**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

**Vietnam National University of Forestry**

**KHOA KINH TẾ & QUẢN TRỊ KINH DOANH**

**............🖎🕮✍...........**



**BÁO CÁO ĐỒ ÁN MÔN   
AN TOÀN BẢO MẬT THÔNG TIN**

**Đề tài: Ứng dụng phòng chat riêng tư   
sử dụng mã hóa RSA**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | **Mai Hà An** |
| Lớp: | **K64 – HTTT** |
| Họ tên sinh viên: | **Vũ Đình Hoàng** |
| Mã sinh viên: | **1951070332** |

Hà Nội – 2023

MỤC LỤC

[DANH SÁCH CÁC BẢNG 2](#_Toc123762615)

[CHƯƠNG I. GIỚI THIỆU VỀ MÃ HÓA RSA 3](#_Toc123762616)

[A. Hệ mã hóa bất đối xứng 3](#_Toc123762617)

[B. Mã hóa khóa công khai 3](#_Toc123762618)

[1. Sinh khóa 3](#_Toc123762619)

[2. Mã hóa và giải mã 4](#_Toc123762620)

[CHƯƠNG II. XÂY DỰNG VÀ DEMO CHƯƠNG TRÌNH 5](#_Toc123762621)

[A. GIỚI THIỆU VỀ CHƯƠNG TRÌNH 5](#_Toc123762622)

[1. Ý tưởng 5](#_Toc123762623)

[2. Cấu trúc thư mục của chương trình 5](#_Toc123762624)

[B. XÂY DỰNG CHƯƠNG TRÌNH 6](#_Toc123762625)

[1. Xây dựng chức năng nhắn tin 6](#_Toc123762626)

[a. Tạo module socket listener bên Server 6](#_Toc123762627)

[b. Tạo module socket listener bên Client 7](#_Toc123762628)

[c. Hiển thị tin nhắn ra màn hình 7](#_Toc123762629)

[2. Xây dựng module mã hóa và cấu hình kết nối với chương trình 8](#_Toc123762630)

[a. Tạo web worker để gói các phương thức của thư viện Jsencrypt 8](#_Toc123762631)

[b. Cấu hình Vue App để giao tiếp với web worker 9](#_Toc123762632)

[3. Trao đổi khóa 9](#_Toc123762633)

[a. Thêm module socket listener bên server để vận chuyển khóa công khai 9](#_Toc123762634)

[b. Tạo cặp khóa 10](#_Toc123762635)

[c. Thêm hàm hỗ trợ liên quan đến khóa công khai 10](#_Toc123762636)

[d. Gửi nhận khóa công khai. 11](#_Toc123762637)

[e. Hiển thị khóa công khai trong giao diện 11](#_Toc123762638)

[4. Mã hóa tin nhắn tin nhắn 12](#_Toc123762639)

[a. Mã hóa tin nhắn trước khi gửi đi 12](#_Toc123762640)

[b. Nhận và giải mã tin nhắn 12](#_Toc123762641)

[c. Hiển thị danh sách tin nhắn 13](#_Toc123762642)

[5. Xây dựng phòng chat 13](#_Toc123762643)

[a. Luồng vào phòng bên server 13](#_Toc123762644)

[b. Luồng vào phòng bên client 14](#_Toc123762645)

[c. Thêm module thông báo 14](#_Toc123762646)

[d. Thêm giao diện vào phòng 14](#_Toc123762647)

[e. Thêm autoscroll 15](#_Toc123762648)

[C. KẾT QUẢ VÀ DEMO CHƯƠNG TRÌNH 15](#_Toc123762649)

[1. Khởi động server 15](#_Toc123762650)

[2. Truy cập vào client 16](#_Toc123762651)

[3. Chọn phòng 16](#_Toc123762652)

[4. Demo nhắn tin 18](#_Toc123762653)

# DANH SÁCH CÁC BẢNG

[Hình II.1 Cấu trúc thư mục của chương trình 5](#_Toc123762591)

[Hình II.2 Tạo module socket listener bên Server 6](#_Toc123762592)

[Hình II.3 Tạo module sockert listener bên Client 7](#_Toc123762593)

[Hình II.4 Hiển thị tin nhắn ra màn hình 7](#_Toc123762594)

[Hình II.5 Tạo web worker để gói các phương thwucs của thư viện JSencrypt 8](#_Toc123762595)

[Hình II.6 Cấu hình Vue App để giao tiếp với web worker 9](#_Toc123762596)

[Hình II.7 Thêm module socket listener bên server đển chuyển khóa công khai 9](#_Toc123762597)

[Hình II.8 Tạo cặp khóa 10](#_Toc123762598)

[Hình II.9 Thêm hàm hỗ trợ liên quan đến khóa công khai 10](#_Toc123762599)

[Hình II.10 Gửi nhận khóa công khai 11](#_Toc123762600)

[Hình II.11 Hiển thị khóa công khai trong giao diện 11](#_Toc123762601)

[Hình II.12 Mã hóa tin nhắn trước khi gửi đi 12](#_Toc123762602)

[Hình II.13 Nhạn và giải mã tin nhắn 12](#_Toc123762603)

[Hình II.14 Hiển thị danh sách tin nhắn 13](#_Toc123762604)

[Hình II.15 Luồng vào phòng bên server 13](#_Toc123762605)

[Hình II.16 Luồng vào phòng bên client 14](#_Toc123762606)

[Hình II.17Thêm module thông báo 14](#_Toc123762607)

[Hình II.18 Thêm giao diện vào phòng 14](#_Toc123762608)

[Hình II.19Thêm autoscroll 15](#_Toc123762609)

[Hình II.20 Demo gửi tin nhắn giữa 2 client trong cùng 1 zoom 15](#_Toc123762610)

[Hình II.21 Kết quả chạy lệnh. Server đã chạy 15](#_Toc123762611)

[Hình II.22 Giao diện tương tác Client 16](#_Toc123762612)

[Hình II.23 Giao diện khi đã kết nối 17](#_Toc123762613)

[Hình II.24 Demo nhắn tin 18](#_Toc123762614)

# GIỚI THIỆU VỀ MÃ HÓA RSA

## Hệ mã hóa bất đối xứng

* Hệ mã hóa bất đối xứng (asymmetric cryptography) hay còn gọi là hệ mã hóa public key là một hệ mã hóa sử dụng một cặp key để mã hóa và giải mã: public key (khóa công khai) dùng để mã hóa và private key (khóa bí mật) để giải mã.
* Trong hệ mã hóa này, bất cứ ai cũng có thể sử dụng public key đã mã hóa bản tin và gửi cho người nhận. Nhưng một điều hiển nhiên là người sở hữu private key sẽ giữ nó cho riêng mình, và do đó, chỉ anh ta mới có giải mã được mà thôi.
* Thông thường thì cặp khóa được sinh này sẽ cố gắng đảm bảo rằng từ public key rất khó (gần như là không thể) truy ra được private key. Vì vậy, bất cứ kẻ tấn công nào nếu có được public key (điều này khá dễ dàng) cũng không thể có được private key để giải mã.
* Tuy nhiên, thực tế phũ phàng hơn rất nhiều. Hoàn toàn không có thuật toán nào đạt được lý tưởng như trên. Các thuật toán sinh khóa hiện nay đều chỉ có thể phòng chống các kỹ thuật crack thông thường sử dụng máy tính cá nhân. Với sự phát triển của các siêu máy tính cũng như các loại máy tính lượng tử, hệ mã hóa bất đối xứng cũng đang phải đối mặt với rất nhiều thách thức.

## Mã hóa khóa công khai

* RSA là một hệ mã hóa bất đối xứng được phát triển bởi Ron Rivest, Adi Shamir và Leonard Adleman (tên của nó cũng chính là tên viết tắt của 3 tác giả này) và được sử dụng rộng rãi trong công tác mã hoá và công nghệ chữ ký điện tử. Trong hệ mã hóa này, public key có thể chia sẻ công khai cho tất cả mọi người. Hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ key, mã hóa và giải mã.

### Sinh khóa

* Mấu chốt cơ bản của việc sinh khóa trong RSA là tìm được bộ 3 số tự nhiên e, d và n sao cho:

Text, letter

Description automatically generated

và một điểm không thể bỏ qua là cần bảo mật cho d sao cho dù biết e, n hay thậm chí cả m cũng không thể tìm ra d được.

* Cụ thể, khóa của RSA được sinh như sau:
  + - Chọn 2 số nguyên tố p và q
    - Tính n = pq. Sau này, n sẽ được dùng làm modulus trong cả public key và private key.
    - Tính một số giả nguyên tố bằng phi hàm Carmichael như sau: λ(n) = BCNN(λ(p), λ(q)) = BCNN(p − 1, q − 1). Giá trị này sẽ được giữ bí mật.
    - Chọn một số tự nhiên e trong khoảng (1, λ(n)) sao cho ƯCLN(e, λ(n)) = 1, tức là e và λ(n) nguyên tố cùng nhau.
    - Tính toán số d sao cho d ≡ 1/e (mod λ(n)) hay viết dễ hiểu hơn thì de ≡ 1 (mod λ(n)). Số d được gọi là số nghịch đảo modulo của e (theo modulo mod λ(n)).
* Public key sẽ là bộ số (n, e), và private key sẽ là bộ số (n, d). Chúng ta cần giữ private key thật cẩn thận cũng như các số nguyên tố p và q vì từ đó có thể tính toán các khóa rất dễ dàng.
* Trong thực hành, chúng ta thường chọn e tương đối nhỏ để việc mã hóa và giải mã nhanh chóng hơn. Giá trị thường được sử dụng là e = 65537. Ngoài ra, chúng ta có thể tính số giả nguyên tố bằng phi hàm Euler φ(n) = (p − 1)(q − 1) và dùng nó như λ(n). Vì φ(n) là bội của λ(n) nên các số d thỏa mãn de ≡ 1 (mod φ(n)) cũng sẽ thỏa mãn d ≡ 1/e (mod λ(n)). Tuy nhiên, một số tác dụng phụ của việc này là d thường sẽ trở nên lớn hơn mức cần thiết.

### Mã hóa và giải mã

* Trong phần này, chúng ta sẽ tìm hiểu cách mã hóa với public key (n, e) và giải mã với private key (n, d).
* Nếu chúng ta có bản rõ M, chúng ta cần chuyển nó thành một số tự nhiên m trong khoảng (0, n) sao cho m, n nguyên tố cùng nhau. Việc này rất dễ dàng thực hiện bằng cách thêm một các kỹ thuật padding. Tiếp theo, chúng ta sẽ má hóa m, thành c như sau:

Text

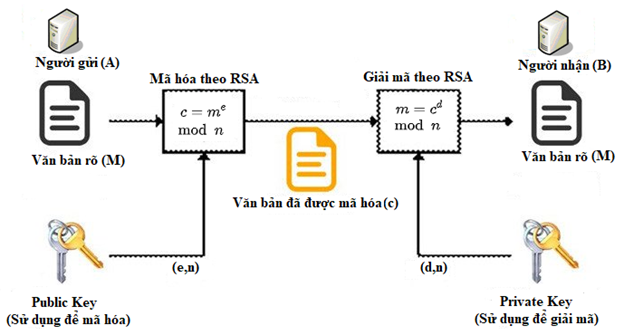
Description automatically generated with medium confidence

* Sau đó giá trị c sẽ được chuyển cho người nhận.
* Ở phía người nhận, họ sẽ giải mã từ c để lấy được m như sau:

Text, letter

Description automatically generated

* Từ m có thể lấy lại được bản tin bằng cách đảo ngược padding



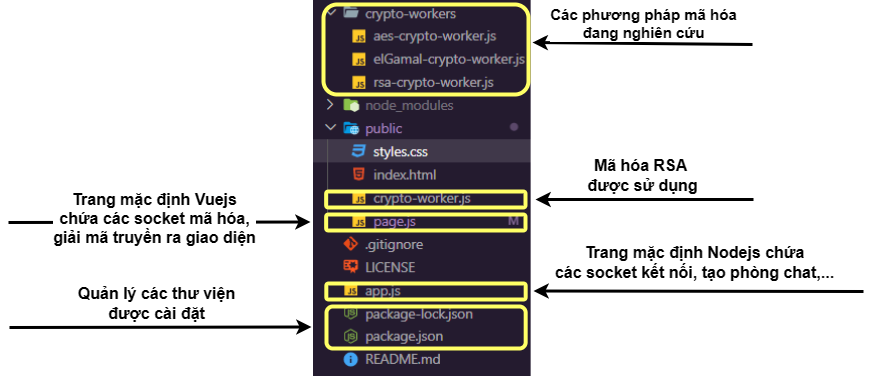
# XÂY DỰNG VÀ DEMO CHƯƠNG TRÌNH

## GIỚI THIỆU VỀ CHƯƠNG TRÌNH

### Ý tưởng

* Ý tưởng của chương trình là tạo một ứng dụng các phòng chat riêng tư online cho nhiều người có thể truy cập để tìm hiểu lẫn nhau cũng như trao đổi làm quen.
* Mỗi phòng chỉ có hai người giúp đảm bảo tính riêng tư của đoạn chat.
* Mỗi người khi vào được cấp một private key được bảo mật trên hệ thống và public key được công khai và đối phương có thể nhìn thấy khi cả hai cùng vao phòng
* Tin nhắn khi gửi sẽ được mã hóa và được bên nhận giải mã và hiển thị lên màn hình
* Số phòng được random hoặc tùy chọn theo thống nhất của mỗi cặp.

### Cấu trúc thư mục của chương trình



Hình II.1 Cấu trúc thư mục của chương trình

## XÂY DỰNG CHƯƠNG TRÌNH

### Xây dựng chức năng nhắn tin

#### Tạo module socket listener bên Server

* Trong file app.js ta thêm đoạn code dưới đây để thêm socket listener
* Đoạn code này sẽ tạo một lắng nghe kết nối và sẽ quản lý bất kì ai kết nối tới server từ front-end của ứng dụng.



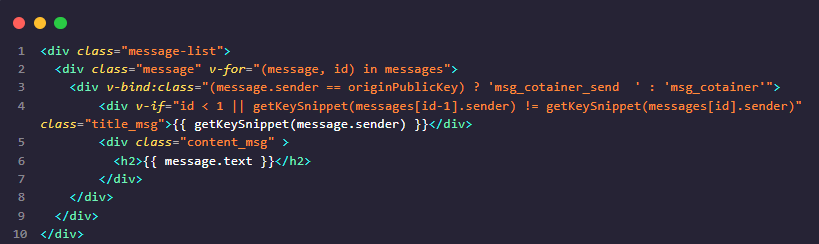
Hình II.2 Tạo module socket listener bên Server

#### Tạo module socket listener bên Client



Hình II.3 Tạo module sockert listener bên Client

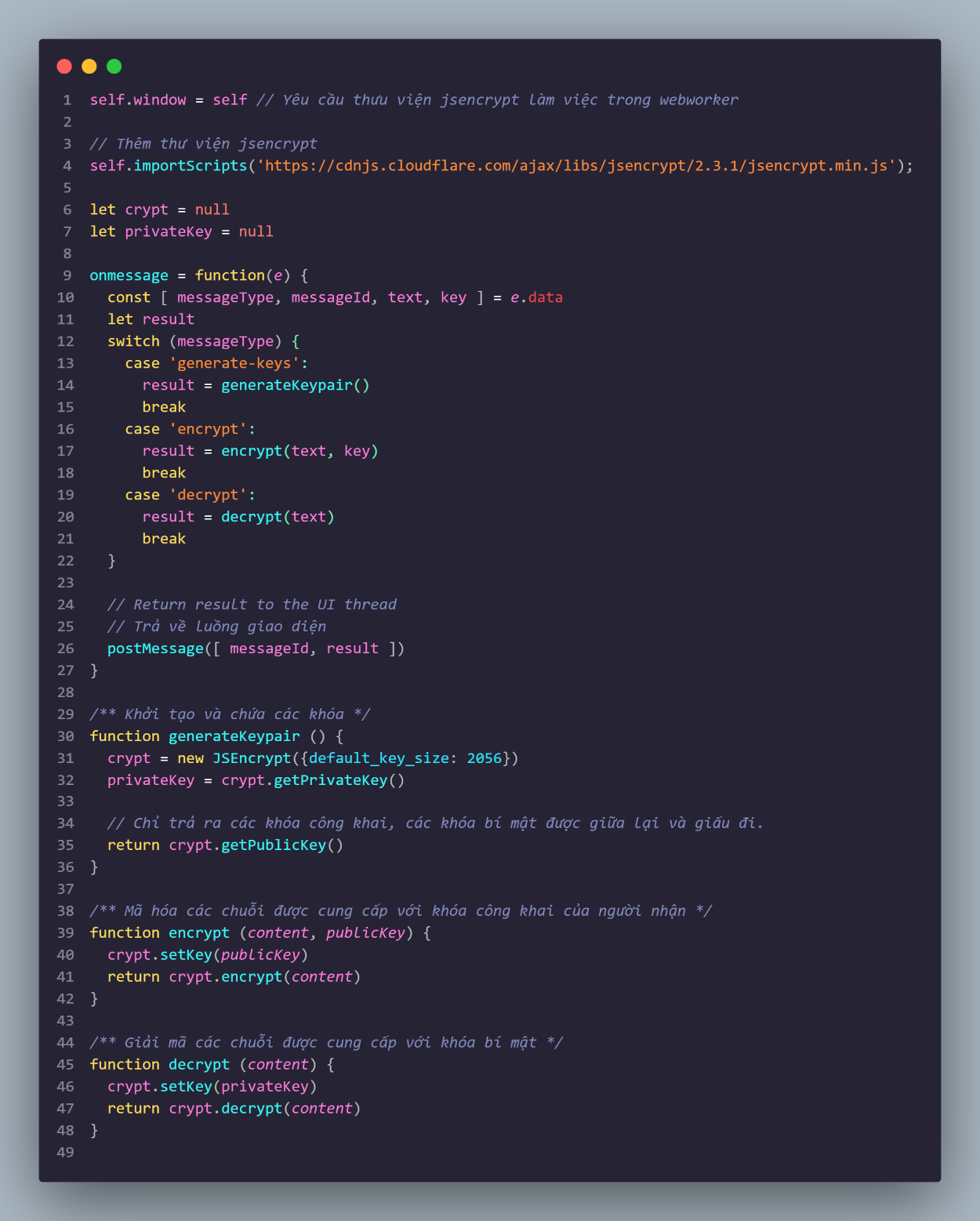
#### Hiển thị tin nhắn ra màn hình



Hình II.4 Hiển thị tin nhắn ra màn hình

### Xây dựng module mã hóa và cấu hình kết nối với chương trình

#### Tạo web worker để gói các phương thức của thư viện Jsencrypt



Hình II.5 Tạo web worker để gói các phương thwucs của thư viện JSencrypt

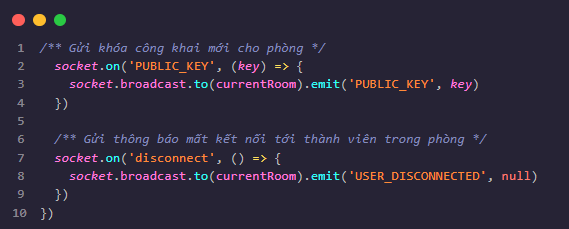
#### Cấu hình Vue App để giao tiếp với web worker



Hình II.6 Cấu hình Vue App để giao tiếp với web worker

### Trao đổi khóa

#### Thêm module socket listener bên server để vận chuyển khóa công khai



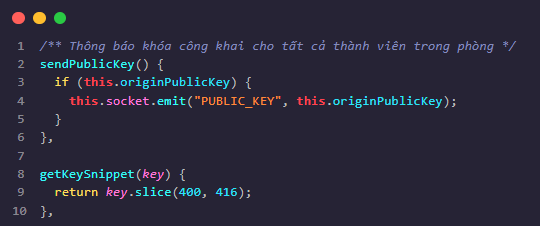
Hình II.7 Thêm module socket listener bên server đển chuyển khóa công khai

#### Tạo cặp khóa



Hình II.8 Tạo cặp khóa

#### Thêm hàm hỗ trợ liên quan đến khóa công khai



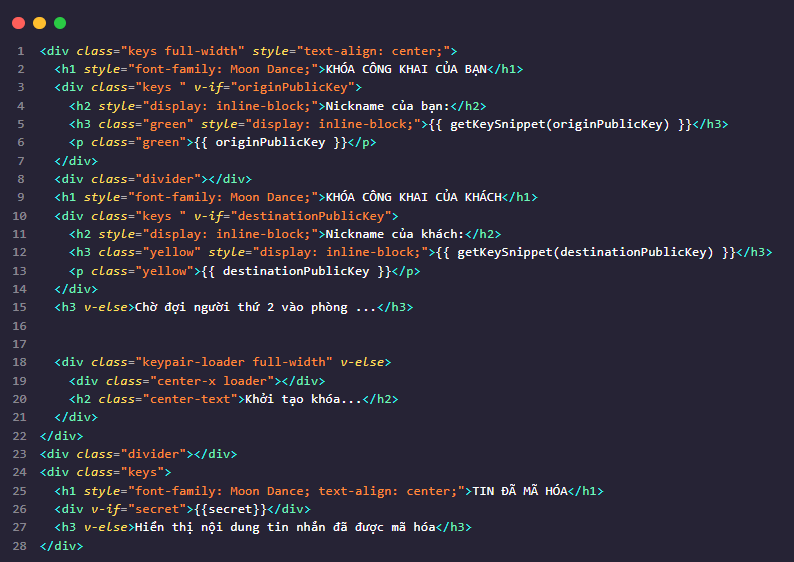
Hình II.9 Thêm hàm hỗ trợ liên quan đến khóa công khai

#### Gửi nhận khóa công khai.



Hình II.10 Gửi nhận khóa công khai

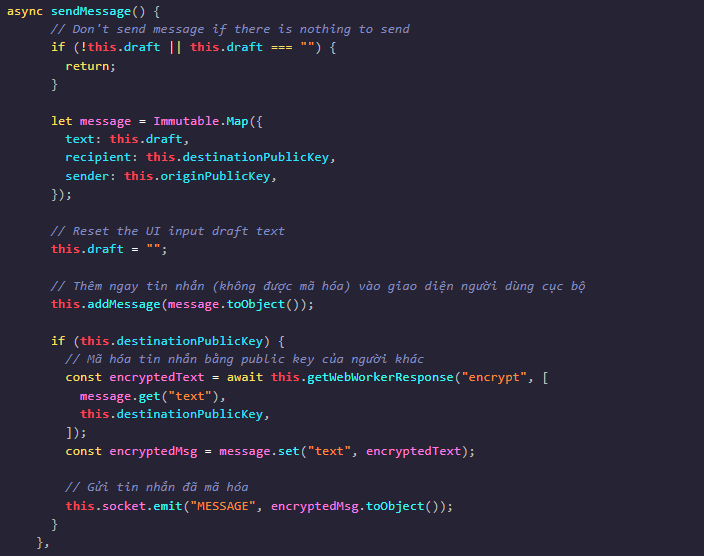
#### Hiển thị khóa công khai trong giao diện



Hình II.11 Hiển thị khóa công khai trong giao diện

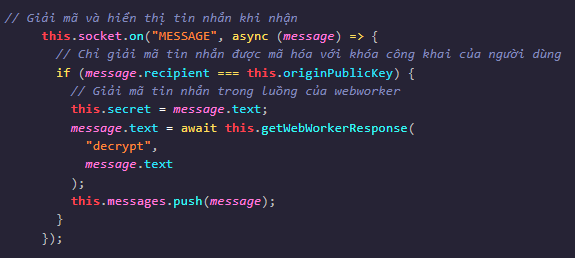
### Mã hóa tin nhắn tin nhắn

#### Mã hóa tin nhắn trước khi gửi đi



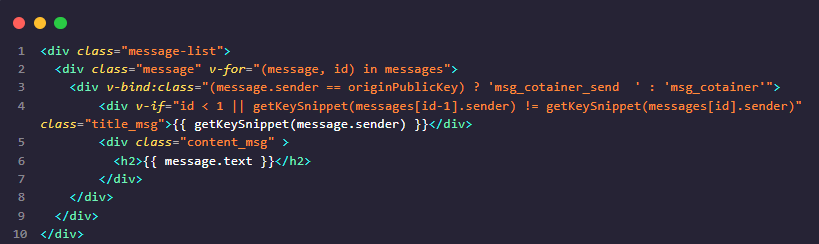
Hình II.12 Mã hóa tin nhắn trước khi gửi đi

#### Nhận và giải mã tin nhắn



Hình II.13 Nhạn và giải mã tin nhắn

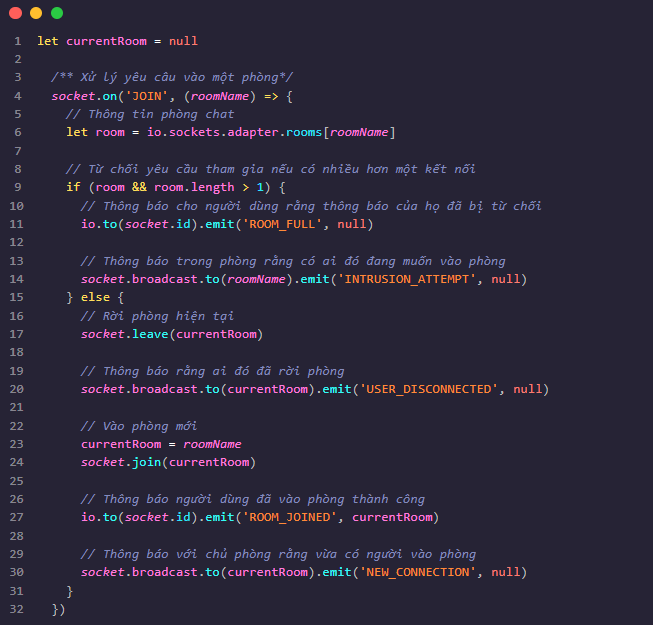
#### Hiển thị danh sách tin nhắn



Hình II.14 Hiển thị danh sách tin nhắn

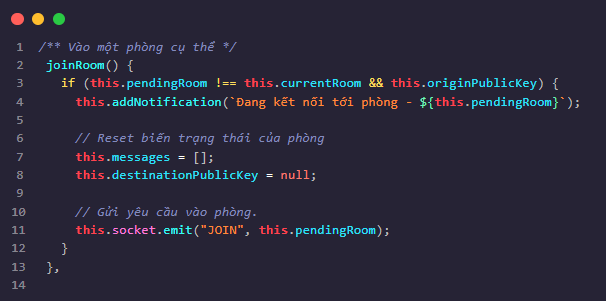
### Xây dựng phòng chat

#### Luồng vào phòng bên server



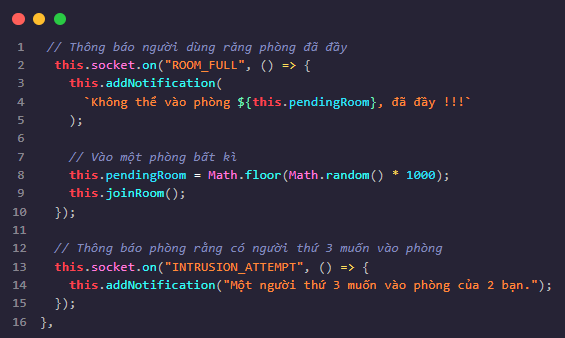
Hình II.15 Luồng vào phòng bên server

#### Luồng vào phòng bên client



Hình II.16 Luồng vào phòng bên client

#### Thêm module thông báo



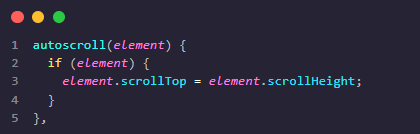
Hình II.17Thêm module thông báo

#### Thêm giao diện vào phòng



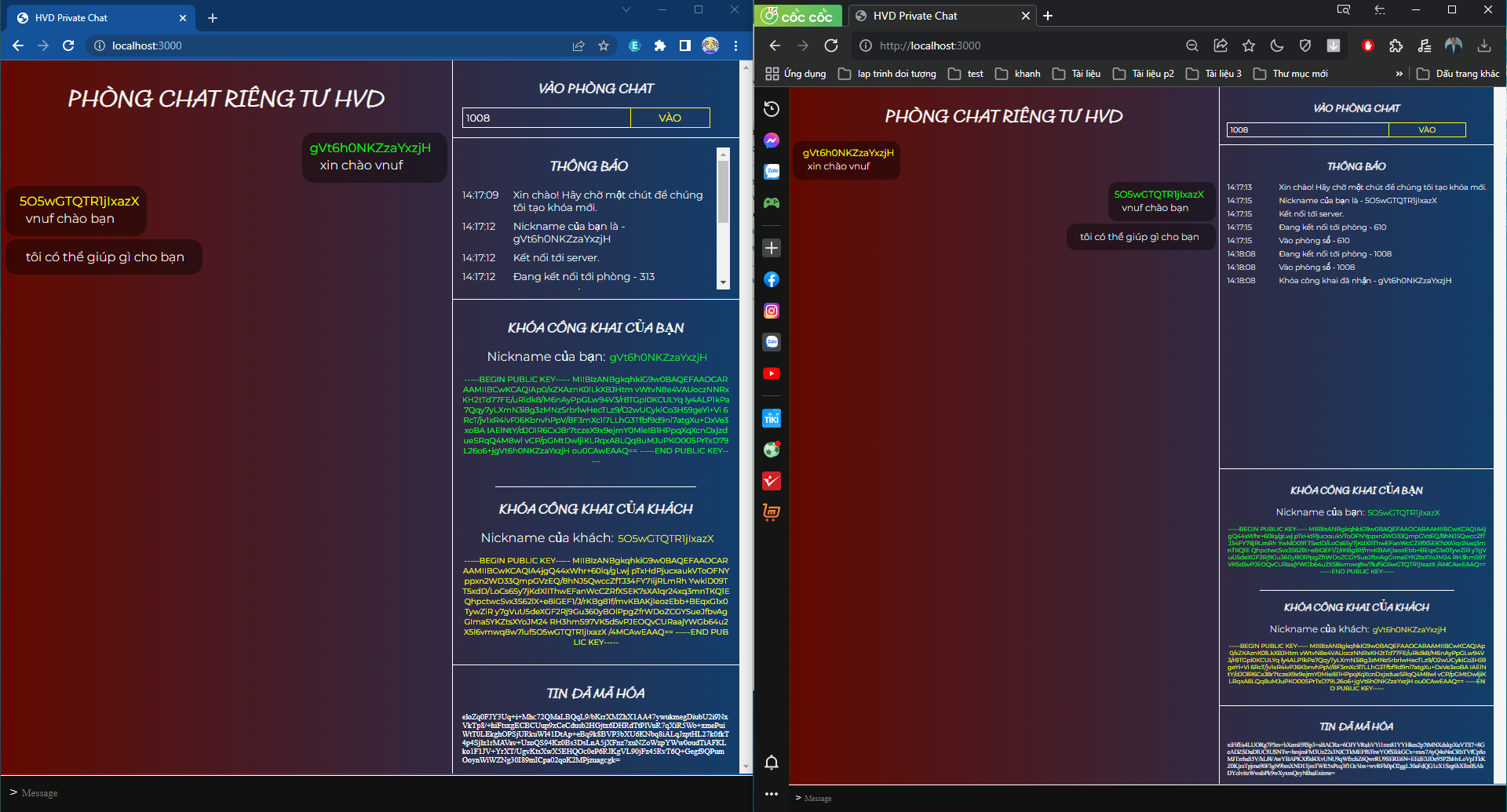
Hình II.18 Thêm giao diện vào phòng

#### Thêm autoscroll



Hình II.19Thêm autoscroll

## KẾT QUẢ VÀ DEMO CHƯƠNG TRÌNH

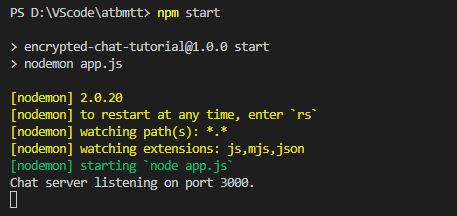


Hình II.20 Demo gửi tin nhắn giữa 2 client trong cùng 1 zoom

* Yêu cầu:
  + - Đã tải project
    - Đã cài đặt nodejs và npm

### Khởi động server

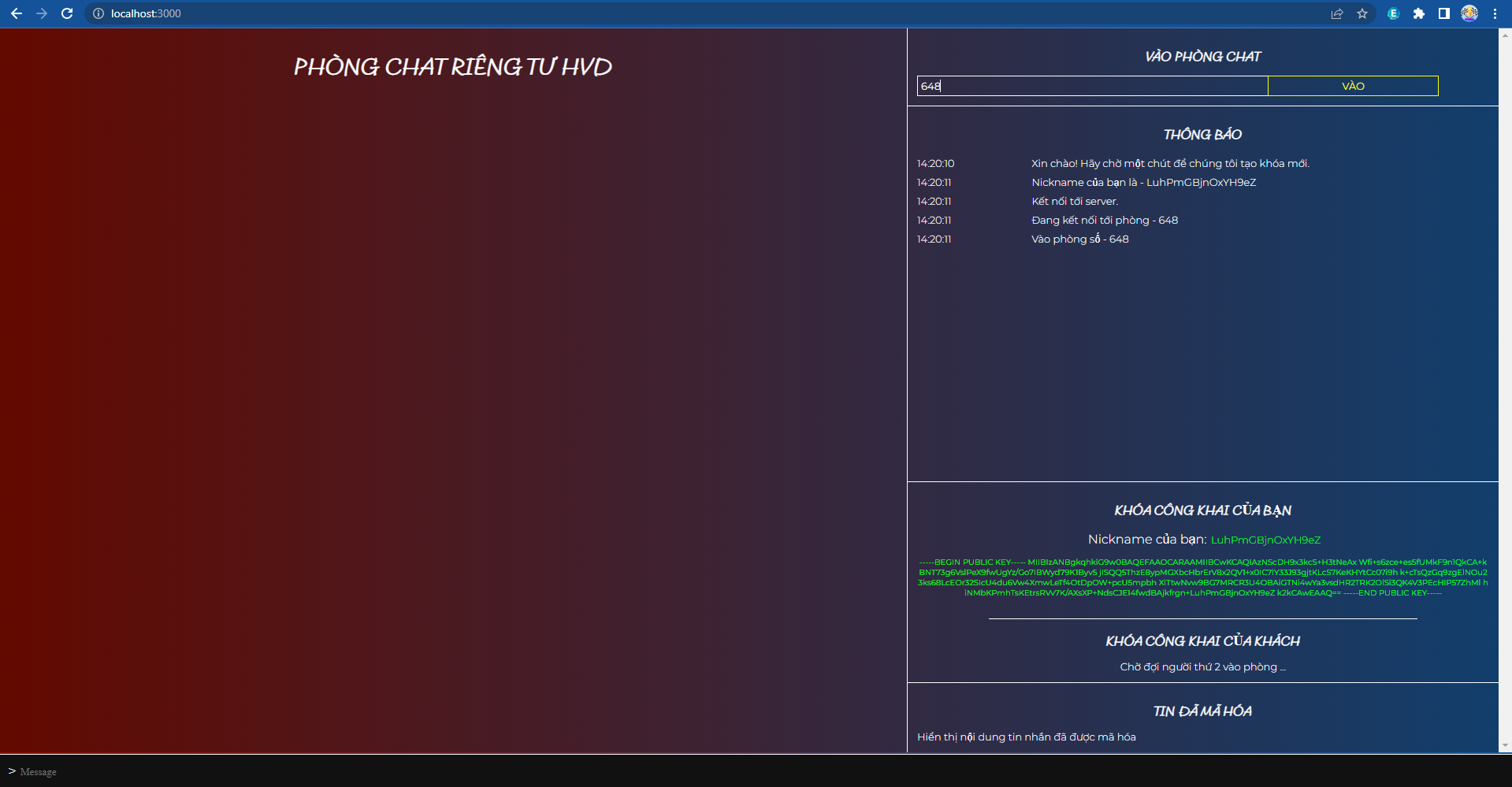
* Truy cập vào thư mục project.
* Mở terminal
* Dùng lệnh
  + - npm install
    - npm start



Hình II.21 Kết quả chạy lệnh. Server đã chạy

### Truy cập vào client

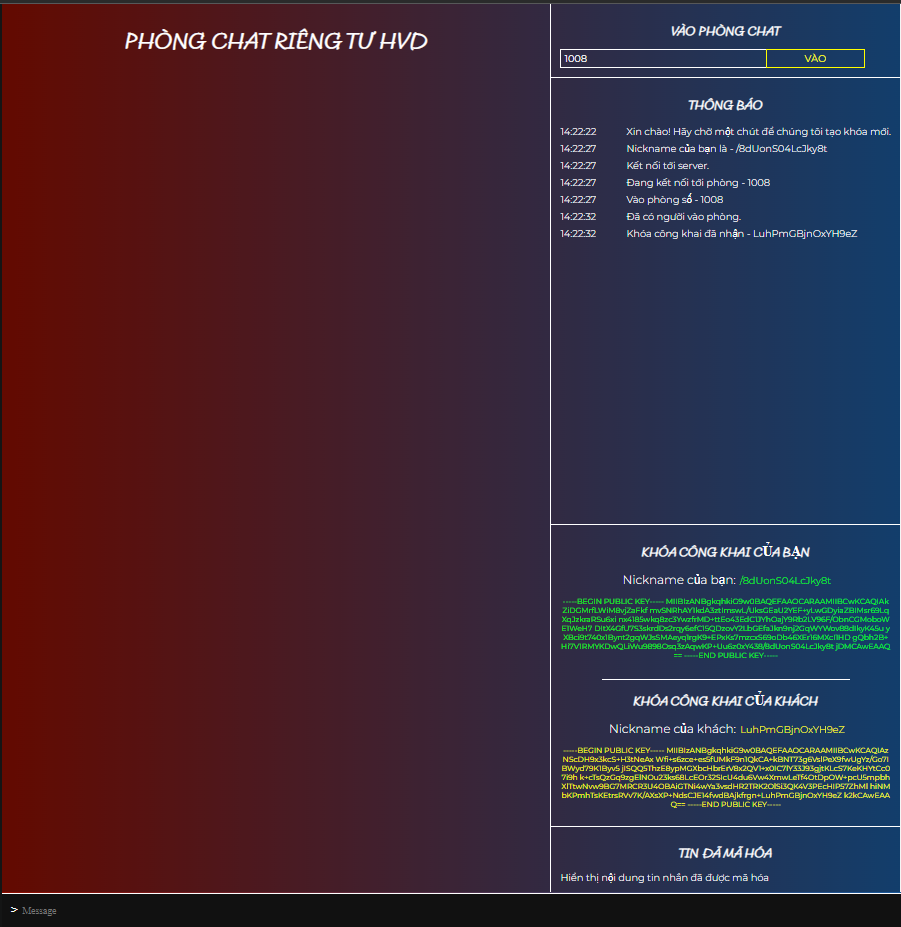
* Sử dụng 2 trình duyệt khác nhau – đại diện cho 2 cá nhân. Truy cập theo đường dẫn localhost:3000



Hình II.22 Giao diện tương tác Client

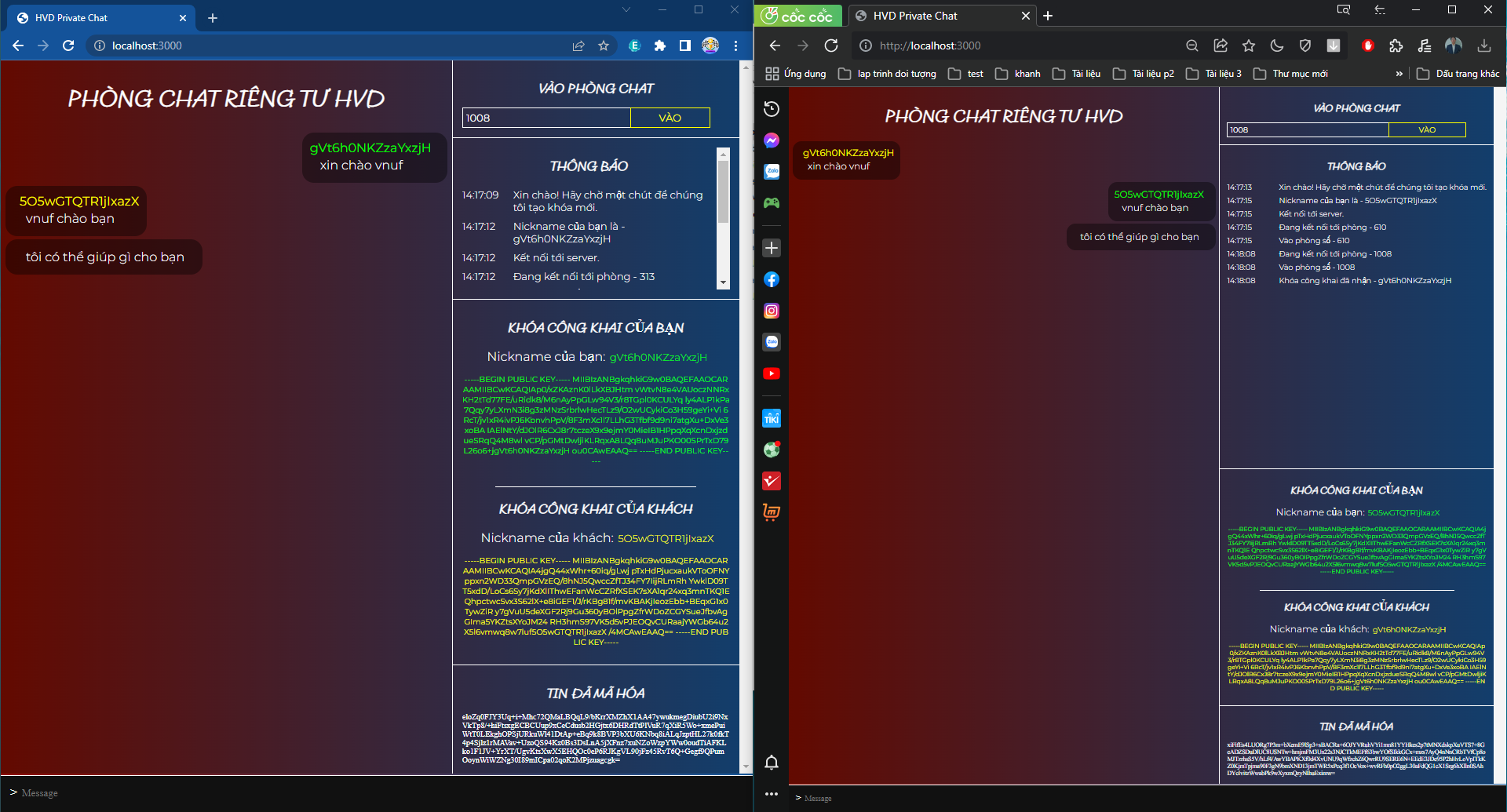
### Chọn phòng

* Hai người cùng truy cập vào một phòng, ở đây cả hai cùng vào phòng 1008, ta được giao diện như sau:L



Hình II.23 Giao diện khi đã kết nối

### Demo nhắn tin



Hình II.24 Demo nhắn tin