

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HCM**  
**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**  
**THIẾT KẾ VÀ XÂY DỰNG WEBSITE CHAT**  
**ONLINE**

**Giáo viên hướng dẫn: Hoàng Văn Dũng**

**Nhóm sinh viên thực hiện: Nhóm 2**

MSSV	Họ và tên	Mức độ đóng góp
20142498	Nguyễn Đình Hiếu	100%
20110505	Trần Tấn Khoa	100%
20161385	Huỳnh Bảo Toàn	100%

**TPHCM, tháng 12 năm 2022**

## LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, thầy Hoàng Văn Dũng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kỹ thuật và kinh nghiệm trong việc thực hiện một dự án web. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

## MỤC LỤC

MỤC LỤC HÌNH ẢNH .....	2
CHƯƠNG 1. MỞ ĐẦU .....	5
1.1. Phát biểu về bài toán: .....	5
1.2. Mục đích, yêu cầu thực hiện: .....	5
1.3. Đối tượng và phạm vi nghiên cứu: .....	6
1.3.1. Đối tượng nghiên cứu: .....	6
1.3.2. Phạm vi nghiên cứu: .....	6
1.4. Giao diện dự kiến: .....	6
1.5. Công cụ và môi trường: .....	6
CHƯƠNG 2. PHÂN CÔNG CÔNG VIỆC .....	7
CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ GIẢI PHÁP .....	8
3.1. Cơ sở lý thuyết: .....	8
3.1.1. HTML, CSS: .....	8
3.1.2. REACTJS: .....	9
3.1.3. Firebase: .....	10
3.2. Phân tích giao diện: .....	11
3.3. Phân tích cơ sở dữ liệu: .....	11
3.3.1. Tổng quan các bảng trong CSDL: .....	11
3.3.2. Mô tả các bảng: .....	12
3.3.3. Mô tả các Field trong bảng: .....	12
3.4. Cấu trúc: .....	14
3.5. Thiết kế giao diện: .....	27
CHƯƠNG 4. CÀI ĐẶT VÀ KIỂM THỬ .....	30
4.1. Thiết kế giao diện: .....	30
4.1.1. Trang chat: .....	30
4.1.2. Đăng nhập, đăng xuất: .....	33
4.1.3. Đăng ký: .....	35
4.1.4. Tìm kiếm bạn bè: .....	36
4.1.5. Thêm bạn bè: .....	37
4.1.6. Thêm, xóa group: .....	38
4.1.7. Thông tin người dùng: .....	40
4.2. Hướng dẫn chạy: .....	41
CHƯƠNG 5. KẾT LUẬN .....	43
5.1. Kết quả đạt được: .....	43
5.2. Định hướng phát triển: .....	43
5.3. Ưu điểm, hạn chế: .....	43
5.3.1. Ưu điểm: .....	44
5.3.2. Hạn chế: .....	44
TÀI LIỆU THAM KHẢO .....	45

## MỤC LỤC HÌNH ẢNH

Hình 1: Database của Webchat .....	12
Hình 2: Trang đăng nhập .....	27
Hình 3: Trang home .....	27
Hình 4: Trang đăng kí tài khoản .....	28
Hình 5: Trang chat .....	28
Hình 6: Trang danh sách bạn bè .....	29
Hình 7: Trang danh sách nhóm .....	29
Hình 8: Trang chat sau khi load toàn bộ danh sách tin nhắn và list tin nhắn .....	30
Hình 9: Form mời bạn vào group .....	31
Hình 10: Thông báo lỗi khi mời thành viên đã có trong nhóm .....	31
Hình 11: Tìm kiếm bạn bè để mời chat .....	32
Hình 12: Form đổi tên phòng và mô tả .....	33
Hình 13: Login thông qua google .....	34
Hình 14: Thẻ hiện hồ sơ, cài đặt, đăng xuất .....	34
Hình 15: Form đăng nhập vào GITCHAT .....	35
Hình 16: Đăng ký tài khoản .....	36
Hình 17: Danh sách bạn bè và tìm kiếm thành viên để mời vào nhóm .....	37
Hình 18: Form tìm kiếm bạn bè theo số điện thoại .....	38
Hình 19: Thẻ hiện lời mời kết bạn .....	39

Hình 20: Thêm phòng và hiển thị danh sách phòng .....	40
Hình 21: Form thông báo xóa phòng .....	40
Hình 22: Form thể hiện thông tin cá nhân .....	41

## **MỤC LỤC BẢNG BIỂU**

Bảng 1: Bảng phân công nhiệm vụ .....	7
Bảng 2: Bảng mô tả các trang .....	11
Bảng 3: Mô tả các bảng sử dụng trong GitChat .....	12
Bảng 4: Mô tả các Field trong bảng user .....	13
Bảng 5: Mô tả các Field trong bảng addfriend .....	13
Bảng 6: Mô tả các Field trong bảng rooms .....	14
Bảng 7: Mô tả các Field trong bảng messages .....	14
Bảng 8: Mô tả các Field trong bảng friend .....	14
Bảng 9: Bảng cấu trúc của hệ thống GitChat .....	27
Bảng 10: Bảng thiết kế giao diện .....	29

# CHƯƠNG 1. MỞ ĐẦU

## 1.1. Phát biểu về bài toán:

Trong thời đại 4.0 ngày nay, chúng ta đang chứng kiến sự phát triển vượt bậc của internet trên toàn thế giới. Cùng với sự phát triển đó biến chúng thành một chiếc chìa khóa giúp chúng ta giải quyết được nhiều nhu cầu trong cuộc sống. Một trong những nhu cầu đó là trao đổi thông tin và liên lạc một cách nhanh chóng tiện lợi. Điều này đã thúc đẩy sự phát triển của rất nhiều phần mềm hỗ trợ cho việc đó. Điển hình cho các phần mềm này là các web chat, app chat chẳng hạn như facebook, instagram, zalo,... Nhưng có một vấn đề đặt ra là các ứng dụng trên phải cài đặt trên các thiết bị của chúng ta trong khi các tính năng của các ứng dụng này đều có thể làm được trên web. Web đem lại cho chúng ta sự tiện lợi, chỉ với một ứng dụng duy nhất có kết nối mạng chỉ cần nhập đường link của các trang web ta có thể truy cập và sử dụng những tính năng mà trang web đó đem lại. Điều đó mang lại sự thuận tiện cho người dùng và đồng thời là sự tiết kiệm tài nguyên máy của ta một cách đáng kể. Qua những vấn đề trên nên nhóm em quyết định thiết kế và xây dựng một website cho phép chúng ta nhắn tin trực tiếp trên đó với đề tài “Xây dựng và thiết kế website chat online”. Với mục tiêu đề ra website sẽ có những phân tương tự như những ứng dụng đã đề ra ở trên, cùng với đó là áp dụng những công nghệ mới giúp mục đích hoàn thiện hơn các chức năng và giao diện của web chat online.

## 1.2. Mục đích, yêu cầu thực hiện:

Đề tài **Thiết kế và xây dựng website chat online** đặt ra các vấn đề trọng tâm về mục tiêu mà sinh viên nghiên cứu cần thực hiện được bao gồm:

Vấn đề đầu tiên được đưa ra về yếu tố lý thuyết, sinh viên áp dụng những kiến thức, nghiên cứu hiểu được tổng quan và khả năng ứng dụng vào trong bài làm. Song

song với việc áp dụng công nghệ được lựa chọn để sử dụng, nhóm sinh viên cần tìm hiểu thêm về cách xây dựng giao diện đẹp, thân thiện với người dùng.

Vấn đề thứ hai là nâng cao khả năng làm việc nhóm, tìm hiểu và thảo luận về đề tài, các vấn đề đưa ra từ đó tìm ra câu trả lời. Xây dựng thái tích cực trong việc học, làm đồ án.

### **1.3. Đối tượng và phạm vi nghiên cứu:**

#### **1.3.1. Đối tượng nghiên cứu:**

Xây dựng trang web chat bằng ReactJS. Kết nối với cơ sở dữ liệu bằng FireBase.

#### **1.3.2. Phạm vi nghiên cứu:**

Thiết kế và xây dựng website chat online.

### **1.4. Giao diện dự kiến:**

Để thích hợp với một ứng dụng chat thì nhóm dự kiến sẽ xây dựng các trang bao gồm trang chủ là nơi giới thiệu thông tin website, trang login để đăng nhập, trang đăng ký để tạo tài khoản, trang chat để thực hiện các chức năng liên quan đến chat. Trong trang chat này gồm 3 thành phần: chat thể hiện các đoạn chat, friends thể hiện danh sách bạn bè và lời mời kết bạn, groups thể hiện danh sách group tồn tại có thể thêm và xóa.

### **1.5. Công cụ và môi trường:**

- IDE: Visual Studio Code.
- Ngôn ngữ: REACTJS, HTML, CSS cho font-end, firebase cho backend.

## CHƯƠNG 2. PHÂN CÔNG CÔNG VIỆC

STT	Danh sách thành viên	Công việc thực hiện được	Tỉ lệ hoàn thành
1	Nguyễn Đình Hiếu	Xây dựng giao diện, viết báo cáo, xây dựng các chức năng trang chat, đăng nhập, đăng ký, kết nối firebase.	100%
2	Trần Tấn Khoa	Xây dựng giao diện, xây dựng các chức năng trang friend, group, kết nối firebase, viết báo cáo.	100%
3	Huỳnh Bảo Toàn	Xây dựng giao diện, xây dựng các chức năng trang friend, group, kết nối firebase, viết báo cáo.	100%

Bảng 1: Bảng phân công nhiệm vụ



## CHƯƠNG 3. PHÂN TÍCH, THIẾT KẾ GIẢI PHÁP

### 3.1. Cơ sở lý thuyết:

#### 3.1.1. HTML, CSS:

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu tiêu chuẩn để tạo các trang Web. Mô tả cấu trúc của một trang web, bao gồm một loạt các phần tử và các phần tử HTML cho trình duyệt biết cách hiển thị nội dung. CSS (Cascading Style Sheets) mô tả cách các phần tử HTML hiển thị trên màn hình laptop, điện thoại,... HTML và CSS độc lập với nhau, CSS không nên viết bên trong HTML và ngược lại. HTML đại diện cho nội dung và CSS thể hiện sự xuất hiện của nội dung đó.

✓ Ưu điểm:

- + Được sử dụng rộng rãi với rất nhiều nguồn tài nguyên mở và hoàn toàn miễn phí. Được hỗ trợ cùng một cộng đồng sử dụng vô cùng lớn đằng sau nó.
- + Hoạt động được trên hầu hết mọi trình duyệt hiện nay.
- + Tiếp cận và học HTML, CSS khá đơn giản.
- + Chuẩn của web được vận hành bởi World Wide Web Consortium (W3C).
- + Dễ dàng tích hợp với các ngôn ngữ backend như Node.js, java,...

✓ Nhược điểm:

- + Ngôn ngữ này chỉ được áp dụng chủ yếu cho trang web còn đối với các tính năng động của trang web thì cần phải kết hợp JavaScript. Bên cạnh đó, muốn hiện những dữ liệu trong cơ sở dữ liệu lên màn hình thì cần phải kết hợp với backend.
- + Phải tạo các trang web riêng lẻ ngay cả khi các phần tử trong trang web đó giống nhau.
- + Phải có khả năng thiết kế tốt để trang web trở nên đẹp hơn.

### 3.1.2. REACTJS:

REACTJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện hiệu website thì chúng cũng đem lại những ưu, nhược điểm khác nhau cho người sử dụng. Sau đây ta tìm hiểu về một vài ưu nhược của chúng.

✓ Ưu điểm:

- + Hỗ trợ chúng ta rất nhiều trong việc chia cấu trúc file để có thể code dễ dàng hơn. Cung cấp rất nhiều loại hook thay thế cho JavaScript một cách tiện lợi nhanh chóng.

- + Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì ta không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML.

- +Tái sử dụng các Component: nếu xây dựng một component tốt, ta có thể sử dụng lại chúng vào nhiều trang web hoặc layout khác nhau thay cho việc làm lại từ đầu.

- + Ngoài ra còn được sử dụng cho cả Mobile application.

- + Debug nhanh chóng và dễ dàng: Facebook đã phát hành 1 Chrome extension hỗ trợ việc debug khi gặp lỗi. Phần nào giúp ta tăng hiệu suất khi xây dựng trang web.

✓ Nhược điểm:

- + Tích hợp Reactjs vào các framework MVC truyền thống nên yêu cầu phải cấu hình lại toàn bộ file.

- + React khá nặng so với một vài framework khác (Khoảng 35kb so với 39kb của Angular).

- + Khó tiếp cận cho người mới học Web.

### **3.1.3. Firebase:**

Firebase là Backend-as-a-Service (Baas) nền tảng chứa các API đơn giản và mạnh mẽ giúp phát triển trang web và ứng dụng di động mà không cần Server hay Backend. Firebase hay Google firebase được phát triển bởi Google chức năng như một dịch vụ cơ sở dữ liệu có hệ thống máy chủ mạnh mẽ, chạy trên nền Cloud – Điện toán đám mây.

Cũng giống như ReactJS, HTML, CSS thì firebase cũng có những ưu và nhược điểm riêng của chúng.

#### **✓ Ưu điểm:**

- + Cho phép chúng ta tạo tài khoản sử dụng một cách dễ dàng.
- + Cung cấp nhiều các xác thực khác nhau khi login vào một trang web như google, facebook, số điện thoại,...
- + Có Firestore database cho phép chúng ta lưu trữ dữ liệu một cách dễ dàng nhanh chóng thay thế cho phía backend rất hữu ích.
- + Cho phép chúng ta sử dụng miễn phí.
- + Có docs hướng dẫn sử dụng một cách chi tiết dễ dàng.

#### **✓ Nhược điểm:**

- + Tuy cho phép chúng ta sử dụng miễn phí nhưng giới hạn sử dụng dữ liệu chẳng hạn như cho phép chúng ta lưu trữ tối đa 1 GiB, cho phép đọc 50k dữ liệu mỗi ngày, ghi 20k dữ liệu mỗi ngày, 20k delete mỗi ngày và muốn nhiều hơn thì phải trả thêm phí. Tuy nhiên chi phí không quá mắc.
- + Khó tiếp cận đối với những người mới học web.

### **3.2. Phân tích giao diện:**

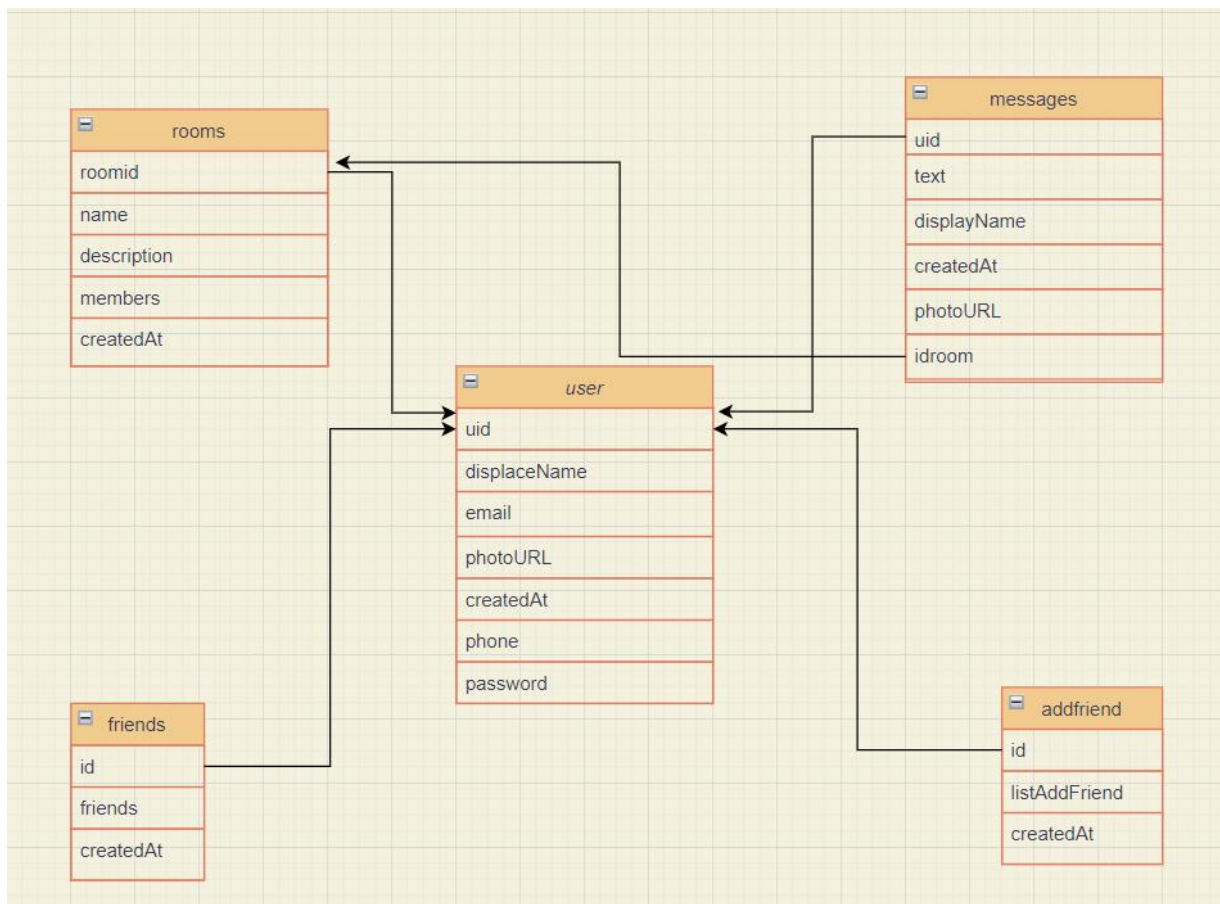
STT	Giao diện	Chức năng
1	Trang Login	Nơi người dùng đăng nhập bằng tài khoản của mình thông qua tài khoản đã đăng ký, thông qua facebook hoặc gmail
2	Trang đăng kí tài khoản	Nơi người dùng có thể đăng kí tài khoản bằng số điện thoại hoặc có thể đăng nhập bằng gmail hoặc facebook
3	Trang chat	Nơi hiển thị danh sách tin nhắn và chi tiết của tin nhắn được chọn, cho phép kết bạn, tạo các cuộc trò chuyện với bạn bè hoặc các hội nhóm. Bên cạnh đó còn có thể phát triển một số tính năng khác.
4	Trang Home	Giới thiệu về trang web, cho phép điều hướng sang trang đăng nhập và đăng ký.

Bảng 2: Bảng mô tả các trang

### 3.3. Phân tích cơ sở dữ liệu:

#### 3.3.1. Tổng quan các bảng trong CSDL:

Trang web có cơ sở dữ liệu đơn giản để lưu trữ thông tin các user, các room, các message và list friend của user đó. Sau đây chúng ta đi tìm hiểu kỹ hơn về từng bảng.



Hình 1: Database của Webchat

### 3.3.2. Mô tả các bảng:

STT	Tên bảng	Mục đích
1	User	Thông tin về tài khoản đăng nhập
2	Rooms	Thông tin về phòng chat
3	Message	Thông tin về tin nhắn của User
4	addfriend	Chứa thông tin về danh sách bạn bè gửi lời mời kết bạn
5	friend	Chứa thông tin về danh sách bạn bè

Bảng 3: Mô tả các bảng sử dụng trong GitChat

### 3.3.3. Mô tả các Field trong bảng:

STT	Tên trường	Kiểu dữ liệu	Mục đích
1	uid	string	Mã số người dùng
2	displayName	string	Tên hiển thị của người dùng
3	email	string	Địa chỉ Email người dùng
4	photoURL	string	Link ảnh của người dùng
5	createdAt	timestamp	Ngày tạo người dùng
6	phone	string	Số điện thoại tạo tài khoản
7	password	string	Mật khẩu để đăng nhập

Bảng 4: Mô tả các Field trong bảng user

TT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	string	Id user
2	listAddFriend	array	Id của các friend gửi lời mời kết bạn
3	createdAt	timestamp	Thời gian tạo bảng

Bảng 5: Mô tả các Field trong bảng addfriend

STT	Tên trường	Kiểu dữ liệu	Mục đích
1	name	string	Tên phòng

2	description	string	Mô tả phòng
3	members	array	Chứa Id của user người chat
4	createdAt	timestamp	Ngày, thời gian tạo phòng
5	roomId	string	Id của phòng chat

Bảng 6: Mô tả các Field trong bảng rooms

STT	Tên trường	Kiểu dữ liệu	Mục đích
1	uid	string	ID sở hữu đoạn chat
2	text	string	Nội dung chat
3	displayName	string	Tên hiển thị
4	createdAt	timestamp	Ngày, thời gian tạo
5	photoURL	string	Ảnh đại diện của user
6	idroom	string	Id phòng hiển thị đoạn chat

Bảng 7: Mô tả các Field trong bảng messages

TT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	string	Id user
2	friends	array	Id các friend
3	createdAt	timestamp	Thời gian tạo bảng

Bảng 8: Mô tả các Field trong bảng friend

### 3.4. Cấu trúc:

PAGES	
Chat	<pre>Chat()  const [tab,setTab] = useState(0)  const chat = useRef()  const friend = useRef()  const group = useRef()  const setting = useRef()  const componentSetting = useRef()  Export default Chat  tonggle()  =&gt; render ra giao diện cho trang chat</pre>
Home	<pre>Home()  =&gt; Xây dựng trang home  export default Home</pre>
Login	<pre>const handleLogin = async (provider) =&gt; {      const { additionalUserInfo, user } = await     auth.signInWithPopup(provider);      if (additionalUserInfo?.isNewUser) {          addDocument('users', {              displayName: user.displayName,              email: user.email,              photoURL: user.photoURL,              uid: user.uid,              providerId: additionalUserInfo.providerId,</pre>



```

});

addDocument('friend',{
  id: user.uid,
  friends: [],
});

addDocument('addfriend',{
  id: user.uid,
  listaddfriend: [],
});
}
};

```

=> login thông qua facebook và google, nếu login thành công và là người mới thì thêm vào trong cơ sở dữ liệu.

```

const handleBtnLogin = () => {
  if(phoneInput === undefined || passwordInput === undefined)
  {
    alert("Hãy nhập đầy đủ số điện thoại và mật khẩu!!!")
  }
  else{
    let loginUser = db.collection('users');
    loginUser.where('phone', '==', phoneInput);
    loginUser = loginUser.where('password', '==', passwordInput);
    getDocs(loginUser)
      .then((snapshot) => {

```

	<pre> const {displayName,photoURL,uid, phone}=snapshot.docs[0].data()  const userLogin=[displayName,photoURL,uid, phone]  const jsonUser=JSON.stringify(userLogin)  localStorage.setItem("user", jsonUser);  setUser(snapshot.docs[0]?.data())  })  .catch((error) =&gt; {      alert("Số điện thoại hoặc mật khẩu không chính xác!!!")      console.log(error)  });  }  }; </pre> <p>=&gt; Login thông qua số điện thoại: kiểm tra nếu trùng số điện thoại và mật khẩu thì cho vào còn các trường hợp khác in ra thông báo.</p>
Signup	<pre> const onClose = () =&gt; {      setPreview(null);  };  const onCrop = (view) =&gt; {      setPreview(view);  };  const createRandom = () =&gt; {      var randomstring = "";      var characters = 'QWERTYUIOPASDFGHJKLZXCVBNM123456789qwertyuiopasdfghj klzxcvbnm'; </pre>

	<pre> for (var i, i = 0; i &lt; 28; i++) {     randomstring += characters.charAt(Math.floor(Math.random() * 28)); }  return randomstring;  }; =&gt; Tạo ra chuỗi random  const handlesubmit = () =&gt; {  if((document.getElementsByClassName(`\${cx('name')}`)[0].value===''    document.getElementsByClassName(`\${cx('phone')}`)[0].value===''    document.getElementsByClassName(`\${cx('pass')}`)[0].value===''    document.getElementsByClassName(`\${cx('repass')}`)[0].value===''))  {     alert("Hãy điền đầy đủ các trường") }  else  {     if (document.getElementsByClassName(`\${cx('pass')}`)[0].value === document.getElementsByClassName(`\${cx('repass')}`)[0].value ) {         const uid=createRandom();         addDocument('users', {             displayName: document.getElementsByClassName(`\${cx('name')}`)[0].value,             phone: document.getElementsByClassName(`\${cx('phone')}`)[0].value,             photoURL: preview, </pre>
--	---

```

        uid: uid,
        providerId: 'phone',
        password:
document.getElementsByClassName(`${cx('pass')}`)[0].value,
    });
    addDocument('friend',{
        id: uid,
        friends: [],
    });
    addDocument('addfriend',{
        id: uid,
        listaddfriend: [],
    });
    alert('Đăng ký thành công');
    navigate("/login");
} else {
    alert('Mật khẩu không trùng khớp');
}
}
};

```

=> Kiểm tra nếu không điền đủ các ô và không trùng pass thì in ra thông báo còn nếu đăng ký thành công thì chuyển sang trang login đồng thời thêm thông tin vào trong cơ sở dữ liệu.

```
//
```

```

const handleLogin = async (provider) => {
    const { additionalUserInfo, user } = await

```

	<pre> auth.signInWithPopup(provider);  if (additionalUserInfo?.isNewUser) {   addDocument('users', {     displayName: user.displayName,     email: user.email,     photoURL: user.photoURL,     uid: user.uid,     providerId: additionalUserInfo.providerId,   });   addDocument('friend',{     id: user.uid,     friends: [],   });   addDocument('addfriend',{     id: user.uid,     listaddfriend: [],   }); }  }; </pre> <p>=&gt; xử lý cho việc login thông qua google và facebook khi đứng tại trang signup.</p>
<b>LAYOUT COMPONENTS</b>	
Button	Button() Export default Button
Service	- Xử lý việc add thành viên nhóm export const addGroup = (user,idgroup) =>

	<pre> { let collectionUser = db.collection('rooms'); getDocs(collectionUser) .then(res =&gt; { const movss = res.docs.find(doc =&gt; doc.id === idgroup); const dataa = movss.data().members;   if(dataa.includes(user)) {     alert("User đã có trong nhóm")   }    else   { const dataaa = [...dataa,user]; db.collection('rooms').doc(movss.id).update({   createdAt: movss.data().createdAt,   description: movss.data().description,   members: dataaa,   name: movss.data().name, })   )   alert('Thêm thành viên thành công')   } }).catch(err =&gt; err) } </pre>
Friends	<p>Friends()</p> <p>Export default Friends</p> <ul style="list-style-type: none"> <li>- Hàm chấp nhận kết bạn</li> </ul> <pre> const acpfriend = (useraddfr) =&gt; {   updateFriend(uid,useraddfr.uid) } </pre> <ul style="list-style-type: none"> <li>- Hàm xóa bạn bè</li> </ul> <pre> const deleteaddfriend = (useraddfr) =&gt; {   updateListAddFriend(uid, useraddfr.uid); } </pre>

	<p>- Khi nhấn vào tin nhắn với bạn bè thì hiển thị những chức năng giống với messchat. Tuy nhiên sẽ xóa đi một số tính năng chẳng hạn như thêm bạn bè vào cuộc trò chuyện,...</p>
Group	<p>Group()</p> <p>Export default Group</p> <ul style="list-style-type: none"> <li>- Hàm hiển thị Form thêm Group, tắt hiển thị các nhóm</li> </ul> <p>showModal()</p> <ul style="list-style-type: none"> <li>- Xử lý thêm nhóm mới vào Database</li> </ul> <p>handleOk()</p> <ul style="list-style-type: none"> <li>- Hàm hủy việc thêm nhóm</li> </ul> <p>handleCancel()</p> <ul style="list-style-type: none"> <li>- Hàm hiển thị Form xóa Group</li> </ul> <p>DelRoom()</p> <ul style="list-style-type: none"> <li>- Hàm hủy hiển thị việc xóa Group</li> </ul> <p>handlePerformCancel()</p> <ul style="list-style-type: none"> <li>- Hàm xử lý việc xóa Group ra khỏi Database</li> </ul> <p>handlePerformOK()</p> <ul style="list-style-type: none"> <li>- Form xóa phòng</li> </ul> <p>&lt;Form form={form} layout="vertical" className="perform-del" style={{ display: 'none' }}&gt;&lt;/Form&gt;</p> <ul style="list-style-type: none"> <li>- Form thêm phòng</li> </ul> <p>&lt;Form form={form} layout="vertical" className="test" style={{ display: 'none' }}&gt;&lt;/Form&gt;</p>

Header	Header () Export default Header
HeaderChat	<pre> const choose = useRef();  const handleInfor = () =&gt; {   if (choose.current.style.display === 'inline-block') {     choose.current.style.display = 'none';   } else {     choose.current.style.display = 'inline-block';   } };  const {   user: { displayName, photoURL }, } = useContext(AuthContext);  const handlelogout=()=&gt;{   auth.signOut(); }  HeaderChat () Export default HeaderChat </pre>
messChat	<ul style="list-style-type: none"> <li>- Lấy ra thông tin của Room</li> </ul> <pre> Const selectedRoom = React.useMemo(() =&gt; {   return rooms.find((room) =&gt; {     if (room.id === roomid) {       return room;     }   }); }, [rooms, roomid]); </pre> <ul style="list-style-type: none"> <li>- Hàm xử lý gửi tin nhắn và lưu thông tin tin nhắn lên Database</li> </ul> <pre> handlesubmit() </pre>



	<ul style="list-style-type: none"> <li>- Hàm xử lý để hiện ra Form mời thành viên showInvite()</li> <li>- Hàm hủy thao tác Rename handleCancelRename()</li> <li>- Hàm Update tên Group  handleCancelOK()</li> <li>- Hàm xử lý việc Update tên mô tả handleOkDes ()</li> <li>- Hàm hủy thao tác Rename mô tả  handleCancelDes()</li> <li>- Hiện thị Avatar các thành viên đang có trong nhóm</li> </ul> <pre> &lt;Avatar.Group size="small" maxCount={2}&gt;   { members.map((member, index) =&gt; (   &lt; StatusAvatar     status="green"     size={30}     name={member.displayName}     githubHandle={member.photoURL}   /&gt; ))} &lt;/Avatar.Group&gt; const handlesubmit = () =&gt; {   if(chat!=="")   {     addDocument('messages', {       text: chat,       uid,       photoURL,       displayName,       idroom: roomid,     });   } } </pre>
--	--

```

        document.getElementsByClassName(`${cx('input-
chat')}`)[0].value = "";
        setChat("")
    }
};
const handleKeyDown = (event) => {
    if (event.key === 'Enter') {
        if(chat!=="")
        {
            addDocument('messages', {
                text: chat,
                uid,
                photoURL,
                displayName,
                idroom: roomid,
            });
            document.getElementsByClassName(`${cx('input-
chat')}`)[0].value = "";
            setChat("")
        }
    }
}
=> Kiểm tra nếu như nhập rỗng thì không cho gửi, ngược lại gửi thành
công.

let listchat = db.collection('messages');

//listchat = listchat.orderBy('createdAt', 'asc');
listchat = listchat.where('idroom', '==', roomid);
//
react.useEffect(() => {
    onSnapshot(listchat, (snapshot) => {
        const chat = snapshot.docs.map((doc) => ({
            ...doc.data(),
            id: doc.id,

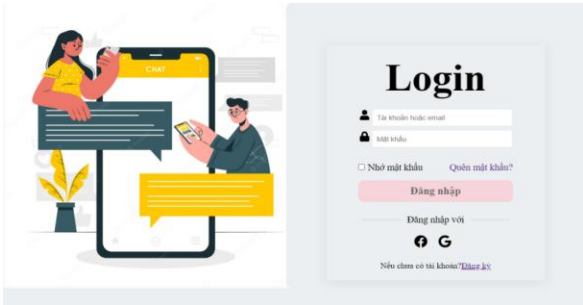

```

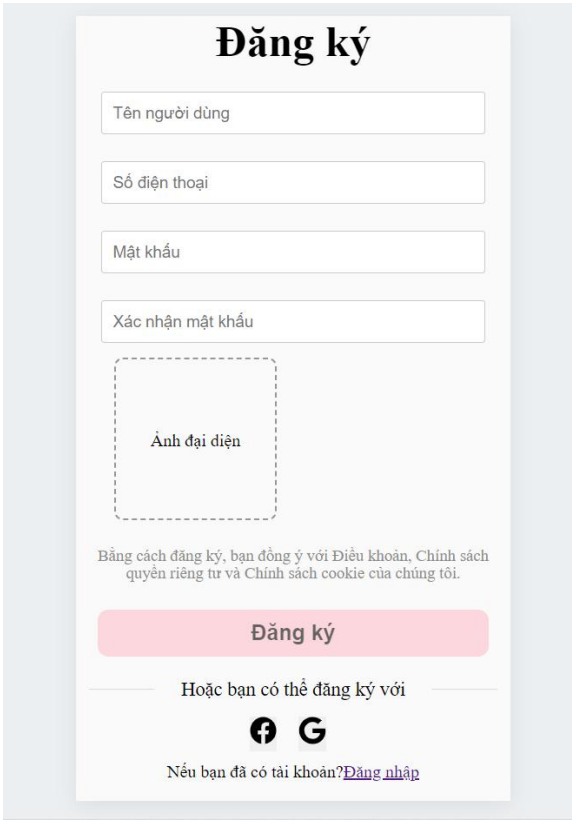

	<pre>         ));         setDocuments(chat);     });     }, [roomId]);     documents.sort((a,b)=&gt;a.createdAt-b.createdAt) =&gt; Lấy toàn bộ tin nhắn và hiển thị ra màn hình. </pre>
ModalInviterMembers	<ul style="list-style-type: none"> <li>- Xử lý thao tác mời thành viên handleInviteMember()</li> <li>- Tắt hiển thị Form mời thành viên handleOpenInviteModal()</li> </ul>
settingChat	<ul style="list-style-type: none"> <li>- Hiển thị thông tin phòng đang được chọn</li> </ul> <pre> const selectedRoom = react.useMemo(() =&gt; {     return rooms.find((room) =&gt; {         if (room.id === roomId) {             return room;         }     }); }, [rooms, roomId]); settingChat () Export default settingChat </pre>
ShowChat	<pre> const {     user: { photoURL, uid }, } = react.useContext(AuthContext); const roomsCondition = react.useMemo(() =&gt; {     return { </pre>

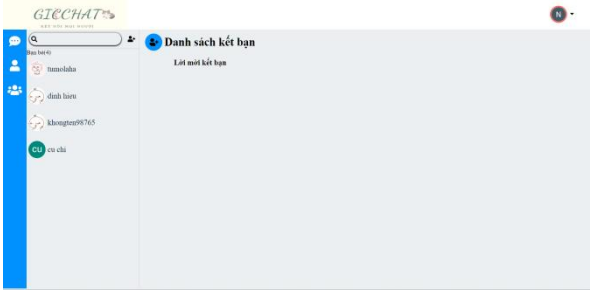
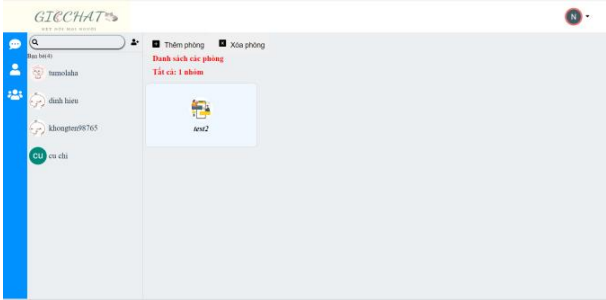
	<pre> fieldName: 'members', operator: 'array-contains', compareValue: uid, }; }, [uid]); ShowChat () Export default ShowChat </pre>
--	---

Bảng 9: Bảng cấu trúc của hệ thống GitChat

### 3.5. Thiết kế giao diện:

STT	Trang	Giải thích thiết kế
1	<p>Hình 2: Trang đăng nhập</p> 	<ul style="list-style-type: none"> <li>- 2 textbox để nhập tên đăng nhập và mật khẩu</li> <li>- Check box để yêu cầu ghi nhớ mật khẩu.</li> <li>- Button Đăng nhập để gọi Controller Login.</li> <li>- 2 button dùng để đăng nhập bằng gmail và facebook.</li> </ul>
2	<p>Hình 3: Trang home</p> 	<p>Trang home thực hiện giới thiệu về trang web và điều hướng đến trang đăng nhập cũng như đăng kí tài khoản.</p>

3	<p>Hình 4: Trang đăng kí tài khoản</p> 	<ul style="list-style-type: none"> <li>- 4 textbox để nhập tên người dùng, số điện thoại, mật khẩu, xác nhận mật khẩu</li> <li>- Button Đăng ký để gọi Controller đăng ký.</li> <li>- 2 button dùng để đăng ký bằng gmail và facebook.</li> <li>- Thẻ đăng nhập để quay lại trang đăng nhập</li> </ul>
4	<p>Hình 5: Trang chat</p> 	<ul style="list-style-type: none"> <li>- Thẻ hiển thị thông tin tài khoản đang đăng nhập</li> <li>- Ta có thể tìm kiếm chức năng thông qua icon kính lúp</li> <li>- Chứa các icon như danh sách bạn bè, hội nhóm.</li> <li>- Thẻ hiển thị nội dung đoạn chat, các user trong room.</li> <li>- Các chức năng như tìm kiếm tin nhắn, được phép sửa đổi tên phòng và description của phòng, cho phép mời thêm</li> </ul>

		friend vào room.
5	<p>Hình 6: Trang danh sách bạn bè</p> 	<ul style="list-style-type: none"> <li>- Thể hiện danh sách lời mời kết bạn chứa các button xóa, đồng ý.</li> <li>- Thể hiện danh sách bạn bè.</li> <li>- Cho phép tìm kiếm bạn bè có trong danh sách và hiển thị tin nhắn đối với bạn bè đó.</li> </ul>
6	<p>Hình 7: Trang danh sách nhóm</p> 	<ul style="list-style-type: none"> <li>- Trang danh sách hội nhóm đã tham gia</li> <li>- Hiển thị danh sách bạn bè và tìm kiếm bạn bè.</li> <li>- Thêm xóa các phòng.</li> </ul>

Bảng 10: Bảng thiết kế giao diện

## CHƯƠNG 4. CÀI ĐẶT VÀ KIỂM THỬ

### 4.1. Thiết kế giao diện:

Ngoài việc thiết kế hoàn chỉnh giao diện thì nhóm em đã hoàn thành một số tính năng sau. Ứng với mỗi tính năng sẽ có những trường hợp kiểm tra thích hợp và những trường hợp thường là những trường hợp bị lỗi nhất mà nhóm đã phát hiện và sửa lỗi để có đầu vào phù hợp.

#### 4.1.1. Trang chat:



Hình 8: Trang chat sau khi load toàn bộ danh sách tin nhắn và list tin nhắn

Khi nhập nội dung tin nhắn và nhấn nút gửi hoặc enter thì tin nhắn sẽ hiển thị lên giao diện người dùng theo phía người nhận người gửi. Trong trường hợp tin nhắn được gửi là rỗng thì sẽ không gửi được tin nhắn.

Cho phép thêm người vào cuộc hội thoại bằng cách nhấn vào nút mời. Khi nhấn vào nút mời sẽ hiển thị lên một form cho phép ta nhập email hoặc số điện

thoại đại diện cho user đó. Khi nhấn vào tìm kiếm thì sẽ hiện thị lên thông tin của user kèm theo đó là nút mời user vào trong đoạn chat.



Hình 9:Form mời bạn vào group

Ví dụ như hình bên khi nhập số điện thoại 0123456789 thì hiện lên thông tin user có tên là “Dinh Hieu” và nút mời. Nếu nhập sai hoặc để trống thì sẽ không hiển thị lên gì cả. Nếu user đã tồn tại trong room thì khi nhấn mời sẽ hiển thị lên thông báo “user đã có trong nhóm”.



Hình 10: Thông báo lỗi khi mời thành viên đã có trong nhóm




# Mời bạn

Tìm theo

Phone

Email

0123456789



định hieu

Mời

Hủy

Tìm kiếm

Hình 11: Tìm kiếm bạn bè để mời chat

Bên phải của màn hình chính trong hình có biểu tượng cho phép chỉnh sửa thông tin phòng. Khi nhấn vào tên và mô tả sẽ hiển thị lên form như các hình bên dưới cho phép ta nhập vào tên phòng mới và mô tả mới cho phòng. Trong trường hợp tên phòng hoặc mô tả trống nhưng vẫn nhấn vào nút xác nhận thì sẽ hiển thị lên một thông báo “Hãy nhập tên phòng!!!” hoặc “Hãy nhập mô tả!!!” và không cập nhập lại tên phòng.



Bạn muốn thay đổi

Tên

Mô tả

### Đổi tên nhóm



Bạn có chắc chắn muốn đổi tên nhóm, khi xác nhận tên nhóm mới sẽ hiển thị với tất cả các thành viên

Hủy

Xác nhận

### Đổi mô tả



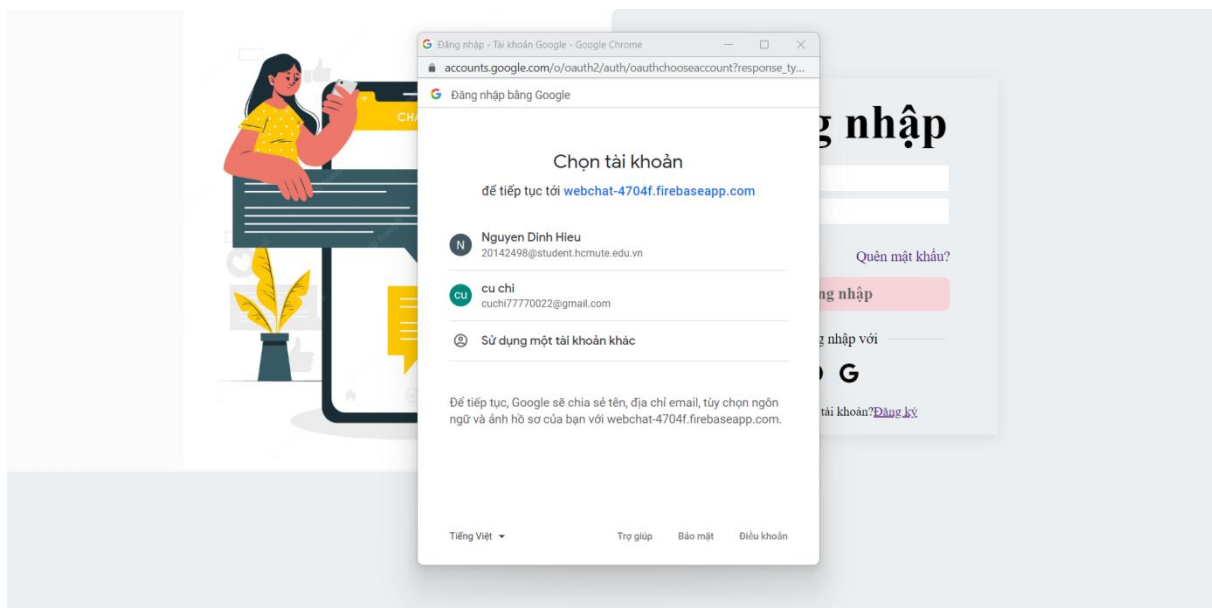
Bạn có chắc chắn muốn đổi mô tả, khi xác nhận tên nhóm mới sẽ hiển thị với tất cả các thành viên

Hủy

Xác nhận

Hình 12: Form đổi tên phòng và mô tả

#### 4.1.2. Đăng nhập, đăng xuất:



Hình 13: Login thông qua google

Login qua google và sau khi login thành công thì sẽ hiện lên thông tin của user trong phần header chat như hình bên.



Hình 14: Thẻ hiển thị hồ sơ, cài đặt, đăng xuất

Bên cạnh đó cũng cho phép người dùng đăng xuất sau khi login. Đối với login thông qua google và facebook firebase sẽ duy trì login cho đến khi ta đăng xuất. Tuy

nhiên một bất lợi đó là khi login qua facebook hiện tại chỉ cho phép login thông qua facebook đã được lưu vào trong firebase mà không dùng được cho facebook nào khác.

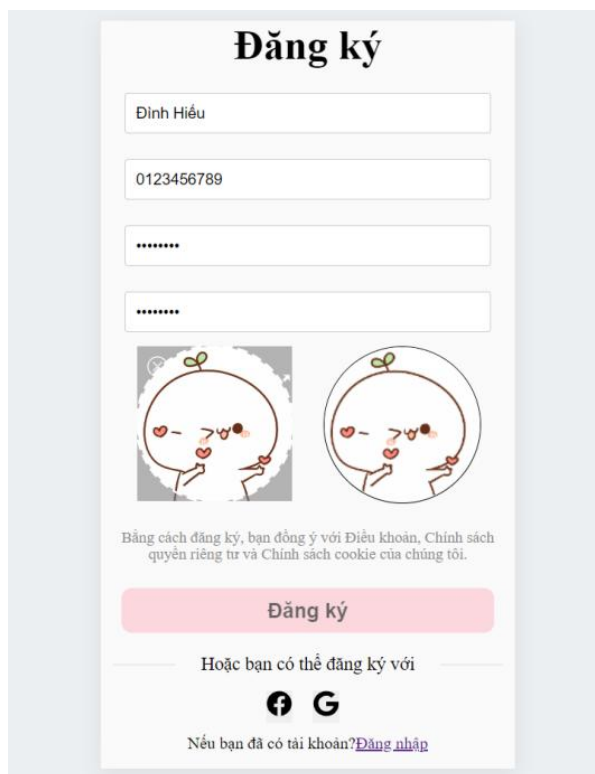
Bên cạnh việc login thông qua facebook và google, ta còn có thể login thông qua số điện thoại mà đã đăng ký tài khoản. Đối với login thông qua số điện thoại, cho phép lưu thông tin đăng nhập để khi lần sau vào không cần đăng nhập lại.

Khi đăng nhập thông qua số điện thoại mật khẩu, khi người dùng nhập đúng thì sẽ dẫn người dùng đến trang chat. Nhưng nếu không nhập gì cả hoặc nhập sai thông tin thì sẽ hiển thị lên thông báo đăng nhập không thành công.

The image shows a login form titled "Đăng nhập" (Login) in a large, bold, black serif font. Below the title are two input fields: the first is preceded by a person icon and labeled "Số điện thoại" (Phone number); the second is preceded by a lock icon and labeled "Mật khẩu" (Password). Below these fields is a checkbox labeled "Nhớ mật khẩu" (Remember password). A large, rounded pink button with the text "Đăng nhập" is positioned below the checkbox. Underneath the button is the text "Đăng nhập với" (Login with) followed by two circular icons: the Facebook 'f' logo and the Google 'G' logo. At the bottom of the form, the text "Nếu chưa có tài khoản?" (If you don't have an account?) is followed by a purple, underlined link that says "Đăng ký" (Sign up).

Hình 15: Form đăng nhập vào GITCHAT

#### 4.1.3. Đăng ký:



The screenshot shows a registration form with the title "Đăng ký" at the top. It contains four input fields: the first is labeled "Định Hiều" (likely a typo for "Họ Tên"), the second contains the number "0123456789", and the next two are masked with "\*\*\*\*\*". Below the fields are two identical cartoon illustrations of a character with a heart on its chest. Under the illustrations, there is a line of text: "Bằng cách đăng ký, bạn đồng ý với Điều khoản, Chính sách quyền riêng tư và Chính sách cookie của chúng tôi." Below this is a pink button labeled "Đăng ký". At the bottom, there is a link "Đăng nhập" preceded by the text "Nếu bạn đã có tài khoản?".

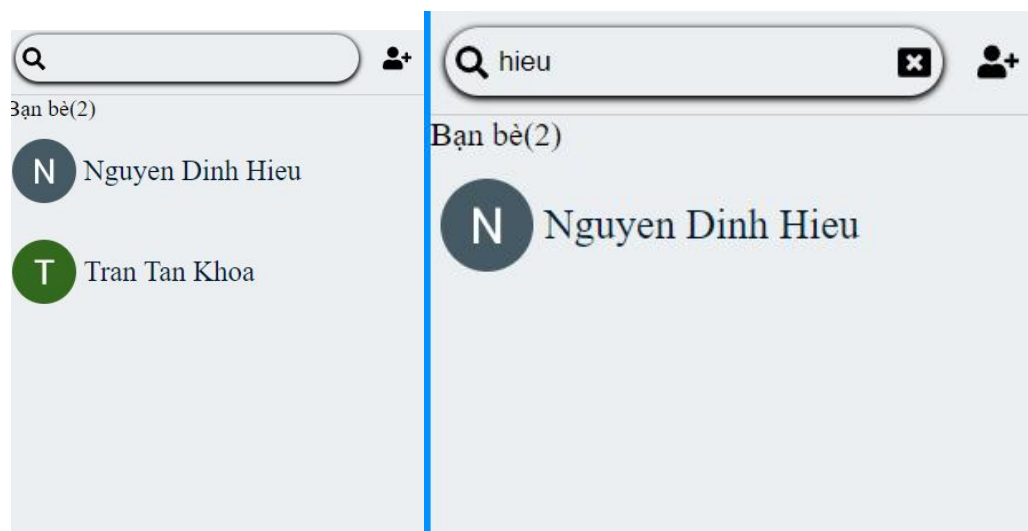
Hình 16:Đăng ký tài khoản.

Khi nhấn đăng ký thì lập tức user sẽ được thêm vào trong cơ sở dữ liệu, và sẽ chuyển trang sang trang login cho phép chúng ta login tài khoản đã tạo. Nếu nhập thiếu trường thì sẽ in ra thông báo đăng ký thất bại, nếu nhập không trùng mật khẩu cũng sẽ thông báo không đăng ký thất bại.

#### 4.1.4. Tìm kiếm bạn bè:

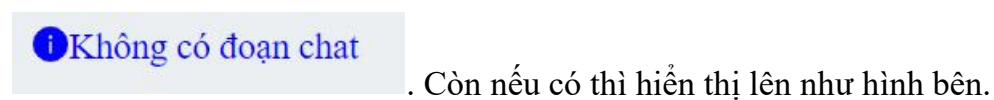
Khi vào trong trang bạn bè sẽ hiển thị toàn bộ danh sách bạn bè. Bên trên danh sách bạn bè cho phép chúng ta tìm kiếm bạn bè theo tên như hình bên. Khi nhập vào “hieu” thì sẽ cập nhập lại danh sách bạn bè như hình bên. Bên cạnh đó, mỗi lần nhập sẽ có icon hiện bên ô nhập khi nhấn vào sẽ xóa các ký tự trong ô search. Khi nhấn vào từng bạn bè thì sẽ hiển thị lên toàn bộ cuộc trò chuyện với người bạn đó. Đồng thời tại

cuộc trò chuyện này vẫn cho phép nhắn tin lẫn nhau, tuy nhiên không thể thêm bạn bè khác vào cuộc trò chuyện được nữa.



Hình 17: Danh sách bạn bè và tìm kiếm thành viên để mời vào nhóm

Nếu lựa chọn vào bạn bè mà chưa từng có cuộc hội thoại thì sẽ hiển thị lên thông báo



Hình 18: Khi nhấn vào tin nhắn với bạn có tên là “hieu”

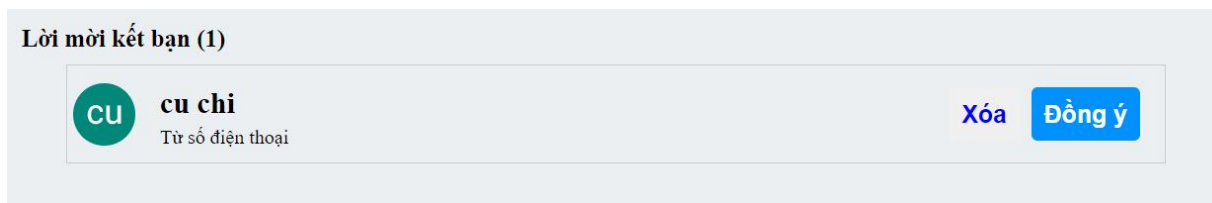
#### 4.1.5. Thêm bạn bè:

Bên phải ô tìm kiếm còn có một icon khi nhấn vào icon đó sẽ hiển thị lên form cho phép tìm kiếm bạn bè theo số điện thoại và email như hình bên. Khi nhập vào số điện thoại hay email sẽ hiển thị thông tin người dùng tương ứng với thông tin nhập vào. Khi nhấn vào nút thêm bạn bè thì bên người dùng tương ứng sẽ hiển thị lên màn hình lời mời kết bạn. Đồng thời khi thêm bạn bè mới sẽ tạo ra phòng chat giữa hai người. Trường hợp nếu đã từng là tồn tại phòng chat giữa hai người thì khi đồng ý kết bạn sẽ không tạo ra phòng mới nữa. Khi đã đồng ý kết bạn ta có thể hủy kết bạn bằng cách tìm kiếm bạn bè theo thông tin liên hệ của bạn qua số điện thoại hoặc email sau đó nhấn vào bạn bè thì lập tức sẽ hủy kết bạn.



The form is titled "Thêm bạn" (Add friend) and includes a close button (X). It has two search options: "Số Điện Thoại" (Phone Number) and "Email". Below these is a text input field containing "0123456789" and a close button (X). Underneath is a placeholder for a profile picture, labeled "định hieu" (profile picture). At the bottom, there are four buttons: "Nhắn Tin" (Send Message), "Kết bạn" (Add Friend), "Hủy" (Cancel), and "Tìm kiếm" (Search).

Hình 18: Form tìm kiếm bạn bè theo số điện thoại



Hình 19: Thể hiện lời mời kết bạn

Như hình trên khi login user tương ứng thì hiển thị lời mời kết bạn cho phép ta đồng ý hoặc xóa lời mời đi. Nếu đồng ý thì lập tức user này sẽ được thêm vào danh sách bạn bè còn nếu xóa đi thì sẽ không tồn lại lời mời nữa.

#### 4.1.6. Thêm, xóa group:

Khi nhấn vào “Thêm phòng” trên màn hình thì sẽ hiển thị ra form cho phép ta nhập tên phòng và mô tả của phòng. Sau khi nhấn xác nhận sẽ hiển thị toàn bộ các phòng ra màn hình. Trong trường hợp một trong thông tin để trống thì sẽ không thêm được phòng. Đồng thời để có thể thêm được phòng thì người dùng khi tạo phòng bắt buộc phải thêm hai người khác vào cuộc trò chuyện.







Hình 20: Thêm phòng và hiển thị danh sách phòng

Khi nhấn vào phòng sau đó nhấn vào nút xóa phòng thì sẽ hiển thị lên một thông báo xác nhận ta sẽ xóa phòng đó. Khi nhấn có thì phòng lập tức sẽ bị xóa và hiển thị lại danh sách các phòng đang có đồng thời thông báo xóa phòng thành công. Và nếu user không phải là chủ phòng thì sẽ không xóa được phòng.



Hình 21: Form thông báo xóa phòng

#### 4.1.7. Thông tin người dùng:

Khi nhấn vào nút mũi tên chỉ xuống bên phải ảnh đại diện ở góc trên bên phải màn hình sẽ hiển thị lên một form. Trong form đó có thông tin hồ sơ của ta và đăng xuất. Khi nhấn vào thông tin hồ sơ sẽ xuất hiện một form hiển thị các thông tin cơ bản như ảnh đại diện, tên sử dụng, liên hệ là email hoặc số điện thoại đăng ký cho tài khoản đó.



Hình 22: Form thể hiện thông tin cá nhân

#### 4.2. Hướng dẫn chạy:

Để chạy được đồ án này đầu tiên cần phải cài ứng dụng visual studio code trên máy. Sau khi cài xong trên thanh công cụ chọn terminal -> new terminal, tại màn hình terminal gõ lệnh “git clone https://github.com/nguyendinhhiu12345/Webchat.git” nếu đã cài git trên máy để cài đặt đồ án về máy. Nếu không ta có thể vào link github được

cung cấp nhấn vào  như hình và download ZIP về máy.

Sau khi đã có đồ án trên máy thì mở ứng dụng visual studio code và mở terminal gõ lệnh “npm install” hoặc “npm i” để cài đặt các thư viện hỗ trợ. Sau đó, gõ

lệnh “npm start” chờ đến lúc trang web hiển thị thì có thể sử dụng. Tuy nhiên, vì sử dụng firebase nên cần có mạng để có thể lấy được dữ liệu lưu trữ.

Tên miền “http://localhost:3000” sẽ hiển thị lên trang chủ, “http://localhost:3000/chat” sẽ hiển thị lên trang chat, “http://localhost:3000/login” là đường dẫn vào trang đăng nhập, “http://localhost:3000/signup” là đường dẫn vào trang đăng ký.

## CHƯƠNG 5. KẾT LUẬN

### 5.1. Kết quả đạt được:

Về cơ bản nhóm tự nhận xét chương trình tạo ra đã giải quyết được những yêu cầu đề ra. Nhóm đã sử dụng reactjs và firebase để xây dựng nên giao diện website và thực hiện được các tính năng đề ra như đăng nhập được thông qua số điện thoại, facebook, google, đăng ký được tài khoản thông qua số điện thoại,... Hiểu rõ hơn về ngôn ngữ đang sử dụng. Bên cạnh đó, nhóm còn nâng cao kỹ năng làm việc nhóm, hiểu rõ hơn về các thành viên. Từ đó, nâng cao được nhiều kỹ năng cho bản thân mình.

### 5.2. Định hướng phát triển:

Trong quá trình thực hiện cũng gặp phải nhiều khó khăn trong việc code chẳng hạn như firebase thay đổi liên tục nhóm phải tìm hiểu về những sự thay đổi đó để thay đổi code cho phù hợp. Gặp nhiều lỗi phát sinh có những lỗi sửa được nhưng cũng có lỗi chưa sửa được nhóm đã cố gắng tìm cách khắc phục. Sử dụng quá số lượng data mà firebase cho phép và khắc phục bằng cách phải tạo ra cái mới để sai cùng với nhiều những khó khăn khác. Tuy nhiên nếu có nhiều thời gian hơn nhóm sẽ tiếp tục phát triển theo một số định hướng như sau:

- Tối ưu hóa code để thời gian chạy tốt hơn, dễ hiểu hơn thuận tiện cho việc phát triển sau này.
- Thiết kế lại giao diện đẹp hơn, các tính năng theo tác dễ hơn.
- Cấu trúc file dễ hiểu, dễ nhìn hơn.
- Tìm ra những trường hợp bị lỗi và sửa lại chúng. Sau đó phát triển thêm những tính năng mới, áp dụng thêm những công nghệ mới về chẳng hạn như boot chat tự động trả lời,...

### 5.3. Ưu điểm, hạn chế:

Bên cạnh việc xây dựng trang web thì nhóm em nhận thấy được những ưu nhược điểm như sau.

#### **5.3.1. Ưu điểm:**

Xây dựng hoàn thiện trang giao diện web với đầy đủ tính năng đưa ra một cách hợp lý, phù hợp với người sử dụng dễ thao tác. Đăng nhập với nhiều phương thức giúp dễ dàng kết nối với mọi người hơn.

#### **5.3.2. Hạn chế:**

Vẫn còn lỗi ở một vài trường hợp có thể nhóm vẫn chưa tìm ra. Chỉ phù hợp sử dụng được cho website và chưa thiết kế để phù hợp với điện thoại. Firebase thường xuyên cập nhập những cái mới làm cho ta cũng phải cập nhập lại những câu lệnh một cách hợp lý. Trong một số trường hợp bị lỗi làm cho lượng dữ liệu bị giới hạn so với firebase cho phép không sử dụng thêm được nữa. Login qua facebook chỉ cho phép một tài khoản facebook được đăng nhập vào.

## TÀI LIỆU THAM KHẢO

- [1]. <https://firebase.google.com/docs> , hướng dẫn sử dụng firebase
- [2]. [https://tenten.vn/tin-tuc/firebase-la-gi-nen-tang-tu-google/#:~:text=Firebase%20l%C3%A0%20g%C3%AC%3F%20Firebase%20l%C3%A0%20Backend-as-a-Service%20%28Baas%29%20n%E1%BB%81n,tr%C3%AA%20n%E1%BB%81n%20Cloud%20%E2%80%93%20%20i%E1%BB%87n%20to%20A1n%20%C4%91%C3%A1m%20m%C3%A2y,firebase là gì?](https://tenten.vn/tin-tuc/firebase-la-gi-nen-tang-tu-google/#:~:text=Firebase%20l%C3%A0%20g%C3%AC%3F%20Firebase%20l%C3%A0%20Backend-as-a-Service%20%28Baas%29%20n%E1%BB%81n,tr%C3%AA%20n%E1%BB%81n%20Cloud%20%E2%80%93%20%20i%E1%BB%87n%20to%20A1n%20%C4%91%C3%A1m%20m%C3%A2y,firebase%20la%20gi%3F)
- [3]. <https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7> , Bắt đầu với Reactjs.
- [4]. <https://reactjs.org/docs/add-react-to-a-website.html> , hướng dẫn cụ thể về reactjs.
- [5]. <https://bizflycloud.vn/tin-tuc/html-la-gi-20211119162558126.htm> , HTML là gì? Ưu nhược điểm và vai trò của HTML trong lập trình web?