

TRƯỜNG ĐẠI HỌC QUỐC TẾ MIỀN ĐÔNG

KHOA: KỸ THUẬT

CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM



BÁO CÁO ĐỒ ÁN THỰC TẬP

TỐT NGHIỆP

**ĐỀ TÀI: WEBSITE HỖ TRỢ QUẢN LÝ VÀ CHẤM ĐIỂM ĐỒ
ÁN**

Sinh viên thực hiện: Đàm Đức Duy - 1331209041

Nguyễn Quang Huy - 1331200013

Huỳnh Minh Quân – 1331200025

Giáo viên hướng dẫn: ThS. Tất Quảng Phát

Bình Dương, 11/2017

[illegible]

Lời nói đầu

Tại các trường đại học ở Việt Nam, việc đăng kí và chấm điểm những loại đồ án khác nhau vô cùng phức tạp, khó thống kê và đòi hỏi rất nhiều kỹ năng của người quản lý. Ngoài ra, từ trước đến nay, việc quản lý và chấm điểm chủ yếu sử dụng những phương pháp thủ công. Do đó, khi quản lý hay tìm kiếm mất rất nhiều thời gian và công sức mà hiệu quả công việc đem lại không cao, đôi khi còn xảy ra sai sót, mất mát dữ liệu không đáng có. Từ những điều trên, việc đòi hỏi phải có một hệ thống chuyên dụng trợ giúp cho công việc quản lý và chấm điểm đồ án của sinh viên là một nhu cầu tất yếu để đảm bảo hiệu quả và tiết kiệm thời gian trong công việc.

Nhận biết được những khó khăn và nhu cầu hiện nay, chúng em đã nghiên cứu, tìm hiểu và phát triển “Website hỗ trợ quản lý và chấm điểm đồ án” trên nền tảng ASP.NET Core 2.0 kết hợp với framework Angular 2 và thiết kế theo tiêu chuẩn RESTful API.

Để hoàn thành được đồ án này, em xin chân thành cảm ơn thầy Tats Quảng Phát, người đã tận tình hướng dẫn, chỉ bảo, động viên và hỗ trợ chúng em trong suốt quá trình thực hiện đồ án này. Mặc dù đã rất cố gắng nhưng do hạn chế về thời gian và trình độ nên đồ án không tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm và đóng góp ý kiến của Thầy và các bạn.

Tp. Bình Dương, tháng 11, năm 2017

Sinh viên thực hiện

Đàm Đức Duy

Huỳnh Minh Quân

Nguyễn Quang Huy

Mục lục

CHƯƠNG I: GIỚI THIỆU VỀ ASP.NET CORE 2.0.....	6
1. ASP.NET CORE LÀ GÌ?	6
2. ASP.NET CORE 2.0.....	7
CHƯƠNG II: GIỚI THIỆU VỀ ANGULAR 2.....	9
1. ANGULAR 2 LÀ GÌ?.....	9
2. ĐẶC TÍNH CỦA ANGULAR.....	10
3. ĐIỂM MẠNH CỦA ANGULAR 2	10
4. SO SÁNH ANGULAR 2 VỚI NHỮNG FRAMEWORK KHÁC.....	11
CHƯƠNG III: GIỚI THIỆU VỀ RESTFUL API	13
1. RESTFUL API LÀ GÌ?	13
2. RÀNG BUỘC VÀ QUY ƯỚC	13
CHƯƠNG IV: MỘT SỐ CÔNG NGHỆ ĐƯỢC SỬ DỤNG.....	15
1. JSON WEB TOKEN.....	15
2. SIGNALR	18
3. GZIP.....	19
4. GITHUB API – STATISTICS.....	21
CHƯƠNG V: PHÂN TÍCH CHỨC NĂNG, HỆ THỐNG VÀ CƠ SỞ DỮ LIỆU CỦA WEBSITE.....	22
1. PHÂN TÍCH TÌNH HÌNH VÀ ĐƯA RA GIẢI PHÁP.....	22
2. GIỚI THIỆU VỀ QUY TRÌNH HOẠT ĐỘNG CỦA WEBSITE	22
3. NHỮNG CHỨC NĂNG CHÍNH CỦA WEBSITE.....	23
4. MÔ HÌNH CƠ SỞ DỮ LIỆU	25
CHƯƠNG VI: KẾT QUẢ ĐẠT ĐƯỢC	26
3. GIAO DIỆN TRANG ĐĂNG NHẬP.....	26
4. ĐĂNG KÝ ĐỒ ÁN VÀ CHẤP NHẬN ĐƠN ĐĂNG KÝ.....	29
5. TẠO NHÓM CHO SINH VIÊN VÀ GIAO DIỆN CỦA NHÓM.....	30
6. TẠO HỘI ĐỒNG CHẤM ĐIỂM CHO NHÓM.....	31
7. THÊM SINH VIÊN, GIẢNG VIÊN VÀ ĐỒ ÁN VÀO HỆ THỐNG.....	32
CHƯƠNG VII: TỔNG KẾT	34
1. CHỨC NĂNG ĐẠT ĐƯỢC.....	34
2. NHỮNG MẶT HẠN CHẾ	34
3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI.....	34
CHƯƠNG VIII: TÀI LIỆU THAM KHẢO.....	35

Chú thích hình ảnh

Hình 1: .NET Core 2.0	7
Hình 2: Angular 2.0	9
Hình 3: Mô hình Scale của Restful API	14
Hình 4: Kiến trúc của JWT	15
Hình 5: So sánh JWT và Session.....	16
Hình 6: Access token trên browser.....	17
Hình 7: Mô hình của SignalR	18
Hình 8: Trước khi sử dụng Gzip.....	19
Hình 9: Sau khi sử dụng Gzip	20
Hình 10: Giao diện khi sử dụng Github API.....	21
Hình 11: Quy trình hoạt động của website.....	22
Hình 12: Biểu đồ Usecase của website.....	24
Hình 13: Mô hình cơ sở dữ liệu của website.....	25
Hình 14: Giao diện Login của website	26
Hình 15: Giao diện của Admin.....	27
Hình 16: Giao diện của giảng viên	27
Hình 17: Giao diện của sinh viên	28
Hình 18: Giao diện của đơn đăng ký	29
Hình 19: Giao diện xem xét đơn đăng ký của giảng viên	29
Hình 20: Giao diện tạo nhóm	30
Hình 21: Giao diện thông tin của nhóm	30
Hình 22: Giao diện tạo hội đồng chấm điểm.....	31
Hình 23: Giao diện thêm sinh viên.....	32
Hình 24: Giao diện thêm đề án.....	33
Hình 25: Giao diện thêm giảng viên.....	33

CHƯƠNG I: GIỚI THIỆU VỀ ASP.NET CORE 2.0

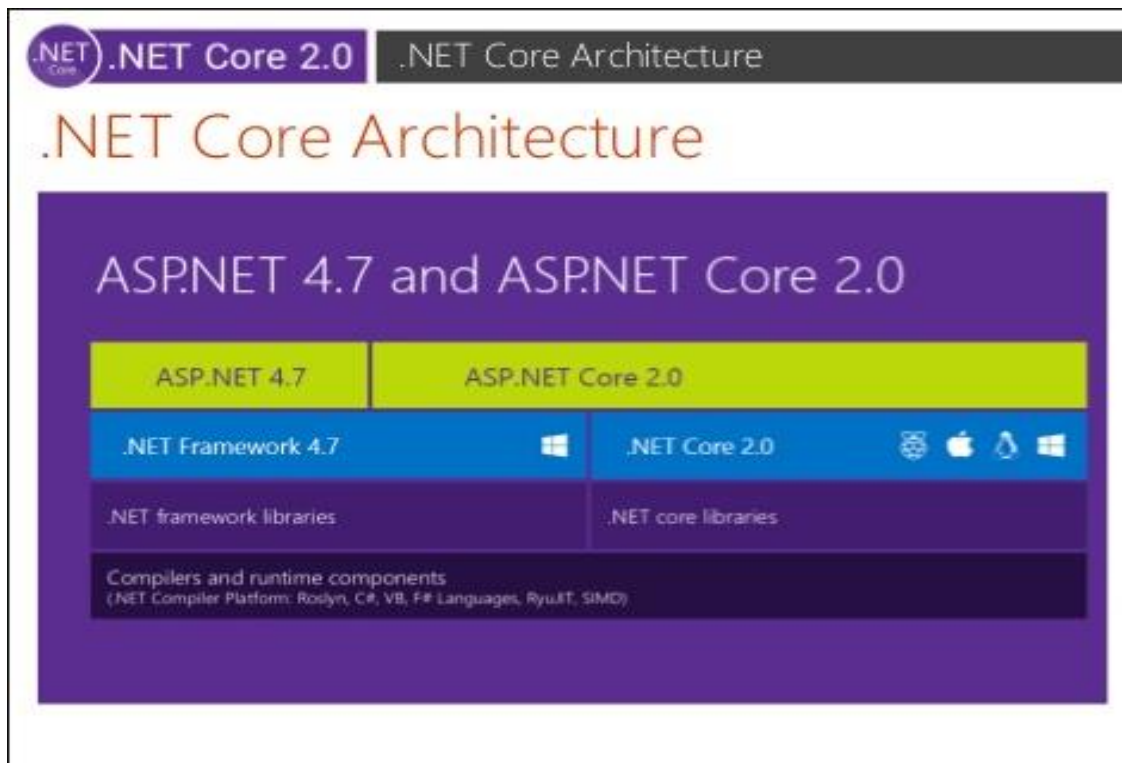
1. ASP.NET CORE LÀ GÌ?

ASP.NET Core là một nền tảng mã nguồn mở, cross-platform framework cho việc xây dựng và kết nối các ứng dụng web hiện đại chẳng hạn như ứng dụng web, Internet of Thing, Mobile Backend - Web API. ASP.NET Core cung cấp một kiến trúc để tối ưu hóa việc xây dựng các ứng dụng đám mây (microsoft azure) hoặc các ứng dụng độc lập. Bạn có thể phát triển ứng dụng ASP.NET Core đa nền tảng (Window, Mac, Linux), với mã nguồn mở tại Github.

Khả năng chạy trên đa nền tảng tuyệt vời của ASP .Net Core là không bàn cãi. Bên cạnh đó, ưu điểm về tốc độ và các tính năng mới của ASP .NET core cũng đáng để lưu ý như:

- HTTP handler gọn nhẹ hơn
- Modular HTTP request pipeline thông qua Middle-Ware
- Tích hợp sẵn DI (Dependency Injection)
- Razor view engine hỗ trợ Tag Helpers
- Cấu trúc Project có thay đổi. Thay vì cấu hình thông số của project qua tập trình *.csproj. Thì .NET Core cho phép cấu hình qua project.json sẽ giúp dễ đọc và thao tác hơn.
- Chạy trên nhiều hệ điều hành
- Mã nguồn mở

2. ASP.NET CORE 2.0



Hình 1: .NET Core 2.0

Microsoft vừa cho ra mắt phiên bản cải tiến mới của ASP.NET core. Ở phiên bản mới là ASP.NET Core 2.0 không chỉ tốc độ nhanh hơn so với phiên bản 1.x mà nó cũng có một số tính năng mới.

ASP.NET Core 2.0 có gì mới

- .NET Standard mở rộng thêm các API và khả năng sử dụng đầy đủ các thư viện: hỗ trợ đến 32.000 API. Việc này giúp cho việc nâng cấp code có sẵn từ phiên bản .NET cũ mà không cần phải thay đổi nhiều code.

- Hỗ trợ cú pháp Razor trên các view: hỗ trợ tạo ra một razor view mà không cần tạo một controller. Mặc dù có thể tạo ra 1 class kế thừa từ PageModel, nó cơ bản là một controller và class model riêng biệt cho page đó. Việc này vô cùng hữu ích trong một số trường hợp và có thể giúp tách nhỏ các controller phức tạp.

-Mở rộng hỗ trợ nhiều hệ điều hành khác nhau: Linux giờ đây được xem như một hệ điều hành riêng biệt (không phân biệt theo từng phiên bản phân phối như trước). Chuyển tương tự với Windows và macOS.

-ASP.NET Core metapackage: một metapackage mới bao gồm những package được tạo ra và hỗ trợ bởi đội ASP.NET Core và đội Entity Framework Core. Do đó tất cả chức năng đều được tích hợp trong package `Microsoft.AspNetCore.All`.

-Cập nhật Configuration: IConfiguration được mặc định thêm vào ASP.NET Core 2.0 giúp lấy các giá trị cấu hình được dễ dàng hơn.

-Cập nhật Authentication: một mô hình xác thực mới hỗ trợ cho những ứng dụng sử dụng DI. Ngoài ra, hỗ trợ những template cấu hình chứng thực cho Web App và Web API sử dụng Azure AD B2C.

-Cập nhật Identity: tạo sự dễ dàng khi xây dựng Web API sử dụng Identity. Chúng ta có thể lấy token để truy cập Web API bằng cách sử dụng Microsoft Authentication Library (MSAL).

-SPA template: hỗ trợ những template của Angular và React, Aurelia và knockout.

-Nâng cấp kestrel: giờ đây, chúng ta có thể thêm những giới hạn như: số lượng kết nối tối đa, dung lượng tối đa của request và dung lượng tối thiểu của yêu cầu trong class `KestrelServerOptions`.

-Tự động sử dụng anti-forgery tokens: ASP.NET Core mặc định giúp html mã hóa nội dung, nhưng với phiên bản mới này, ASP.NET Core sẽ sử dụng anti-forgery tokens và xác nhận lại trên các thao tác POST và các trang không có cấu hình bổ sung. Việc này nhằm tránh các cuộc tấn công cross-site request forgery (XSRF).

-Automatic precompilation: giảm dung lượng khi publish và thời gian khởi động.

CHƯƠNG II: GIỚI THIỆU VỀ ANGULAR 2

1. ANGULAR 2 LÀ GÌ?

Angular 2 là 1 framework UI phát triển trên nền JavaScript của Google để xây dựng những ứng dụng Web trên Desktop và Android. Kế thừa các đặc điểm của AngularJS và phát triển một phương thức tiếp cận việc xây dựng ứng dụng hoàn toàn mới, phương pháp hướng Component.

Nó được xây dựng dựa trên Javascript. Chúng ta có thể dùng nó để xây dựng 1 ứng dụng client side thú vị dùng HTML, CSS và Javascript.



Hình 2: Angular 2.0

Angular 2 có rất nhiều cải tiến so với Angular 1 để dễ dàng học và phát triển các ứng dụng quy mô doanh nghiệp.

Với Angular 2 thì chúng ta dễ dàng xây dựng được 1 ứng dụng có thể dễ dàng mở rộng, bảo trì, kiểm nghiệm và chuẩn hóa ứng dụng của mình.

2. ĐẶC TÍNH CỦA ANGULAR

- Angular 2 là nhanh hơn và dễ dàng hơn so với Angular 1.
- Nó hỗ trợ tất các phiên bản của trình duyệt và cũng hỗ trợ các trình duyệt cũ bao gồm IE9 + và Android 4.1 trở lên.
- Nó là một framework đa nền tảng.
- Angular 2 chủ yếu tập trung vào các ứng dụng di động.
- Cấu trúc mã là rất đơn giản hơn so với phiên bản trước.
- Typing sử dụng TypeScript, Dart, JavaScript

3. ĐIỂM MẠNH CỦA ANGULAR 2

- Phát triển trên nền tảng hiện đại hơn, nhiều tính năng, và dễ dàng cho người mới học hơn là Angular 1.x và cũng dễ dàng hơn cho các lập trình viên lâu năm.
- Mặc dù được viết lại hoàn toàn, Angular2 vẫn giữ lại các khái niệm cốt lõi và các quy tắc của Angular 1.x. Điều đó có nghĩa là các lập trình viên giỏi với Angular1 sẽ dễ dàng chuyển đổi sang Angular2 hơn là các thư viện khác như ReactJS....
- Angular2 được viết với TypeScript, một superset của JavaScript, nó triển khai nhiều tính năng của ES2016 và các tính năng của TypeScript.
- Angular2 được thiết kế cho mobile từ ban đầu. Bỏ qua một bên các vấn đề về sức mạnh xử lý, các thiết bị di động có các tính năng khác và sự giới hạn đó phân biệt chúng với máy tính truyền thống. Giao diện chạm, màn hình nhỏ, phần cứng đã được tính toán trong Angular2. Máy tính để bàn cũng sẽ được cải tiến về hiệu suất và khả năng đáp ứng.

- Sử dụng hệ thống module ECMA Script 6 (ES6), và các công cụ đóng gói hiện đại như Webpack hoặc SystemJS. Module được tùy biến theo cách riêng của Angular để dễ dàng viết các Generic cho JS để cắm vào Angular. Hệ thống module mới giúp dễ dàng phát triển các dự án lớn hiệu quả hơn.

4. SO SÁNH ANGULAR 2 VỚI NHỮNG FRAMEWORK KHÁC

So sánh với phiên bản trước

	Angular 1.x	Angular 2
Scope lồng nhau	Tránh sử dụng	Đã bỏ
Directives và Controllers	Sử dụng kết hợp	Component directives
Controller và triển khai service	Functions	ES6 classes
Hệ thống module	Angular's modules	ES6 modules
Yêu cần dịch	Không	TypeScript

So sánh với các framework khác

Framework	Angular	React	Ember
Định nghĩa	Superheroic JavaScript MVW Framework	Một thư viện Javascript dùng xây dựng giao diện người dùng và nhiều hơn thế nữa	Một framework để tạo một ứng dụng web cạnh tranh.

Ra mắt	Được ra mắt vào năm 2009 bởi Miško Hevery	Được tạo ra bởi Jordan Walke, bắt nguồn từ năm 2013	Ban đầu được tạo ra SproutCore năm 2007 bởi Yehuda Katz, được Facebook mua lại và đổi tên thành EmberJS vào năm 2011
Homepage	angularjs.org	reactjs.net	emberjs.com
Github	https://github.com/angular/angular.js	https://github.com/facebook/react	https://github.com/emberjs/ember.js
Bug report	https://github.com/angular/angular.js/issues	https://github.com/facebook/react/issues	https://github.com/node-ember
Licenses	MIT	MIT	BSD-3-Clause
Những website nổi tiếng sử dụng	Youtube, Vevo, Freelancer, Istockphoto, Weather, Sky Store	Facebook, Instagram, Khan Academy, New York Times, Airbnb, Flipkart, Sony Lifelog	Apple Music, Yahoo!, LinkedIn, TinderBox, Netflix, Groupon
Lý tưởng cho	Xây dựng các ứng dụng web tương tác và chủ động.	Các ứng dụng web lớn có hiển thị dữ liệu thay đổi thường xuyên	Dynamic SPAs

CHƯƠNG III: GIỚI THIỆU VỀ RESTFUL API

1. RESTFUL API LÀ GÌ?

Những khái niệm đầu tiên về REST (**RE**presentational **State** **T**ransfer) được đưa ra vào năm 2000 trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Trong luận văn ông giới thiệu khá chi tiết về các ràng buộc, quy ước cũng như cách thức thực hiện với hệ thống để có được một hệ thống REST.

Hiểu đơn giản nó là một bộ các ràng buộc và quy ước, khi áp dụng đầy đủ vào hệ thống của bạn thì ta có 1 hệ thống REST.

2. RÀNG BUỘC VÀ QUY ƯỚC

Hệ thống hoạt động theo mô hình client-server, trong đó server là tập hợp các service nhỏ lắng nghe các request từ client. Với từng request khác nhau thì có thể một hoặc nhiều service xử lý.

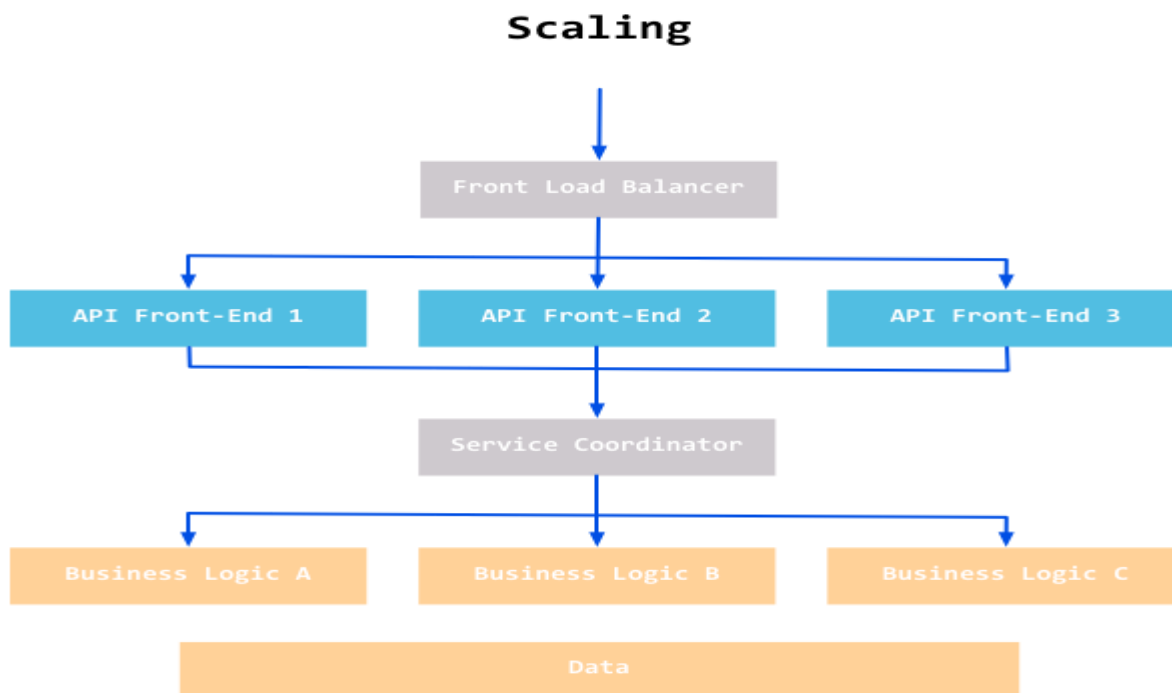
Stateless (phi trạng thái). Đơn giản server và client không lưu trạng thái của nhau dẫn theo mỗi request lên server thì client phải đóng gói thông tin đầy đủ để server hiểu được. Điều này giúp hệ thống của bạn dễ phát triển, bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client. Hệ thống phát triển theo hướng này có ưu điểm nhưng cũng có khuyết điểm là gia tăng lượng thông tin cần truyền tải giữa client và server.

Khả năng caching: Các response có thể lấy ra từ cache. Bằng cách cache các response, server giảm tải việc xử lý request, còn client cũng nhận được thông tin nhanh hơn.

Chuẩn hóa các interface: Đây là một trong những đặc tính quan trọng của hệ thống REST. Bằng cách tạo ra các quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống, bạn đã đơn giản hóa việc client có thể tương tác với server. Các quy ước này áp dụng cho toàn bộ các service giúp cho người sử dụng hệ thống của bạn dễ dụng hơn. Dễ hiểu hơn trên hệ

thông bạn đặt ra 1 chuẩn API để người dùng dù là mobile, web đều có thể kết nối vào được. Hệ thống REST có yếu điểm ở đây vì khi chuẩn hóa rồi ta không thể tối ưu từng kết nối.

Phân lớp hệ thống: trong hệ thống REST bạn chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp bạn giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần:



Hình 3: Mô hình Scale của Restful API

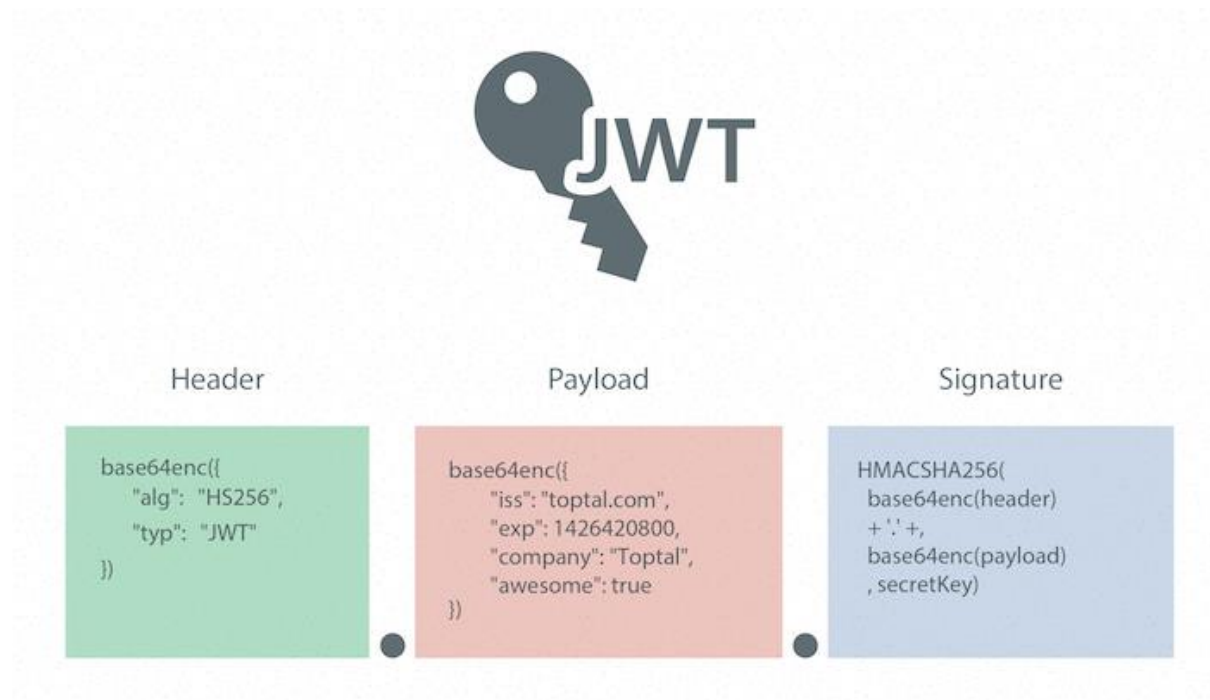
Trong đồ án này, chúng em sử dụng ASP.NET Core 2.0 để thực hiện, xử lý bên dưới server và Angular 2 để truy xuất dữ liệu, gửi những request từ phía client.

CHƯƠNG IV: MỘT SỐ CÔNG NGHỆ ĐƯỢC SỬ DỤNG

Để hoàn thành Website này, việc áp dụng những công nghệ mới hoặc công nghệ hỗ trợ đóng một phần vô cùng quan trọng. Sau đây sẽ là một số công nghệ, thủ thuật được sử dụng nhằm nâng cao chất lượng của hệ thống cũng như cải thiện tốc độ của Website.

1. JSON WEB TOKEN

JSON Web Token (JWT) là 1 tiêu chuẩn mở ([RFC 7519](#)) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng **HMAC** hoặc **RSA**.



Hình 4: Kiến trúc của JWT

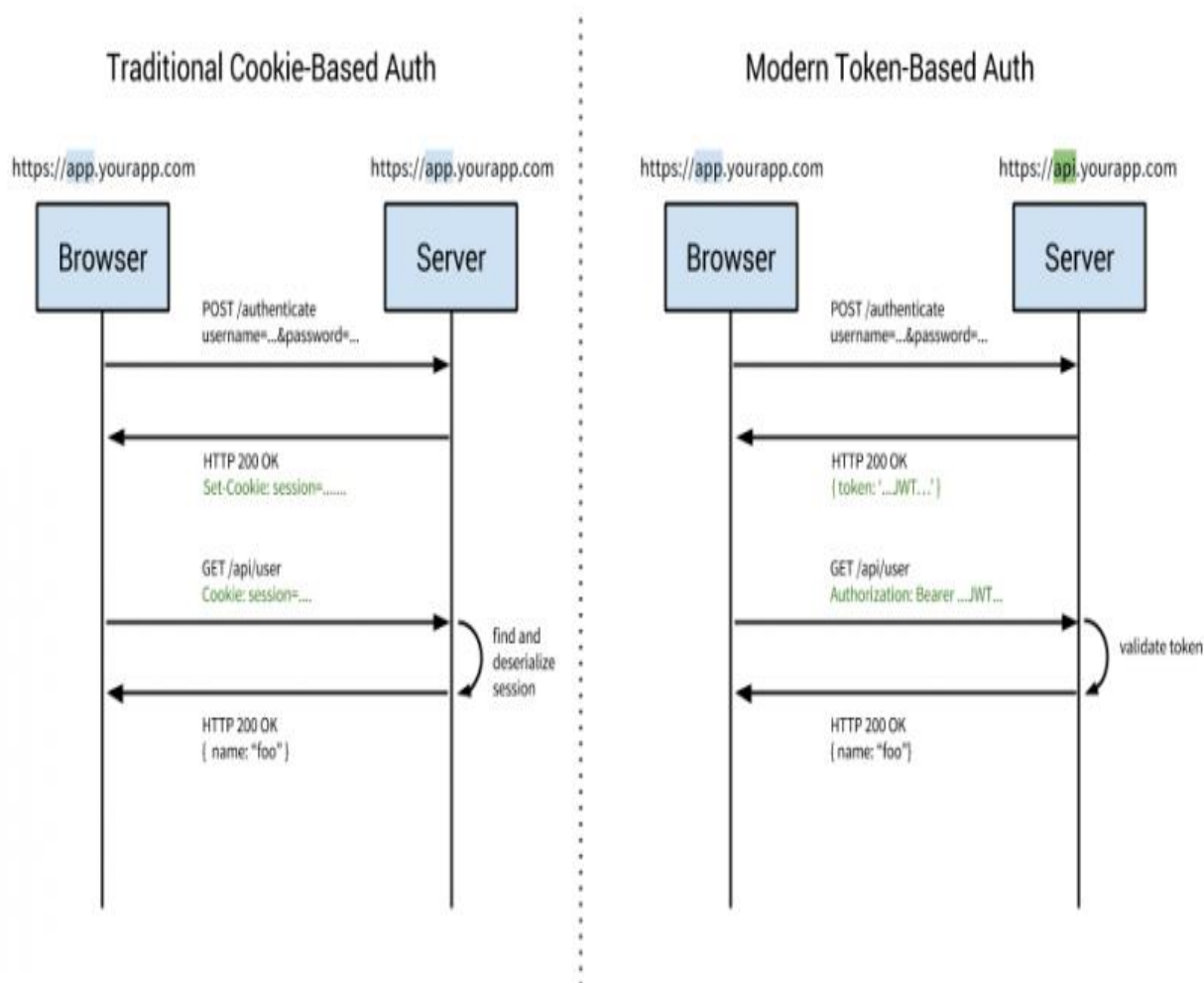
KỊCH BẢN DÀNH CHO JSON WEB TOKEN

Authentication: Đây là kịch bản phổ biến nhất cho việc sử dụng JWT. Một khi người dùng đã đăng nhập vào hệ thống thì những request tiếp theo từ phía người dùng sẽ chứa thêm mã JWT, cho phép người dùng quyền truy cập vào các đường dẫn, dịch vụ, và tài

nguyên mà cần phải có sự cho phép nếu có mã Token đó. Phương pháp này không bị ảnh hưởng bởi Cross-Origin Resource Sharing (CORS) do nó không sử dụng cookie.

Trao đổi thông tin: JSON Web Token là 1 cách thức không tồi để truyền tin an toàn giữa các thành viên với nhau, nhờ vào phần "chữ ký" của nó. Phía người nhận có thể biết được người gửi là ai thông qua phần chữ ký. Ngoài ra, chữ ký được tạo ra bằng việc kết hợp cả phần header, payload lại nên thông qua đó ta có thể xác nhận được chữ ký có bị giả mạo hay không.

JWTs VS SESSION



Hình 5: So sánh JWT và Session

ÁP DỤNG VÀO WEBSITE

Với mỗi lần người dùng đăng nhập vào website họ sẽ nhận được một access token. Mỗi request từ client phải kèm theo access token này thì mới có thể hoạt động được và truy cập được những dịch vụ ở server.

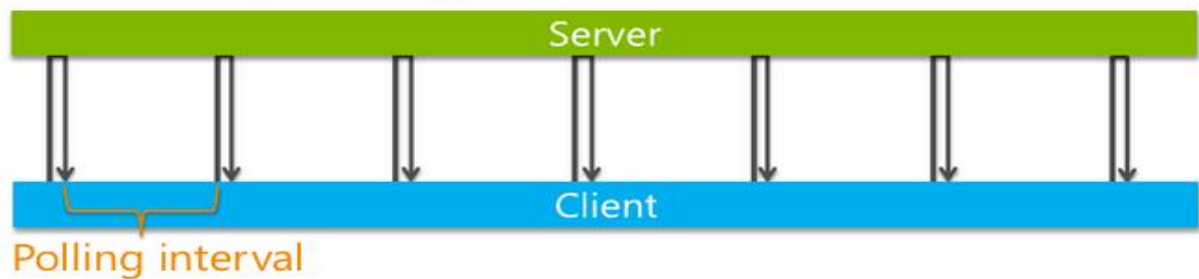


Hình 6: Access token trên browser

2. SIGNALR

Vào năm 2005, giới lập_trình_web làm quen với AJAX. Browser dùng JavaScript gửi request lên server. Server trả về và browser tự động cập nhật nội dung cần làm mới mà không cần phải reload lại cả trang web. Nếu một web site cần hiển thị trong thời gian thực hoặc gần như vậy đồ thị lên xuống của các mã cổ phiếu. Chúng sẽ nghĩ đến lập trình để browser đều đặn gửi yêu cầu lấy giá cổ phiếu mới nhất từ web server -> Cơ chế pooling. Nếu số lượng browser đồng thời cùng truy cập đến hàng nghìn, thì web server sẽ phải phục vụ rất vất vả cho dù chỉ có một vài mã cổ phiếu thay đổi điểm, những vẫn phải trả lời và gửi trả tất cả các mã cổ phiếu. Thay vì để client hỏi server (pooling) liên tục một cách mù quáng. Tại sao không để server nếu có thay đổi sẽ báo lại cho client, và sẽ chỉ gửi dữ liệu cần thiết để client xử lý (notification).

Periodic polling



Hình 7: Mô hình của SignalR

Ngày càng nhiều ứng dụng web cần áp dụng cơ chế này, ví dụ Twitter, LiveScore, Stock Trading, chat... SignalR là một thư viện lập trình do 2 thành viên của đội phát triển ASP.net Microsoft phát triển. SignalR cho phép lập trình viên tạo kênh liên lạc giữa server và client theo cơ chế notification.

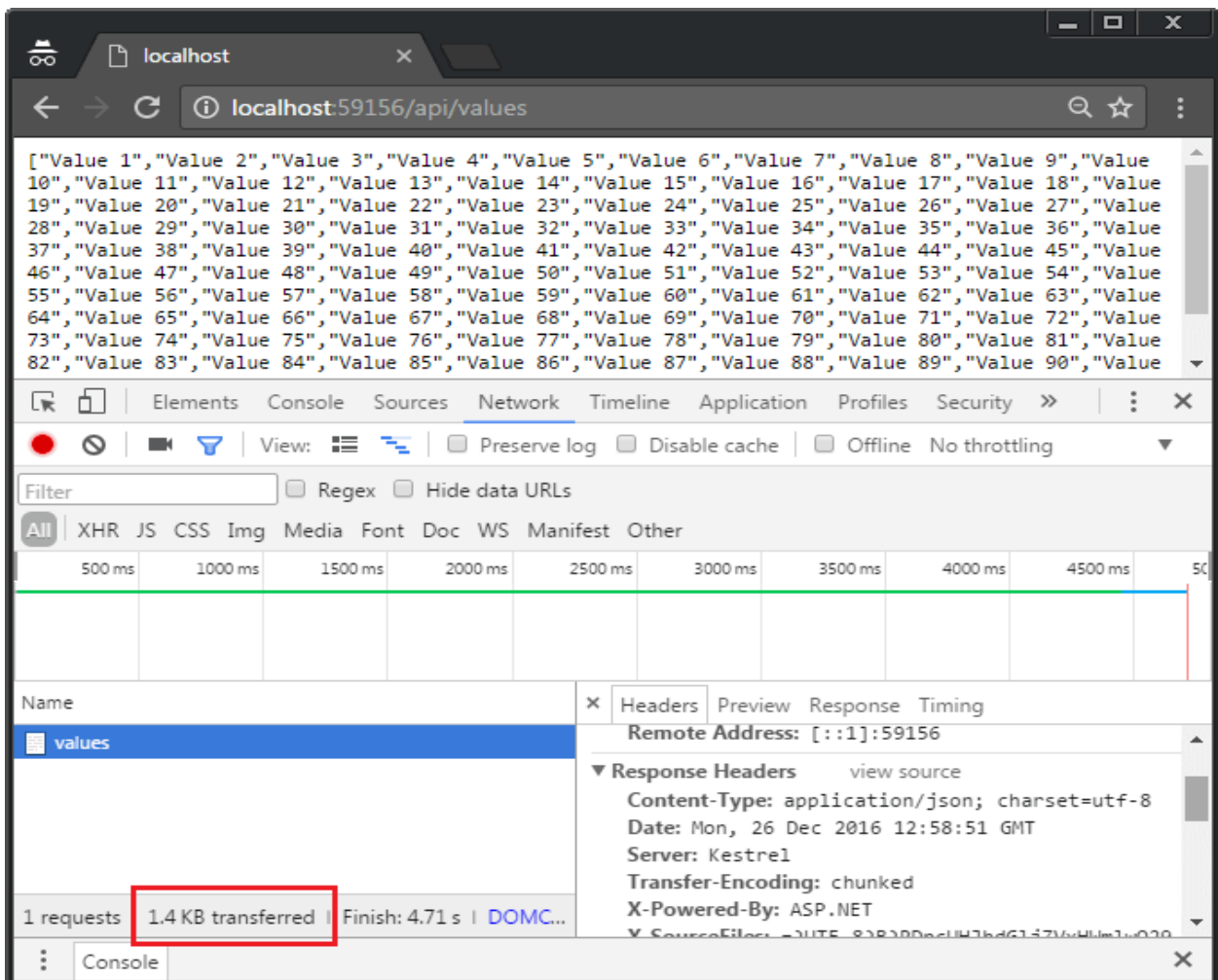
Trong đồ án này, chúng em đã sử dụng SignalR để hỗ trợ website có thể chạy realtime, cập nhật dữ liệu theo thời gian thực.

3. GZIP

Tất cả những trình duyệt hiện nay đều hỗ trợ nén response, vì vậy, tại sao lại không tận dụng tính năng này để giảm thiểu kích cỡ của những response. Việc này sẽ giúp giảm tải cho server và tăng tốc độ dữ liệu truyền từ server về client.

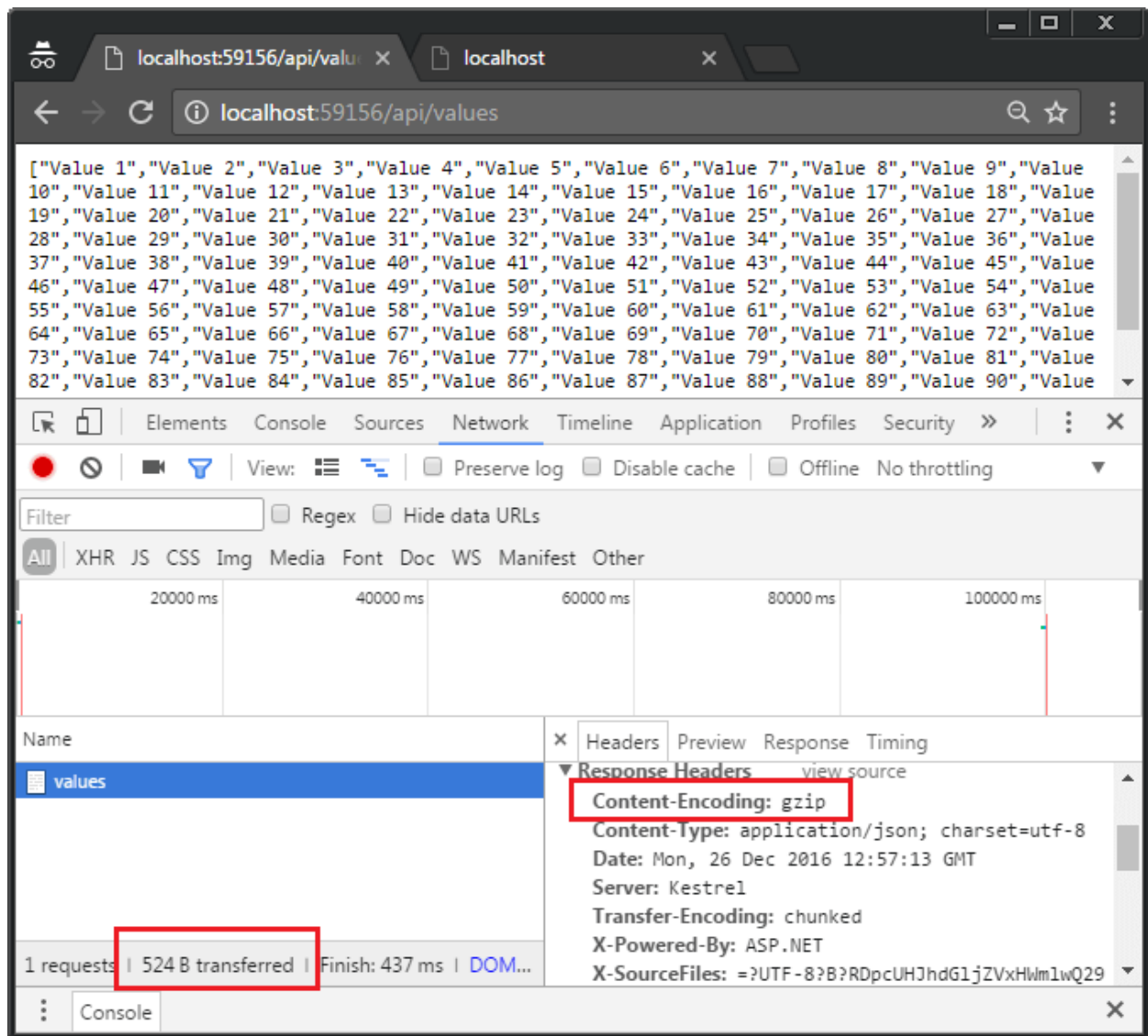
Đặc biệt hơn, trong ASP.NET Core có sẵn một thư viện có tên là `Microsoft.AspNetCore.ResponseCompression`

Dưới đây hai hình ảnh cho thấy sự khác biệt trong kích thước của response. Hình ảnh đầu tiên không có nén, trong đó kích thước phản hồi là 1,4 KB.



Hình 8: Trước khi sử dụng Gzip

Và đây là kết quả, sau khi sử dụng gzip. Kích thước phản hồi giảm xuống còn 524 byte. Và cũng có thể dễ dàng thấy được Content-Encoding được đặt thành gzip trong response.



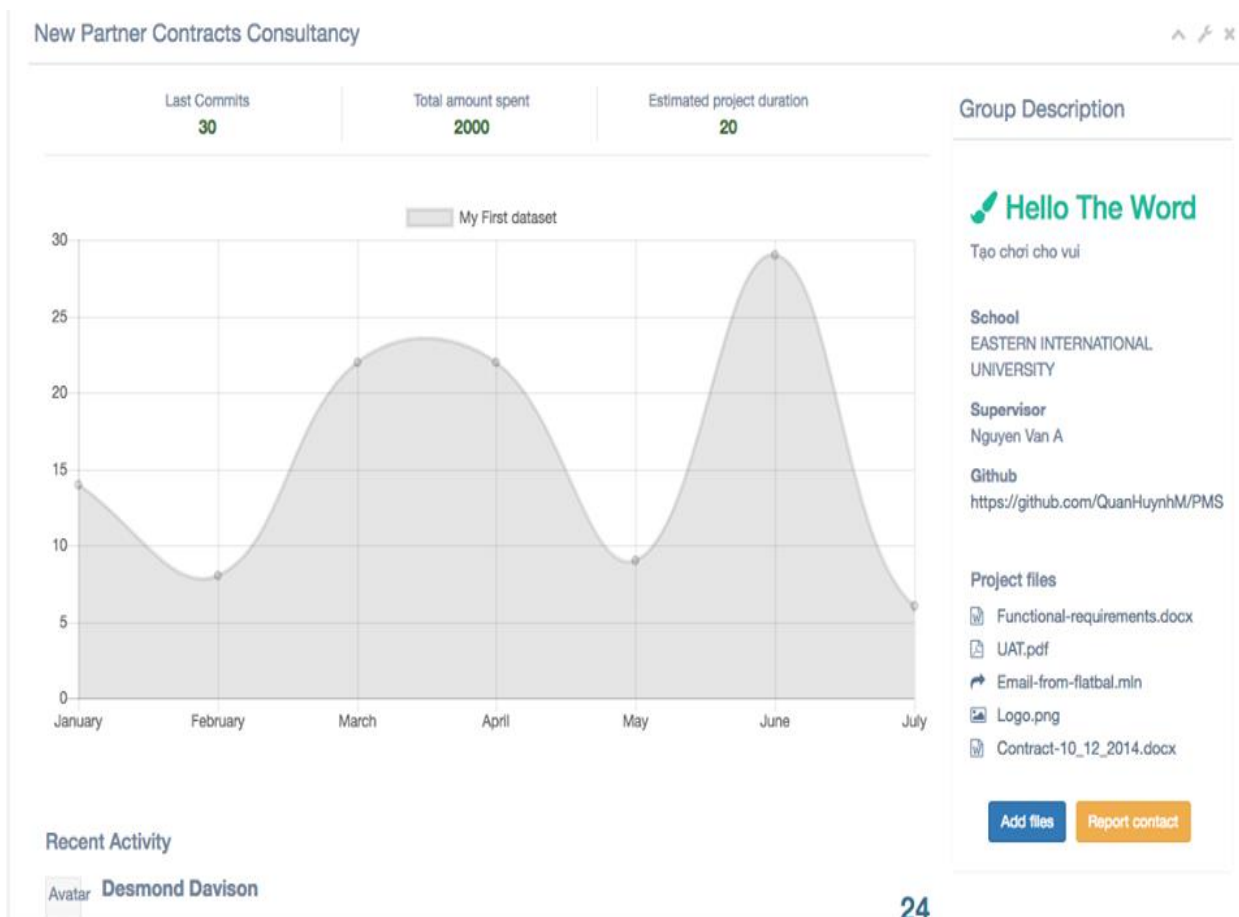
Hình 9: Sau khi sử dụng Gzip

4. GITHUB API – STATISTICS

Api này cho phép chúng ta tìm, khai thác dữ liệu của những repository nhằm mục đích dùng để hiển thị thành những loại đồ thị thống kê khác nhau. Những đồ thị này được gọi là Repository graph.

Repository graph hỗ trợ xem và phân tích dữ liệu cho mỗi repository bằng những thông tin như: lượng truy cập, những dự án trong mỗi repository, số lượng người commit v.v...

Dưới đây là một ví dụ về sử dụng Github API – Statistics để vẽ một Repository graph về lượng commit trong một dự án.



24

Hình 10: Giao diện khi sử dụng Github API

CHƯƠNG V: PHÂN TÍCH CHỨC NĂNG, HỆ THỐNG VÀ CƠ SỞ DỮ LIỆU CỦA WEBSITE

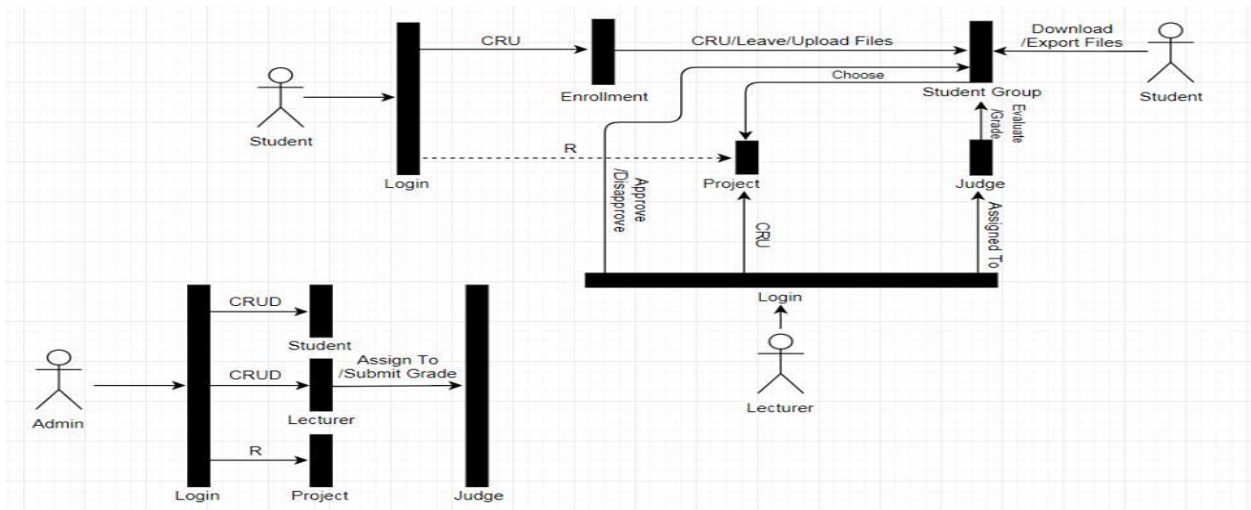
1. PHÂN TÍCH TÌNH HÌNH VÀ ĐƯA RA GIẢI PHÁP

Như đã nêu ở trên, một hệ thống hỗ trợ giúp đỡ sinh viên trong việc đăng ký, quản lý đề án cũng như chấm điểm cũng như hỗ trợ giảng viên theo dõi và chấm điểm được dễ dàng đang trở thành một vấn đề rất khó giải quyết đối với những trường đại học ở Việt Nam hiện nay.

Giải pháp: một website có đầy đủ chức năng, hỗ trợ một quy trình nhất quán từ lúc giảng viên đưa ra đề án, sinh viên đăng ký, theo dõi quá trình thực hiện và cuối cùng là đưa ra hội đồng để chấm điểm. Đồng thời phải có chế độ của Administrator, giúp cho nhà trường dễ dàng thống kê dữ liệu, quản lý, theo dõi và xem xét việc vận hành của quy trình này.

2. GIỚI THIỆU VỀ QUY TRÌNH HOẠT ĐỘNG CỦA WEBSITE

Dưới đây là hình mô tả quy trình hoạt động chính trên website



Hình 11: Quy trình hoạt động của website

Giải thích:

Sinh viên là người tạo ra những đơn đăng ký (Enrollment) để chọn loại đồ án và giảng viên sẽ hướng dẫn mình. Ngoài ra, những sinh viên khác trong cùng một nhóm có thể upload và download những tài liệu mà đồng đội của họ đã đưa lên kho lưu trữ của nhóm

Giảng viên sẽ là người xem xét, kiểm tra những đơn đăng ký dành cho chính giảng viên đó có hợp lệ hay không. Nếu không vi phạm bất kỳ quy tắc nào, họ sẽ chấp nhận (Accepted) và sau đó xếp những sinh viên thành nhiều nhóm (Group) để làm những đồ án khác nhau. Bên cạnh đó, sau một quá trình thực hiện, giảng viên có thể xem xét tình hình, kết quả hiện tại của nhóm để quyết định có đưa nhóm ra hội đồng để chấm điểm hay không.

Admin ở đây là người có quyền hành cao nhất. Admin sẽ là người tạo ra danh sách sinh viên, giảng viên và đồng thời cùng với giảng viên đưa ra những đồ án mới. Đồng thời, hội đồng chấm điểm gồm giáo viên, tỉ lệ điểm số, chức danh của từng người cũng do chính admin tạo ra và phân chia dành cho những nhóm đã được giảng viên chấp thuận đưa ra hội đồng. Sau khi hội đồng chấm các nhóm làm đồ án, điểm số đó phải qua một bước xem xét cuối cùng của Admin trước khi chính thức có giá trị.

3. NHỮNG CHỨC NĂNG CHÍNH CỦA WEBSITE

a. Giảng viên

- Tạo ra những đồ án và đề tài, dành cho sinh viên.
- Xác nhận sinh viên có đủ điều kiện thực hiện đồ án.
- Tạo nhóm, thêm sinh viên vào nhóm và chọn đề tài cho nhóm.
- Tham gia vào hội đồng để chấm điểm đồ án cho sinh viên.

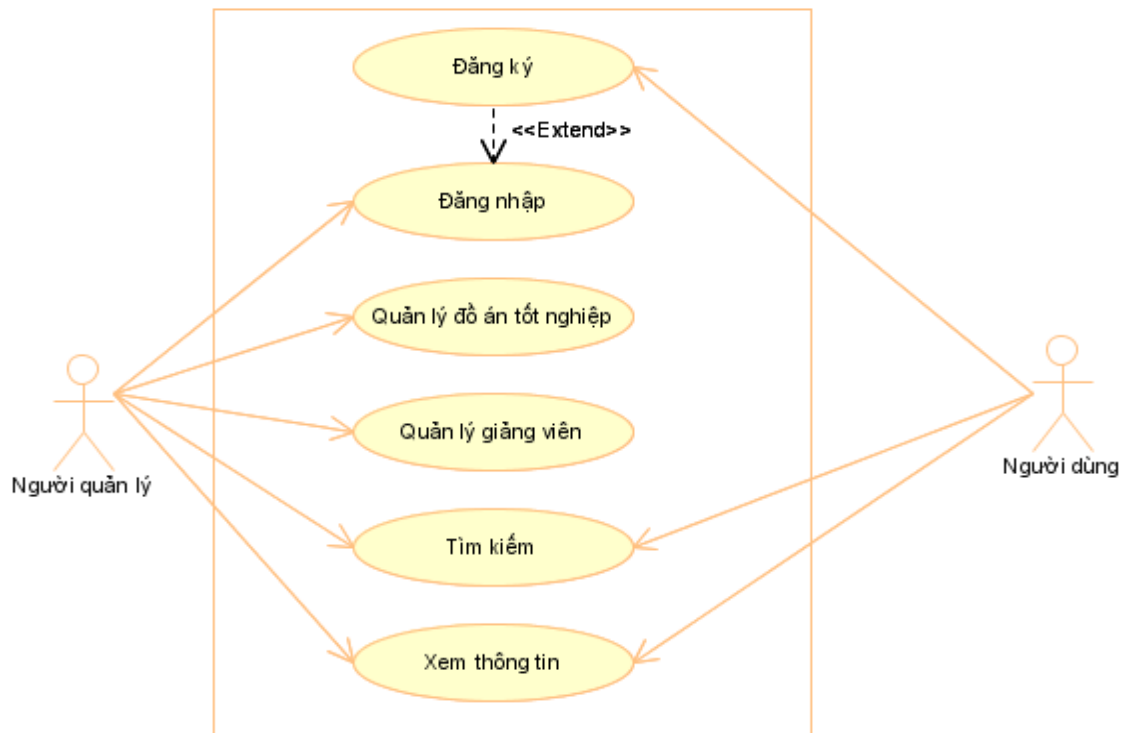
b. Sinh viên

- Đăng kí đồ án.
- Thực hiện đồ án dưới sự theo dõi của giảng viên.

c. Admin

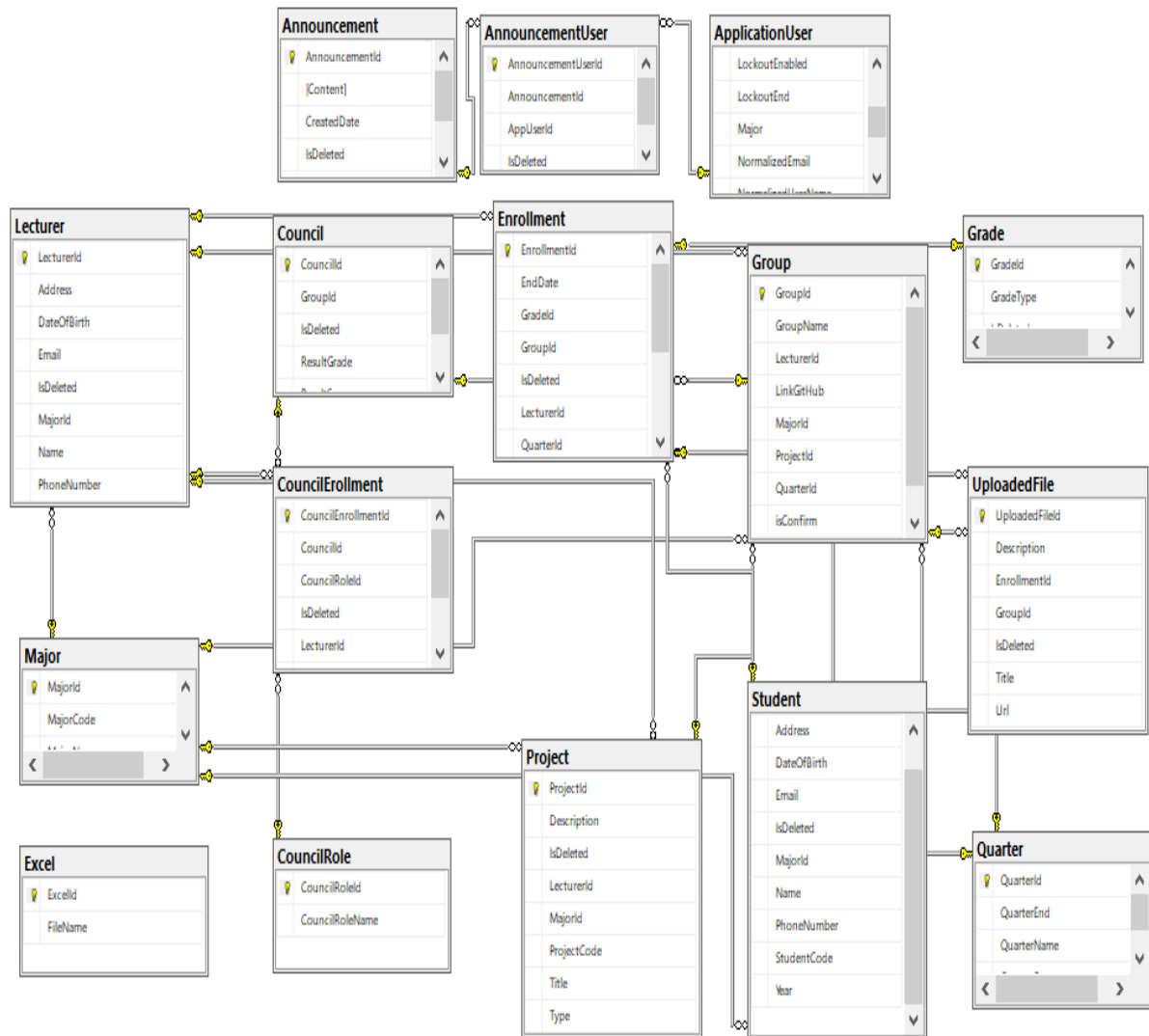
- Có quyền cao nhất, theo dõi toàn bộ quá trình.
- Thêm, sửa, xóa sinh viên, giảng viên và đồ án.
- Chỉnh sửa và đưa giảng viên vào hội đồng chấm điểm.
- Là người xác nhận điểm cuối cùng dành cho sinh viên.
- Thống kê, thu thập tất cả tài liệu, dữ liệu liên quan.

4. BIỂU ĐỒ USECASE



Hình 12: Biểu đồ Usecase của website

5. MÔ HÌNH CƠ SỞ DỮ LIỆU

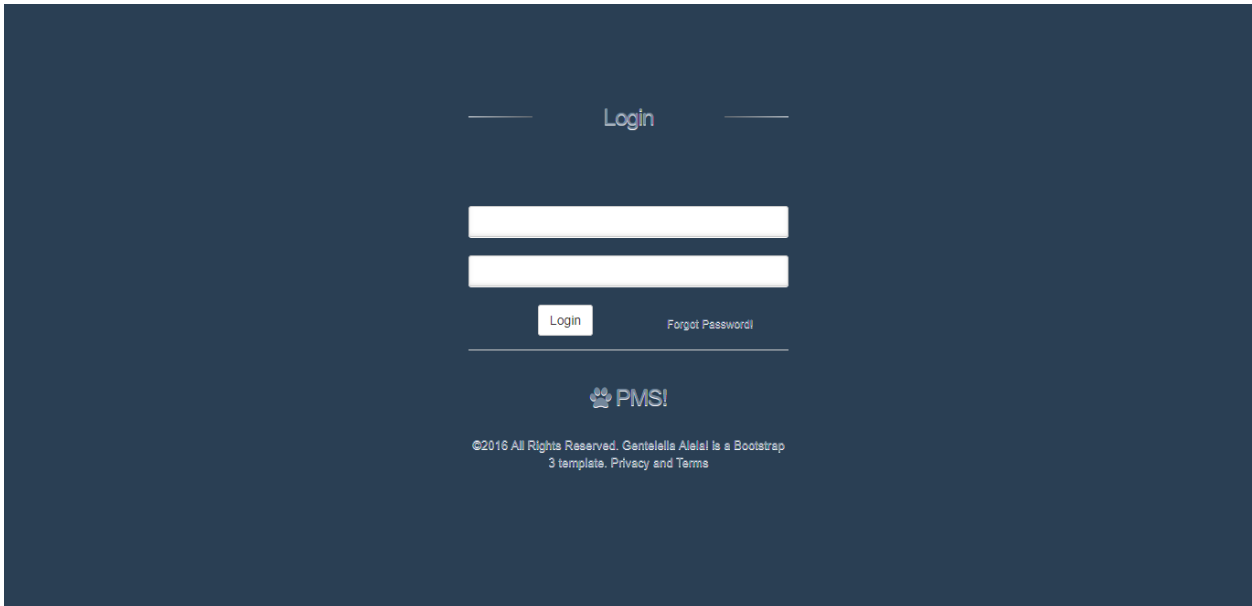


Hình 13: Mô hình cơ sở dữ liệu của website

CHƯƠNG VI: KẾT QUẢ ĐẠT ĐƯỢC

3. GIAO DIỆN TRANG ĐĂNG NHẬP

Dưới đây là giao diện của trang chủ - cũng chính là trang đăng nhập vào website



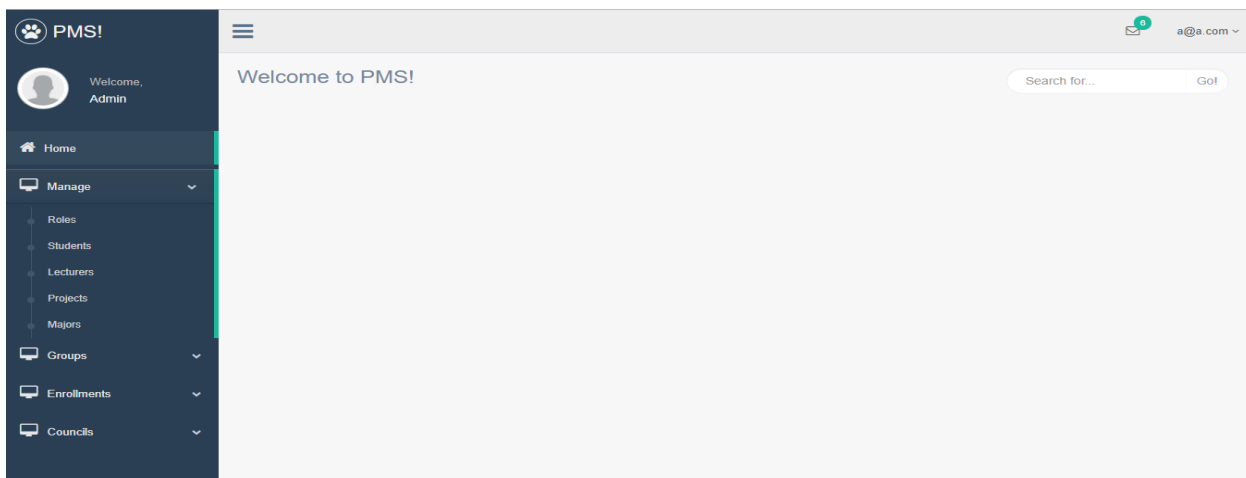
Hình 14: Giao diện Login của website

Tùy theo role của tài khoản mà người dùng đăng nhập, giao diện của website sẽ hiển thị những chức năng mà người dùng đó có quyền truy cập

Admin: do admin là người có quyền cao nhất trong website, nên trong giao diện sau khi đăng nhập chứa đầy đủ những chức năng chính của website như quản lý:

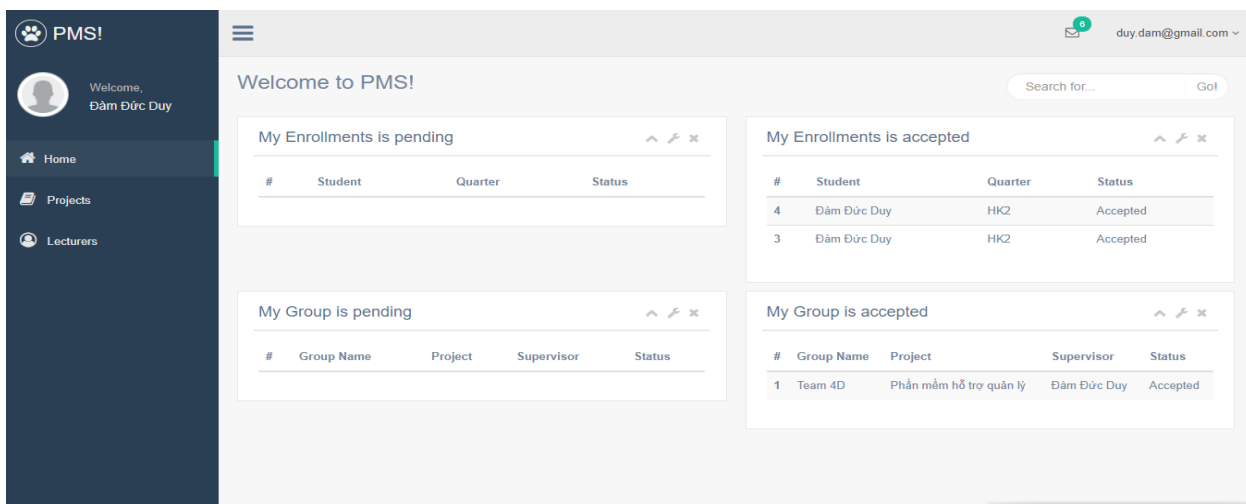
- Role: quyền hạn của tài khoản, của giảng viên, sinh viên.
- Students: hiển thị danh sách sinh viên của trường đại học.
- Lecturers: hiển thị danh sách giảng viên của trường đại học.
- Projects: danh sách những đồ án được chia theo nhiều chuyên ngành và nhiều loại khác nhau.
- Majors: hiển thị danh sách chuyên ngành của trường đại học.

- Groups: hiển thị danh sách những nhóm làm đồ án.
- Enrollments: hiển thị danh sách những đơn xin đăng ký làm đồ án của sinh viên.
- Councils: hiển thị danh sách hội đồng của trường đại học. Ngoài ra, hỗ trợ chức năng tạo hội đồng chấm điểm cho đồ án.



Hình 15: Giao diện của Admin

Giảng viên: là người xem xét, đưa lên những đồ án mới và chấp thuận cho những đơn xin làm đồ án của sinh viên. Bên cạnh đó, giảng viên cũng là người chọn những sinh viên để đưa vào nhóm làm đồ án cùng nhau.



Hình 16: Giao diện của giảng viên

Sinh viên: từ giao diện của sinh viên, họ có thể xem được danh sách những đồ án mà các giảng viên đã đưa ra. Đồng thời, tạo các đơn đăng kí để gửi cho Admin. Sau khi Admin xem xét sự hợp lệ cũng như những điều kiện tiên quyết có được thỏa mãn, những đơn xin này sẽ được tự động gửi về cho giáo viên.

The screenshot displays the PMS! student interface. On the left is a dark blue sidebar with the PMS! logo, a user profile for Huỳnh Phan Thanh Tú, and navigation links for Home, Projects, and Lecturers. The main content area has a light gray header with a welcome message and a search bar. Below the header, there are two main sections: 'My Enrollments' and 'My Groups'. 'My Enrollments' contains a table with three rows of enrollment data. 'My Groups' contains a table with one row of group data.

Welcome to PMS!

Search for... Go!

My Enrollments

#	Lecturer	Quarter	Status
3	Đàm Đức Duy	HK2	Accepted
2	Nguyễn Đình Trung	HK2	Accepted
1	Lý Mạnh Hùng	HK2	Pending

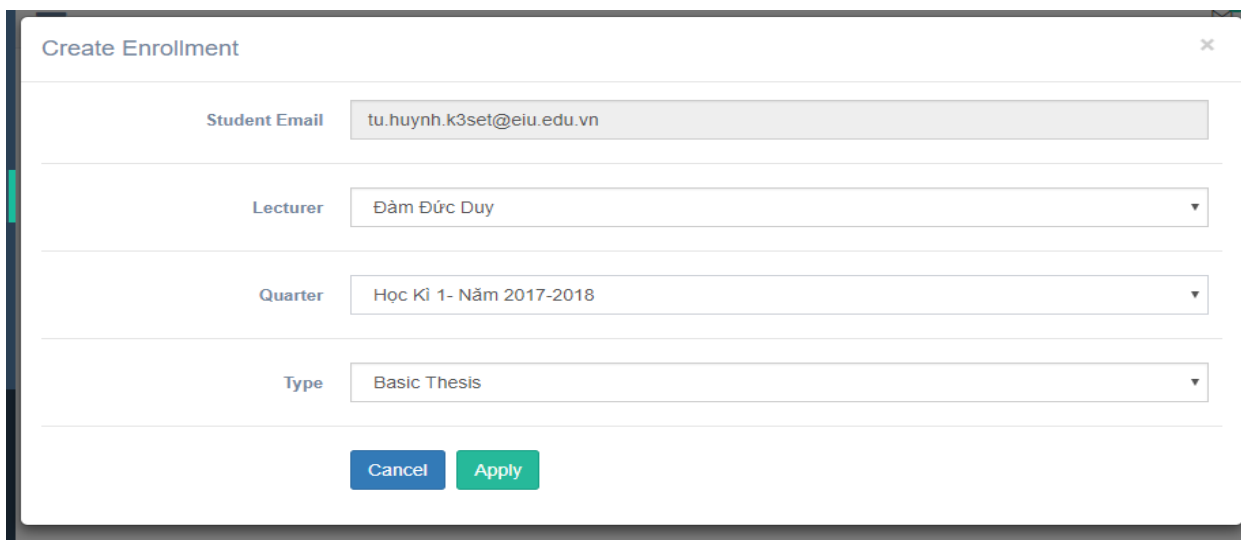
My Groups

#	Group Name	Project	Supervisor	Status
1	Team 4D	Phần mềm hỗ trợ quản lý	Đàm Đức Duy	Accepted View

Hình 17: Giao diện của sinh viên

4. ĐĂNG KÝ ĐỒ ÁN VÀ CHẤP NHẬN ĐƠN ĐĂNG KÝ

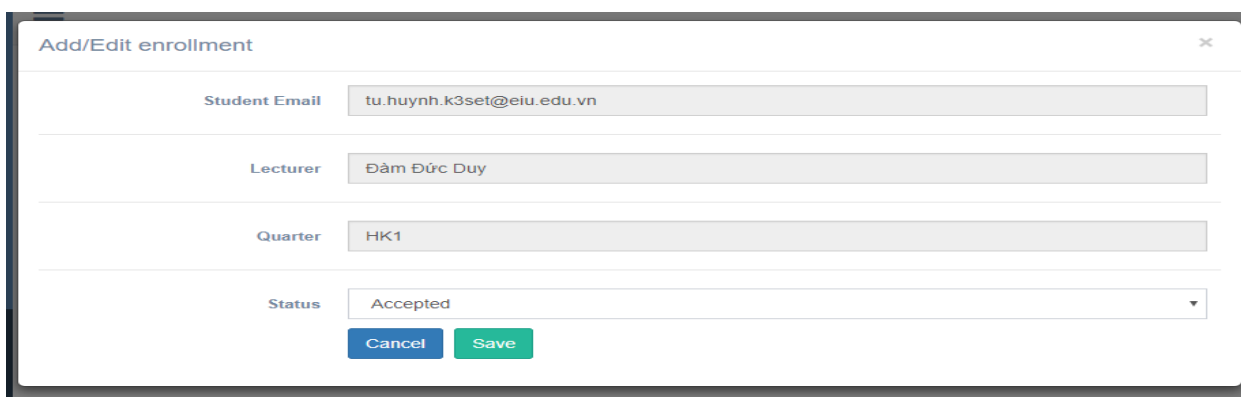
Bước đầu tiên khi sinh viên muốn thực hiện đồ án, họ phải tạo một đơn đăng ký chứa thông tin về loại đồ án họ muốn thực hiện, học kì thực hiện và giảng viên hướng dẫn chính.



The screenshot shows a web form titled "Create Enrollment" with a close button (X) in the top right corner. The form contains four input fields: "Student Email" with the value "tu.huynh.k3set@eiu.edu.vn", "Lecturer" with a dropdown menu showing "Đàm Đức Duy", "Quarter" with a dropdown menu showing "Học Kì 1- Năm 2017-2018", and "Type" with a dropdown menu showing "Basic Thesis". At the bottom of the form are two buttons: "Cancel" (blue) and "Apply" (green).

Hình 18: Giao diện của đơn đăng ký

Sau đó đơn đăng ký này sẽ được gửi đến giảng viên mà sinh viên mong muốn hướng dẫn mình. Giảng viên sẽ là người quyết định có phê duyệt đơn đăng ký này hay không dựa trên nhiều tiêu chuẩn khác nhau như: những môn tiên quyết, đồ án trước đó, kiến thức của sinh viên v.v... Chỉ những đơn đăng ký nào được giảng viên chấp thuận mới có giá trị và có thể đưa vào nhóm sinh viên.

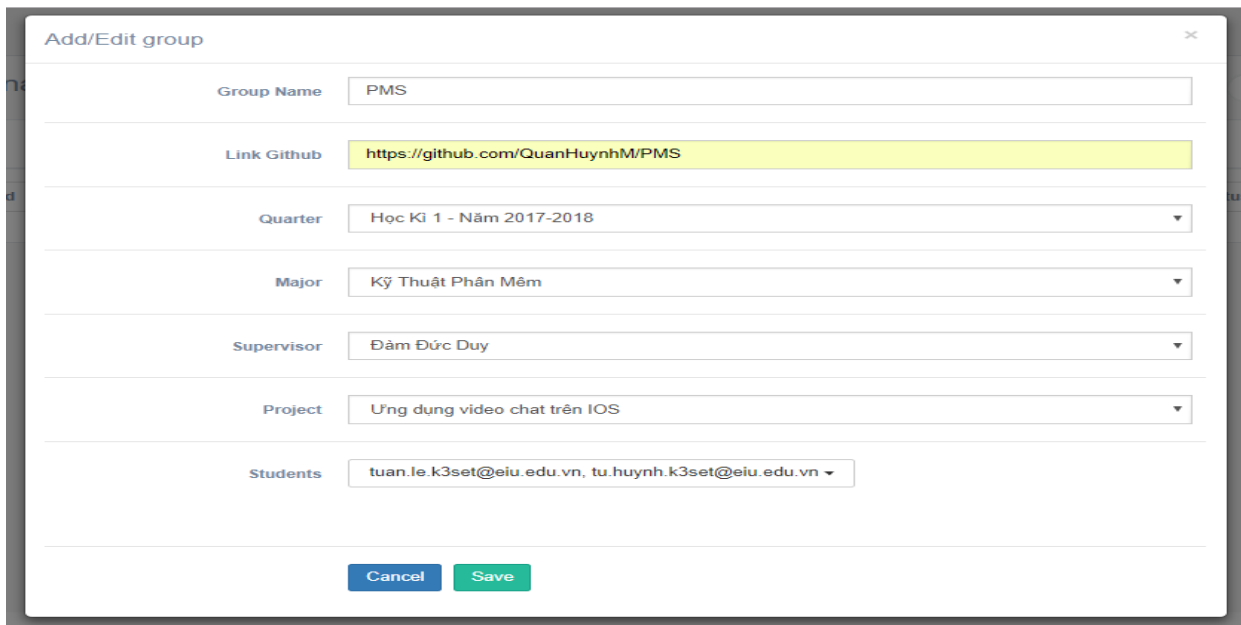


The screenshot shows a web form titled "Add/Edit enrollment" with a close button (X) in the top right corner. The form contains four input fields: "Student Email" with the value "tu.huynh.k3set@eiu.edu.vn", "Lecturer" with a dropdown menu showing "Đàm Đức Duy", "Quarter" with a dropdown menu showing "HK1", and "Status" with a dropdown menu showing "Accepted". At the bottom of the form are two buttons: "Cancel" (blue) and "Save" (green).

Hình 19: Giao diện xem xét đơn đăng ký của giảng viên

5. TẠO NHÓM CHO SINH VIÊN VÀ GIAO DIỆN CỦA NHÓM

Giảng viên sẽ là người phân chia, chọn những sinh viên phù hợp để đưa vào những nhóm nhất định để làm những đồ án phù hợp với đơn đăng ký.



Add/Edit group

Group Name: PMS

Link Github: <https://github.com/QuanHuynhM/PMS>

Quarter: Học Kỳ 1 - Năm 2017-2018

Major: Kỹ Thuật Phần Mềm

Supervisor: Đàm Đức Duy

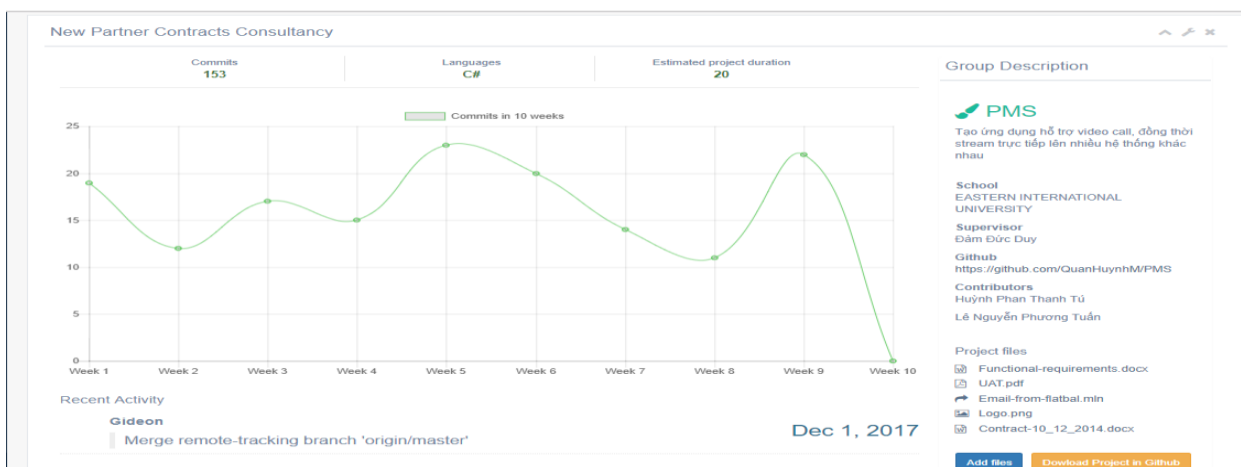
Project: Ứng dụng video chat trên IOS

Students: tuan.le.k3set@eiu.edu.vn, tu.huynh.k3set@eiu.edu.vn

Buttons: Cancel, Save

Hình 20: Giao diện tạo nhóm

Sau khi nhóm được tạo thành công, cả giảng viên và sinh viên có thể theo dõi thông tin, tiến trình thực hiện thông qua những biểu đồ được tạo ra bằng Github API, đồng thời upload lên những tài liệu cần thiết lên nhóm này.

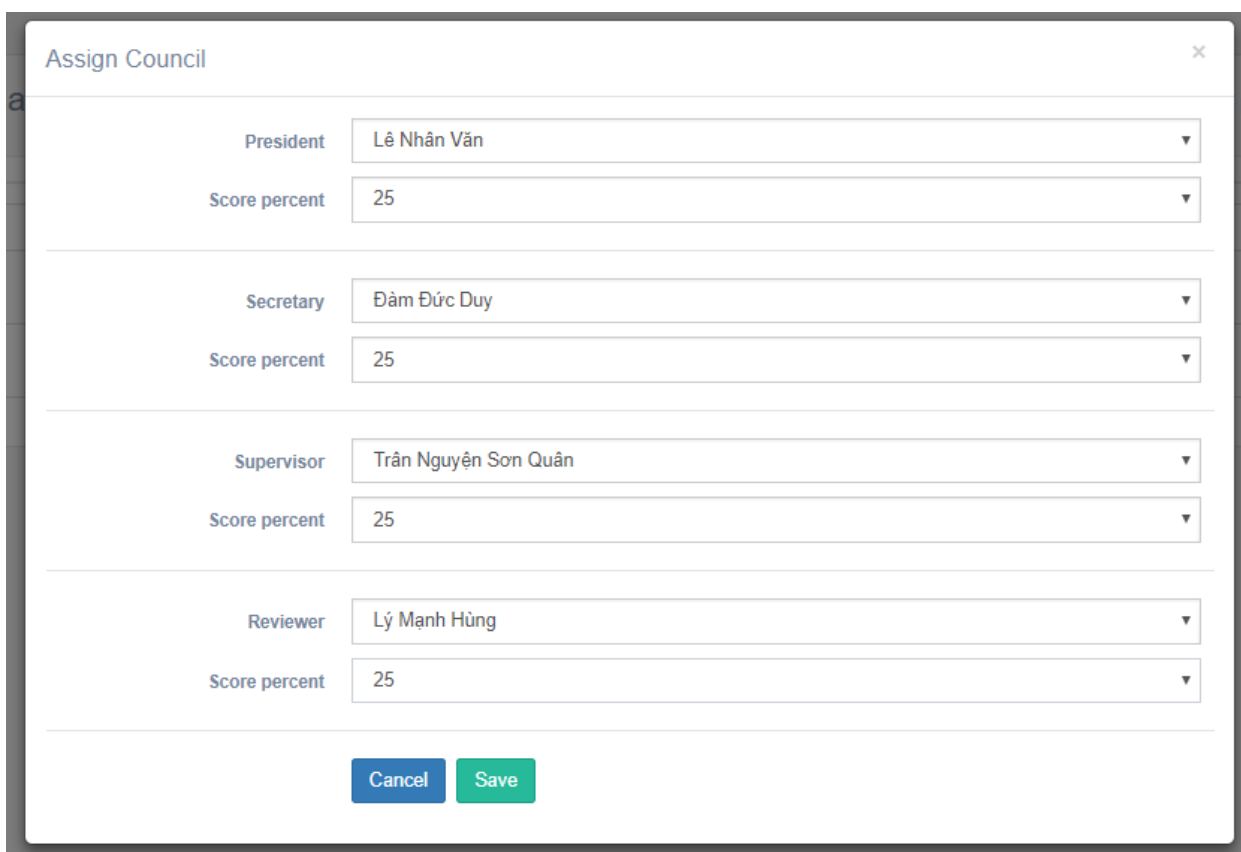


Hình 21: Giao diện thông tin của nhóm

6. TẠO HỘI ĐỒNG CHẤM ĐIỂM CHO NHÓM

Sau một quãng thời gian thực hiện đồ án, nếu giảng viên thấy nhóm đã đủ điều kiện để báo cáo trước hội đồng thì họ sẽ phê duyệt, chấp thuận cho nhóm. Danh sách những nhóm được chấp thuận sẽ được gửi đến Admin và cũng chính Admin sẽ là người tạo ra hội đồng.

Hội đồng sẽ gồm có 4 chức vụ và sẽ có những tỉ lệ chấm điểm khác nhau tùy thuộc vào sự phân chia, tính toán của Admin.



The screenshot shows a web form titled "Assign Council" with a close button (X) in the top right corner. The form is divided into four sections, each representing a role in the council. Each section contains a label for the role, a dropdown menu for selecting a name, and a dropdown menu for selecting a score percent. The roles and their assigned names are: President (Lê Nhân Văn), Secretary (Đàm Đức Duy), Supervisor (Trần Nguyễn Sơn Quân), and Reviewer (Lý Mạnh Hùng). All score percent values are set to 25. At the bottom of the form, there are two buttons: "Cancel" (blue) and "Save" (green).

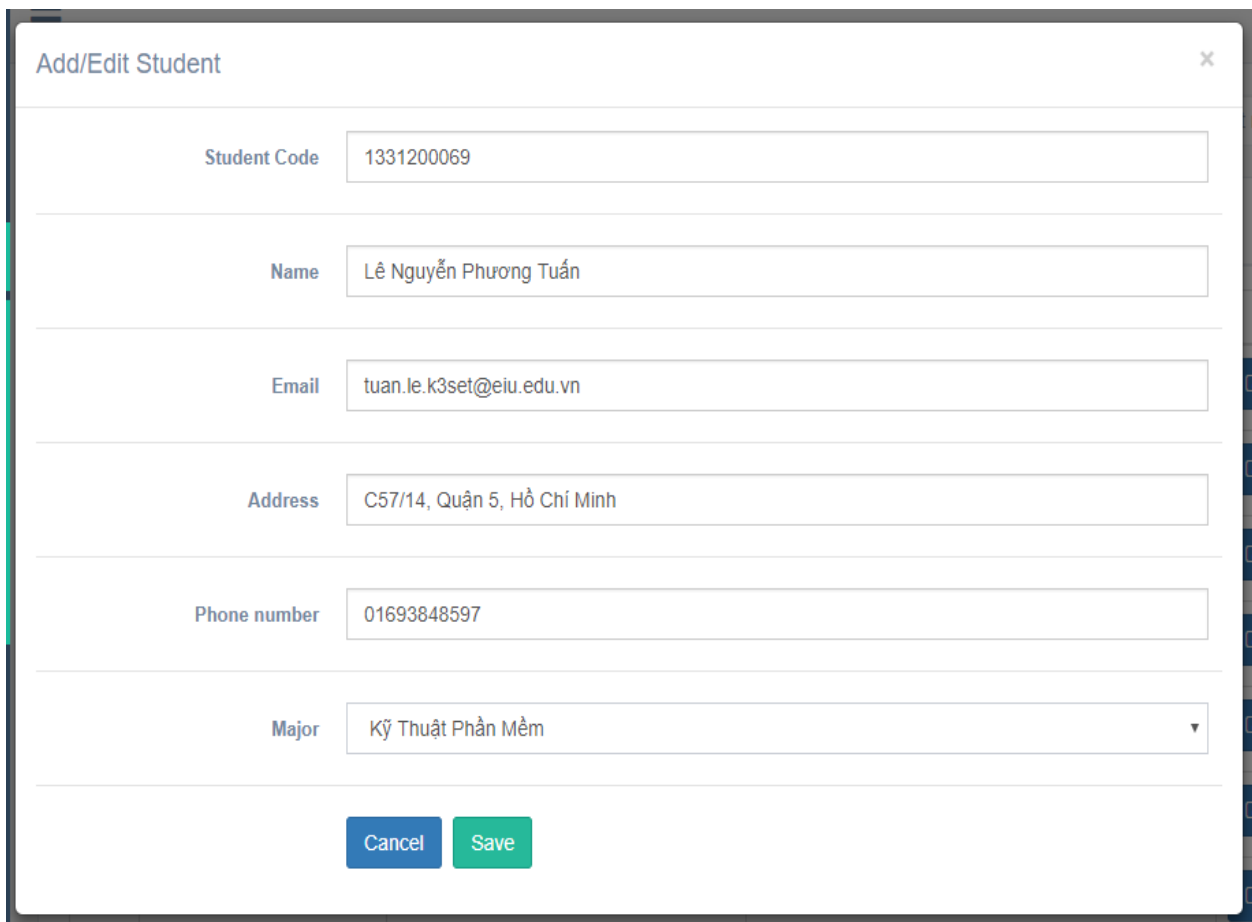
Role	Name	Score percent
President	Lê Nhân Văn	25
Secretary	Đàm Đức Duy	25
Supervisor	Trần Nguyễn Sơn Quân	25
Reviewer	Lý Mạnh Hùng	25

Hình 22: Giao diện tạo hội đồng chấm điểm

7. THÊM SINH VIÊN, GIẢNG VIÊN VÀ ĐỒ ÁN VÀO HỆ THỐNG

Đây là chức năng dành riêng cho Admin. Tuy nhiên, đối với đồ án, giảng viên cũng là người có thể đưa ra nhưng vẫn phải được Admin xem xét chấp nhận trước khi hiển thị lên trên website.

Thêm sinh viên vào hệ thống



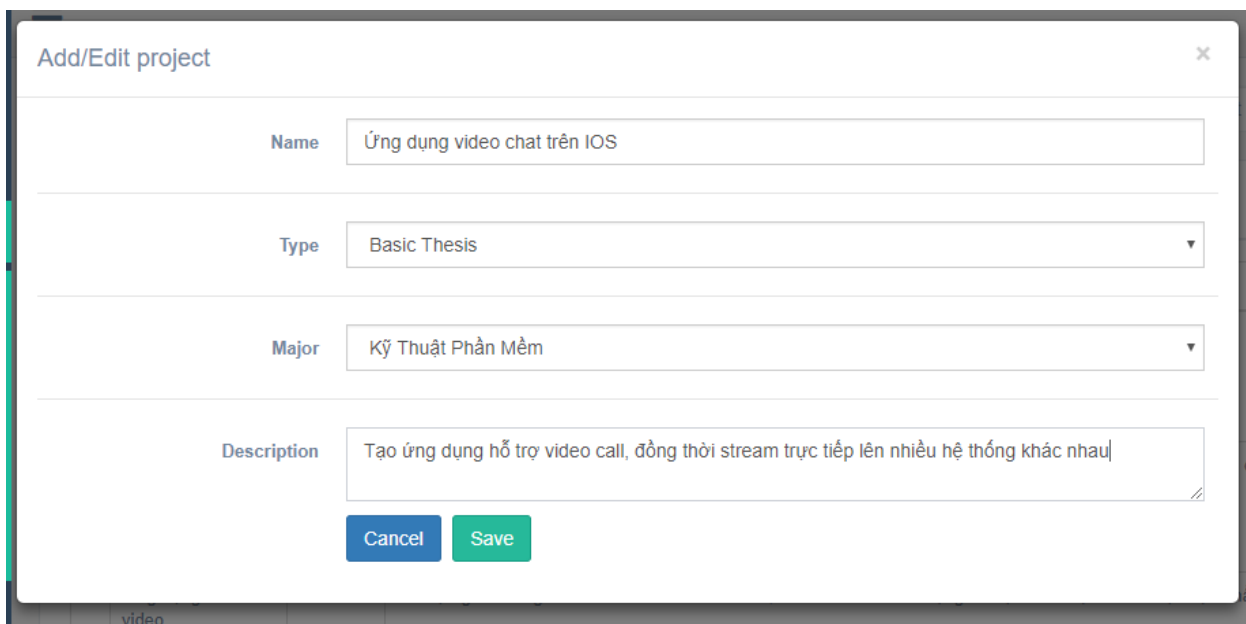
The screenshot shows a web form titled "Add/Edit Student" with a close button (X) in the top right corner. The form contains the following fields:

- Student Code:** Input field containing "1331200069".
- Name:** Input field containing "Lê Nguyễn Phương Tuấn".
- Email:** Input field containing "tuan.le.k3set@eiu.edu.vn".
- Address:** Input field containing "C57/14, Quận 5, Hồ Chí Minh".
- Phone number:** Input field containing "01693848597".
- Major:** Dropdown menu with "Kỹ Thuật Phần Mềm" selected.

At the bottom of the form are two buttons: "Cancel" (blue) and "Save" (green).

Hình 23: Giao diện thêm sinh viên

Thêm đồ án vào hệ thống



Add/Edit project

Name: Ứng dụng video chat trên IOS

Type: Basic Thesis

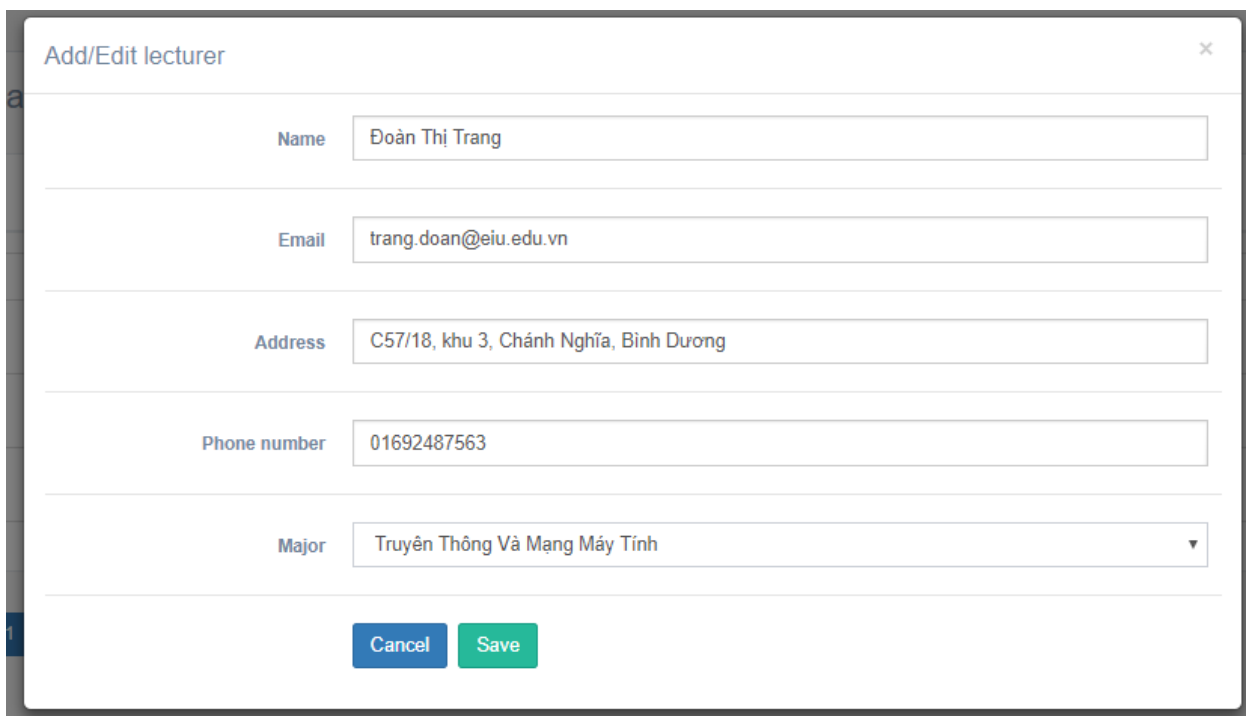
Major: Kỹ Thuật Phần Mềm

Description: Tạo ứng dụng hỗ trợ video call, đồng thời stream trực tiếp lên nhiều hệ thống khác nhau

Cancel Save

Hình 24: Giao diện thêm đồ án

Thêm giảng viên vào hệ thống



Add/Edit lecturer

Name: Đoàn Thị Trang

Email: trang.doan@eiu.edu.vn

Address: C57/18, khu 3, Chánh Nghĩa, Bình Dương

Phone number: 01692487563

Major: Truyền Thông Và Mạng Máy Tính

Cancel Save

Hình 25: Giao diện thêm giảng viên

CHƯƠNG VII: TỔNG KẾT

1. CHỨC NĂNG ĐẠT ĐƯỢC

- Website có giao diện, dễ dàng thao tác, làm việc.
- Sử dụng Website để đăng ký, chấm điểm đồ án
- Có thể thêm sinh viên, giảng viên, đồ án vào hệ thống của website
- Tích hợp nhiều công nghệ mới, hỗ trợ realtime, bảo mật và sử dụng những API hiệu quả.
- Website được phân chia thành hai phần front-end và back-end song song với nhau theo mô hình Restful API.

2. NHỮNG MẶT HẠN CHẾ

- Tốc độ xử lý, lưu trữ, truy xuất dữ liệu đôi lúc còn khá chậm.
- Chưa hỗ trợ nhiều ngôn ngữ.
- Chưa có chức năng liên kết được đến cơ sở dữ liệu của trường đại học.

3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

- Cải thiện tốc độ xử lý Back-End và Front-End.
- Cải thiện giao diện, hỗ trợ nhiều ngôn ngữ
- Cập nhật thêm nhiều chức năng mới như người dùng có thể liên lạc, trò chuyện được với nhau.
- Thêm tính năng của quản lý, phân công công việc trong nhóm.

CHƯƠNG VIII: TÀI LIỆU THAM KHẢO

[1] <https://docs.microsoft.com/en-us/aspnet/core/>

[2] <https://techmaster.vn/posts/33959/khai-niem-ve-json-web-token>

[3] Mosh Hamedani, Build a real-world App with Angular 2 and ASP.NET Core

[4] Adam Freeman, Pro ASP.NET Core MVC