TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN: LẬP TRÌNH WEB NÂNG CAO**

**TÌM HIỂU VỀ GIAO THỨC WEB SOCKET**

**VÀ XÂY DỰNG MỘT TRANG WEB CHO PHÉP NGƯỜI DÙNG CHAT TRỰC TUYẾN**

Người hướng dẫn: **ThS. LÊ VĂN HÒA**

Người thực hiện: **NGÔ MINH TOÀN –** **51801031**

**HUỲNH AN LẠC - 51800691**

**LÊ QUANG LINH -** **51800423**

Khoá  **: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN: LẬP TRÌNH WEB NÂNG CAO**

**TÌM HIỂU VỀ GIAO THỨC WEB SOCKET**

**VÀ XÂY DỰNG MỘT TRANG WEB CHO PHÉP NGƯỜI DÙNG CHAT TRỰC TUYẾN**

Người hướng dẫn: **ThS. LÊ VĂN HÒA**

Người thực hiện: **NGÔ MINH TOÀN –** **51801031**

**HUỲNH AN LẠC - 51800691**

**LÊ QUANG LINH -** **51800423**

Khoá  **: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

Với tình cảm chân thành, chúng em xin bày tỏ lòng biết ơn tới trường Đại học Tôn Đức Thắng, đã tạo một môi trường học tập năng đông, tuyệt vời với một cơ sở vật chất đầy đủ tiện nghi và đội ngũ giảng viên kinh nghiệm, nhiệt huyết với nghề, thân thiện, tận tâm vì sinh viên, đặc biệt là Giảng viên ThS. Lê Văn Hòa đã tận tình chỉ dạy và giúp đỡ em hoàn thành bài tiểu luận môn Lập trình web nâng cao. Em cũng xin cảm ơn các thầy cô đã giảng dạy chúng em, cung cấp cho chúng em những kiến thức quý báu và bổ ích.

Do giới hạn kiến thức và khả năng lý luận của bản thân còn nhiều thiếu sót và hạn chế, kính mong sự chỉ dẫn và đóng góp của các thầy cô giáo để bài báo cáo của chúng em được hoàn thiện hơn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của ThS. Lê Văn Hòa. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 10 tháng 05 năm 2021*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Ngô Minh Toàn*

*Huỳnh An Lạc*

*Lê Quang Linh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Trong cuộc sống hiện tại, nhu cầu giao tiếp thông qua các nền tảng mạng xã hội, các ứng dụng web chat real-time ngày càng tăng. Từ khi các dịch vụ, ứng dụng trao đổi thông tin trực tuyến thông qua mạng Internet ra đời thì mọi người có thể dễ dàng liên hệ, trao đổi, trò truyện với nhau mọi lúc mọi nơi. Giao thức WebSocket được biết đến là một giao thức được sử dụng rộng rãi cho việc phát triển ứng dụng real-time. Trong bài tiểu luận này chúng ta sẽ cùng tìm hiểu về giao thức Websocket và xây dựng một trang web cho phép người dùng chat trực tuyến. Nội dung bài tiểu luận gồm các phần như sau:

* Chương 1: Mở đầu
* Lý do chọn đề tài
* Phương pháp nghiên cứu
* Chương 2: Cơ sở lý thuyết
* Tìm hiểu về socket và các khái niệm liên quan
* Tìm hiểu về giao thức Websocket
* Chương 3: Thực nghiệm
* Xây dựng trang web cho phép người dùng chat trực tuyến
* Chương 4: Tổng kết
* Kết quả nguyên cứu
* Kinh nghiệm đúc kết

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc71575044)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc71575045)

[TÓM TẮT iv](#_Toc71575046)

[MỤC LỤC 1](#_Toc71575047)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc71575048)

[CHƯƠNG 1 – MỞ ĐẦU 4](#_Toc71575049)

[1.1 Lý do chọn đề tài 4](#_Toc71575050)

[1.2 Phương pháp nghiên cứu 4](#_Toc71575051)

[CHƯƠNG 2 – CƠ SỞ LÍ THUYẾT 4](#_Toc71575052)

[2.1 Tìm hiểu chung về Socket và các khái niệm liên quan 4](#_Toc71575053)

[2.1.1 Tổng quan về socket 4](#_Toc71575054)

[2.1.2 Phân loại Socket 5](#_Toc71575055)

[2.1.2.1 Stream Socket 6](#_Toc71575056)

[2.1.2.2 Datagram Socket 7](#_Toc71575057)

[2.2 Websocket 9](#_Toc71575058)

[2.2.1 Web socket là gì? 9](#_Toc71575059)

[2.2.2 Lịch sử Websocket 10](#_Toc71575060)

[2.2.3 Ưu nhược điểm của Websocket 11](#_Toc71575061)

[2.2.4 Ứng dụng của Websocket 11](#_Toc71575062)

[CHƯƠNG 3 – THỰC NGHIỆM (DEMO) 12](#_Toc71575063)

[3.1 Source code 12](#_Toc71575064)

[3.1.1 Phía server 12](#_Toc71575065)

[3.1.2 Phía client 15](#_Toc71575066)

[3.2 Demo 18](#_Toc71575067)

[3.2.1 Đăng nhập 18](#_Toc71575068)

[3.2.2 Đăng ký 19](#_Toc71575069)

[3.2.3 Màn hình trang chủ 20](#_Toc71575070)

[3.4 Màn hình chat chi tiết 21](#_Toc71575071)

[CHƯƠNG 4 – TỔNG KẾT 22](#_Toc71575072)

[4.1 Kết quả nguyên cứu 22](#_Toc71575073)

[4.2 Kinh nghiệm đúc kết được 22](#_Toc71575074)

[TÀI LIỆU THAM KHẢO 23](#_Toc71575075)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1 Giao thức TCP và UDP 5](#_Toc71575076)

[Hình 2.2: Stream Socket 7](#_Toc71575077)

[Hình 2.1 Datagram Socket 8](#_Toc71575078)

[Hình 2.5 Ông Michael Carter 10](#_Toc71575079)

[Hình 3.1 thư viện Socket.io 12](#_Toc71575080)

[Hình 3.2 Đăng nhập 18](#_Toc71575081)

[Hình 3.3 Đăng ký 19](#_Toc71575082)

[Hình 3.4 Màn hình trang chủ 20](#_Toc71575083)

[Hình 3.5 Màn hình chat chi tiết 21](#_Toc71575084)

CHƯƠNG 1 – MỞ ĐẦU

1.1 Lý do chọn đề tài

Hiện nay với sự phát triển nhanh chóng của công nghệ thông tin và Internet, việc ứng dụng giao tiếp thời gian thực trên các niền tảng mạng xã hội và trang web vào công việc và học tập ngày càng phổ biến. Việc đó trở thành một phần không thể thiếu của cuộc sống hiện đại. Để tạo ra các ứng dụng web tương tác thời gian thực thì một không thể không nhắc đến giao thức Websocket - một giao thức được sử dụng rộng rãi cho việc phát triển ứng dụng real-time.

Nhận thấy được ý nghĩa của Websocket trong việc phát triển của công nghệ thông tin, đưa công nghệ thông tin tiếp cận gần hơn với nhiều đối tượng người dùng trong xã hội. Nhóm chúng em quyết định chọn đề tài “Tìm hiểu về giao thức Websocket và xây dựng một trang web cho phép người dùng chat trực tuyến”. Thông qua nguyên cứu chủ đề lần này, chúng em xin trình bày những hiểu biết và tìm hiểu của bản thân về Websocket.

1.2 Phương pháp nghiên cứu

Dựa vào những kiến thức học tập trên lớp từ giảng viên và nhưng kiến thức tìm hiểu được từ sách báo, các bài viết trên các diễn đàn và mạng xã hội trực tuyến. Từ đó trắc lọc, suy luận, tổng kết và trình bày lại những gì đã nguyên cứu, tìm hiểu được.

CHƯƠNG 2 – CƠ SỞ LÍ THUYẾT

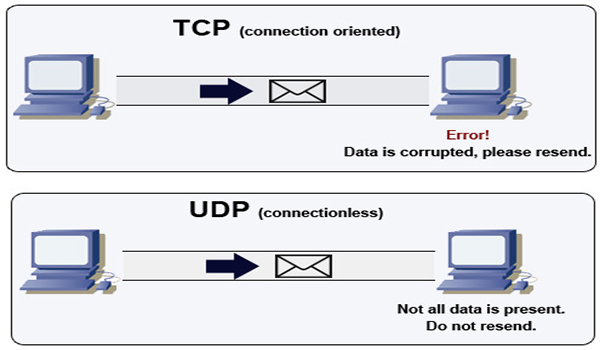
2.1 Tìm hiểu chung về Socket và các khái niệm liên quan

2.1.1 Tổng quan về socket

***Định nghĩa:*** Socket là điểm cuối end-point trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server.

Socket là một điểm cuối của liên kết giao tiếp hai chiều giữa hai chương trình chạy trên mạng, để 2 process nói chuyện được với nhau cần thông qua một socket. Socket chính là cửa giao tiếp giữa tầng ứng dụng (Application layer) và tầng giao vận (Transport layer). Có thể nói, Socket là giao diện do ứng dụng tạo ra trên máy trạm, được quản lí bởi hệ điều hành thông qua đó các ứng dụng có thể gửi thông điệp đến hoặc nhận thông điệp từ các ứng dụng khác. Socket sẽ được ràng buộc với một mã số cổng (được gọi là Port number) để giúp tầng giao vận định danh được ứng dụng nhận hoặc gửi thông điệp.

2.1.2 Phân loại Socket



Hình 2.1 Giao thức TCP và UDP

Tầng giao vận có 2 giao thức là TCP (Ttransmission Control Protocol) và UDP (User Datagram Protocol), cho nên socket cơ bản cũng có 2 loại: Stream Socket sử dụng TCP và Datagram Socket sử dụng UDP.

TCP/IP (Transmission Control Protocol (TCP)/ Internet Protocol (IP)) là giao thức cài đặt truyền thông, chồng giao thức mà hầu hết các mạng máy tính ngày nay đều sử dụng để kết nối. Đây là 2 giao thức đầu tiên trên thế giới được định nghĩa. **IP cho phép máy tính chuyển tiếp gói tin tới một máy tính khác. Thông qua một hoặc nhiều khoảng (chuyển tiếp) gần với người nhận gói tin. TCP sẽ giúp kiểm tra lỗi của các gói tin. Sau đó gửi yêu cầu truyền lại nếu có lỗi được tìm thấy.**

UDP (User Datagram Protocol) là một phần của bộ giao thức Internet được sử dụng bởi các chương trình chạy trên các máy tính khác nhau trên mạng. Khác với TCP/IP, UDP được sử dụng để gửi các gói tin ngắn gọi là datagram, cho phép truyền nhanh hơn. Tuy nhiên, UDP không cung cấp kiểm tra lỗi nên không đảm bảo toàn vẹn dữ liệu.

2.1.2.1 Stream Socket

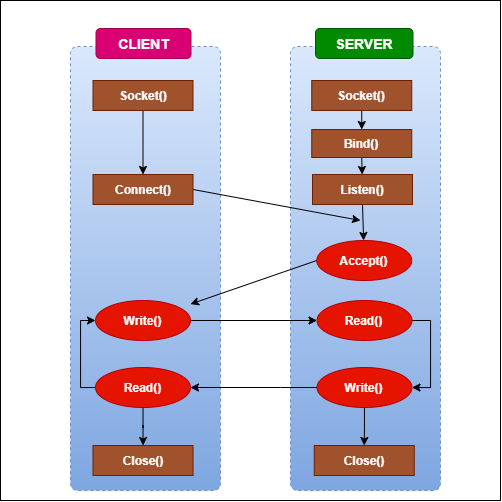
Stream Socket (TCP): Dựa trên giao thức TCP, việc truyền dữ liệu chỉ thực hiện giữa hai quá trình đã thiết lập kết nối. Hình thức này được gọi là Socket hướng kết nối.

**Ưu điểm:**

* Dữ liệu truyền đi được chính xác và nhanh chóng.
* Mỗi thông điệp gửi đi đều có xác nhận trả về kết quả của quá trình truyền tải thông điệp để thông báo cho người dùng biết.

**Nhược điểm:**

* Giữa máy chủ và máy nhận chỉ có 1 IP, nên khi kết nối, 1 máy phải chờ máy còn lại chấp nhận kết nối.



Hình 2.2: Stream Socket

2.1.2.2 Datagram Socket

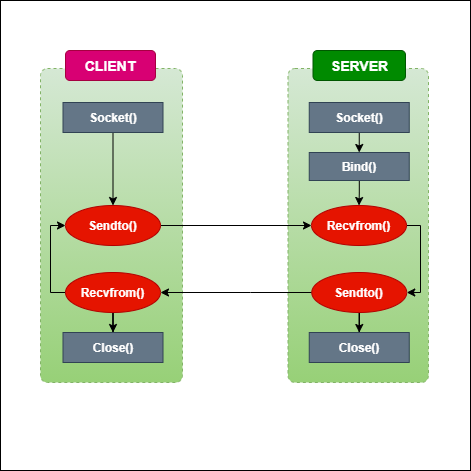
**Datagram Socket (UDP):** Dựa trên giao thức UDP, việc truyền dữ liệu không yêu cầu có sự kết nối giữa hai quá trình. Hình thức này được gọi là socket không hướng kết nối.

**Ưu điểm:**

* Quá trình kết nối và truyền tải thông tin đơn giản, không cần thực hiện nhiều thao tác.
* Thời gian truyền tải dữ liệu nhanh chóng.

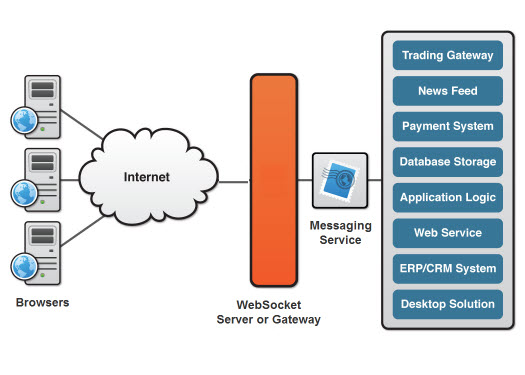
**Nhược điểm:**

* Quá trình truyền thông tin không đảm bảo tính tin cậy, thông tin có thể truyền sai thứ tự hoặc bị lặp.



Hình 2.1 Datagram Socket

2.2 Websocket



Hình 2.4 WebSocket

2.2.1 Web socket là gì?

Websocket là một công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Websocket được thiết kế sử dụng trong các ứng dụng web nhưng lập trình viên vẫn có thể sử dụng nó cho bất cứ ứng dụng nào. Nền tảng của Websocket bao gồm máy khách (Client-side) và máy chủ (Server-side). Hiện tại, công nghệ này phía Client đã được hỗ trợ trên hầu hết các trình duyệt phổ biến khác nhau như: Firefox, Google Chrome, Internet Explorer và Safari…

Bất kỳ trình duyệt nào có hỗ trợ WebSocket cho phép chạy một ứng dụng khách dựa trên HTML và JavaScript bằng cách sử dụng API HTML5 WebSocket. Một kết nối WebSocket bắt đầu với một bắt tay tương thích HTTP ban đầu, đảm bảo khả năng tương thích ngược lại để các kết nối WebSocket có thể chia sẻ các cổng HTTP (80) và HTTPS (443) mặc định. Khi bắt tay thành công, kết nối được mở để trao đổi dữ liệu, cho đến khi một trong hai bên kết thúc kết nối.

2.2.2 Lịch sử Websocket



Hình 2.5 Ông Michael Carter

WebSocket lần đầu tiên được tham chiếu là TCPConnection trong đặc tả HTML5, như một trình giữ chỗ (placeholder) cho API socket dựa trên TCP.

Michael Carter là một nhà công nghệ và doanh nhân khởi nghiệp với tư cách là một kỹ sư mạng. Tháng 6/2008, Michael đã thiết kế giao thức WebSocket ban đầu cho HTML5, một công nghệ được sử dụng trên hơn 2 tỷ thiết bị trên khắp thế giới mỗi ngày. WebSocket giúp dễ dàng xây dựng một lớp ứng dụng thời gian thực mới như trò chơi, mã chứng khoán, phòng trò chuyện và các ứng dụng khác,..

Tên giao thức "WebSocket" được đặt ra bởi Ian Hickson và Michael Carter ngay sau đó thông qua sự hợp tác trên phòng trò chuyện #whatwg IRC, và sau đó được Ian Hickson đưa vào đặc tả HTML5.

Tháng 12 năm 2009, Google Chrome 4 là trình duyệt đầu tiên cung cấp hỗ trợ đầy đủ cho tiêu chuẩn, với WebSocket được bật theo mặc định.

Sự phát triển của giao thức WebSocket sau đó đã được chuyển từ nhóm W3C và WHATWG sang IETF vào tháng 2 năm 2010, và là tác giả của hai bản sửa đổi dưới thời Ian Hickson.

Sau khi giao thức được vận hành và được kích hoạt theo mặc định trên nhiều trình duyệt, RFC đã được hoàn thiện theo Ian Fette vào tháng 12 năm 2011.

2.2.3 Ưu nhược điểm của Websocket

**Ưu điểm:**

* Websockets cung cấp khả năng giao tiếp hai chiều mạnh mẽ, dễ sử lý lỗi và có độ trễ thấp.
* API dễ sử dụng trực tiếp: không yêu cầu một thư viện tốt để xử lý kết nối lại, thời gian chờ (timeout), Ajax request, các tin báo nhận và các dạng truyền tải tùy chọn khác nhau (Ajax long-polling và jsonp polling).
* Không cần có nhiều kết nối.

**Nhược điểm:**

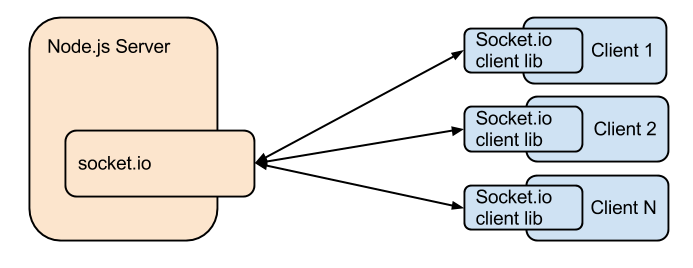
* WebSocket chưa được tất cả các trình duyệt hỗ trợ.
* Phạm vi yêu cầu không có. Lý do là bởi vì WebSocket không phải là HTTP request mà nó là TCP socket, chính vì vậy các dịch vụ có phạm vi yêu cầu như SessionInViewFilter của Hibernate sẽ không dễ dàng để sử dụng.

2.2.4 Ứng dụng của Websocket

Chúng ta có thể dễ dàng nhận thấy các ứng dụng của Websocket trong đời sống thường ngày. Chẳng hạn như: nhắn tín tức thời, trò chơi trực tuyến, giao dịch chứng khoán trực tuyến,..

Các ứng dụng của Websocket ngày càng phổ biến bởi sự phát triển của công nghệ trong xã hội hiện tại. Nhu cầu trao đổi dữ liệu và tương tác trực tuyến trên các trang mạng xã hội, các ứng dụng Web của con người ngày càng tăng, chính vì thế nó cũng là lí do Websocket thông dụng như hiện nay.

CHƯƠNG 3 – THỰC NGHIỆM (DEMO)



Hình 3.1 thư viện Socket.io

Để tạo ra một chương trình chat real-time ta sử dụng Nodejs và cài đặt thư viện socket.io bằng lệnh:

*npm install socket.io*

3.1 Source code

3.1.1 Phía server

const messModel = require('./models/messageModel')

let usersOnline = {}

// Cấu hình socket

const configSocket = (*app*) => {

    const server = require('http').createServer(*app*);

    // Gán socket vào server của ứng dụng

    const io = require('socket.io')(server);

    // Handle client mới (khi người dùng kết nối)

    io.on('connection', (*socket*) => {

        const \_sessionID = *socket*.id

        let userID

*socket*.on('setUser', (*\_userID*)=>{

            // Lưu thông tin người dùng đang online

            userID = *\_userID*

            usersOnline[userID] = \_sessionID

            // Cập nhật trạng thái online cho người dùng khác

            io.emit('updateStatus', usersOnline)

            console.log(`người dùng: ${userID} online`)

        })

        // Nhận tin nhắn

*socket*.on('sendMess', function({*mess*, *otherID*}){

            // Tạo thời gian

            let time\_mess = **new** *Date*(*Date*.now())

            // Lưu nội dung tin nhắn vào database

**new** messModel({fromID: userID, toID: *otherID*, content: *mess*, timeStamp: time\_mess}).save()

            console.log('gui tin nhan den id: '+ *otherID*)

            // Chuyển tiếp tin nhắn cho người dùng kia

            if (usersOnline[*otherID*]){

                io.to(usersOnline[*otherID*]).emit('newMess', {mess, time\_mess: time\_mess.toLocaleTimeString()})

            }

            else{

                console.log(`nguoi dung ${*otherID*} offline`)

            }

        })

        // Nhận được thông tin ngắt kết nối, cập nhật lại danh sách người dùng online

*socket*.on('disconnect', ()=>{

            console.log('got disconnect')

            delete usersOnline[userID]

            io.emit('updateStatus', usersOnline)

        })

    })

    return server

}

*module*.*exports* = {configSocket, usersOnline}

Socket cung cấp 2 methods chính là .on và .emit, trong đó:

* .on('event', callback): để lắng nghe sự kiện 'event' và trả về callback
* .emit('event', data): để gủi đi sự kiện 'event' và data

Khi nhận được sự kiện, data sẽ được đưa vào callback để xử lý

Có 2 sự kiện chính luôn được gọi khi kết nối đến client là:

* *'connection'*
* *'disconnect'*

Khi handle 'connection', data sẽ là socket riêng đến một client đã kết nối.

Khi handle 'disconnect' báo hiệu người dùng đã ngắt kết nối khỏi socket.

3.1.2 Phía client

// Tạo socket

const socket = io()

let user = {id, name, nickname} = document.getElementById('userIn4').dataset

// Kết nối đến server

socket.on('connect', function (){

    const sessionID = socket.id

    // Gủi thông tin người dùng cho server

    socket.emit('setUser', user.id)

    // Nếu người dùng đang chat với 1 người khác

    if(document.getElementById('otherIn4')){

        let other = {id, name, nickname} = document.getElementById('otherIn4').dataset

        // Lắng nghe tin nhắn từ người dùng đang chat

        socket.on('newMess', function({*mess*, *time\_mess*}){

            // Thêm tin nhắn vào box chat

            let messdiv = document.createElement('div')

            messdiv.innerHTML =

                `<li class="you">

                    <div class="entete ten-ben-aside">

                        <span class="status green"></span>

                        <h2>${other.nickname}</h2>

                        <h3>${*time\_mess*}</h3>

                    </div>

                    <div class="message">

                        ${*mess*}

                    </div>

                </li>`

            document.getElementById('chat').append(messdiv.firstChild)

            pageScroll()

        })

        // Lắng nghe, cập nhật danh sách người dùng đang online và offline

        socket.on('updateStatus', (*usersOnline*) => {

            console.log('da nhan tin hieu cap nhat')

            let status = document.getElementsByClassName('userName')

*Array*.prototype.forEach.call(status, *e* => {

                if (*usersOnline*.hasOwnProperty(*e*.dataset.id))

*e*.nextElementSibling.innerHTML = `<span class="status green"></span> online`

                else

*e*.nextElementSibling.innerHTML = `<span class="status orange"></span> offline`

            })

        })

    }

})

// Tự lăn đến tin nhắn mới nhận

function pageScroll() {

    document.getElementById('chat').scrollTo(0, 1000000000)

}

// Gủi tin nhắn đi

function sendMess(*otherID*){

    let mess = document.getElementById('message\_input').value

    if(mess.length <= 0)

        return

    document.getElementById('message\_input').value = ''

    // Gửi tin nhắn bằng sự kiện 'sendMess'

    socket.emit('sendMess', {mess, otherID})

    // Thêm tin nhắn vào box chat

    let messdiv = document.createElement('div')

    messdiv.innerHTML =

        `<li class="me">

            <div class="entete">

                <h3>${**new** *Date*(*Date*.now()).toLocaleTimeString()}</h3>

                <h2>You</h2>

                <span class="status blue"></span>

            </div>

            <div class="message">

                ${mess}

            </div>

        </li>`

    document.getElementById('chat').append(messdiv.firstChild)

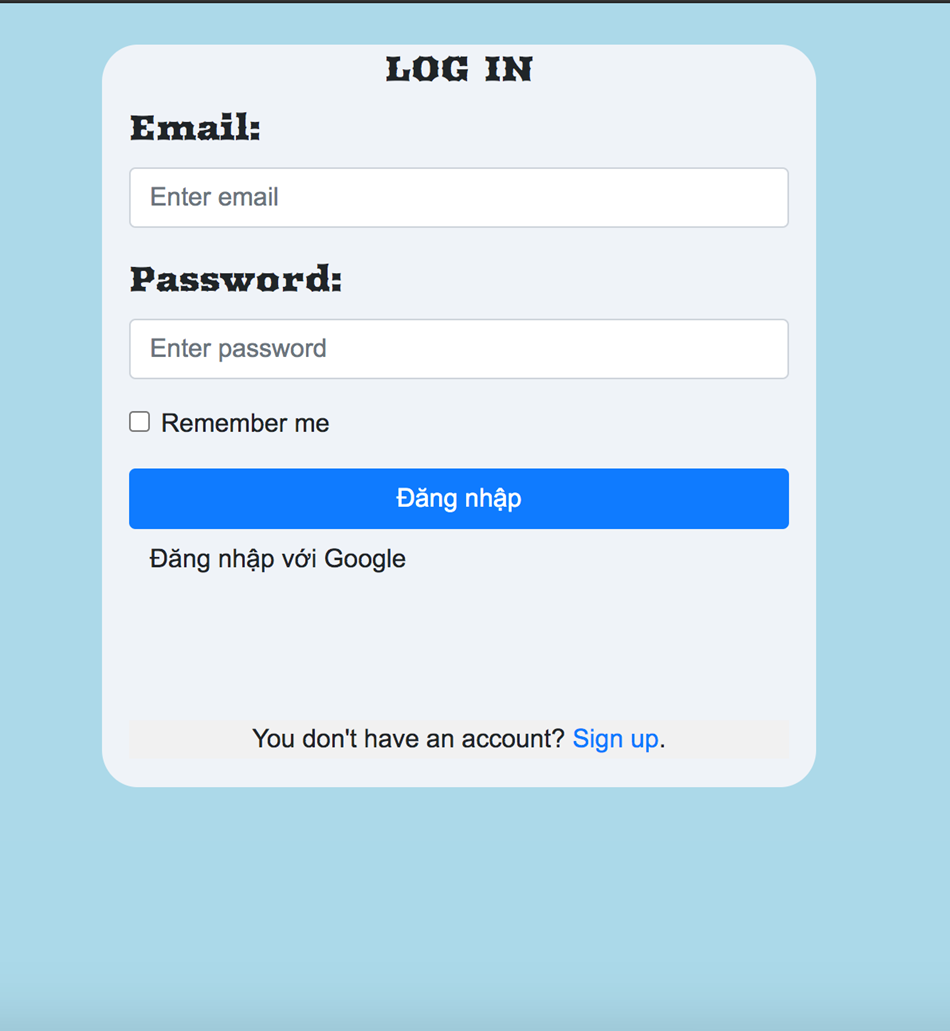
    pageScroll()

}

3.2 Demo

3.2.1 Đăng nhập

Người dùng tiến hành đăng nhập để sử dụng trang web.

****

Hình 3.2 Đăng nhập

3.2.2 Đăng ký

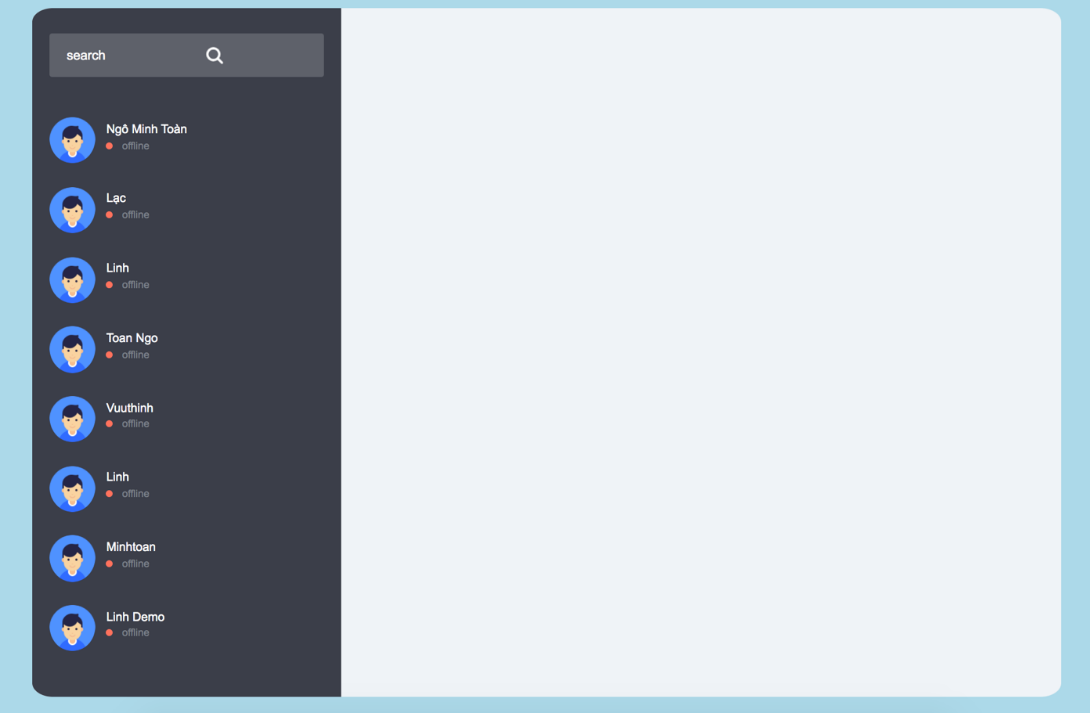
Nếu người dùng chưa có tài khoản. Ấn chọn “Sign In” để tiến hành đăng kí tài khoản.



Hình 3.3 Đăng ký

**3.2.3 Màn hình trang chủ**

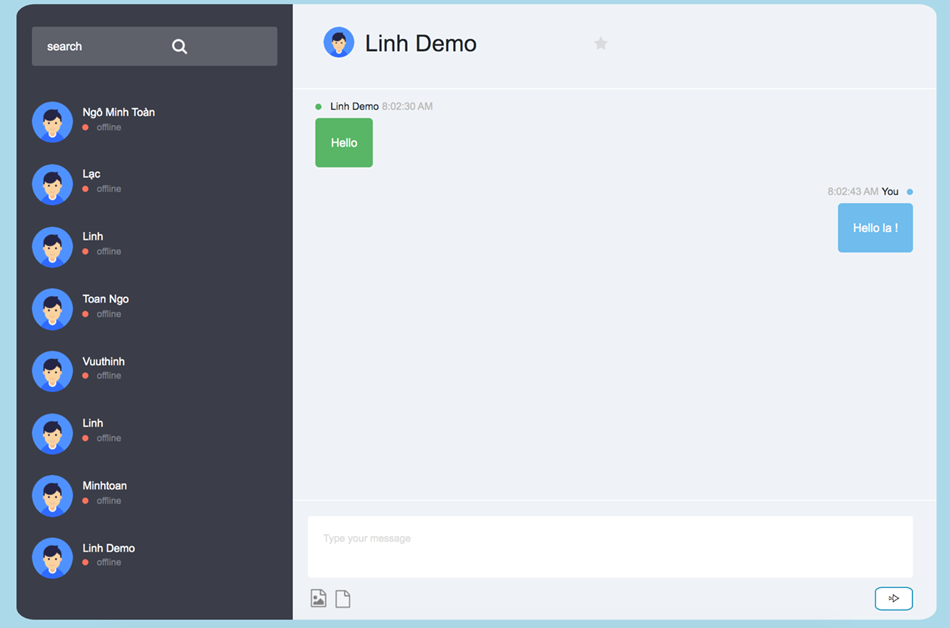
Sau khi đăng nhập thành công. Người dùng sẽ được chuyển đến trang chủ. Ở trang chủ sẽ hiển thị tất cả những tài khoản người dùng khác.



Hình 3.4 Màn hình trang chủ

3.4 Màn hình chat chi tiết

Người dùng ấn chọn người mình mình muốn chat.



Hình 3.5 Màn hình chat chi tiết

CHƯƠNG 4 – TỔNG KẾT

4.1 Kết quả nguyên cứu

Thông qua bài báo cáo, chúng ta đã có cơ hội tìm hiểu kỹ hơn về các khái niệm, ứng dụng của Socket và Websocket. Tự mình lập trình tạo ra chương trình chat trực tuyến bằng cách sử dụng thư viện socket.io trong Nodejs.

4.2 Kinh nghiệm đúc kết được

Trong quá trình thực hiện đề tài tiểu luận, chúng tôi đã đúc kết được nhiều kiến thức bổ ích cũng như kinh nghiệm xử lí đối với các tình huống trong làm việc nhóm. Trao dồi được các kỹ năng mềm như tra cứu tài liệu, đọc hiểu và diễn giải các kiến thức tìm hiểu được thông qua bài báo cáo.

TÀI LIỆU THAM KHẢO

1. <https://wiki.matbao.net/socket-la-gi-khai-niem-can-biet-ve-giao-thuc-tcp-ip-va-udp/>
2. <https://blog.tinohost.com/socket-la-gi/#Phan_loai_Socket>
3. <https://vietwebgroup.vn/blog/websocket-la-gi-tim-hieu-ve-websocket-la-gi.vietweb>
4. <https://viblo.asia/p/tao-mot-ung-dung-chat-don-gian-su-dung-nodejs-va-websocket-3P0lPz64Kox>
5. <https://www.google.com/amp/s/wiki.matbao.net/socket-la-gi-khai-niem-can-biet-ve-giao-thuc-tcp-ip-va-udp/amp/>
6. <https://bizflycloud.vn/tin-tuc/websocket-la-gi-uu-nhuoc-diem-cua-websocket-khi-su-dung-lam-phuong-thuc-giao-tiep-trong-moi-truong-internet-20210122155209401.htm>