TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP TIỂU LUẬN GIỮA KỲ**

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS**

**WEB RTC**

*Người hướng dẫn*: **TS NGUYỄN THANH QUÂN**

*Người thực hiện*: **LÊ MINH KHÔI – 52000774**

**ĐINH VĂN THẮNG – 52000800**

Lớp **: 20050301**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP TIỂU LUẬN GIỮA KỲ**

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS**

**WEB RTC**

*Người hướng dẫn*: **TS NGUYỄN THANH QUÂN**

*Người thực hiện*: **LÊ MINH KHÔI – 52000774**

**ĐINH VĂN THẮNG – 52000800**

Lớp **: 20050301**

Khoá  **: 24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn đến Th Nguyễn Thanh Quân, thầy tận tình truyền dạy những kiến thức, chia sẽ những kinh nghiện của thầy cho chúng em trong môn phát triển ứng dụng web với nodejs. Điều đó rất quan trọng và sẽ là những hành trang cho nghề nghiệp công việc của chúng em sau này.

Chắc hẳn là phần thực hiện và bài báo cáo còn nhiều thiếu sót, mong thầy xemxét và đánh giá để chúng em hoàn thiện hơn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Nguyễn Thanh Quân; 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 19 tháng 4 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Lê Minh Khôi*

*Đinh Văn Thắng*

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

WebRTC là một công nghệ cho phép ứng dụng web truyền dữ liệu âm thanh, video và dữ liệu trực tiếp giữa các trình duyệt khác nhau mà không cần thông qua máy chủ trung gian. Để xây dựng một ứng dụng sử dụng WebRTC, ta có thể sử dụng Node.js và ExpressJS framework để tạo một web service. Ứng dụng sử dụng Websockets để cho phép kết nối thường xuyên giữa các ứng dụng trình khách và trình chủ, gửi các sự kiện như phát sóng âm thanh, video từ phòng họp đến người dùng tham gia. Sử dụng Web Audio API để cung cấp khả năng phát và ghi âm thanh trong trình duyệt web, MediaStream API để truy cập vào phương tiện phát và ghi hình từ camera và microphone của thiết bị sử dụng, Rest API để gửi và nhận yêu cầu HTTP để lưu trữ và phân phối dữ liệu từ một máy chủ API. Ứng dụng cũng có thể sử dụng Rest API để tạo phòng họp và kiểm tra phòng họp tồn tại.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc133083715)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc133083716)

[TÓM TẮT iv](#_Toc133083717)

[MỤC LỤC 1](#_Toc133083718)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc133083719)

[CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI 4](#_Toc133083720)

[1.1 Giới thiệu web RTC. 4](#_Toc133083721)

[1.2 Ưu – nhược điểm của WebRTC. 4](#_Toc133083722)

[1.2.1 Ưu điểm 4](#_Toc133083723)

[1.2.2 Nhược điểm 5](#_Toc133083724)

[1.2.3 Ứng dụng WebRTC 6](#_Toc133083725)

[1.3 Kiến trúc của WebRTC. 7](#_Toc133083726)

[1.3.1 Sơ đồ kết nối giữa Client 7](#_Toc133083727)

[1.3.2 Các phần chính của WebRTC 7](#_Toc133083728)

[1.4 Các công cụ và kỹ thuật được sử dụng trong đề tài WebRTC. 8](#_Toc133083729)

[1.5 Hạn chế và hướng phát triển thêm trong tương lai. 9](#_Toc133083730)

[CHƯƠNG 2 – LÝ THUYẾT 11](#_Toc133083731)

[2.1 Câu 1: Trình bày front-end, back-end và ví dụ. 11](#_Toc133083732)

[2.2 Câu 2: Nodejs và front-end, back-end. 12](#_Toc133083733)

[2.3 Câu 3: Expressjs và front-end, back-end. 13](#_Toc133083734)

[2.4 Câu 4: Làm rõ về expressjs. 14](#_Toc133083735)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

RTC Real-Time Communication

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. 1 Ưu điểm web RTC 5](#_Toc132978428)

[Hình 1. 2 Kiếm trúc web RTC 7](#_Toc132978429)

[Hình 1. 3 Kiếm trúc webRTC 8](#_Toc132978430)

[Hình 2. 1 Ví dụ vể Front-end 11](#_Toc132978433)

[Hình 2. 2 Ví dụ về Back-end 12](#_Toc132978434)

[Hình 2. 3 ví dụ về xử lý các yêu cùa của back-end 13](#_Toc132978435)

[Hình 2. 4 Ví dụ về ExpressJS 14](#_Toc132978436)

[Hình 2. 5 ExpressJS và Nodejs 15](#_Toc132978437)

**DANH MỤC BẢNG**

CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI

1.1 Giới thiệu web RTC.

WebRTC là một bộ các giao diện lập trình ứng dụng web (web API) được viết bằng JavaScript, cho phép truyền tải dữ liệu và giao tiếp trực tiếp giữa các máy tính mà không cần sử dụng plugin hoặc phần mềm hỗ trợ. Thông qua WebRTC, người dùng có thể thực hiện các hoạt động như Video Call, Voice Call trực tiếp giữa các thiết bị, mà không cần thông qua máy chủ trung gian.

1.2 Ưu – nhược điểm của WebRTC.

1.2.1 Ưu điểm

* WebRTC là một công nghệ truyền thông thời gian thực mã nguồn mở miễn phí, và điểm mạnh lớn nhất của nó là không cần đến bất kì plugin hay phần mềm hỗ trợ nào. Điều này giúp tối ưu tốc độ, tiết kiệm chi phí và mang lại sự tiện lợi cho người dùng.
* WebRTC cũng hỗ trợ đa nền tảng, cho phép bất kì trình duyệt hoặc thiết bị nào có hệ điều hành khác nhau tạo kết nối trực tiếp với nhau thông qua real-time voice hoặc video. Nó cũng được mã hóa và xác thực dữ liệu media bằng giao thức SRTP để đảm bảo an toàn cho người dùng.
* WebRTC thích ứng với các điều kiện mạng khác nhau và hỗ trợ việc tích hợp với nhiều kiểu media và các thiết bị đầu cuối khác nhau. Điều này giúp tăng hiệu quả sử dụng băng thông và mang lại trải nghiệm tốt hơn cho người dùng.



Hình 1. 1 Ưu điểm web RTC

* Ngoài ra, WebRTC cũng rất dễ sử dụng và có thể tích hợp vào các dịch vụ web bằng cách sử dụng JavaScript APIs hoặc các Framework có sẵn. Cuối cùng, tính năng video call của WebRTC giúp người dùng tiết kiệm chi phí điện thoại cho mỗi cuộc gọi. Tóm lại, WebRTC là một công nghệ tiên tiến và linh hoạt, giúp người dùng thực hiện các tác vụ truyền thông trực tuyến một cách dễ dàng và tiện lợi.

1.2.2 Nhược điểm

* Một trong những nhược điểm của WebRTC là nó có thể bị cản trở khi tạo kết nối peer-to-peer bởi tường lửa và NAT. Điều này có thể làm giảm hiệu suất truyền tải và gây ra lỗi kết nối.
* Ngoài ra, WebRTC cũng có một nhược điểm khác là nó tạo ra kết nối peer-to-peer giữa các trình duyệt mà không có một cơ chế báo hiệu (signaling) nào được cài đặt sẵn. Điều này có nghĩa là các ứng dụng WebRTC phải sử dụng một cơ chế báo hiệu riêng để trao đổi thông tin và thiết lập kết nối. Việc cài đặt cơ chế báo hiệu này có thể là một thách thức đối với các nhà phát triển ứng dụng WebRTC.

1.2.3 Ứng dụng WebRTC

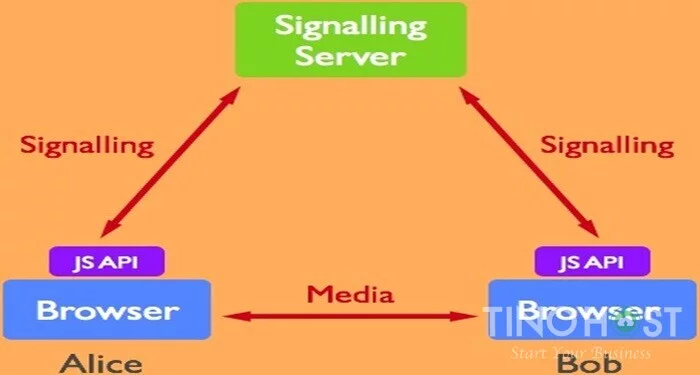
WebRTC là một phương thức kết nối phổ biến hiện nay, được áp dụng trong nhiều trường hợp thuộc các lĩnh vực khác nhau để giải quyết các vấn đề công nghệ. Dưới đây là các trường hợp cần sử dụng WebRTC:

* Tạo ứng dụng trực tuyến như Skype hoặc Google Hangouts mà không cần phải cài đặt phần mềm trên máy tính hoặc thiết bị di động.
* Tích hợp khả năng truyền dẫn trực tiếp vào một trang web hoặc ứng dụng web.
* Tạo hệ thống hỗ trợ trực tuyến cho các cuộc họp trực tuyến, hội thảo hoặc lớp học trực tuyến.
* Tạo hệ thống tư vấn trực tuyến cho các công ty hoặc tổ chức.
* Tích hợp khả năng truyền dẫn trực tiếp vào một ứng dụng di động hoặc trình duyệt di động.
* Tạo ra các ứng dụng trò chơi trực tuyến với khả năng truyền dữ liệu nhanh chóng và truyền tải âm thanh, hình ảnh và video mượt mà giữa các người chơi.
* Tích hợp tính năng gọi điện thoại hoặc video call trực tiếp vào các ứng dụng di động như ứng dụng nhắn tin.
* Tạo ra các ứng dụng liên quan đến IoT (Internet of Things) để truyền tải dữ liệu giữa các thiết bị thông minh như máy giặt, tủ lạnh, đồng hồ thông minh, v.v.
* Tích hợp khả năng chia sẻ màn hình và trình chiếu vào các ứng dụng trực tuyến như các phần mềm hội nghị trực tuyến hoặc hệ thống hỗ trợ khách hàng.
* Tạo ra các ứng dụng trực tuyến để truyền tải âm thanh và video trực tiếp cho các sự kiện trực tuyến như các buổi hòa nhạc, triển lãm, v.v.

1.3 Kiến trúc của WebRTC.

1.3.1 Sơ đồ kết nối giữa Client

* Kiến trúc của WebRTC khá đơn giản, nó bao gồm các Client kết nối với nhau thông qua Server Signalling. Cụ thể sơ hồ được trình bày ngay hình bên dưới đây:



Hình 1. 2 Kiếm trúc web RTC

1.3.2 Các phần chính của WebRTC

* WebRTC cung cấp ba thành phần cơ bản để thực hiện truyền tải âm thanh và hình ảnh trực tuyến: MediaStream, RTCDataChannel và RTCPeerConnection.
* MediaStream là một stream dữ liệu âm thanh và hình ảnh, được khởi tạo bằng cách sử dụng hàm getUserMedia trong trình duyệt để truy cập vào camera và/hoặc microphone. Khi thiết lập kết nối WebRTC với một máy tính khác, MediaStream cho phép truy cập vào stream của máy tính đó.
* RTCDataChannel là một kênh hai chiều cho phép trao đổi dữ liệu thời gian thực peer-to-peer. Nó trao đổi các dữ liệu dạng text, chia sẻ tệp P2P và các loại dữ liệu khác mà không chứa các dữ liệu nghe nhìn. RTCDataChannel đảm bảo trao đổi thông tin an toàn, đáng tin và nhanh chóng.
* RTCPeerConnection là API quan trọng giúp kết nối MediaStream và RTCDataChannel trở thành WebRTC. Nó được sử dụng để cài đặt các cuộc gọi video và voice call cho việc truyền tải. RTCPeerConnection cung cấp các phương thức để kết nối, duy trì kết nối và đóng kết nối khi không còn cần thiết sử dụng. Với ba thành phần này, WebRTC cung cấp một giải pháp toàn diện cho truyền tải âm thanh và hình ảnh trực tuyến.



Hình 1. 3 Kiếm trúc webRTC

1.4 Các công cụ và kỹ thuật được sử dụng trong đề tài WebRTC.

* Sử dụng Node.js và Express JS framework để xây dựng một web service.
* Axios: Được sử dụng để thực hiện các yêu cầu HTTP tới API Metered.
* Websockets: Cho phép kết nối thường xuyên giữa các ứng dụng trình khách và trình chủ. Được sử dụng để gửi các sự kiện như phát sóng âm thanh, video từ phòng họp đến người dùng tham gia.
* Web Audio API: Cung cấp khả năng phát và ghi âm thanh trong trình duyệt web. Được sử dụng để truyền dữ liệu âm thanh từ phòng họp đến người dùng.
* MediaStream API: Cho phép web ứng dụng truy cập vào phương tiện phát và ghi hình từ camera và microphone của thiết bị sử dụng. Được sử dụng để lấy truyền dữ liệu video từ người dùng tham gia.
* Rest API: Cho phép ứng dụng gửi và nhận yêu cầu HTTP để lưu trữ và phân phối dữ liệu từ một máy chủ API. Được sử dụng để tạo phòng họp, kiểm tra phòng họp tồn tại ...

1.5 Hạn chế và hướng phát triển thêm trong tương lai.

* Hạn chế
  + Độ trễ: Truyền tải video và âm thanh trực tiếp giữa các trình duyệt và thiết bị di động thông qua WebRTC có thể gặp phải vấn đề độ trễ. Điều này có thể ảnh hưởng đến trải nghiệm người dùng và làm cho các cuộc gọi trở nên khó chịu.
  + Kết nối Internet: WebRTC đòi hỏi kết nối Internet ổn định để truyền tải dữ liệu một cách mượt mà. Nếu kết nối Internet không ổn định, chất lượng video và âm thanh có thể bị giảm.
* Hướng phát triển thêm trong tương lai:
  + Cải thiện độ trễ: Các nhà phát triển có thể tìm cách giảm độ trễ bằng cách sử dụng các giải pháp tối ưu hóa mạng như cân bằng tải, đa đường truyền và giảm thiểu độ trễ.
  + Tăng cường tính bảo mật: Các nhà phát triển có thể tăng cường tính bảo mật của ứng dụng bằng cách sử dụng các tính năng mã hóa và xác thực người dùng.
  + Tích hợp các tính năng mới: Các nhà phát triển có thể tích hợp các tính năng mới như chia sẻ màn hình, trình chiếu và tích hợp IoT để cải thiện trải nghiệm người dùng.
  + Tích hợp trí tuệ nhân tạo: Các nhà phát triển có thể tích hợp trí tuệ nhân tạo để cải thiện trải nghiệm người dùng và tối ưu hóa chất lượng video và âm thanh.

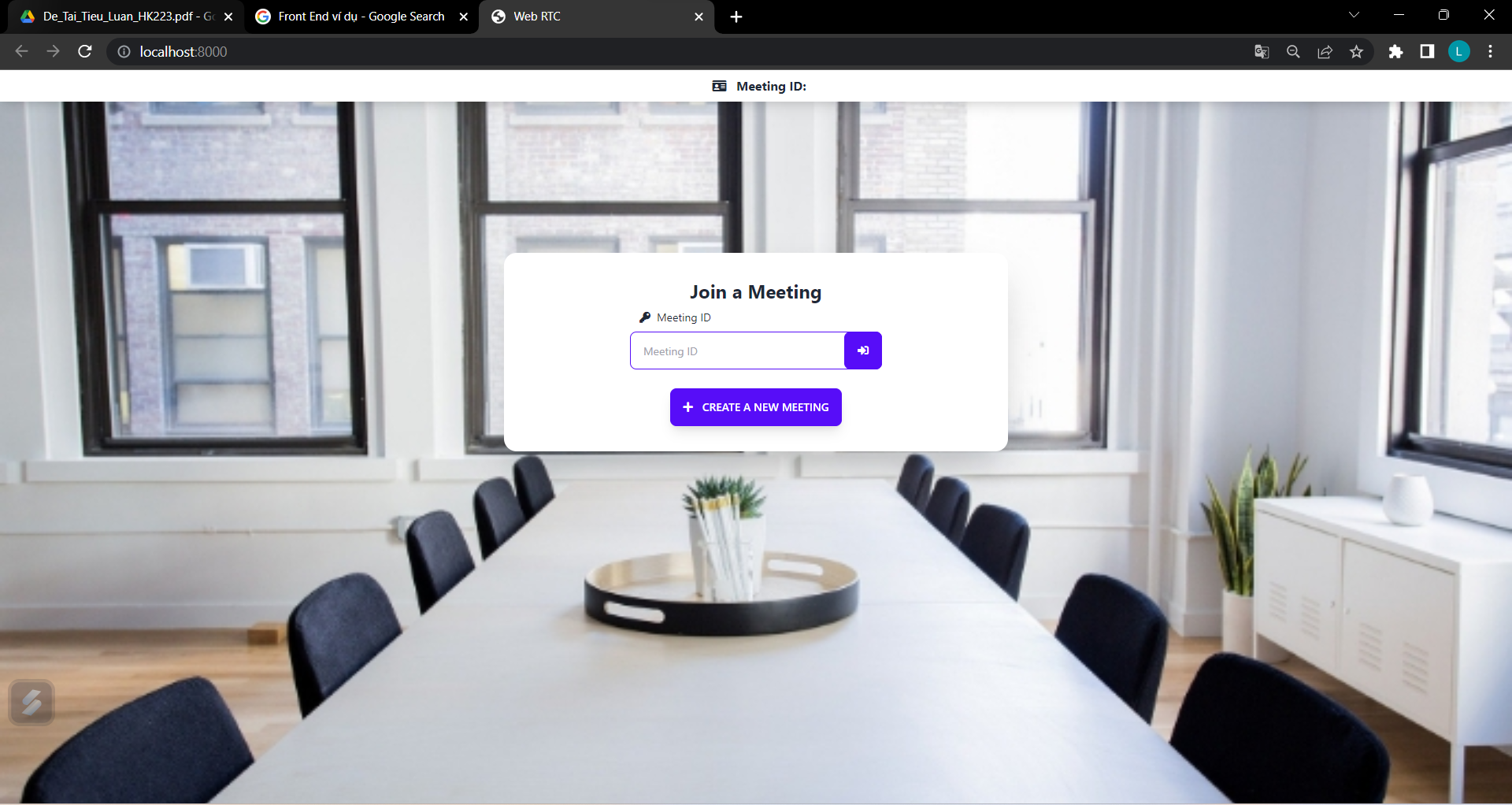
CHƯƠNG 2 – LÝ THUYẾT

2.1 Câu 1: Trình bày front-end, back-end và ví dụ.

Front-end là phần giao diện người dùng của một ứng dụng web, bao gồm các thành phần mà người dùng thấy và tương tác trực tiếp, chẳng hạn như nút bấm, trường nhập liệu, hình ảnh, v.v. Front-end thường được viết bằng các ngôn ngữ web như HTML, CSS và JavaScript.

Back-end là phần của ứng dụng web chạy trên máy chủ, xử lý các yêu cầu được gửi từ Front-end và cung cấp các dữ liệu và thông tin cần thiết cho Front-end. Back-end thường được viết bằng các ngôn ngữ lập trình như PHP, Python, Ruby, v.v.

Ví dụ: Khi bạn truy cập vào một trang web, trình duyệt của bạn hiển thị phần giao diện người dùng (Front-end) của trang đó.



Hình 2. 1 Ví dụ vể Front-end

Khi bạn nhấp vào nút "Create a new metting", thông tin đó được gửi đến máy chủ của trang web để xử lý (Back-end).

Text

Description automatically generated

Hình 2. 2 Ví dụ về Back-end

Sau đó, máy chủ trả về kết quả cho Front-end một phòng họp để hiển thị cho người dùng.

2.2 Câu 2: Nodejs và front-end, back-end.

NodeJS là một nền tảng lập trình được xây dựng trên nền tảng JavaScript, cho phép lập trình viên sử dụng cùng một ngôn ngữ để xây dựng cả Front-end và Back-end cho các ứng dụng web. Tuy nhiên, NodeJS thường được sử dụng chủ yếu để xây dựng Back-end, vì nó cung cấp các tính năng hỗ trợ cho việc xử lý yêu cầu và phản hồi trên máy chủ.

A screenshot of a computer

Description automatically generated with medium confidence

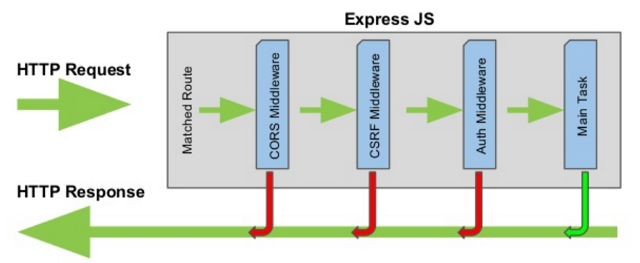
Hình 2. 3 ví dụ về xử lý các yêu cùa của back-end

Việc sử dụng NodeJS để xây dựng Back-end giúp lập trình viên có thể sử dụng các thư viện và công cụ hỗ trợ để xử lý và lưu trữ dữ liệu, xử lý các yêu cầu từ phía Front-end và tạo ra các API để cung cấp dữ liệu cho các ứng dụng khác. Điều này giúp cho việc phát triển ứng dụng web trở nên dễ dàng hơn và giảm thiểu thời gian và chi phí cho việc phát triển.

2.3 Câu 3: Expressjs và front-end, back-end.

ExpressJS là một framework phổ biến được sử dụng để xây dựng Back-end cho các ứng dụng web. Tuy nhiên, ExpressJS cũng có thể được sử dụng để xây dựng Front-end, nhưng điều này không phải là mục đích chính của nó.

ExpressJS cung cấp các tính năng hỗ trợ cho việc xử lý yêu cầu và phản hồi trên máy chủ, quản lý các API, sử dụng các thư viện cho việc xử lý dữ liệu, và nhiều tính năng khác để phục vụ cho việc xây dựng Back-end của ứng dụng web. ExpressJS cũng hỗ trợ các thư viện middleware để tăng cường tính bảo mật, nâng cao hiệu suất và cải thiện trải nghiệm người dùng.



Hình 2. 4 Ví dụ về ExpressJS

Tuy nhiên, ExpressJS không phải là một công cụ lý tưởng để xây dựng Front-end. Để xây dựng Front-end, thường sử dụng các thư viện và framework như ReactJS, AngularJS, VueJS, hoặc các công cụ khác để tạo ra các giao diện người dùng tương tác và trực quan. Các thư viện và framework này cung cấp các tính năng hỗ trợ để xử lý các sự kiện người dùng, tương tác với các API Back-end, và quản lý trạng thái của các thành phần trên trang web.

2.4 Câu 4: Làm rõ về expressjs.

ExpressJS là một framework cho Back-end của ứng dụng web, không phải là ngôn ngữ lập trình hay công cụ xây dựng web. ExpressJS là một framework mã nguồn mở do được phát triển và phát hành dưới giấy phép mã nguồn mở MIT. Nó được xây dựng trên nền tảng NodeJS và cung cấp các tính năng hữu ích cho việc xây dựng các ứng dụng web.



Hình 2. 5 ExpressJS và Nodejs

Bất kỳ lập trình viên nào cũng có thể tùy chỉnh và phát triển các tính năng mới cho ExpressJS phù hợp với nhu cầu của mình. Đồng thời, cộng đồng lập trình viên sử dụng ExpressJS cũng đóng góp vào việc cải thiện và phát triển framework.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. <https://bkhost.vn/blog/webrtc/#khi_nao_can_su_dung_webrtc>.
2. <https://wiki.tino.org/webrtc-la-gi/#ftoc-heading-4>.

**Tiếng Anh**