảo ngược (reverse engineering) để tạo mô hình từ cơ sở dữ liệu hiện có, và chuyển tiếp (forward engineering) để triển khai mô hình thành cơ sở dữ liệu. Trong dự án Instagram clone,

MySQL Workbench được sử dụng để thiết kế và quản lý cơ sở dữ liệu MySQL, bao gồm các bảng như users, posts, comments, và messages.



#### GitHub:

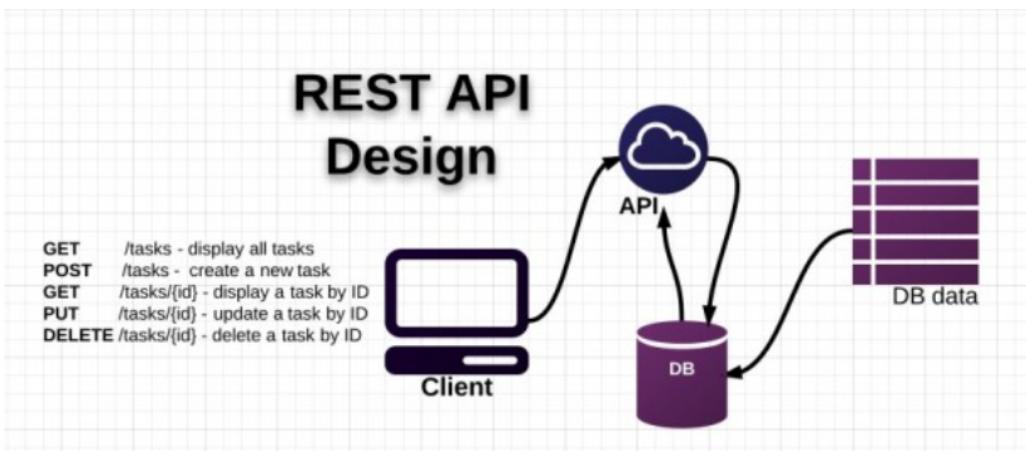
- GitHub là một nền tảng quản lý mã nguồn phổ biến, cho phép các lập trình viên chia sẻ, cộng tác và quản lý phiên bản mã nguồn một cách hiệu quả. Sự phát triển của GitHub bắt đầu vào ngày 19 tháng 10 năm 2007, và trang web chính thức được ra mắt vào tháng 4 năm 2008 bởi Tom Preston-Werner, Chris Wanstrath, và PJ Hyett. Microsoft đã mua lại GitHub vào tháng 6 năm 2018. Trong dự án Instagram clone, GitHub được sử dụng để lưu trữ mã nguồn, quản lý phiên bản, và hỗ trợ làm việc nhóm giữa các thành viên phát triển frontend và backend.

### 6.3 Kiến trúc phần mềm

#### RESTful API:

- **Khái niệm:** RESTful API là một tiêu chuẩn được sử dụng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) nhằm mục đích quản lý các tài nguyên (resource) một cách hiệu quả. Nó tập trung vào các tài nguyên hệ thống như tệp văn bản, hình ảnh, âm thanh, video, hoặc dữ liệu động, bao gồm các trạng thái tài nguyên được định dạng và truyền tải thông qua giao thức HTTP. RESTful API cho phép các ứng dụng giao tiếp với nhau thông qua các phương thức HTTP

chuẩn, giúp đơn giản hóa việc truy cập và xử lý dữ liệu trên các nền tảng khác nhau.



### Diễn giải các thành phần

- **API (Application Programming Interface):** API là một tập hợp các quy tắc và cơ chế cho phép một ứng dụng hoặc thành phần tương tác với một ứng dụng hoặc thành phần khác. API đóng vai trò như một cầu nối, giúp các ứng dụng trao đổi dữ liệu với nhau một cách dễ dàng. Dữ liệu trả về từ API thường được định dạng ở các kiểu phổ biến như JSON hoặc XML, phù hợp để tích hợp vào các ứng dụng web hoặc di động.
- **REST (REpresentational State Transfer):** REST là một kiểu kiến trúc (architectural style) được sử dụng để thiết kế API, dựa trên việc chuyển đổi trạng thái biểu diễn của tài nguyên. REST tận dụng các phương thức HTTP đơn giản như GET, POST, PUT, DELETE để thực hiện các thao tác trên tài nguyên. Thay vì sử dụng một URL để xử lý thông tin phức tạp, REST gửi các yêu cầu HTTP đến một URL cụ thể nhằm truy xuất hoặc chỉnh sửa dữ liệu, giúp giao tiếp giữa các hệ thống trở nên trực quan và hiệu quả.
- **RESTful API:** RESTful API là một tiêu chuẩn thiết kế API cho các ứng dụng web, tập trung vào việc quản lý các tài nguyên (resource) và cho phép các ứng dụng khác nhau (web, mobile, v.v.) giao tiếp với nhau. Chức năng quan trọng nhất của RESTful API là quy định cách sử dụng các phương thức HTTP (như GET để lấy dữ liệu, POST để tạo mới, PUT để cập nhật, DELETE để xóa) và cách định dạng URL để truy cập các tài nguyên. RESTful API không quy định logic mã nguồn của ứng dụng và không bị giới hạn bởi ngôn ngữ lập trình, do đó bất kỳ

ngôn ngữ hoặc framework nào (như Java với Spring Boot, JavaScript với Node.js) cũng có thể được sử dụng để thiết kế một RESTful API.

# 7 Các tính năng được xây dựng

## Các tính năng được xây dựng

### Quản lý bài post

- **Thêm bài post:** Đăng tải bài viết kèm hình ảnh, video, văn bản. Hình ảnh và video được upload lên Cloudinary thông qua API POST /api/post.
- **Chia sẻ bài post:** Thả cảm xúc (like), hiển thị danh sách bài viết, chia sẻ bài post.
- **Bình luận bài post:** Hỗ trợ văn bản, emoji (dùng @emoji-mart/react), và hình ảnh (tích hợp Cloudinary).
- **Xóa bài post:** Người dùng có thể xóa bài post của mình thông qua API DELETE /api/post/{id}.

### Quản lý bình luận

- **Thêm comment:** Thả cảm xúc (like), trả lời comment (hỗ trợ comment lồng nhau).
- **Lọc comment:** Theo nội dung, thời gian, hoặc người bình luận, giúp người dùng dễ dàng tìm kiếm bình luận.
- **Xóa comment:** Người dùng có thể xóa bình luận của mình hoặc của người khác (nếu là chủ bài post).

### Quản lý tin nhắn

- **Gửi tin nhắn:** Hỗ trợ văn bản, hình ảnh.
- **Hiển thị trạng thái:** Hiển thị trạng thái tin nhắn (đã gửi, đã xem), trạng thái hoạt động của người dùng (online/offline).

- **Tạo nhóm chat:** Hỗ trợ tạo nhóm chat với tối đa 100 người, tích hợp với module **Friends** để thêm thành viên.

### Quản lý hồ sơ cá nhân

- **Chỉnh sửa thông tin:** Cập nhật avatar (upload lên AWS S3), chỉnh sửa thông tin như tên, email, mô tả.
- **Xem danh sách bài post:** Hiển thị tất cả bài post của người dùng, cho phép chỉnh sửa hoặc xóa bài viết.
- **Xem thông tin công khai:** Hiển thị thông tin hồ sơ (avatar, số lượng bài post, số lượng bạn bè) cho người dùng khác xem.

### Quản lý danh sách bạn bè

- **Gửi, hủy, xác nhận:** Gửi lời mời kết bạn, hủy lời mời, hoặc xác nhận lời mời từ người khác. Khi xác nhận, hệ thống tự động tạo cuộc trò chuyện (tích hợp với module **Message**).
- **Chặn bạn:** Chặn người dùng, xem danh sách bạn bè, tìm kiếm bạn bè theo tên hoặc email.

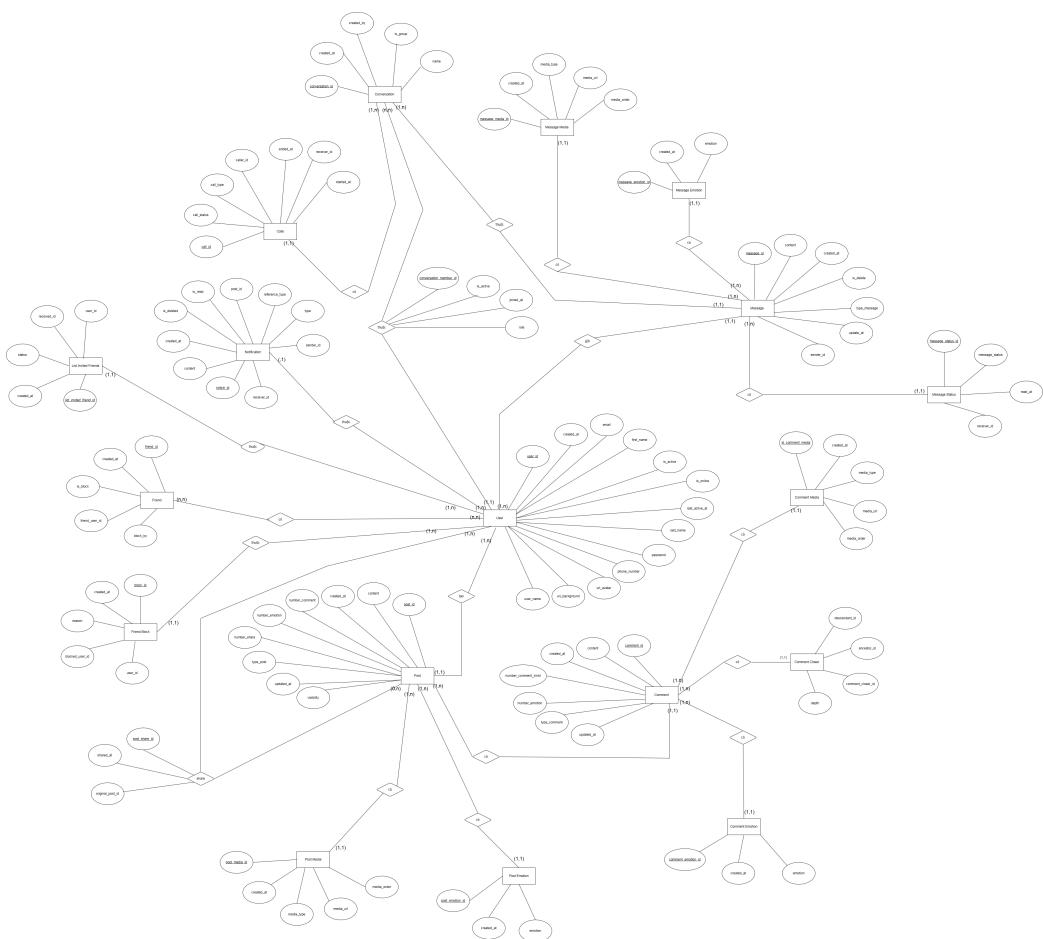
### Quản lý thông báo

- **Gửi thông báo thời gian thực:** Thông báo khi có lời mời kết bạn, bình luận mới, hoặc tin nhắn mới, sử dụng WebSocket và RabbitMQ để xử lý bất đồng bộ.
- **Hiển thị danh sách thông báo:** Hiển thị tất cả thông báo của người dùng, cho phép đánh dấu đã đọc hoặc xóa thông báo.

### Tìm kiếm

- **Tìm kiếm người dùng:** Tìm kiếm người dùng theo tên hoặc email, hiển thị danh sách kết quả.
- **Tìm kiếm bài post:** Tìm kiếm bài post theo nội dung hoặc hashtag.

# 8 Sơ đồ ERD



## 8.1 Entity User

Tên thuộc tính	Kiểu dữ liệu	Mô tả
----------------	--------------	-------

<code>userId</code>	<code>int</code>	Khóa chính, tự động tăng
<code>userName</code>	<code>String</code>	Tên đăng nhập (không trùng)
<code>firstName</code>	<code>String</code>	Tên
<code>lastName</code>	<code>String</code>	Họ
<code>email</code>	<code>String</code>	Email (không trùng)
<code>password</code>	<code>String</code>	Mật khẩu đã mã hóa
<code>phoneNumber</code>	<code>String</code>	Số điện thoại
<code>isActive</code>	<code>Boolean</code>	Trạng thái hoạt động
<code>lastActiveAt</code>	<code>LocalDateTime</code>	Thời gian hoạt động gần nhất
<code>createdAt</code>	<code>LocalDateTime</code>	Thời điểm tạo tài khoản
<code>isOnline</code>	<code>Boolean</code>	Dang online hay không
<code>urlAvatar</code>	<code>String</code>	URL ảnh đại diện
<code>urlBackground</code>	<code>String</code>	URL ảnh nền

Mối quan hệ:

- User có nhiều ConversationMember.
- User có nhiều Role (ManyToMany).

## 8.2 Entity Conversation

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<code>conversationId</code>	<code>int</code>	Khóa chính, tự động tăng
<code>name</code>	<code>String</code>	Tên cuộc trò chuyện
<code>isGroup</code>	<code>Boolean</code>	Có phải nhóm không
<code>createdBy</code>	<code>int</code>	ID người tạo cuộc trò chuyện
<code>createdAt</code>	<code>LocalDateTime</code>	Thời gian tạo

Mối quan hệ:

- Conversation có nhiều ConversationMember.

## 8.3 Entity ConversationMember

Tên thuộc tính	Kiểu dữ liệu	Mô tả
conversationMemberId	int	Khóa chính, tự động tăng
user	User	Tham chiếu đến User
conversation	Conversation	Tham chiếu đến Conversation
joinedAt	LocalDateTime	Thời điểm tham gia

Mối quan hệ:

- **ConversationMember** thuộc về một **User**.
- **ConversationMember** thuộc về một **Conversation**.

## 8.4 Entity Post

Tên thuộc tính	Kiểu dữ liệu	Mô tả
postId	int	Khóa chính, tự động tăng
user	User	Người dùng đăng bài
content	String	Nội dung bài viết (tối đa 5000 ký tự)
updatedAt	LocalDateTime	Thời gian cập nhật bài viết
createdAt	LocalDateTime	Thời gian tạo bài viết
numberEmotion	int	Số lượt cảm xúc
numberComment	int	Số lượt bình luận
numberShare	int	Số lượt chia sẻ
visibility	PostVisibilityEnum	Quyền riêng tư của bài viết
typePost	PostTypeEnum	Loại bài viết (ảnh, video, văn bản)
commentList	List<Comment>	Danh sách bình luận của bài viết
postMediaList	List<PostMedia>	Danh sách media (ảnh, video)
postEmotionList	List<PostEmotion>	Danh sách cảm xúc của bài viết

## 8.5 Entity Message

Tên thuộc tính	Kiểu dữ liệu	Mô tả
messageId	int	Khóa chính, tự động tăng

<code>conversationId</code>	<code>Conversation</code>	Cuộc trò chuyện chứa tin nhắn
<code>sender</code>	<code>User</code>	Người gửi tin nhắn
<code>typeMessage</code>	<code>MediaTypeEnum</code>	Loại tin nhắn (văn bản, hình ảnh, video)
<code>content</code>	<code>String</code>	Nội dung tin nhắn (kiểu TEXT)
<code>updateAt</code>	<code>LocalDateTime</code>	Thời gian cập nhật tin nhắn

## 8.6 Entity Comment

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<code>commentId</code>	<code>Integer</code>	Khóa chính, tự động tăng
<code>post</code>	<code>Post</code>	Bài viết mà bình luận thuộc về
<code>user</code>	<code>User</code>	Người dùng viết bình luận
<code>content</code>	<code>String</code>	Nội dung bình luận (kiểu TEXT)
<code>typeComment</code>	<code>CommentTypeEnum</code>	Loại bình luận
<code>createdAt</code>	<code>LocalDateTime</code>	Thời gian tạo bình luận
<code>numberEmotion</code>	<code>Integer</code>	Số lượt cảm xúc của bình luận
<code>numberCommentChild</code>	<code>Integer</code>	Số lượng phản hồi (bình luận con)
<code>updatedAt</code>	<code>LocalDateTime</code>	Thời gian cập nhật bình luận
<code>commentEmotionList</code>	<code>List&lt;CommentEmotion&gt;</code>	Danh sách cảm xúc của bình luận
<code>commentMediaList</code>	<code>List&lt;CommentMedia&gt;</code>	Danh sách media đính kèm bình luận
<code>ancestors</code>	<code>List&lt;CommentCloser&gt;</code>	Các tổ tiên (bình luận cha)
<code>descendants</code>	<code>List&lt;CommentCloser&gt;</code>	Các hậu duệ (bình luận con)

## 8.7 Entity Notification

Tên thuộc tính	Kiểu dữ liệu	Mô tả
<code>noticeId</code>	<code>int</code>	Khóa chính, tự động tăng
<code>receiver</code>	<code>User</code>	Người nhận thông báo
<code>sender</code>	<code>User</code>	Người gửi thông báo
<code>type</code>	<code>String</code>	Loại thông báo

postId	int	ID bài viết liên quan đến thông báo
content	String	Nội dung thông báo

## 8.8 Entity Friend

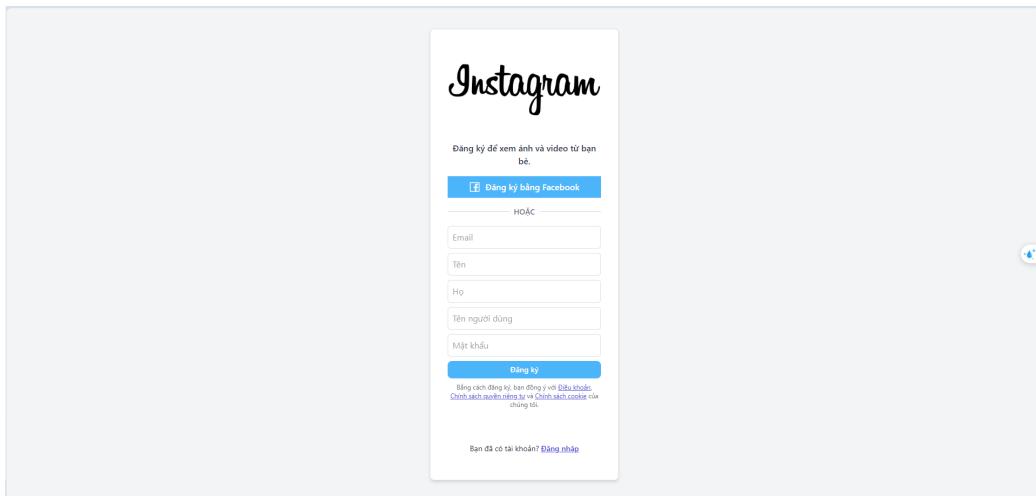
Tên thuộc tính	Kiểu dữ liệu	Mô tả
friendId	Integer	Khóa chính, tự động tăng
user	User	Người dùng A
friend	User	Người bạn của người dùng A
createdAt	LocalDateTime	Thời gian kết bạn
isBlock	Boolean	Có bị chặn hay không (true/false)
blockBy	User	Người đã chặn (nếu có)

## 8.9 Entity PostEmotion

Tên thuộc tính	Kiểu dữ liệu	Mô tả
postEmotionId	Integer	Khóa chính, tự động tăng
post	Post	Bài viết được cảm xúc
user	User	Người dùng thực hiện cảm xúc
emotion	String	Loại cảm xúc (ví dụ: like, love, haha, angry,...)
createdAt	LocalDateTime	Thời gian tạo cảm xúc

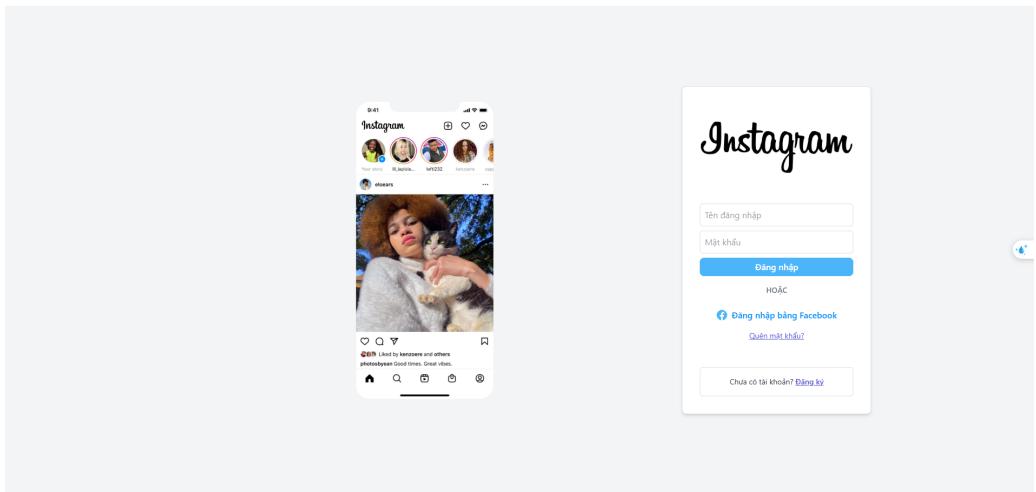
# 9 Kết quả đạt được

## 9.1 Đăng ký



Hình 9.1: Đăng ký

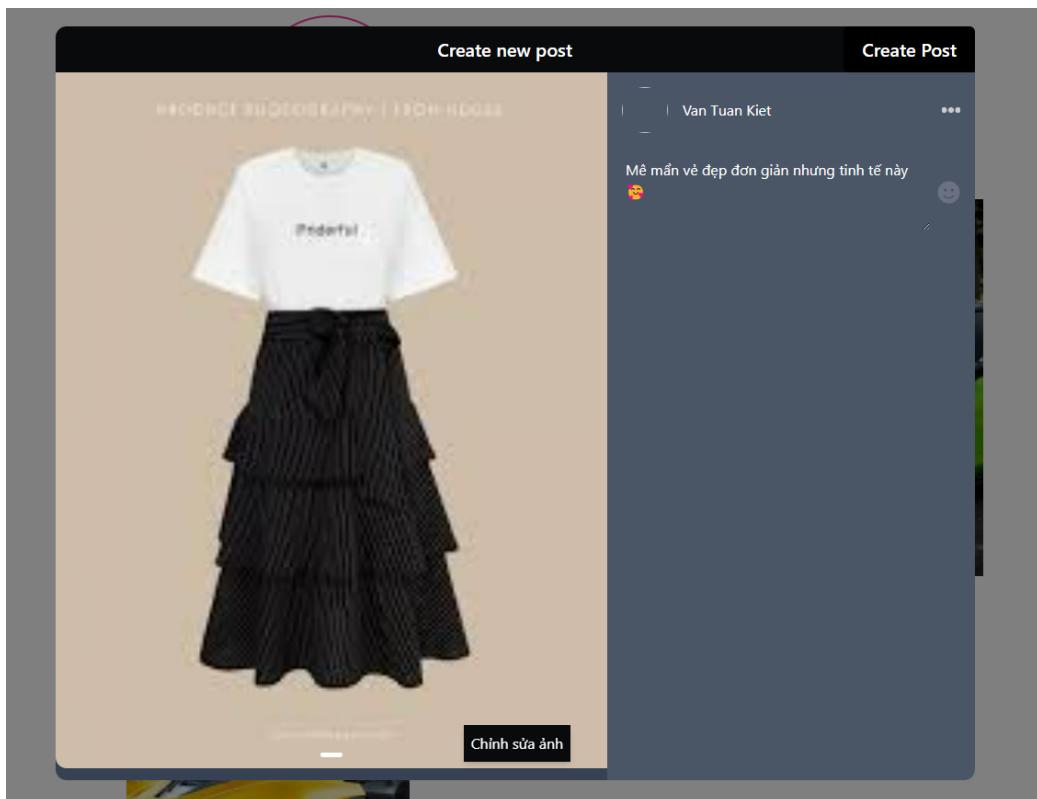
## 9.2 Đăng nhập



Hình 9.2: Đăng nhập

## 9.3 Chức năng đăng bài viết

### 9.3.1 Hình ảnh



Hình 9.3: Tạo vài viết hình ảnh



Hình 9.4: Kết quả tạo bài viết hình ảnh

### 9.3.2 Video

## 9.4 Chức năng tương tác

### 9.4.1 Like



Van Tuan Kiet Phong cách tối giản nhưng vẫn cuốn hút. ✨

Hình 9.5: Ảnh trước like  
30



Van Tuan Kiet 7 phút trước

...



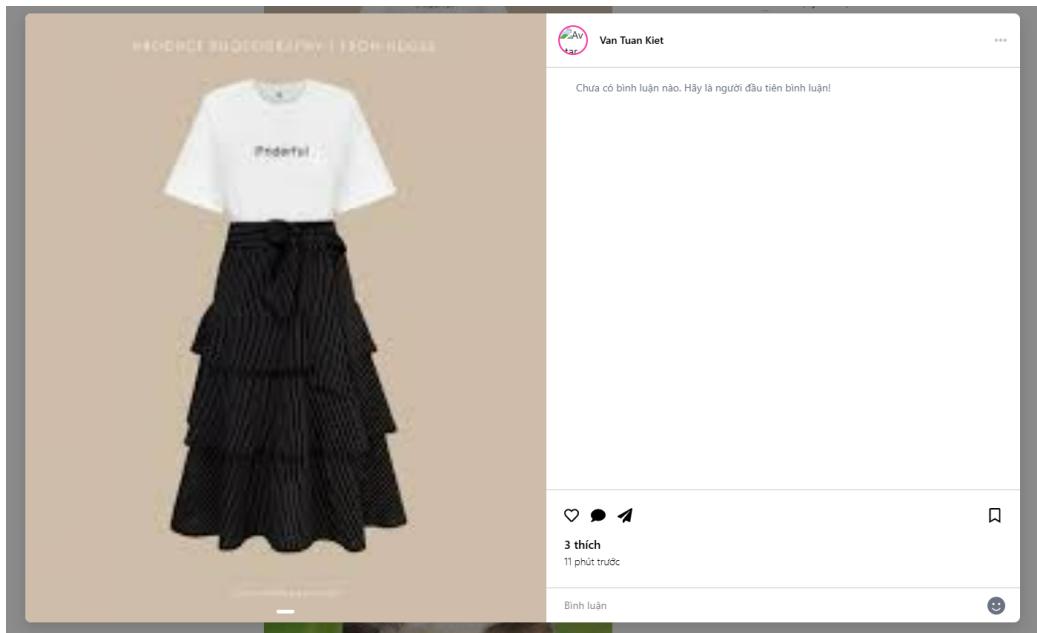
0 thích

Van Tuan Kiet Phong cách tối giản nhưng vẫn cuốn hút. ✨

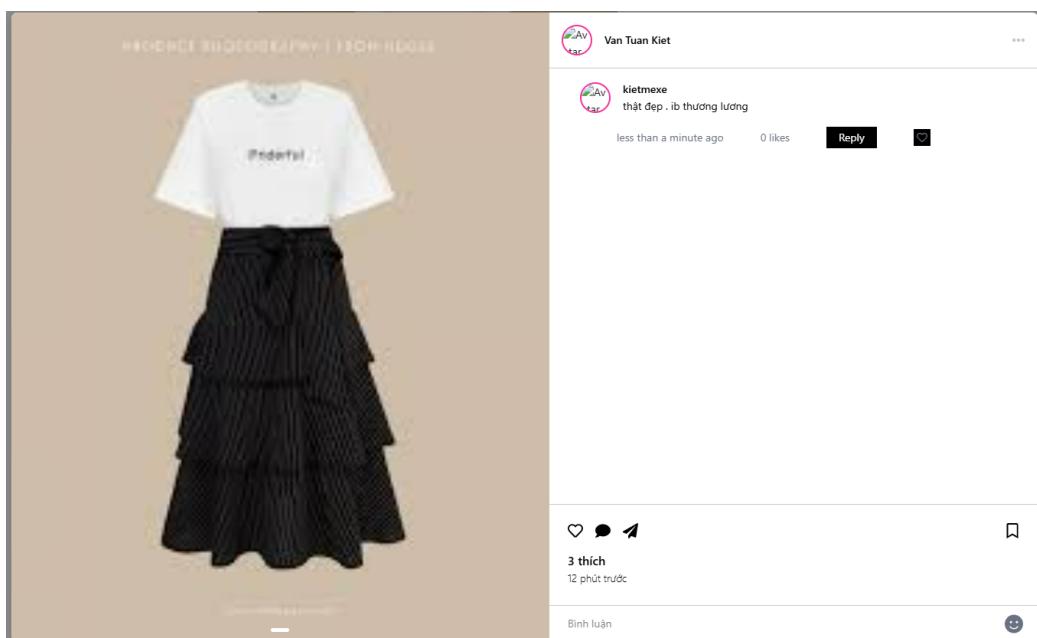
[Xem thêm 0 bình luận](#)

Hình 9.6: Ảnh sau like

#### 9.4.2 Comment



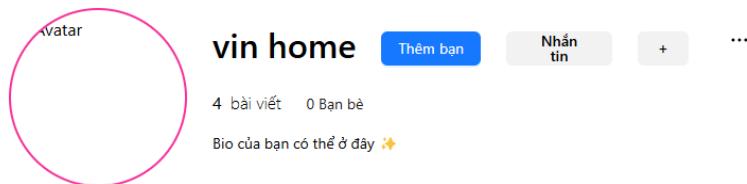
Hình 9.7: Ảnh chưa bình luận



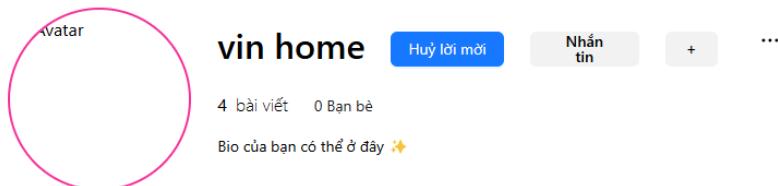
Hình 9.8: Ảnh đã bình luận

## 9.5 Chức năng quản lí bạn bè

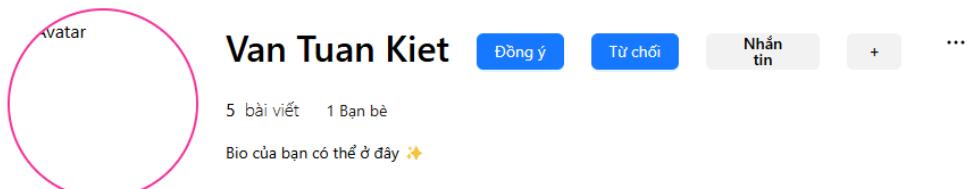
### 9.5.1 Thêm bạn



Hình 9.9: Ảnh thêm bạn



Hình 9.10: Ảnh huỷ lời mời



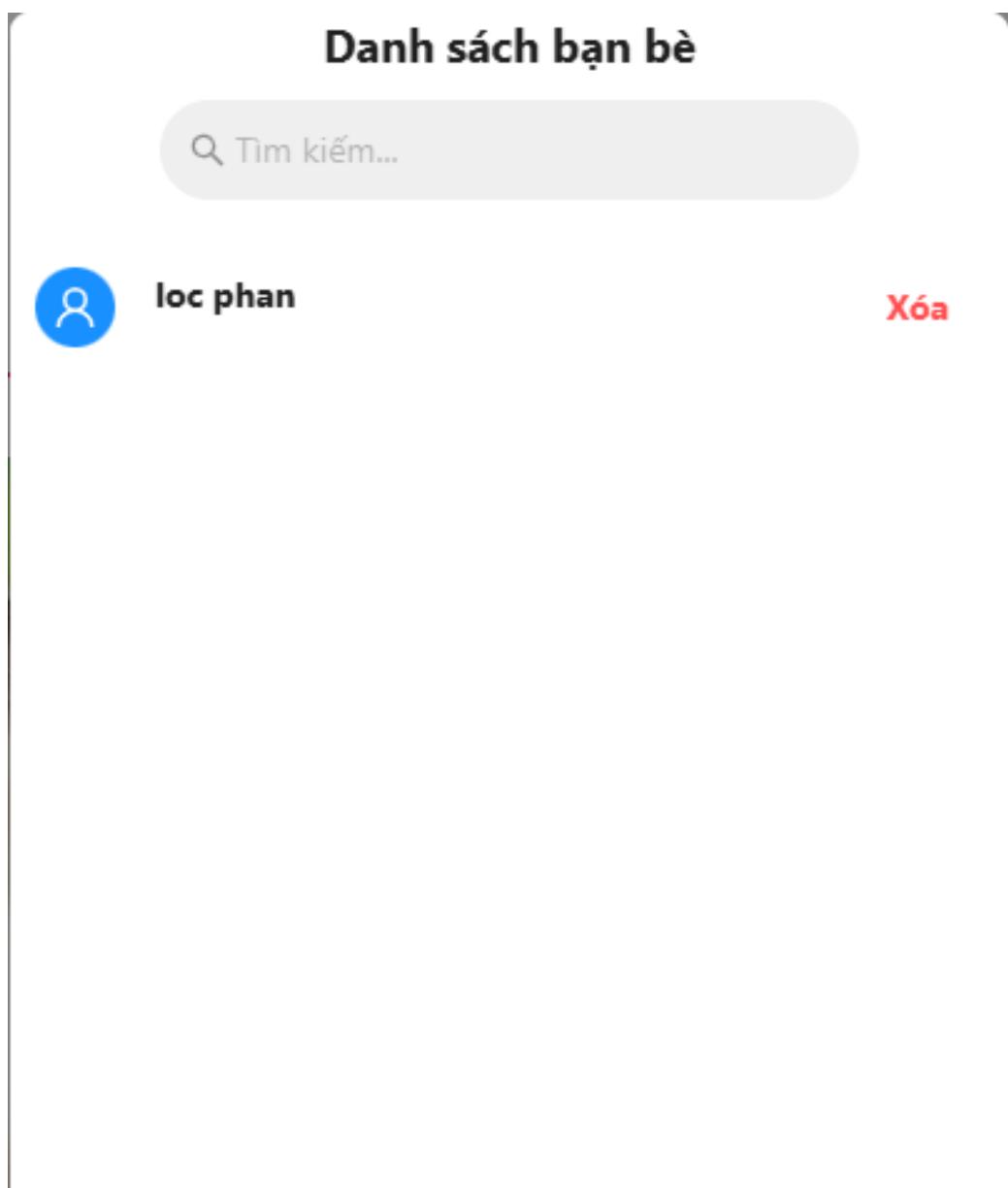
Hình 9.11: Ảnh chấp nhận , từ chối lời mời

### 9.5.2 Huỷ kết bạn



Hình 9.12: Ảnh đã là bạn bè , có thẻ xoá bạn

### 9.5.3 Danh sách bạn bè



Hình 9.13: Ảnh danh sách bạn bè

## 9.6 Chức năng quản lý hồ sơ cá nhân

### 9.6.1 Cập nhật thông tin người dùng

#### Chỉnh sửa thông tin cá nhân

The screenshot shows a user profile editing interface. The fields filled in are:

- Tên đăng nhập: locmecay
- Họ: loc
- Tên: phan
- Email: cay11@gmail.com
- Số điện thoại: 0899899899

At the bottom right are two buttons: "Cập nhật" (Update) in green and "Hủy" (Cancel) in orange.

## Chỉnh sửa thông tin cá nhân

Tên đăng nhập

Họ

Tên

Email

Số điện thoại

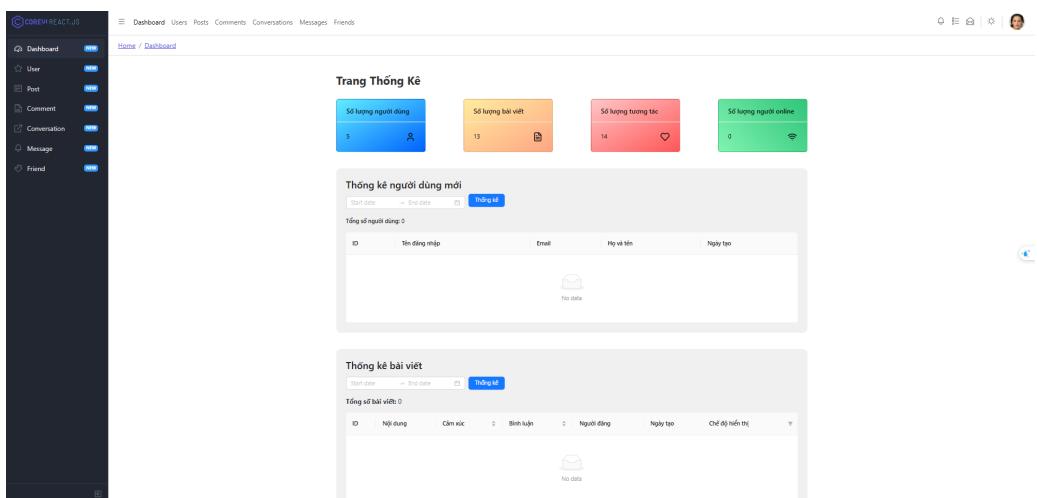
<sup>37</sup>  
Hình 9.15: Kết quả đổi tên

## 9.6.2 Cập nhật ảnh đại diện

Hình 9.16: Ảnh trước khi cập nhật

Hình 9.17: Ảnh sau khi cập nhật

## 9.7 Trang Admin



Hình 9.18: Ảnh trang admin

### 9.7.1 quản lí user

The screenshot shows the 'Danh Sách Người Dùng' (User List) page. It features a table with columns: ID, Họ (Name), Tên (Name), Email, Trạng Thái (Status), Khoá (Lock), Role, and Hành Động (Actions). The table contains five user entries. At the bottom, there is a search bar 'Tim kiem nguoi dung...', a 'Thêm Người Dùng' (Add User) button, and pagination controls 'Trước 1 Sau'.

ID	Họ	Tên	Email	Trạng Thái	Khoá	Role	Hành Động
1	user	new	user@gmail.com	Offline	<input checked="" type="checkbox"/>	User	Xem Sửa
2	Van Tuan	Kiet	xe@gmail.com	Offline	<input checked="" type="checkbox"/>	User	Xem Sửa
3	mai phuc	lam	cay@gmail.com	Offline	<input checked="" type="checkbox"/>	User	Xem Sửa
4	loc	hoang	cay1@gmail.com	Offline	<input checked="" type="checkbox"/>	User	Xem Sửa
5	vin	home	shooping@gmail.com	Offline	<input checked="" type="checkbox"/>	User	Xem Sửa

Hình 9.19: Ảnh trang quản lí user

### 9.7.2 quản lí post

The screenshot shows the CoreUI React JS application interface. On the left is a dark sidebar with navigation links: Dashboard, User, Post (selected), Comment, Conversation, Message, and Friend. The main content area has a header with 'Dashboard' and other navigation items like 'Users', 'Posts', 'Comments', 'Conversations', 'Messages', and 'Friends'. Below the header is a search bar labeled 'Tim kiếm bài post...'. The central part of the screen is titled 'Danh Sách Bài Post' (Post List) and contains a table with the following columns: ID, Nội dung (Content), Emotion, Comment, Share, Chế độ (Mode), and Hành động (Actions). The table lists 13 posts with various details. At the bottom right of the table are buttons for 'Trước' (Previous), 'Sau' (Next), and a small icon.

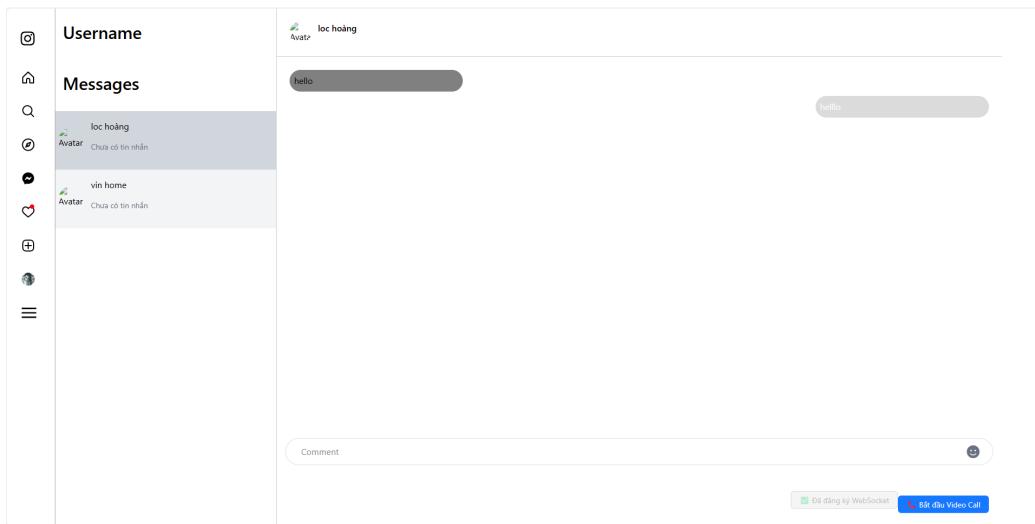
ID	Nội dung	Emotion	Comment	Share	Chế độ	Hành động
13	Phong cách tối giản nhưng vẫn cuốn hút.	3	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
12	Một thiên thần nhỏ giữa cô xanh muối.	1	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
11	Vẻ đẹp hoang dã đầy uy lực.	0	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
10	Một chủ sở đỗ dễ thương đang nhâm nhi bữa ăn của mình giữa khung cảnh thiên nhiên tươi đẹp.	0	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
9	tôi mê cay nhưng lại yêu động vật	0	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
8	chinh phục tốc độ với tên nào he he	2	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
7	Bàm mệt tốc độ, chinh phục mọi giới hạn.	0	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
6	ngẫu không hả các bạn	1	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
5	xe của tôi đẹp không hehe	0	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>
4	Một ngày đẹp trời, bình yên đến lạ.	2	0	0	PUBLIC	Xem <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Xem</span> <span style="background-color: #f0ad4e; color: white; padding: 2px 5px;">Sửa</span>

Hình 9.20: Ảnh trang quản lí user

### 9.8 Nhắn tin real time

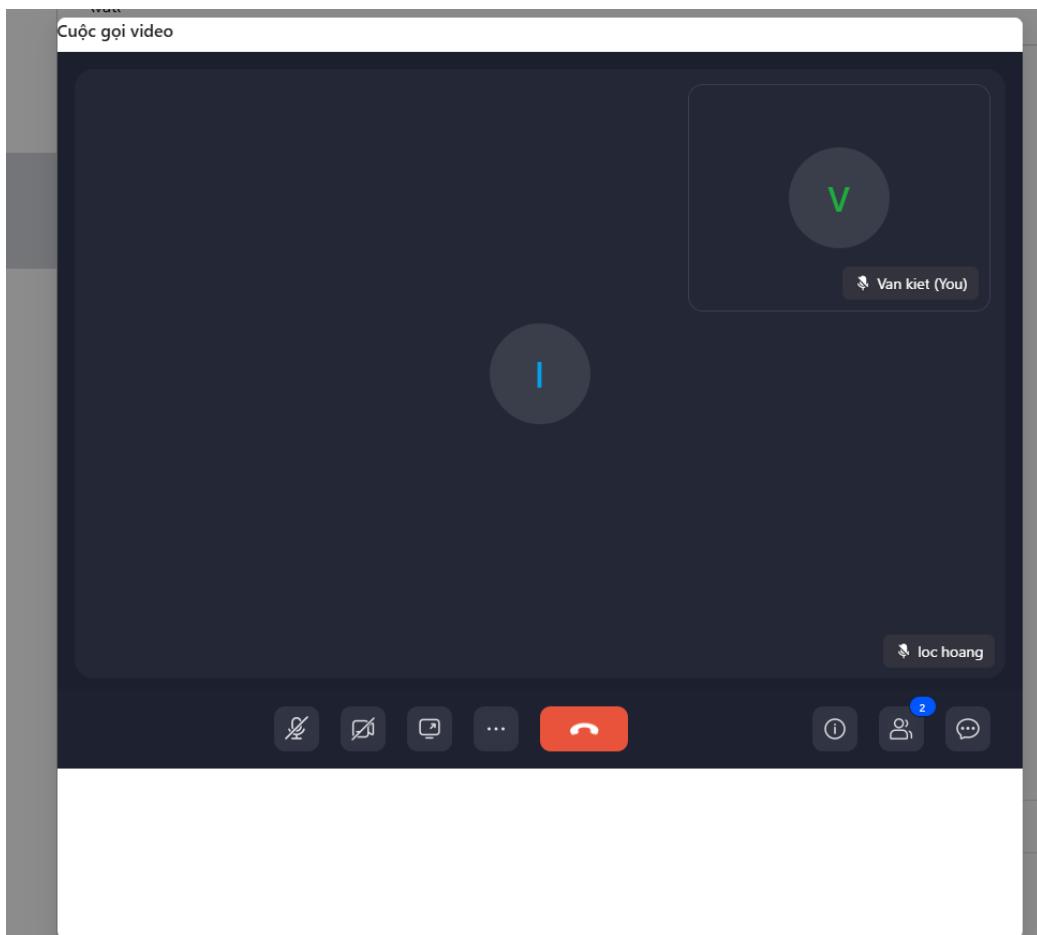
The screenshot shows a real-time messaging interface within the CoreUI React JS application. On the left is a sidebar with icons for User, Post, Comment, Conversation, Message, and Friend. The main area has a header with 'Username' and 'Tin nhắn'. Below the header, a message from 'Van Tuan Kiet' is shown: 'Avatar hello'. A text input field at the bottom is labeled 'Bình luận' (Comment) and features a smiley face emoji. At the bottom right are buttons for 'Đăng ký WebSocket' (WebSocket registration) and 'Bắt đầu Video Call' (Video call start).

Hình 9.21: Ảnh tin nhắn ở người 1

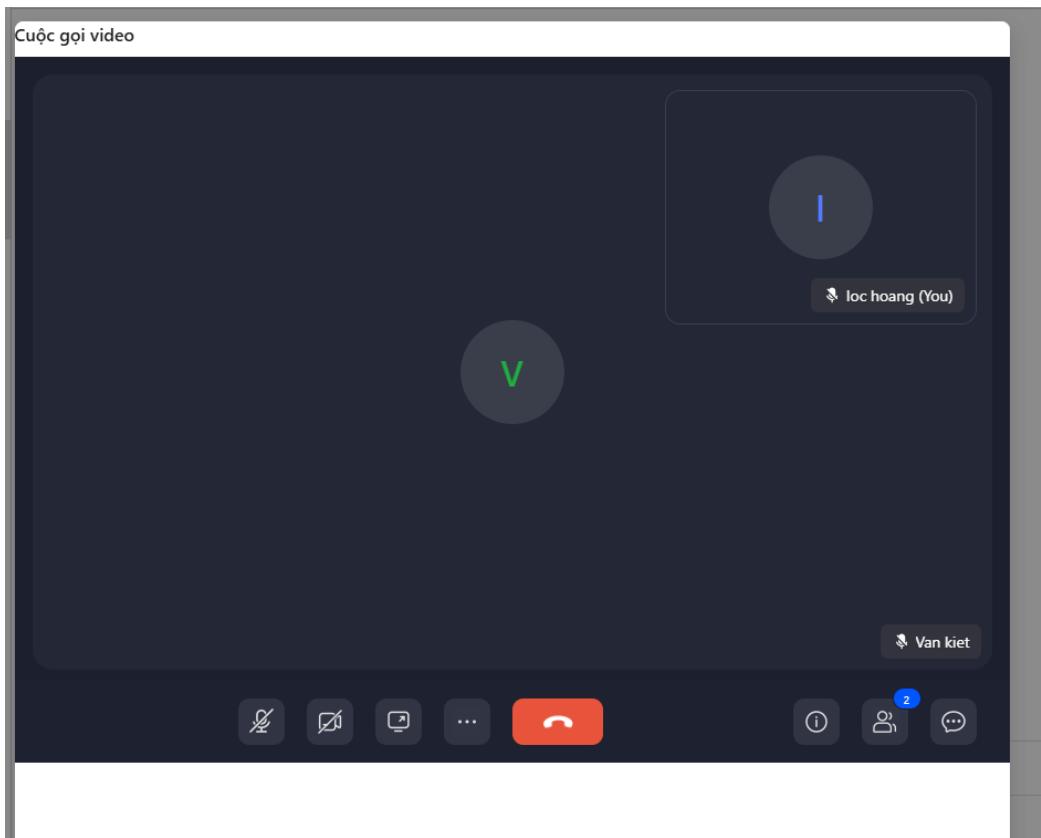


Hình 9.22: Ảnh tin nhắn ở người 2

## 9.9 Gọi real time

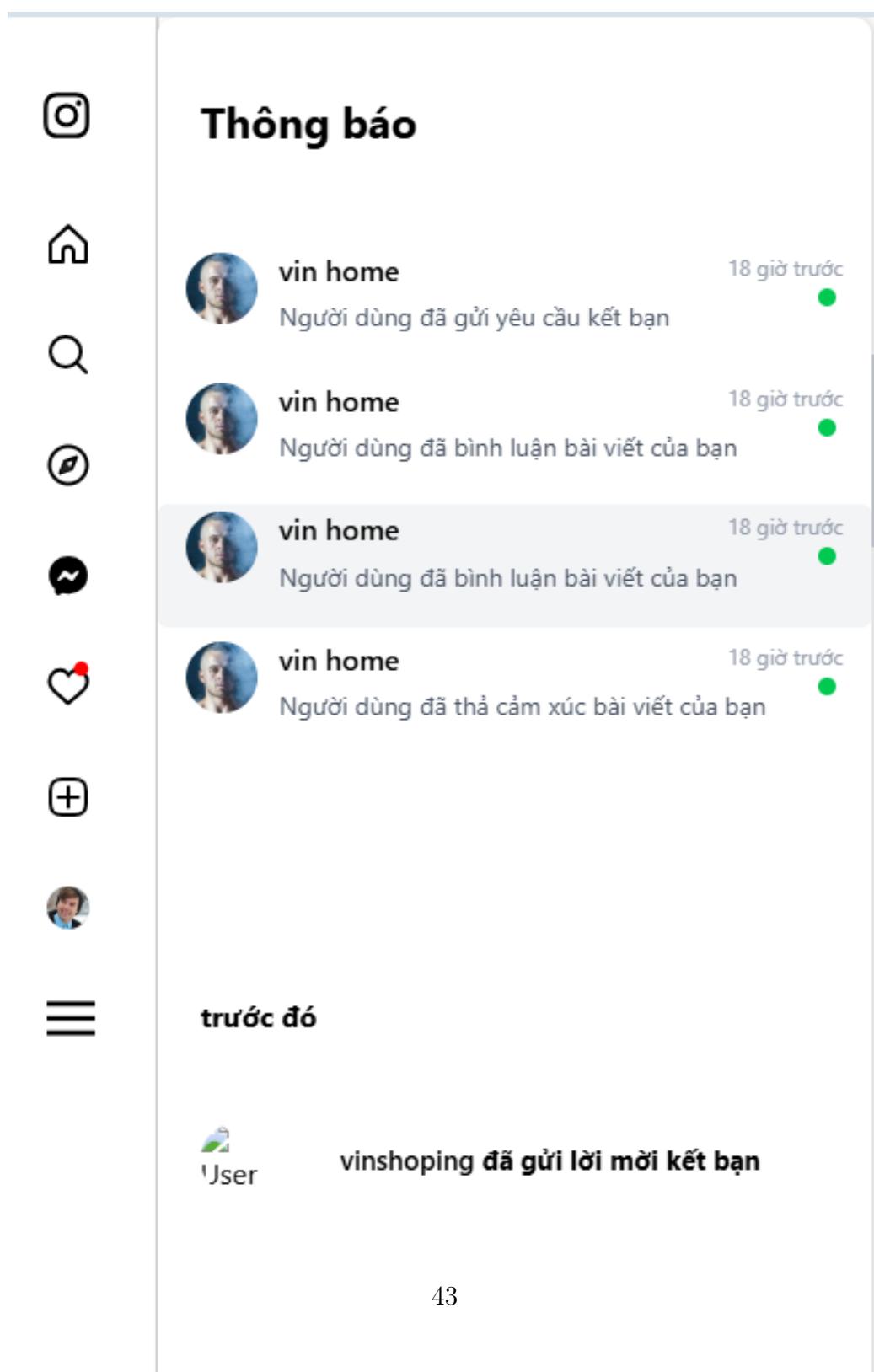


Hình 9.23: Ảnh gọi real time ở người 1



Hình 9.24: Ảnh gọi real time ở người 2

## 9.10 Thông báo



Hình 9.25: Ảnh thông báo khi có người bình luận , gửi kết bạn

# 10 Hướng phát triển

## 10.1 Hạn chế

Mặc dù hệ thống đã đạt được một số kết quả khả quan, tuy nhiên vẫn còn tồn tại một số hạn chế như sau:

- **Hệ thống vẫn có độ trễ khá cao** : các request , response phản hồi khá là chậm
- **Hạn chế về giao diện** : Giao diện còn khá thô sơ

## 10.2 Hướng phát triển trong tương lai

Để nâng cao hiệu quả và ứng dụng thực tế của hệ thống, trong tương lai có thể triển khai các hướng phát triển sau:

- **Cải thiện độ trễ** : Nghiên cứu thêm về các phương pháp cải thiện độ trễ