**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**🙡🙞🕮🙞🙣**

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: MẠNG MÁY TÍNH**

**ĐỀ TÀI: Lập trình SOCKET ứng dụng chat**

**GIẢNG VIÊN HƯỚNG DẪN: PGS.TS: TRẦN QUANG VINH**

**NHÓM SINH VIÊN THỰC HIỆN:**

1. MAI VĂN KIÊN
2. TRẦN TUẤN MINH
3. NGÔ KHÁNH MAY
4. TRẦN ĐỨC TÂM

**HÀ NỘI – 2022**

­­­­NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

....................................................................................................................................................

**GIẢNG VIÊN HƯỚNG DẪN**

**Mục Lục**

[LỜI CẢM ƠN 4](#_Toc128860493)

[TÓM TẮT 5](#_Toc128860494)

[Chương 1: TỔNG QUAN 6](#_Toc128860495)

[1.1 ĐẶT VẤN ĐỀ 6](#_Toc128860496)

[1.2 MỤC TIÊU 6](#_Toc128860497)

[1.3 NỘI DUNG NGHIÊN CỨU 6](#_Toc128860498)

[Chương 2: CƠ SỞ XÂY DỰNG ỨNG DỤNG 7](#_Toc128860499)

[2.1 GIỚI THIỆU PHẦN MỀM 7](#_Toc128860500)

[**2.1.1 Giới thiệu về NodeJs** 7](#_Toc128860501)

[**2.1.2 Giới thiệu về SOCKET** 7](#_Toc128860502)

[**2.1.3 Giới thiệu về socket.io** 9](#_Toc128860503)

[**2.1.4 Giới thiệu về React** 11](#_Toc128860504)

[**2.1.5 Giới thiệu về MongoDB** 11](#_Toc128860505)

[2.2 XÂY DỰNG ỨNG DỤNG 12](#_Toc128860506)

[**2.2.1 Xây dựng giao diện** 12](#_Toc128860507)

[**2.2.2 Xây dựng code phía server bằng nodejs và socket.io** 15](#_Toc128860508)

[**2.2.3 Chạy dự án trên local** 16](#_Toc128860509)

[**2.2.4 Kết quả chạy thực tế** 17](#_Toc128860510)

[**2.2.5 Kết quả đạt được** 18](#_Toc128860511)

[Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 19](#_Toc128860512)

[4.1 KẾT LUẬN 19](#_Toc128860513)

[4.2 HƯỚNG PHÁT TRIỂN 19](#_Toc128860514)

LỜI CẢM ƠN

Trước tiên, nhóm em xin gửi lời cảm ơn tới thầy hướng dẫn PGS.TS. Trần Quang Vinh đã rất tận tình chỉ bảo nhóm em trong quá trình thực hiện. Nhóm em cũng xin cảm ơn các Thầy/Cô Khoa Công nghệ Điện – Điện Tử, các cán bộ Khoa Phòng Trường Đại học mở Hà Nội đã quan tâm, tạo điều kiện giúp đỡ nhóm em trong quá trình học tập và nghiên cứu tại Trường.

Trân trọng cảm ơn!

# **TÓM TẮT**

Trong thời đại phát triển của công nghệ thông tin hiện nay, do nhu cầu đòi hỏi cao nên các ứng dụng nhắn tin ngày càng phổ biến và được phát triển trên cả nền tảng web và app di động. Ví dụ như Messenger, Zalo, WhatApps, Telegram... Trong giới hạn đè tài được giao là thiết kế ứng dụng chat sử dụng Socket, nhóm em đã sử dụng Javascript với thư viện React để viết giao diện, platform Nodejs, cơ sở dữ liệu MongoDB cùng với thư viện socket.io để thực hiện. Ứng dụng được lập trình và chạy trên giao diện web. Chúng em cảm ơn Thầy Trần Quang Vinh đã giúp đỡ chúng em trong quá trình thực hiện đề tài.

**Nội dung chính của đề tài**

* Sử dụng React để viết giao diện web
* Sử dụng Nodejs và socket.io để tạo kết nối realtime từ client với server
* Sử dụng MongoDB để lưu trữ tài khoản người dùng, và thông tin các phòng chat

# **Chương 1: TỔNG QUAN**

* 1. **ĐẶT VẤN ĐỀ**

Việc sử dụng ứng dụng chat để giao tiếp trực tiếp đã trở thành một phần không thể thiếu trong cuộc sống hiên đại. Tuy nhiên, việc xây dựng một ứng dụng chat đáp ứng được yêu cầu của người dùng với tốc độ và tính bảo mật cao là một thách thức lớn. Nodejs là một nền tảng lập trình phổ biến, được sử dụng cho các ứng dụng web có hiệu suất cao.

Do đó, đề tài xây dựng ứng dụng chat sử dụng nodejs là một vấn đề đáng quan tâm

Khi dự án thành công sẽ là tiền đề để phát triển các ứng dụng cho cuộc sống thường ngày.

**1.2 MỤC TIÊU**

Thiết kế giao diện cho ứng dụng chat sử dụng react

Kết nối với server nodejs bằng socket.io

Kết nối tới cơ sở dữ liệu MongoDB lưu trữ thông tin người dùng

**1.3 NỘI DUNG NGHIÊN CỨU**

Tìm hiểu về nodejs

Tìm hiểu về thư viện react

Tìm hiểu về socket.io

Tìm hiểu về cơ sở dữ liệu MongoDB

# **Chương 2: CƠ SỞ XÂY DỰNG ỨNG DỤNG**

## 2.1 GIỚI THIỆU PHẦN MỀM

### **2.1.1 Giới thiệu về NodeJs**

**NodeJs** là một môi trường runtime chạy Javascript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

Ưu và nhược điểm của nodejs

Ưu điểm

+ IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.

+ Sử dụng Javascript – một ngôn ngữ lập trình dễ đọc.

+ Chia sẻ cùng code ở cả phía client và server.

+ NPM ( Node Package Manager ) và module Node đang phát triển mạnh.

+ Cộng đồng hộ trợ mạnh.

+ Cho phép stream file có kích thước lớn.

Nhược điểm

+ Không có khả năng mở rộng, vì vậy khó tận dụng được lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.

+ Khó thao tác với cơ sở dữ liệu.

+ Cần đòi hỏi có kiến thức về javascript.

+ Không phù hợp với các tác vụ đòi nhiều CPU.

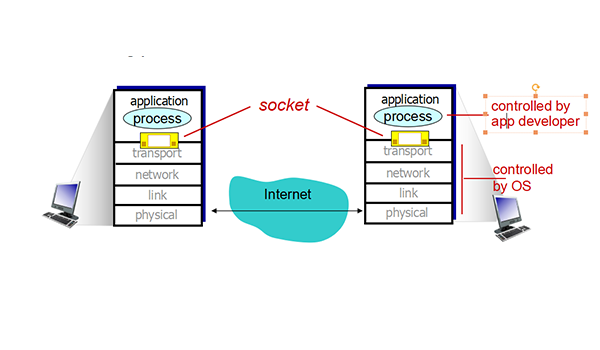
### **2.1.2 Giới thiệu về SOCKET**

- Khái niệm socket

**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 giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

Một chức năng khác của socket là giúp các tầng **TCP** hoặc **TCP Layer** định danh ứng dụng mà dữ liệu sẽ được gửi tới thông qua sự ràng buộc với một cổng port (thể hiện là một con số cụ thể), từ đó tiến hành kết nối giữa client và server.

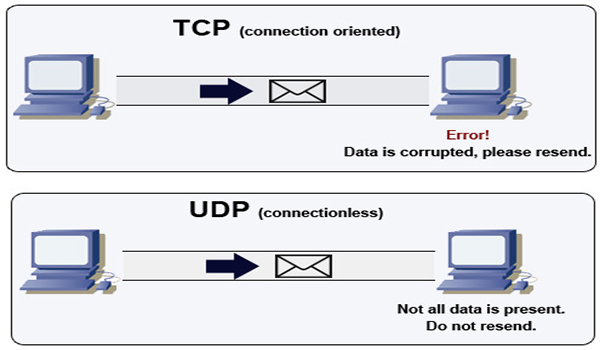
* Tại sao dùng socket



Ưu điểm lớn nhất của socket là hỗ trợ hầu hết các hệ điều hành bao gồm MS Windows, Linux,… Ngoài ra, socket cũng được sử dụng với nhiều ngôn ngữ lập trình, gồm C, C++, Java, Visual Basic, Visual C++,… nên nó có thể tương thích với hầu hết mọi đối tượng người dùng với những cấu hình máy khác nhau.

Đặc biệt, người dùng cũng có thể chạy cùng một lúc nhiều socket liên tục, giúp nâng cao hiệu suất làm việc, cũng như tiết kiệm thêm nhiều thời gian và công sức hơn.

* Socket hoạt động như thế nào



Như đã đề cập trước đó, chức năng của **socket** là kết nối giữa client và server thông qua **TCP/IP** và**UDP** để truyền và nhận giữ 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 của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

2 ứng dụng cần truyền thông tin phải đáp ứng điều kiện sau thì socket mới có thể hoạt động:

* 2 ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau
* Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

- Phân loại socket

**Socket**được chia làm 4 phân loại khác nhau. Dưới đây, hãy cùng Mắt Bão tìm hiểu chi tiết về:

* Stream Socket
* Datagram Socket
* Websocket
* Unix socket

### **2.1.3 Giới thiệu về socket.io**

- Socket.io là gì

Để xây dựng một ứng dụng realtime cần sử dụng socketio. Socketio sẽ giúp các bên ở những địa điểm khác nhau kết nối với nhau, truyền dữ liệu ngay lập tức thông qua server trung gian. Socketio có thể được sử dụng trong nhiều ứng dụng như chat, game online, cập nhật kết quả của một trận đấu đang xảy ra,...

Socketio không phải là một ngôn ngữ, mà chỉ là 1 công cụ giúp thực hiện những ứng dụng realtime. Vì thế, không thể sử dụng socketio để thay thế hoàn toàn cho một ngôn ngữ, mà phải sử dụng kết hợp với một ngôn ngữ khác. Ngôn ngữ đó có thể là php, [asp.net](http://asp.net/), nodejs,...

* Cách sử dụng socket.io

Cấu trúc một ứng dụng realtime sử dụng socket bao gồm 2 phần: phía server, phía client.

1. Phía server Đây là nơi sẽ cài đặt socket io. Ngôn ngữ để dựng server có thể là php, [asp.net](http://asp.net/), nodejs,... Tuy nhiên, tùy vào ngôn ngữ lựa chọn mà cách cấu trúc server khác nhau. Ở đây, nếu được thì khuyến khích sử dụng nodejs để dựng server, vì như vậy có thể cài trực tiếp socketio vào cùng một server. Nếu sử dụng php thì phải cài thêm những package khác, hoặc phải chuẩn bị riêng server để chạy socketio.
2. Phía client: Ở phía client sẽ xây dựng giao diện người dùng. Ở đây có thể sử dụng js, hoặc các thư viện của js như jquery,... Nói chung là ngôn ngữ gì cũng được.
3. Những thứ cần chuẩn bị để sử dụng socketio

* PC: Mac hoặc Win đều được
* Download phần mềm socketio

1. Cài đặt socketio trên server nodejs Tạo thư mục Demo (Có thể đặt tên bất kỳ) Mở màn hình terminal (Mac)/ cmd (Win), cd đến thư mục Demo. Tại màn hình cmd, gõ câu lệnh: npm init; rồi nhấn enter. Khi đó hệ thống sẽ chạy và yêu cầu nhập tên dự án, nhập tên dự án bất kỳ. Ở các setting khác thì enter bỏ qua, khi hệ thống hiển thị Yes or No? thì gõ Yes để cài đặt.

Sau khi hoàn thành bước cài đặt ở trên, hệ thống sẽ tạo ra file package.json. Đây là file dùng để cài đặt cấu hình server. Tiếp theo, cần đặt những package cần thiết trên server để có thể làm web và ứng dụng realtime. Để cài những package này mở màn hình cmd > cd đến thư mục Demo > gõ dòng lệnh npm install express ejs socketio Khi cài đặt thành công, hệ thống sẽ tự tạo thư mục nodemodules

- Cơ chế lắng nghe, truyền dữ liệu của socket.io

Để lắng nghe data, ta sử dụng câu lệnh socket.on(), để phát dữ liệu thì sử dụng lệnh socket.emit() .

Ví dụ, client gửi 1 đoạn chat đi, thì khi đó ở phía server cần viết code để nhận dữ liệu đoạn code đó và truyền dữ liệu chat đó đi đến các server khác. Đồng thời ở ở phía client cũng cần viết code để gửi và nhận dữ liệu từ server.

### **2.1.4 Giới thiệu về React**

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production,và [www.instagram.com](http://www.instagram.com/) được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM. Trươc khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

### **2.1.5 Giới thiệu về MongoDB**

* MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.
* MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
* Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là **collection** thay vì bảng
* So với RDBMS thì trong MongoDB **collection** ứng với **table**, còn **document** sẽ ứng với **row** , MongoDB sẽ dùng các document thay cho row trong RDBMS.
* Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.
* Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

## 

## 2.2 XÂY DỰNG ỨNG DỤNG

### **2.2.1 Xây dựng giao diện**

**Giao diện ứng dụng chat được viết bằng react một thư viện để viết frontend của javascript**

**Cấu trúc giao diện gồm các phần:**

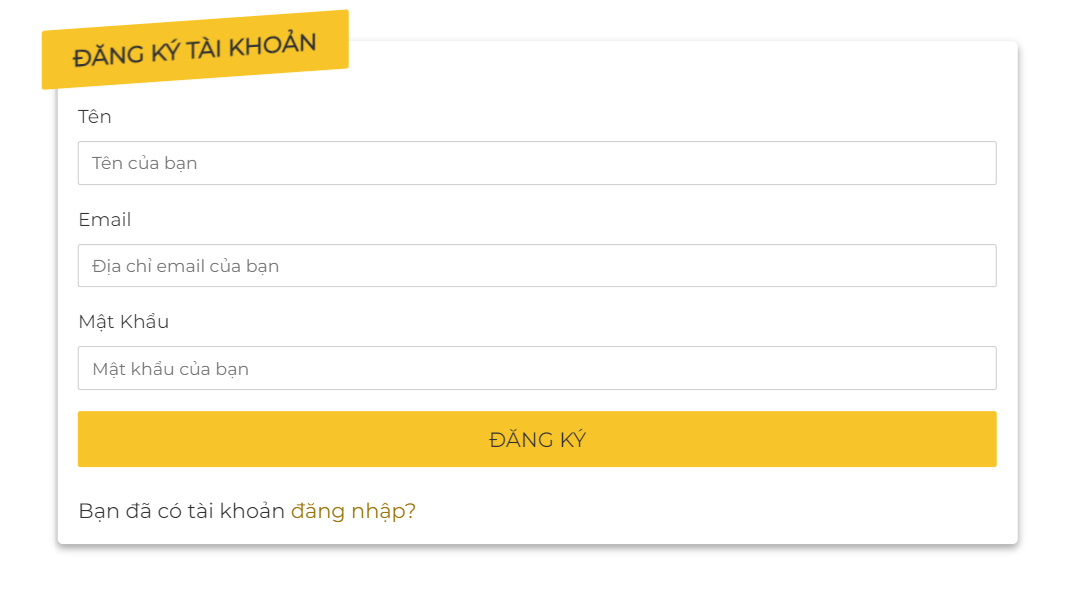
* **Đăng ký tài khoản**
* **Đăng nhập**
* **Xem tất cả các phòng và tạo phòng**
* **Phòng chat**

**+ Đăng ký tài khoản**

**Giao diện bao gồm:**

**Tên ứng dụng: Đăng ký**

**From đăng ký: Gồm 3 ô input để nhập tên, email, mật khẩu, một button để đăng ký tài khoản**

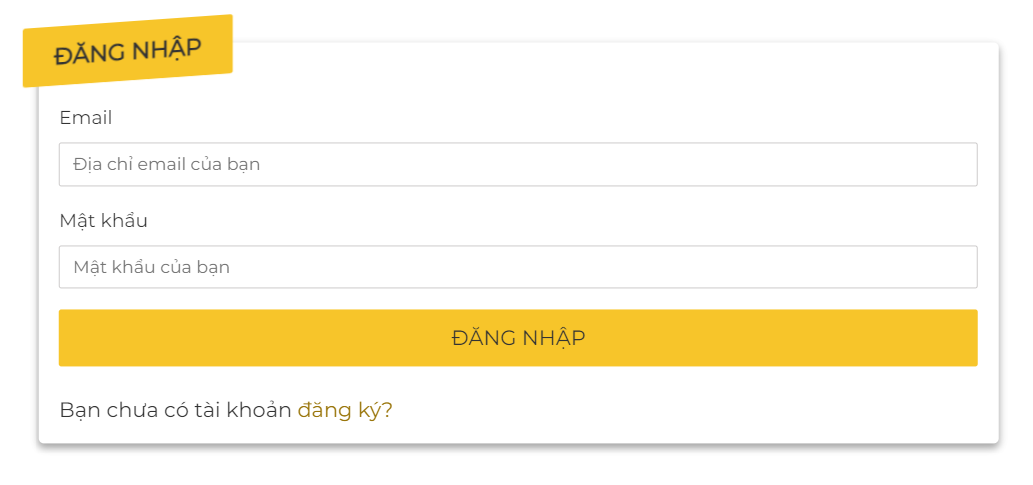
****

**+ Đăng nhập**

**Giao diện bao gồm:**

**Tên ứng dụng: Đăng nhập**

**From đăng nhập: Gồm 2 ô input để nhập tên và email, một button để đăng nhập**

****

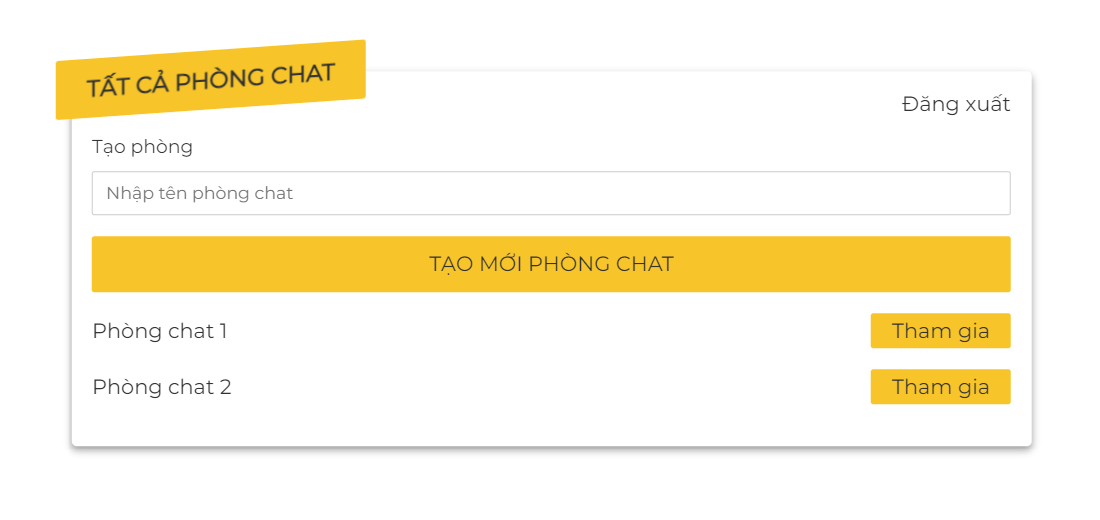
+ Dashboard

Giao diện bao gồm:

Tên ứng dụng: Dashboard

From gồm 1 ô input để tạo mới một phòng

Danh sách tất cả các phòng chat hiện có



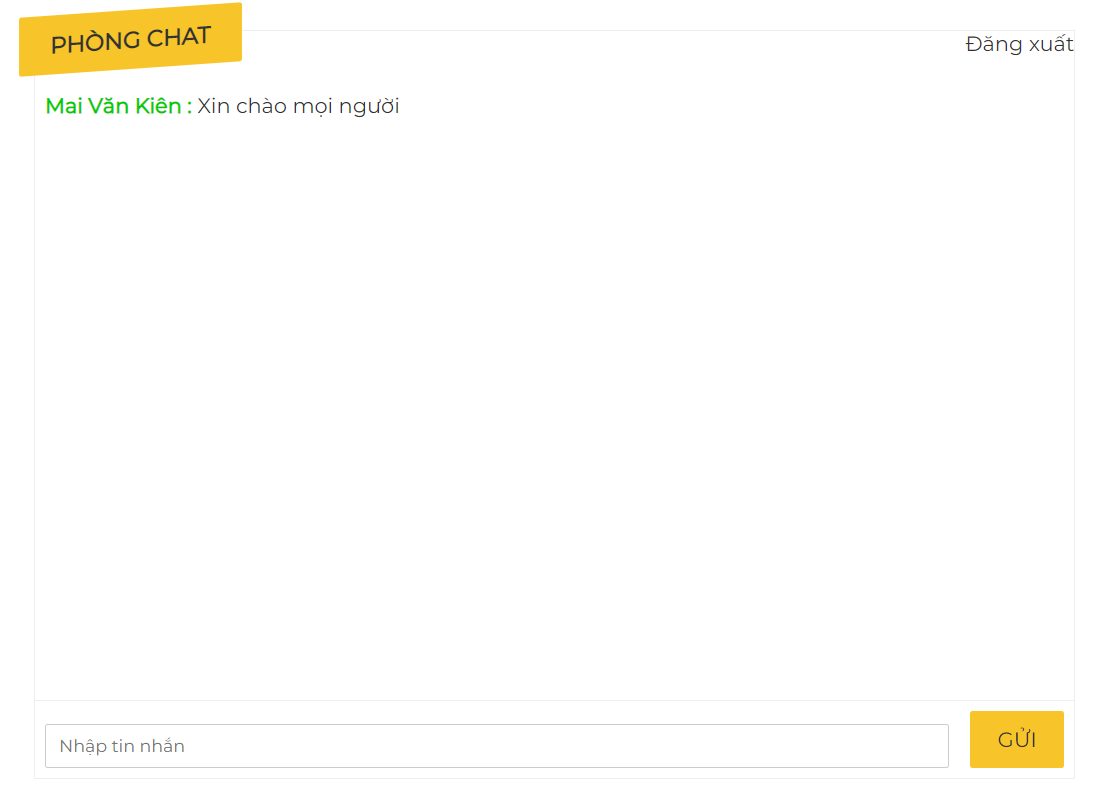
+ Phòng chat

Giao diện ứng dụng gồm:

Tên ứng dụng: Phòng chat

From gồm 1 ô input để nhập tin nhắn và một button để gửi tin nhắn

Danh sách các tin nhắn đã gửi



**2.2.2 Xây dựng code phía server bằng nodejs và socket.io**

Xây dựng Server

* Một server cơ bản được xây dựng dựa trên framework Express

const express = require(‘express’);

const app = express();

const server = require(‘http’).Server(app);

const io = require(‘socket.io’)(server);

server.listen(3000, () => {

console.log(“Server listening on port 3000”);

}

Kết nối với cơ sở dữ liệu MongoDB sử dụng thư viện mongoose

require(‘dotenv’).config();

const mongoose = require(‘mongoose’);

mongoose.connect(process.env.DATABASE, {

useUnifiedTopology: true,

useNewUrlParser: true,

});

mongoose.connection.on(‘err’, (err) => {

console.log(‘Error’, err.message);

})

mongoose.connection.once(‘open’, () => {

console.log(‘Success’);

})

Kết nối tới socket và lắng nghe các sự kiện

io.on(‘connection’, (socket) => {

socket.on(‘disconnected’, () => {

console.log(‘disconnected’, socket.userId);

})

socket.on(‘joinRoom’, ({chatroomId}) => {

socket.join(chatroomId);

})

socket.on(‘leaveRoom’, ({chatroomId}) => {

socket.leave(chatroomId);

})

socket.on(‘chatroomMessage’, async ({chatroomId, message}) => {

if(message.trim().length > 0) {

const user = await User.findOne({\_id: socket.userId});

const newMessage = new Message({

chatroom: chatroomId,

user: socket.userId,

message

});

io.to(chatroomId).emit(‘newMessage’, {

message,

name: user.name,

userId: scoket.userId

});

await newMessage.save()

}

})

})

### **2.2.3 Chạy dự án trên local**

Yêu cầu

* Cần cài đặt một số phần mềm, công cụ sau:

+ Visual studio code hoặc một số phần mềm soạn thảo code khác

+ Nodejs ( dùng để chạy code trên máy)

+ MongoDBCompass hoặc có thể dùng docker để chạy Database

+ Git (không bắt buộc)

Tiến hành chạy dự án

* Clone code trên github: git clone <https://github.com/maivankien/Chat-app>

nếu máy không cài git thì có thể tải file ZIP về giải nén

* Dùng Visual studio code mở thư mục code mới tải về
* Mở MongoDBCompass lên và chạy trên port tùy chỉnh
* Tạo 2 file .env lần lượt trong 2 thư mục server và client
* Cấu hình 2 file .env dựa theo file .env.example
* Mở 2 terminal lên tại thư mục gốc của source code
* Di chuyển thư mục của terminal thứ nhất vào thư mục server bằng lệnh : cd server
* Tiếp tục chạy lệnh : npm install và chờ cho các package được cài đặt
* Di chuyển thư mục của terminal thứ hai vào thư muc client bằng lệnh : cd client
* Tiếp tục chạy lệnh : npm install và chờ cho các package được cài đặt
* Quay lại thư mục thứ nhất chạy lệnh : npm start
* Tiếp tục chạy lệnh npm start ở terminal thứ hai
* Ứng dụng chạy thành công sẽ được truy cập ở địa chỉ: <http://localhost:3000>

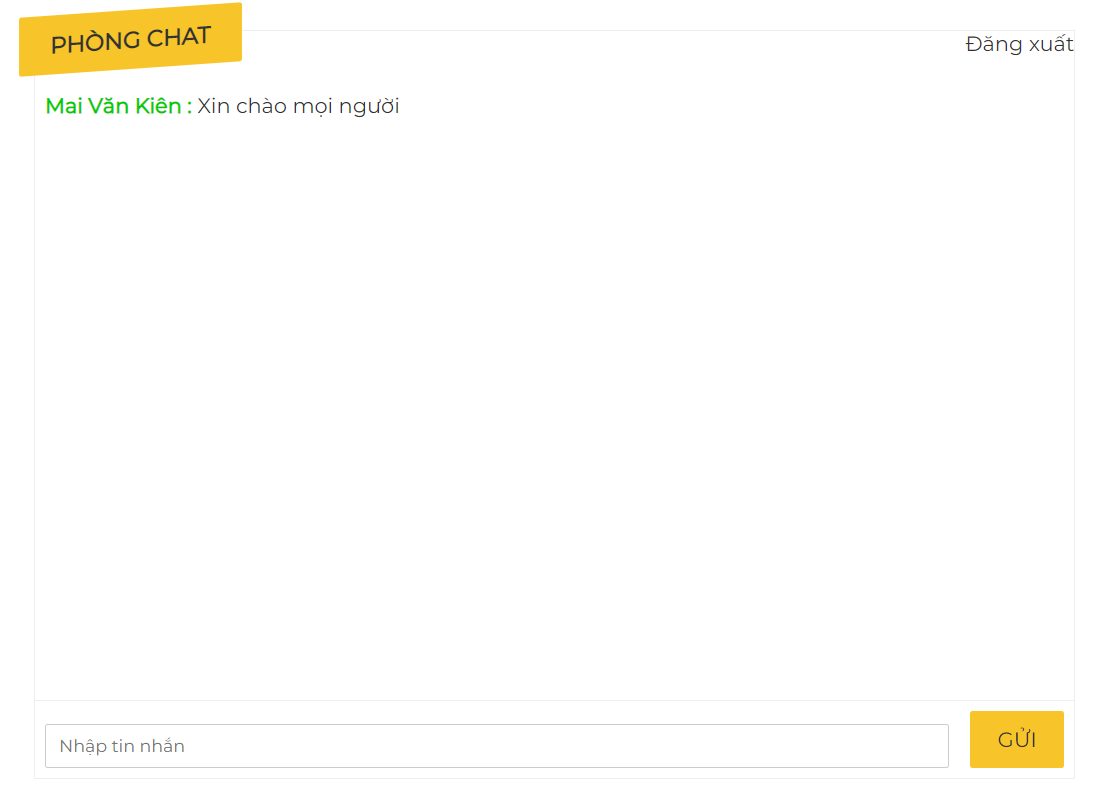
### **2.2.4 Kết quả chạy thực tế**

- Deploy ứng dụng server lên render

- Deploy database MongoDB lên MongoDB Cloud

- Deploy ứng dụng client lên vercel

- Trỏ ứng dụng lên tên miền: <http://www.maivankien.online>



### **2.2.5 Kết quả đạt được**

- Biết cách thiết kế giao diện web cơ bản sử dụng React

- Biết cách lập trình Nodejs sử dụng framework Express

- Biết cách tạo một server socket bằng thư viện socket.io

- Biết cách kết nối tới cơ sở dữ liệu MongoDB và đọc ghi dữ liệu

- Hiểu được cách hoạt động của Socket

- Biết cách xây dựng một ứng dụng chat realtime cơ bản

# **Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## 4.1 KẾT LUẬN

Sau khoảng thời gian khoảng 1 tháng nghiên cứu và tìm hiểu, chúng em đã hoàn thành bài tập lớn và chạy ứng dụng theo những yêu cầu đã đặt ra ban đầu. Trong quá trình thực hiện chúng em đã thu hoặc được những kết quả nhất định.

- Sản phẩm đạt yêu cầu cơ bản của một ứng dụng chat.

- Đăng ký đăng nhập tài khoản dễ dàng.

- Chat với nhiều phòng chat khác nhau.

## 4.2 HƯỚNG PHÁT TRIỂN

- Nâng cấp thêm một số tính năng như:

+ Kết bạn, chat với từng người, muốn vào phòng chat thì phải có mật khẩu phòng chat, hiện thị được các tin nhắn cũ trước đây, xóa tin nhắn, xóa phòng chat

**TÀI LIỆU THAM KHẢO**

**1. Sách tham khảo**

[1]. Mạng máy tính của TS. Nguyễn Tấn Hiệp – Trường đại học công nghệ thông tin, Đại học quốc gia TP. HCM.

[2]. Mạng máy tính của PGS.TS Trần Đình Hậu – Khoa Công nghệ thông tin và Truyền thông, Trường đại học bách khoa Hà Nội.

[3]. Mạng máy tính của PGS.TS Vũ Văn San – Khoa Công nghệ thông tin và truyền thông, Trường đại học bách khoa Hà Nội.

**2. Trang web tham khảo**

[1]. Reactjs.org

[2]. Socket.io

[3]. Nodejs.org

[4]. Mongodb.com