

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN  
LẬP TRÌNH WEB NÂNG CAO**

# **WEB SOCKET**

*Người hướng dẫn:* **GV Mai Văn Mạnh**

*Người thực hiện:* **Nguyễn Hữu Tài – 518H0558**

**Nguyễn Thành Long – 518H0217**

**Lớp : 18H50205**

**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**



**BÀI TẬP LỚN  
LẬP TRÌNH WEB NÂNG CAO**

# **WEB SOCKET**

*Người hướng dẫn:* **GV Mai Văn Mạnh**

*Người thực hiện:* **Nguyễn Hữu Tài – 518H0558**

**Nguyễn Thành Long – 518H0217**

**Lớp : 18H50205**

**Khoá : 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

## LỜI CẢM ƠN

Đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc đến Trường Đại học Tôn Đức Thắng và khoa Công Nghệ Thông Tin đã tạo điều kiện để lớp học được mở cũng như cung cấp mọi điều kiện tốt nhất cho khóa học. Đặc biệt, chúng em xin chân thành cảm ơn Giảng viên Mai Văn Minh đã quan tâm, giúp đỡ, hướng dẫn em mọi lúc để chúng em có thể hoàn thành bài báo cáo trong thời gian qua.

Bài báo cáo này là sản phẩm của riêng chúng em, nên không thể tránh khỏi những sai sót. Kính mong nhận được sự chỉ bảo đóng góp ý kiến của các thầy cô, cũng như mọi người để chúng em ngày càng hoàn thiện thêm kiến thức.

Chúng Em xin chân thành cảm ơn!

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của GV Mai Văn Mạnh;. 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 tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Thành Long*

*Nguyễn Hữu Tài*

## **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**

Đây là ứng dụng chat trực tuyến áp dụng web socket.

## MỤC LỤC

### Contents

LỜI CẢM ƠN .....	1
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	3
Phản xác nhận của GV hướng dẫn.....	3
Phản đánh giá của GV chấm bài .....	3
TÓM TẮT .....	4
MỤC LỤC.....	1
CHƯƠNG 1 .....	2
GIỚI THIỆU WEBSOCKET.....	2
CHƯƠNG 2 .....	5
ÁP DỤNG WEBSOCKET VÀO ỨNG DỤNG CHAT.....	5
Models/ mess.js.....	6
Users/users.js.....	7
Server.js.....	8
Tài Liệu Tham Khảo .....	13

## CHƯƠNG 1

### GIỚI THIỆU WEBSOCKET



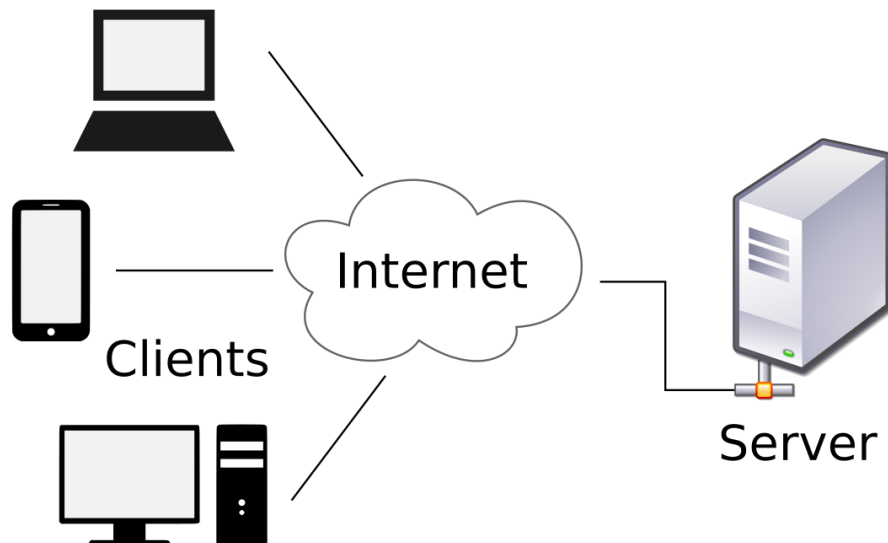
Websocket là giao thức hỗ trợ giao tiếp hai chiều giữa client và server để tạo một kết nối trao đổi dữ liệu. Giao thức này không sử dụng HTTP mà thực hiện nó qua TCP.. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.

Ở đây Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.



Đồng thời Websocket có thể được sử dụng cùng lúc nhiều lần để tiết kiệm thời gian và nâng cao năng suất công việc.

Bên cạnh đó, Websocket có 1 ưu điểm rất lớn là hỗ trợ hầu hết các hệ điều hành như MS Windows, Linux, ... với nhiều loại ngôn ngữ lập trình gồm C, C++, Java, NodeJs, ... điều này giúp cho nó dễ dàng tương thích với hầu hết mọi cấu hình máy khác nhau.



Chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận dữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng (nếu trên cùng 1 máy thì không được trùng số hiệu với nhau) của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

Websocket mang lại rất nhiều ưu điểm khác chẳng hạn như :

- + Tăng tốc độ truyền tải thông tin giữa 2 chiều.
- + Dễ phát hiện và xử lý trong trường hợp có lỗi xảy ra
- + Dễ dàng sử dụng, không cần cài đặt thêm các phần mềm bổ sung khác
- + Không cần sử dụng nhiều phương pháp kết nối khác nhau

Bên cạnh các ưu điểm thì dĩ nhiên WebSocket cũng có các nhược điểm của riêng nó như:

- Chưa hỗ trợ trên tất cả các trình duyệt
- Với các dịch vụ có phạm vi yêu cầu, Websocket chưa hỗ trợ hoàn toàn.

Cách thức hoạt động của Websocket gồm:

1. Client sẽ gửi yêu cầu khởi tạo kết nối đến server
2. Server kiểm tra và gửi lại kết quả chấp nhận yêu cầu kết nối từ Client
3. Kết nối được tạo ra
4. Quá trình gửi dữ liệu(dữ liệu chính là các Ws frame) giữa 2 chiều có thể được thực hiện.

## CHƯƠNG 2

# ÁP DỤNG WEBSOCKET VÀO ỨNG DỤNG CHAT

Socket.io là 1 thư viện dùng để dựng nên ứng dụng realtime như app chat trực tuyến thông qua việc truyền dữ liệu ngay lập tức từ server trung gian.

Thông thường với socketio gồm 2 phần đó là:

- Client
- Server

Client: Nơi mà để tạo giao diện người dùng như ,js,jquery. ..(Trong app tại e sẽ áp dụng js )

Server:Nơi dùng để cài đặt socketio.Ở phần này chúng e áp dụng Nodejs để dựng server vì nó giúp cài trực tiếp socketio vào cùng 1 server khác với php sẽ phải tạo nhiều package khác nhau hoặc thậm chí phải tạo 1 server riêng để chạy socketio.

Những modules thông thường được chạy khi dùng socketio như express,moment,socket.io hoặc thêm mongoose để lưu các tin nhắn vào database.

Thông tin chi tiết về ứng dụng demo em sẽ thể hiện qua video demo, Ở đây e sẽ nêu ra những chi tiết chính của ứng dụng chat.

## Models/ mess.js

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema
const messSchema = new Schema({
  user: {
    type: String,
    required: true
  },
  time: {
    type: String,
    required: true
  },
  content: {
    type: String,
  },
  room: {
    type: String,
    required: true
  }
})

module.exports = mongoose.model('Mess', messSchema)
```

Thông tin của một tin nhắn sẽ bao gồm:

1. User - tên của username
2. Time- Thời gian tạo tin nhắn
3. Content- nội dung tin nhắn
4. Room – Tên của phòng nhắn tin

Users/users.js

```

const users = [];

// Join user to chat
function userJoin(id, username, room) {
  const user = { id, username, room };
  users.push(user);
  return user;
}

// Get current user
function getCurrentUser(id) {
  return users.find(user => user.id === id);
}

// User leaves chat
function userLeave(id) {
  const index = users.findIndex(user => user.id === id);

  if (index !== -1) {
    return users.splice(index, 1)[0];
  }
}

// Get room users
function getRoomUsers(room) {
  return users.filter(user => user.room === room);
}

module.exports = {
  userJoin,
  getCurrentUser,
  userLeave,
  getRoomUsers
};

```

Đây là nơi để tương tác với các user.

Biến “users” là một danh sách sẽ lưu id của các user đang đăng nhập và tương tác với hệ thống.

Hàm `getCurrentUser( id )`: có tham số truyền vào là id, dùng để lấy id của user đang đăng nhập.

Hàm `userLeave( id )`: có tham số truyền vào là id, dùng để xóa id của user khi user đó đăng xuất

Hàm `getRoomUsers( room )`: có tham số truyền vào là room, dùng để lưu phòng khi user chọn phòng.

## Server.js

Đây là nơi Server tương tác với các client và kết nối với mongoose.

- Các hoạt động chính trong server.js

### 1. Connection:

```
io.on('connection', socket => {
  socket.on('joinRoom', async ({ username, room }) => {
    const user = userJoin(socket.id, username, room);

    socket.join(user.room);

    //Load existing message in room
    try {
      let message = await Mess.find();
      message.forEach(mess => {
        if (mess.room == user.room) {
          socket.emit("loadMessage", formatOldMessage(mess.user,
            mess.content, mess.time));
        }
      })
    } catch (err) {
      console.log(err);
    }

    console.log(message);
  })

  // Welcome current user
  socket.emit('message', formatMessage(botName, `Welcome to
    ${user.room}!`));
});
```

```
// Broadcast when a user connects
socket.broadcast
  .to(user.room)
  .emit(
    'message',
    formatMessage(botName, `${user.username} has joined the
chat`)
  );

// Send users and room info
io.to(user.room).emit('roomUsers', {
  room: user.room,
  users: getRoomUsers(user.room)
});
});
```

Khi client vừa đăng nhập, hệ thống sẽ tạo ra 1 đối tượng user bao gồm tên user và phòng chat, sau đó server sẽ thông báo vào phòng chat các client có user mới vào phòng.

Phía client vừa đăng nhập hệ thống sẽ load lại các tin nhắn đã có ở phòng và hiển thị cho client đó

## 2. Chat Message:

```
// Listen for chatMessage
socket.on('chatMessage', async msg => {
  const user = getCurrentUser(socket.id);

  const mess = new Mess({
    user: user.username,
    time: moment().format('h:mm a'),
    content: msg,
    room: user.room,
  });

  try {
    const newMess = await mess.save()
    console.log("mess: ", newMess)
  } catch (e) {
    console.log(e)
  }

  io.to(user.room).emit('message', formatMessage(user.username,
msg));
});
```

Khi server nhận được tin nhắn từ client, server sẽ lưu nó vào mongooseDB để phục vụ cho việc các client khác khi đăng nhập vào cũng sẽ được hiển thị.

Ngoài ra server sẽ tạo 1 tin nhắn với nội dung là tin nhắn từ phía client gửi cho các client còn lại

### 3. Disconnect:

```
socket.on('disconnect', () => {
  const user = userLeave(socket.id);

  if (user) {
    io.to(user.room).emit(
      'message',
      formatMessage(botName, `${user.username} has left the chat`)
    );

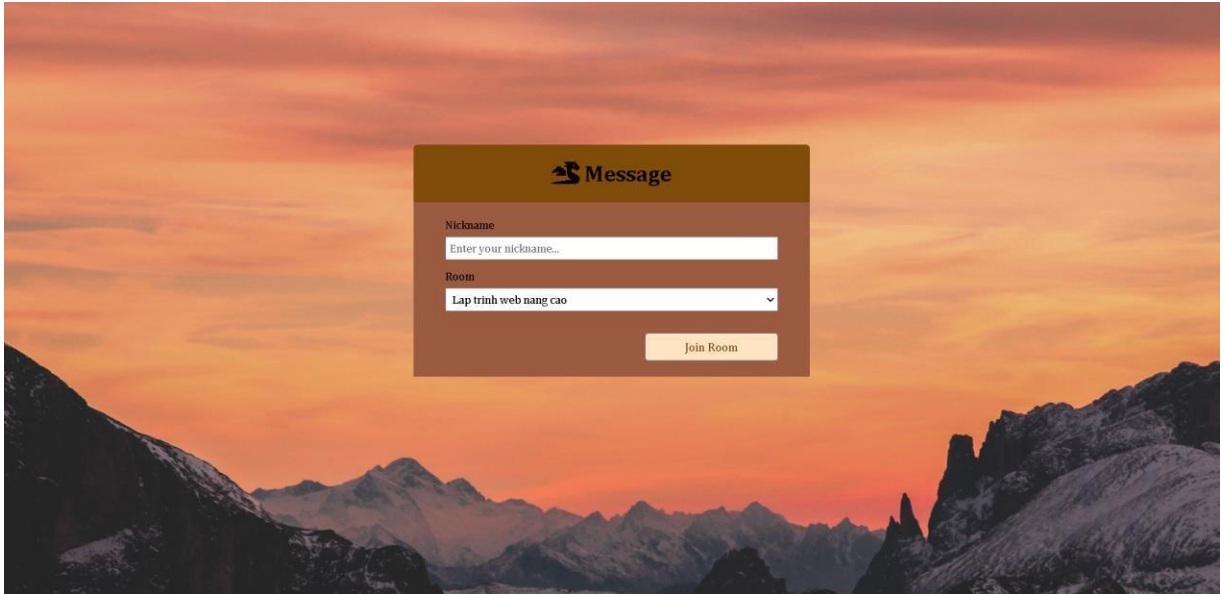
    // Send users and room info
    io.to(user.room).emit('roomUsers', {
      room: user.room,
      users: getRoomUsers(user.room)
    });
  }
});
```

Khi client thoát ra khỏi phòng, server bắt hành động đó và thực hiện 2 công việc:

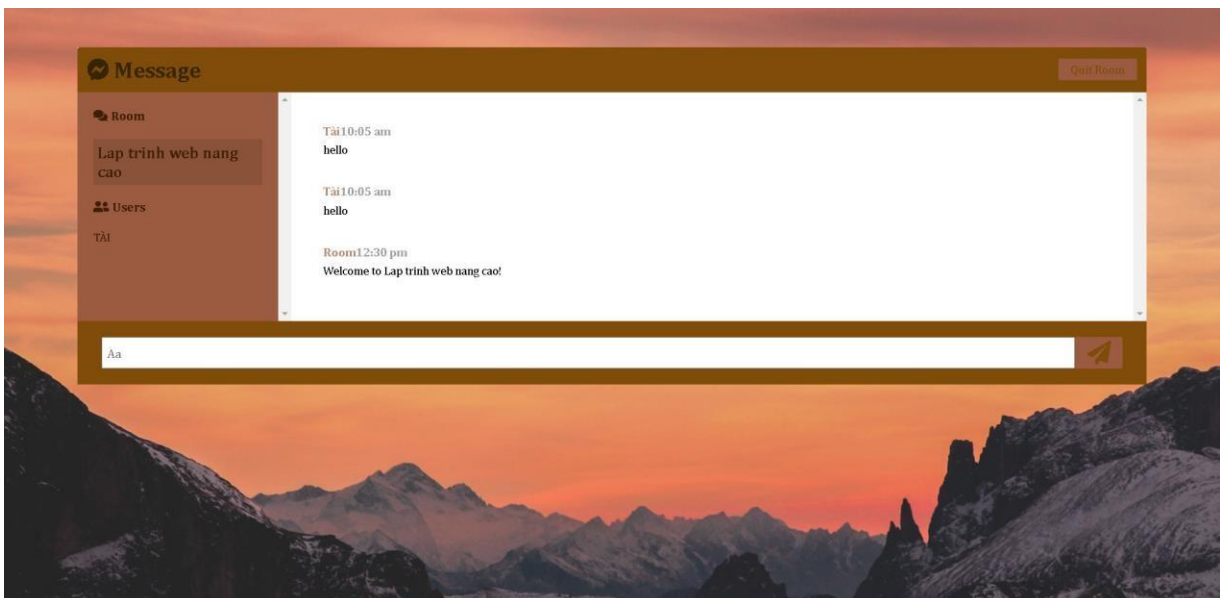
- Xóa user ra khỏi danh sách users đang tương tác với server
- Thông báo cho mỗi client trong phòng bằng tin nhắn ở cửa sổ chat với nội dung có client thoát ra



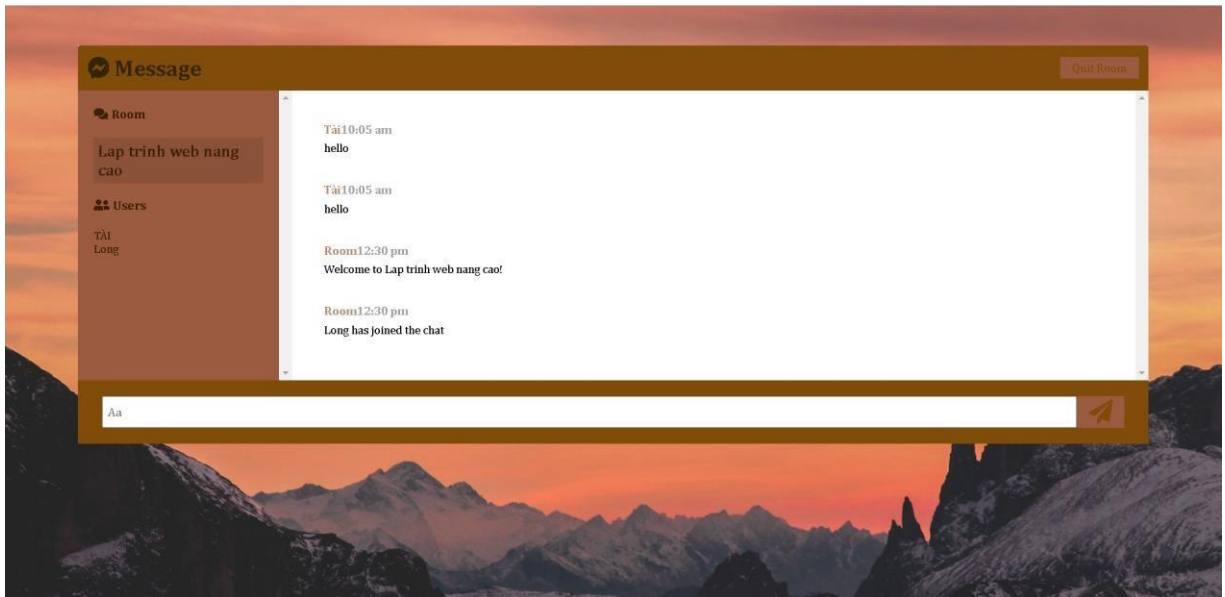
Dưới đây là 1 vài hình ảnh giao diện demo:



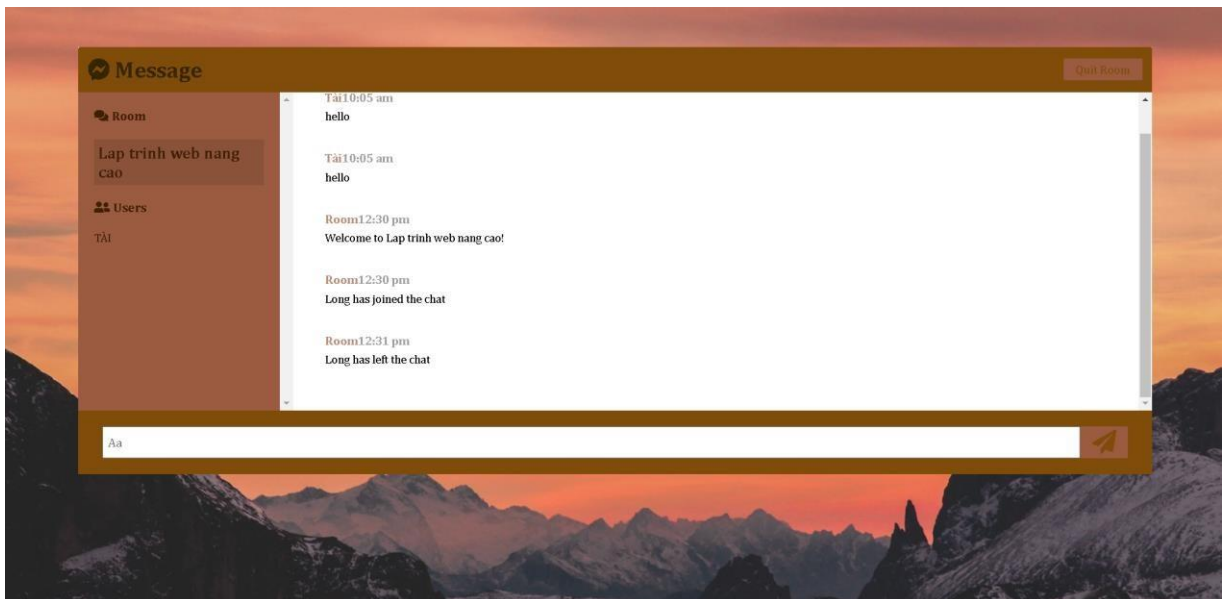
Hình 2.1: Giao diện chính của ứng dụng chat



Hình 2.2 :Giao diện phần chat



Hình 2.3: Khi có 1 user mới tham gia vào



Hình 2.4: Khi 1 user out ra

### **Tài Liệu Tham Khảo**

<https://www.youtube.com/watch?v=rxzOqP9YwmM>

<https://viblo.asia/p/co-ban-ve-socketio-bJzKm0kY59N>