

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ  
XÂY DỰNG ỨNG DỤNG HỢP MẶT TRỰC TUYẾN  
VỚI STRINGEE API**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Trần Quốc Thịnh - Võ Bá Toàn**

**Mã số sinh viên: 60132367 - 60137177**

**Khóa: 60**

**KHÁNH HÒA-2021**

**TRƯỜNG ĐẠI HỌC NHA TRANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**  
**BỘ MÔN HỆ THỐNG THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**  
**XÂY DỰNG ỨNG DỤNG HỢP MẶT TRỰC TUYẾN**  
**VỚI STRINGEE API**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Trần Quốc Thịnh - Võ Bá Toàn

Mã số sinh viên: 61131272 - 60137144

Khánh Hòa, tháng 01/2021

**TRƯỜNG ĐẠI HỌC NHA TRANG****Khoa: Công nghệ Thông tin****PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ****Tên đề tài:** Xây dựng ứng dụng họp mặt trực tuyến với Stringee API**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh**Sinh viên được hướng dẫn:** Trần Quốc Thịnh - Võ Bá Toàn**MSSV:** 61131272 - 60137144**Khóa:** 60**Ngành:** Công nghệ Thông tin

Lần	Ngày	Nội dung	Nhận xét của GVHD
1	7/12/2020	Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sau đó, thảo luận kế hoạch để triển khai phần mềm và hoàn thiện báo cáo kịp tiến độ.	Sinh viên và GVHD trao đổi nội dung của đề tài. Sinh viên có nhiều phương án để triển khai, tuy nhiên cần kiểm thử các phương pháp trước khi đưa ra phương án cuối cùng thông qua việc demo các chức năng.
2	14/12/2020	Sinh viên trình bày ý tưởng và giải pháp của mình. Sinh viên trình bày kết quả mô phỏng cách thức gọi video call.	Sinh viên có ý tưởng rõ ràng về nội dung thực hiện. Phân mô phỏng có kết quả tốt thể hiện được các yêu cầu đặt ra, tuy nhiên về mặt thực tế sản phẩm có thể gặp lỗi, đề nghị tiến hành kiểm thử.
3	21/12/2020	Sinh viên trình bày kết quả mô phỏng và hoàn tất quá trình này. Về mặt giao diện còn đơn giản, song đã đáp ứng được yêu cầu của sản phẩm.	Kết thúc quá trình mô phỏng sinh viên đã đáp ứng đầy đủ các yêu cầu đặt ra. Sinh viên cần hoàn thiện giao diện cũng như tìm hiểu cách thức bảo mật, kiểm tra số lượng kết nối và tìm cách xây dựng máy chủ offline.
4	4/01/2021	Sinh viên nộp bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa. Trình bày các yêu cầu lần trước.	Báo cáo chỉ trình bày chung chung chưa đi vào cụ thể phân tích các yêu cầu của bài toán, hình ảnh, bảng biểu chưa trình bày rõ ràng. Cần hiệu chỉnh theo yêu cầu của GVHD.

5	11/01/2021	Sinh viên nộp bản thảo lần 2 và hoàn thiện khả năng kết nối, bảo mật của phần mềm, các tính năng chia sẻ được bổ sung, thử tính năng gửi nhận tập tin qua ứng dụng.	Báo cáo lần này đã khắc phục được các lỗi của lần trước, tuy nhiên phần phương pháp và kết quả chưa nổi bật, chưa có sự liên kết giữa các phần. Các yêu cầu khác như zoom hay google meet chưa thực hiện được cần thời gian tìm hiểu thêm.
6	18/1/2021	Sinh viên hoàn thiện báo cáo cuối cùng sau khi nhận các ý kiến đóng góp.	Sinh viên đã hoàn thiện báo cáo, sản phẩm đã hoàn thành các chức năng cơ bản, còn một số yêu cầu khác thì do hạn chế của API nên chưa được triển khai. Sẽ cần thêm thời gian

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL):**

Sinh viên thực hiện tốt các yêu cầu của GVHD, trong quá trình thực hiện đề tài có sự liên hệ chặt chẽ với GV. Sinh viên có sự nỗ lực rất lớn trong việc tìm hiểu công nghệ mới và tự mình trang bị các kiến thức đó.

Về nội dung báo cáo đã thỏa mãn các yêu cầu của đề tài như trong đề cương. Về sản phẩm, vì đây là sản phẩm hoàn toàn do SV đề xuất nên GVHD chưa thể giúp được nhiều hơn nhưng SV đã hoàn thành tốt các yêu cầu đề ra.

Về hình thức của báo cáo và sản phẩm, báo cáo trình bày rõ ràng các mục tiêu, phương pháp, kết quả và thảo luận cho sản phẩm. Sản phẩm có giao diện thân thiện, dễ dùng. Các đoạn mã được giải thích đầy đủ và chi tiết.

Điểm hình thức: 9.0/10 Điểm nội dung: 9.0/10 **Điểm tổng kết: 9.0/10**

Đồng ý cho sinh viên: Được bảo vệ: ☒ Không được bảo vệ: ☐

Khánh Hòa, ngày 20 tháng 01 năm 2021

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

## MỤC LỤC

PHẦN 1. GIỚI THIỆU .....	1
1.1. Stringee API .....	2
1.2. Glitch.com .....	3
PHẦN 2. PHƯƠNG PHÁP THỰC HIỆN .....	4
2.1. Tạo tài khoản Stringee và code bằng Glitch.com .....	4
2.2. Cách thức hoạt động của ứng dụng Video Call .....	4
2.2.1. Tạo Room .....	4
2.2.1.1. Tạo RoomToken .....	4
2.2.1.2. Tạo UserToken .....	5
2.2.1.3. Kết nối vào hệ thống.....	6
2.2.2. Join Room .....	6
2.2.2.1. Join Room bằng RoomID .....	6
2.2.2.2. Join Room bằng đường link .....	7
2.2.2.3. Tạo Track.....	7
2.2.2.4. Thêm Track vào Room.....	8
2.2.2.5. Hiện thị Track ra Room.....	8
2.2.2.6. Remove Track .....	9
2.2.3. Tính năng Share màn hình .....	9
PHẦN 3. KẾT QUẢ .....	10
3.1. Tạo account Stringee và ứng dụng hỗ trợ code Glitch.....	10
3.2 Ứng dụng hợp mặt trực tuyến đơn giản với Stringee API .....	10
3.2.1 Tạo Room .....	10
3.2.2 Join Room.....	11
3.3.3 Share màn hình .....	13
PHẦN 4. TỔNG KẾT THỰC TẬP .....	15
TÀI LIỆU THAM KHẢO.....	16

## LỜI NÓI ĐẦU

Trong thời kỳ giãn cách xã hội vì đại dịch Covid19 việc các công ty, doanh nghiệp, trường học phải đối mặt với nhiều khó khăn vì không thể làm việc với nhau trực tiếp. Chính vì vậy các ứng dụng họp mặt trực tuyến được sử dụng nhằm mục đích giúp cho nhiều người cùng kết nối với nhau để họp, bàn công việc và dạy học. Vì thế, đề tài “Xây dựng ứng dụng họp mặt trực tuyến với Stringee API” được tạo ra để mô hình hóa các chương trình như của Google Meet hay Zoom, trên cơ sở đó làm nền tảng cho các đợt phát triển sản phẩm tiếp theo. Sản phẩm sử dụng nền tảng Stringee API cho việc xây dựng ứng dụng. Mặc dù sản phẩm chưa hoàn thiện nhưng đã vận hành cơ bản được tính năng gọi, nhắn tin trực tuyến, chia sẻ video trong quá trình thao tác.

Trải qua một thời gian thực tập, em đã tìm hiểu được khá nhiều lý thuyết thú vị. Hiểu được cách thức vận hành của các ứng dụng lớn như Zoom hay Google Meeting. Biết được nền tảng lý thuyết của API về ưu điểm và nhược điểm cũng như việc API phổ biến như thế nào trong thời đại công nghệ ngày nay. Đặc biệt là em được tìm hiểu thêm về ngôn ngữ lập trình Javascript. Để xây dựng được 1 ứng dụng như Zoom hay Google Meeting thật sự không đơn giản tí nào. Vì thế nếu muốn cải thiện ứng dụng Video Call này thì em phải cần rất nhiều thời gian nghiên cứu, cần tích lũy thêm kinh nghiệm cũng như kỹ năng lập trình để có thể cải thiện ứng dụng tốt hơn.

Toàn bộ đoạn mã của chương trình được tải lên theo địa chỉ:

[https://github.com/thinhdoanvu/ThuctapCoSo2020/tree/main/TranQuocThinh\\_VoBaToan](https://github.com/thinhdoanvu/ThuctapCoSo2020/tree/main/TranQuocThinh_VoBaToan)

## PHẦN 1. GIỚI THIỆU

Trong thời đại công nghệ phát triển mạnh mẽ, rất nhiều ứng dụng được tạo ra nhằm phục vụ người dùng trong nhiều lĩnh vực khác nhau của cuộc sống. Một trong số đó, các ứng dụng họp mặt trực tuyến như Google Meeting, Zoom trở nên phổ biến và được rất nhiều công ty, doanh nghiệp và cả trường học sử dụng trong thời kỳ toàn cầu đang phải chịu ảnh hưởng bởi dịch Covid19.

Trong thời kỳ giãn cách xã hội vì đại dịch Covid19 việc các công ty, doanh nghiệp, trường học phải đối mặt với nhiều khó khăn vì không thể làm việc với nhau trực tiếp. Chính vì vậy các ứng dụng họp mặt trực tuyến được sử dụng nhằm mục đích giúp cho nhiều người cùng kết nối với nhau để họp, bàn công việc và dạy học. Mỗi ứng dụng đều có những ưu điểm và khuyến điểm khác nhau nhưng đều có chung mục đích là đáp ứng yêu cầu họp mặt trực tuyến một cách dễ dàng và nhiều tính năng hỗ trợ đi kèm. Vì thế, đề tài “*Xây dựng ứng dụng họp mặt trực tuyến với Stringee API*” được tạo ra để mô hình hóa các chương trình như của Google Meet hay Zoom, trên cơ sở đó làm nền tảng cho các đợt phát triển sản phẩm tiếp theo.

Mặc dù sản phẩm chưa hoàn thiện như 2 sản phẩm kể trên nhưng đã vận hành cơ bản được tính năng gọi, nhắn tin trực tuyến, chia sẻ video trong quá trình thao tác. Các tính năng bảo mật hay mã hóa cuộc gọi chưa được nghiên cứu cụ thể, số lượng người tham gia cuộc họp chưa được kiểm chứng đầy đủ, máy chủ của ứng dụng còn phụ thuộc vào nền tảng Stringee nên sản phẩm chỉ dừng ở mức độ mô phỏng chức năng.

### 1.1. Stringee API

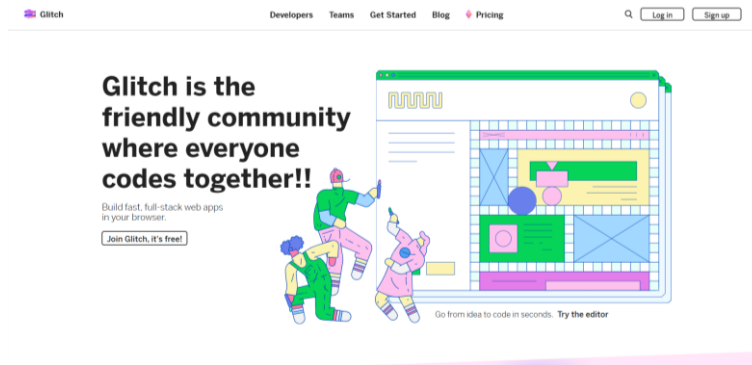


Hình 1.1. Giao diện chính của website Stringee API

Stringee được thành lập năm 2017 bởi Đậu Ngọc Huy và Nguyễn Bá Luân – hai kỹ sư công nghệ tốt nghiệp trường Đại học Bách khoa Hà Nội, từng khởi nghiệp với sản phẩm Bom Chat.

Stringee là 1 nền tảng lập trình giao tiếp ở Việt Nam, cung cấp những API để Video Call, SMS, tổng đài chăm sóc khách hàng. Giúp cho các doanh nghiệp và developer dễ dàng tích hợp những tính năng như Video Call, Video Chat vào hệ thống. Địa chỉ trang chủ của Stringee:

## 1.2. Glitch.com



*Hình 1.2. Giao diện chính của trang web Glitch.com*

Glitch là một nền tảng được xây dựng bởi find folks tại Fog Creek Software, công ty được biết đến với FogBugz, Trello, đồng sáng lập StackOverflow. Hầu hết lập trình viên đều quen thuộc với cái tên Joel Soplisky. Glitch là nền tảng mà người dùng có thể xây dựng ứng dụng Node.js trên cloud, public or private. Hay có thể hiểu đơn giản nó là 1 service để đẩy ứng dụng Node.js lên web.

Khi người dùng muốn thử phát triển một ứng dụng web với Node.js, cần phải cài đặt Server và đăng ký Domain, điều này khá phức tạp và mất nhiều thời gian. Glitch có thể hỗ trợ cả 2 bước trên giúp tiết kiệm thời gian và tài chính trong việc xây dựng một Server web có thể đáp ứng số người dùng tương đối lớn.



## PHẦN 2. PHƯƠNG PHÁP THỰC HIỆN

### 2.1. Tạo tài khoản Stringee và code bằng Glitch.com

**Bước 1.** Truy cập trang web Stringee.com

**Bước 2.** Đăng ký tài khoản Stringee và thực hiện đăng nhập

**Bước 3.** Tạo Project

**Bước 5:** Truy cập trang web Glitch.com và tiến hành Login

**Bước 6:** Sau khi Login thành công - Chọn New Project - hello-webpage

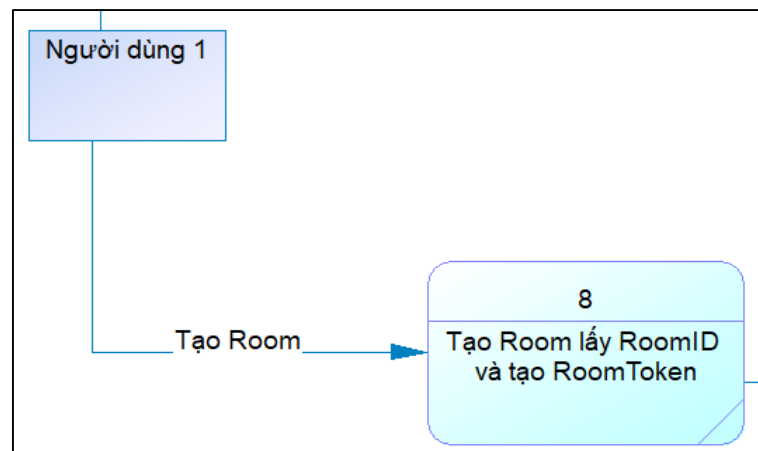
**Bước 7:** Dùng API KEY SID và API KEY SECRET để nhận Project của Stringee và lấy API sẵn có của Stringee.

### 2.2. Cách thức hoạt động của ứng dụng Video Call

#### 2.2.1. Tạo Room

##### 2.2.1.1. Tạo RoomToken

Một cuộc họp trực tuyến cần được bắt đầu với việc khởi tạo một Room với 2 tham số là RoomToken và RoomID. Người khởi tạo được định danh là Người dùng 1.



Hình 2.1 Tạo RoomToken từ RoomID

```

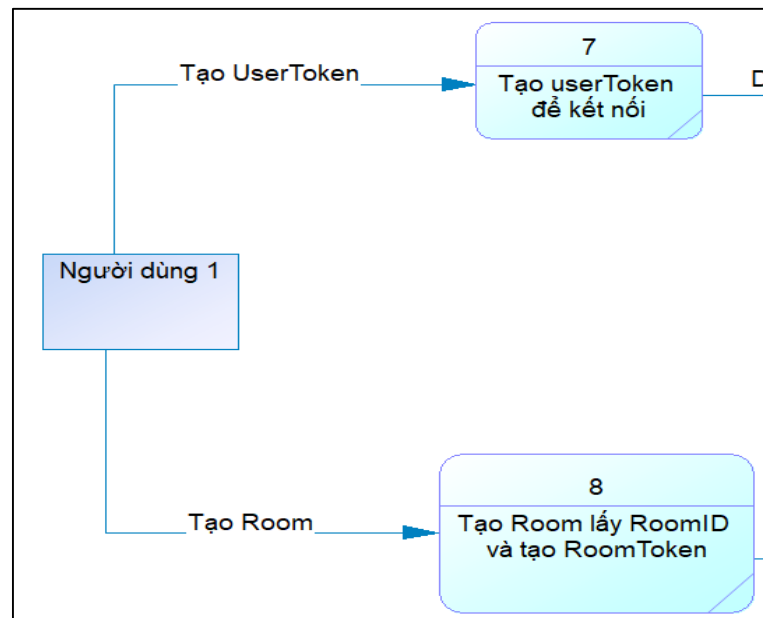
createRoom: async function() {
  const room = await api.createRoom();
  const { roomId } = room;
  const roomToken = await api.getRoomToken(roomId);

  this.roomId = roomId;
  this.roomToken = roomToken;
  console.log({ roomId, roomToken });
  await this.authen();
  await this.publish();
},
  
```

Trong hàm Create Room dòng lệnh `const room = await api.createRoom();` dùng để gọi 1 đối tượng là “room”. Khi createRoom thì đối tượng room này sẽ nhận giá trị từ hàm `await api.createRoom()`. Tiếp đó gọi đối tượng RoomID và gán giá trị của room vào roomID. Gọi hàm roomToken bằng cách lấy giá trị từ roomID thông qua hàm `await api.getRoomToken(roomId)`. Sau khi tạo Room thì đã có RoomID và roomToken, các dòng lệnh tiếp theo được sử dụng nhằm mục đích tái sử dụng khi tiến hành tạo 1 Room mới.

#### 2.2.1.2. Tạo UserToken

Bất kì người dùng nào khi muốn tham gia vào Room, hệ thống cần phải tạo cho mỗi người dùng 1 userToken. Từ đó dùng userToken này để đưa người dùng vào Room



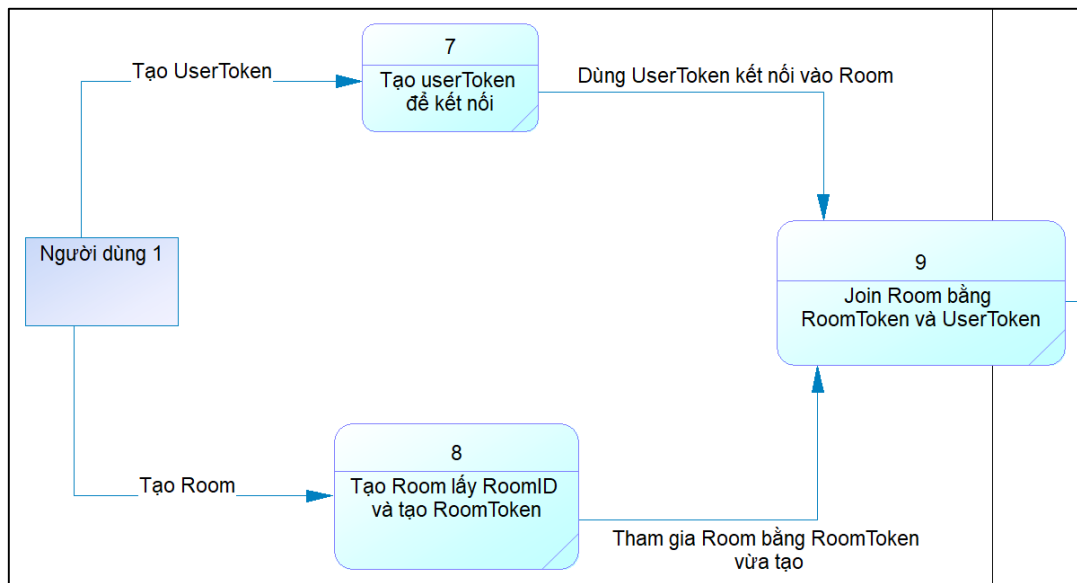
Hình 2.2 Tạo UserToken để kết nối vào hệ thống

```

const userId = `${(Math.random() * 100000).toFixed(6)}`;
const userToken = await api.getUserToken(userId);
this.userToken = userToken;
  
```

Khởi tạo giá trị userID bằng 1 số random. Sau đó gọi đối tượng userToken bằng cách lấy giá trị từ đối tượng userID đã tạo. Cách này được thực hiện bằng hàm `const userToken = await api.getUserToken(userId)`. Và dòng lệnh `this.userToken = userToken;` được gọi lại khi có 1 người khác Join Room, hệ thống sẽ tự động tạo userToken cho các người dùng tiếp theo.

### 2.2.1.3. Kết nối vào hệ thống



Hình 2.3 Join Room

Sau khi có RoomToken và userToken. Hệ thống sẽ nhận 2 giá trị này và đưa vào hệ thống. Hệ thống sẽ tạo 1 client và gọi client đó vào trong Room. Lúc này hệ thống đã nhận được client từ người dùng.

```

if (!this.callClient) {
  const client = new StringeeClient();
  client.on("authen", function(res) {
    console.log("on authen: ", res);
    resolve(res);
  });
  This.callClient = client;
}
this.callClient.connect(userToken);

```

Dòng lệnh `this.callClient.connect(userToken)` được thực hiện khi 1 client đã được tạo từ trước nhận userToken từ người dùng. Từ đó kết nối vào hệ thống. Nếu chưa có client thì hệ thống sẽ tạo client mới thông qua việc tạo đối tượng mới `const client = new StringeeClient()`. Hệ thống sẽ xác thực thông qua dòng lệnh `client.on("authen", function(res))`. Tiếp theo `This.callClient = client` gọi lại client đã được tạo cho các lần khởi tạo tiếp theo.

## 2.2.2. Join Room

### 2.2.2.1. Join Room bằng RoomID

Sau khi tạo Room, người tạo gửi roomID cho những người dùng khác để họ có thể tham gia vào Room.

```

joinWithId: async function() {
    const roomId = prompt("Paste Room ID");
    if (roomId) {
        this.roomId = roomId;
        await this.join();
    }
},

```

Đối tượng roomId nhận giá trị được gửi đến từ ô “Paste RoomID”. Hệ thống sẽ kiểm tra nếu tồn tại RoomID này thì sẽ đưa người dùng vào Room.

#### 2.2.2.2. Join Room bằng đường link

Một cách Join Room khác là Join Room bằng đường link của Room.

```

const urlParams = new URLSearchParams(location.search);
const roomId = urlParams.get("room");
if (roomId) {
    this.roomId = roomId;

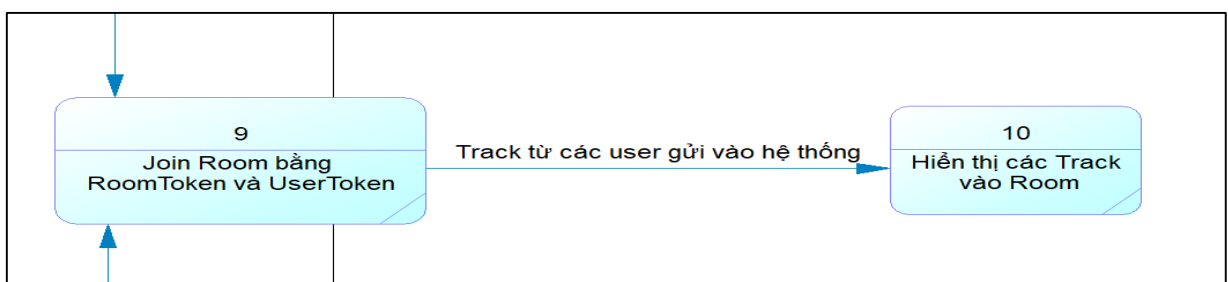
    await this.join();
}

```

Việc tạo ra 1 đường link url của Room được thực hiện khi tiến hành tạo 1 Room mới. Khi muốn Join room bằng link url, hệ thống sẽ kiểm tra xem RoomID này có tồn tại hay không. Nếu có thì thực hiện việc đưa người dùng vào Room.

#### 2.2.2.3. Tạo Track

Để hiển thị được hình ảnh của người dùng thì khi Join Room hệ thống sẽ tạo Track (hình ảnh được lấy từ webcam người dùng mã hóa thành dạng video) sau đó sẽ gửi lên Room cho tất cả người dùng trong Room có thể thấy hình ảnh của nhau.



Hình 2.4 Cách thức gửi hình ảnh từ webcam lên hệ thống

```

const videoElement = localTrack.attach();
this.addVideo(videoElement);
const roomData = await StringeeVideo.joinRoom(
    this.callClient,
    this.roomToken
);
const room = roomData.room;
console.log({ roomData, room });

```

Đối tượng được gọi ở đây là dữ liệu được lấy từ webcam, ma hóa thành dạng Video element. Hàm attach `localTrack.attach()` dùng để lấy video từ videoElement ra. Tiếp theo đó việc thêm video này vào trong hệ thống với dòng lệnh `this.addVideo(videoElement)`. Sau đó tạo cho Video Element này roomToken và userToken để hệ thống đưa Video hiển thị vào Room bằng dòng lệnh `const roomData = await StringeeVideo.joinRoom`. Gán dữ liệu của roomData vào trong room và xuất dữ liệu ra room.

#### 2.2.2.4. Thêm Track vào Room

Sau khi tạo xong Track, việc hiển thị Track này lên Room để tất cả người dùng đang có mặt trong Room thấy.

```
if (!this.room) {
  this.room = room;
  room.clearAllOnMethos();
  room.on("addtrack", e => {
    const track = e.info.track;
    console.log("addtrack", track);
    if (track.serverId === localTrack.serverId) {
      console.log("local");
      return;
    }
    this.subscribe(track);
  });
}
```

Khi chưa tồn tại Room nào. Hệ thống sẽ gọi hàm room khi có 1 Room được tạo. dòng lệnh `room.on("addtrack", e =>` được chạy khi có người nào đó Join vào Room. Khi có 1 Track được tạo thì hệ thống sẽ lấy được thông tin của Track đó bằng dòng lệnh `const track = e.info.track`. Dòng lệnh `if (track.serverId === localTrack.serverId)` được hiểu là nếu Track của mình đã được đưa vào trong Room thì bỏ qua. Còn nếu không thì publish Track của người khác vào trong Room `this.subscribe(track)`

#### 2.2.2.5. Hiển thị Track ra Room

Sau khi hệ thống thực hiện các bước trên. Việc cần làm tiếp theo là publish Track đó ra Room cho tất cả người dùng thấy.

```
publish: async function() {
  const localTrack = await StringeeVideo.createLocalVideoTrack(
    this.callClient,
    {
      audio: true,
      video: true,
```

```

        videoDimensions: { width: 640, height: 360 }
    };

```

Hàm `const localTrack = await StringeeVideo.createLocalVideoTrack` sẽ nhận client và nhận setting của Track bao gồm `audio: true, video: true`, nhận cả hình lẫn tiếng từ client và `videoDimensions: { width: 640, height: 360 }` là khung Video hiển thị lên màn hình với kích thước chiều rộng 640px và chiều cao 360px.

#### 2.2.2.6. Remove Track

Khi có người trong Room tắt trình duyệt. Hệ thống sẽ xóa Track của người dùng đó ra khỏi Room.

```

room.on("removetrack", e => {
    const track = e.track;
    if (!track) {
        return;
    }
    const mediaElements = track.detach();
    mediaElements.forEach(element => element.remove());
});

```

Dòng lệnh `room.on("removetrack", e =>` được chạy khi có người rời khỏi Room. Nếu không có Track thì bỏ qua. Ngược lại gọi đối tượng `mediaElement` gán giá trị của Track vào. Sau đó dùng hàm `mediaElements.forEach(element => element.remove())` để xóa Track ra khỏi Room.

#### 2.2.3. Tính năng Share màn hình

Stringee đã có API hỗ trợ tính năng Share màn hình nên việc cần làm chỉ cần thêm option Share màn hình ở phần hiển thị Track ra Room

```

publish: async function(screenSharing = false) {
    const localTrack = await StringeeVideo.createLocalVideoTrack(
        this.callClient,
        {
            audio: true,
            video: true,
            screen: screenSharing,
            videoDimensions: { width: 640, height: 360 }
        }
    );
    const videoElement = localTrack.attach();      this.addVideo(videoElement);
}

```

## PHẦN 3. KẾT QUẢ

### 3.1. Tạo account Stringee và ứng dụng hỗ trợ code Glitch

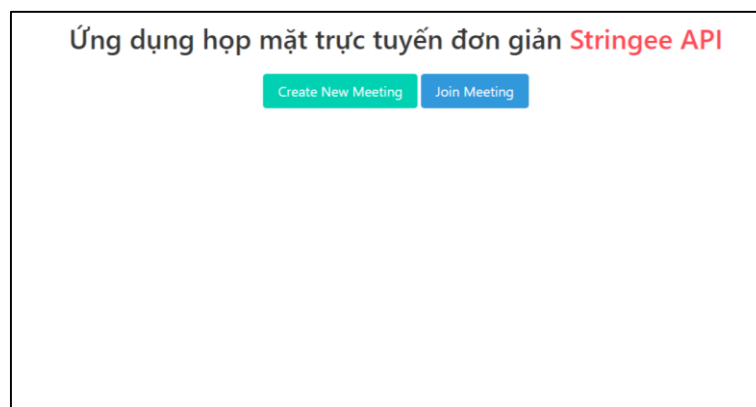
Sử dụng API KEY SID và API KEY SECRET từ account Stringee để đưa vào phần mềm code Glitch.

```
const PROJECT_ID = "SKoRfANJ9fBmYdq38FEjrGt438NqiCgny0";
const PROJECT_SECRET = "WGQ4bUdReFRKbDJoN25Dc0NXMTNhbnIxYXZXNW1Fc1g=";
const BASE_URL = "https://api.stringee.com/v1/room2";
```

Hình 3.1 Paste API KEY SID và API KEY SECRET vào Glitch

API KEY SID và API KEY SECRET được gọi với mục đích giúp phần mềm code Glitch có thể nhận các hàm đã có sẵn của Stringee giúp cho việc tạo ứng dụng trở nên tiện lợi hơn.

### 3.2 Ứng dụng họp mặt trực tuyến đơn giản với Stringee API



Hình 3.2 Giao diện của ứng dụng họp mặt trực tuyến

Giao diện chính của ứng dụng bao gồm 2 nút bấm. Nút “Create New Meeting” có chức năng tạo 1 Room mới. Nút “Join Room” có chức năng tìm kiếm RoomID để thực hiện việc Join Room.

#### 3.2.1 Tạo Room

Việc tạo Room được thực hiện bằng cách chọn “Create New Meeting”. Sau khi thực hiện xong người tạo Room được tự động Join vào Room (Hình 3.3)



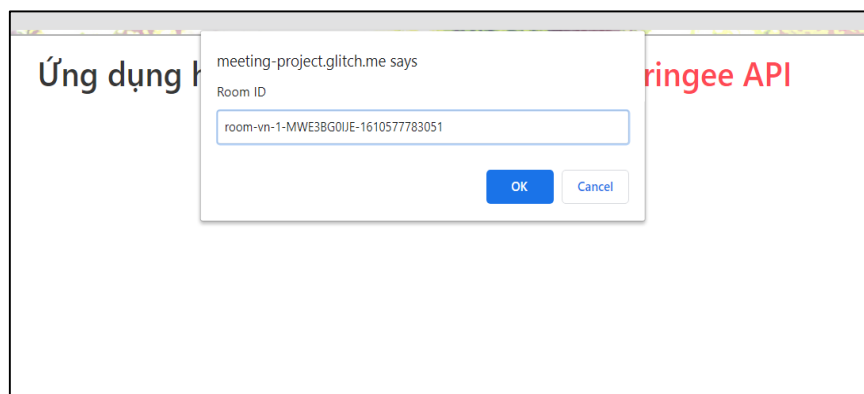
*Hình 3.3 Tạo Room và tự động Join Room*

Sau khi tạo Room, giao diện sẽ hiển thị hình ảnh từ webcam của người tạo Room. Ở giao diện này, có RoomID giúp cho người khác có thể Join vào Room từ nút “Join Meeting” (Hình 3.2). Người dùng cũng có thể gửi đường link của Room đến cho người khác để họ tham gia vào Room.

### 3.2.2 Join Room

Có 2 cách để Join vào Room:

- Dùng RoomID



*Hình 3.4 Join Room bằng RoomID*

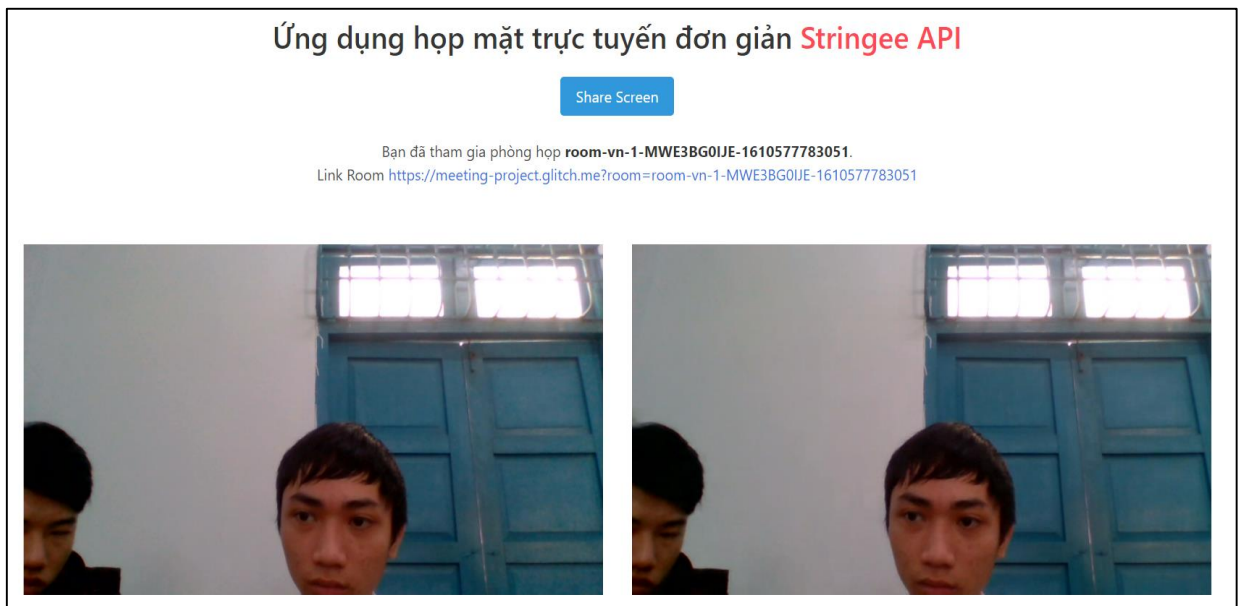
- Dùng đường link của Room



*Hình 3.5 Join Room bằng đường link*

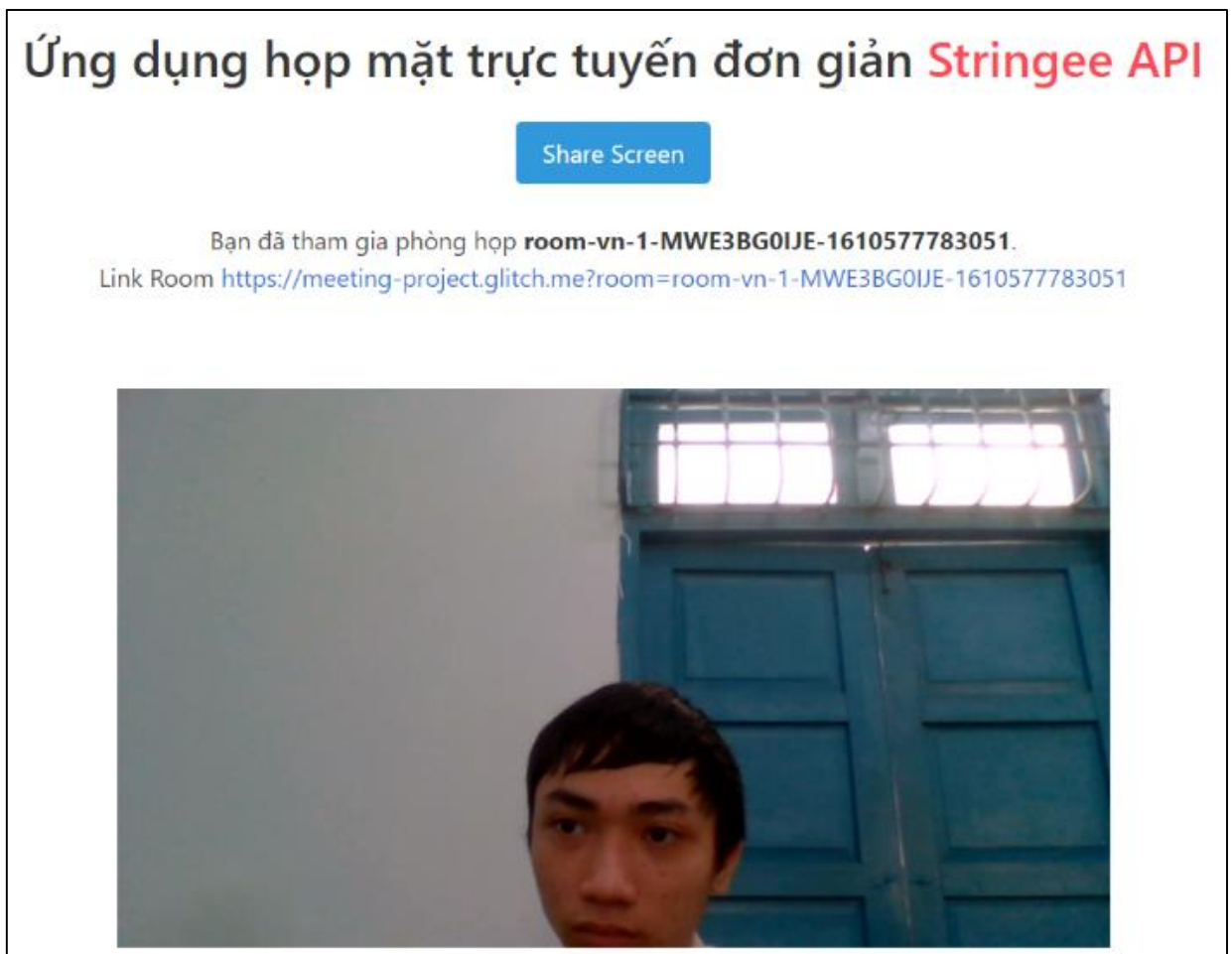


Sau khi Join Room. Track của người thứ 2 được gửi lên Room



Hình 3.6 Giao diện Room khi người thứ 2 Join Room

Sau khi có 1 người dùng rời khỏi phòng họp hệ thống sẽ xóa Track của người dùng đó ra khỏi Room.

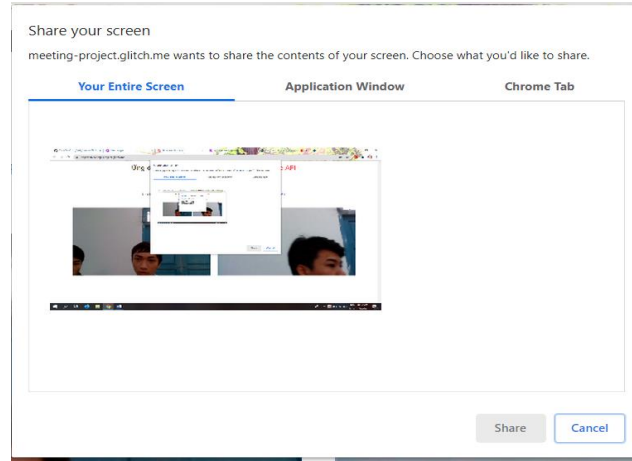


Hình 3.7 Giao diện Room sau khi người dùng thứ 2 rời khỏi Room

### 3.3.3 Share màn hình

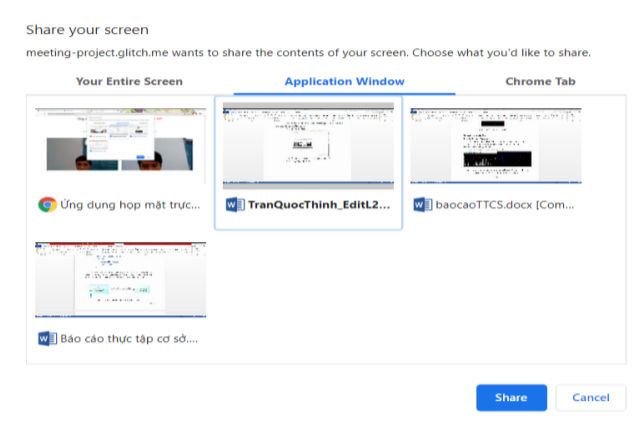
Sau khi chọn vào nút Share Screen. Các tùy chọn để Share Screen xuất hiện. Chọn phương thức Share Screen (Hình 3.8)

- Share toàn bộ màn hình



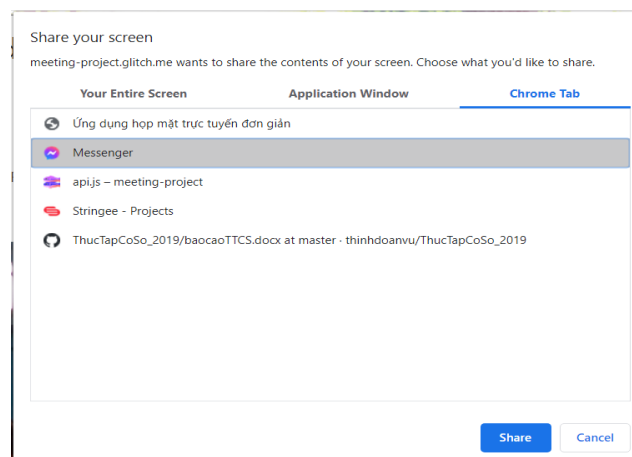
Hình 3.8 (a) Cửa sổ phương thức chọn toàn bộ màn hình

- Share 1 ứng dụng đang hiện hành



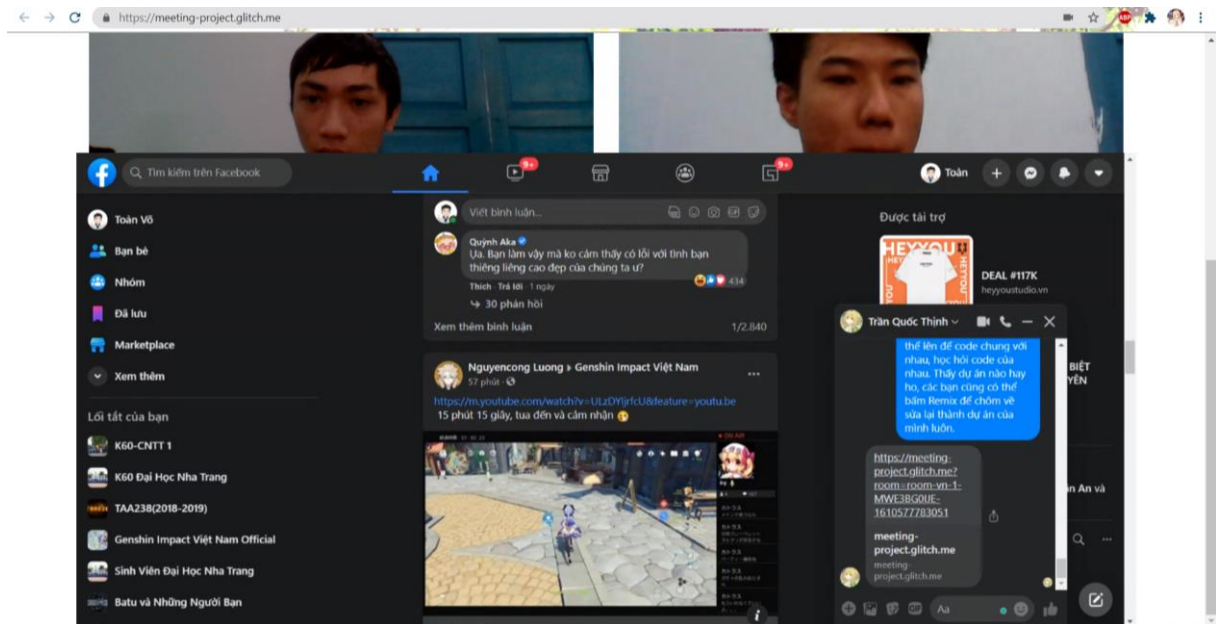
Hình 3.8 (b) Cửa sổ phương thức chọn 1 ứng dụng

- Share 1 tab của Chrome



Hình 3.8 (c) Cửa sổ phương thức chọn 1 tab của Chrome

Sau khi chọn xong phương thức để Share và xác nhận Share. Video hình ảnh sẽ được gửi lên Room.



Hình 3.9 Giao diện Share Screen

## PHẦN 4. TỔNG KẾT THỰC TẬP

### **Kết quả đạt được:**

Trải qua một thời gian thực tập, em đã tìm hiểu được khá nhiều lý thuyết thú vị. Hiểu được cách thức vận hành của các ứng dụng lớn như Zoom hay Google Meeting. Biết được nền tảng lý thuyết của API về ưu điểm và nhược điểm cũng như việc API phổ biến như thế nào trong thời đại công nghệ ngày nay. Đặc biệt là em được tìm hiểu thêm về ngôn ngữ lập trình Javascript. Để xây dựng được 1 ứng dụng như Zoom hay Google Meeting thật sự không đơn giản tí nào. Vì thế nếu muốn cải thiện ứng dụng Video Call này thì em phải cần rất nhiều thời gian nghiên cứu, cần tích lũy thêm kinh nghiệm cũng như kỹ năng lập trình để có thể cải thiện ứng dụng tốt hơn.

### **Ưu điểm của ứng dụng:**

Dễ dùng, dễ xây dựng, không cần bản quyền.

### **Hạn chế:**

Ứng dụng Video Call này chỉ có thể dùng cho web. Chưa sử dụng được trên các nền tảng di động và cũng chưa có ứng dụng cụ thể. Giao diện còn đơn giản, chưa bắt mắt người dùng. Chưa thêm được nhiều tính năng như bật/ tắt webcam tùy ý người sử dụng. Chưa có một số tính năng admin dành cho người host như quyền đuổi người khác ra khỏi phòng, chặn micro của người gây rối...

### **Định hướng phát triển**

Trong tương lai, ứng dụng sẽ còn tiếp tục phát triển, cải thiện hơn để đưa vào sử dụng. Khắc phục những khuyết điểm hiện có và cố gắng phát triển thêm nhiều tính năng mới mà các ứng dụng họp mặt trực tuyến hiện không có.

## TÀI LIỆU THAM KHẢO

Stringee - <https://stringee.com/vi>

Glitch - <https://glitch.com/>

Hướng dẫn từ A-Z tích hợp Video Call với Stringee SDK - <https://viblo.asia/p/huong-dan-a-z-tich-hop-video-call-api-voi-stringee-sdk-YWOZrbVPZQ0>

Source Code - <https://github.com/stringeecom/web-sdk-conference-samples>