**ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**Võ Anh Tuấn**

**XÂY DỰNG ỨNG DỤNG CHIA SẺ KIẾN THỨC**

**BẰNG LARAVEL VÀ VUEJS**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: Nguyễn Hồng Phương**

**Sinh viên thực hiện: Võ Anh Tuấn**

**HÀ NỘI – 2019**

**LỜI CẢM ƠN**

Trước hết, tôi xin phép gửi lời cảm ơn chân thành nhất đến quý thầy giáo, cô giáo, đội ngũ giảng viên trường đại học Bách khoa Hà Nội đã hướng dẫn, giảng dạy và truyền đạt những bài học quý báu cho tôi trong công việc và cuộc sống. Đó không chỉ là kiến thức về chuyên ngành, mà hơn thế, đó còn là những bài học sâu sắc trong cuộc sống, là những kinh nghiệm quý giá, vô cùng cần thiết cho cả những năm tháng ngồi trên ghế giảng đường đại học và cả cho công việc trong tương lai. Là một sinh viên của trường đại học Bách khoa Hà Nội, bản thân tôi luôn tự hào vì điều đó, vì tôi là sinh viên của trường đại học Bách khoa Hà Nội, ngôi trường đại học có bề dày lịch sử, với đội ngũ giảng viên giỏi, đầy lòng nhiệt huyết. Cùng với đó là hệ thống cơ sở vật chất trường học đầy đủ, được đảm bảo phục vụ tốt nhất cho quá trình học tập và nghiên cứu của sinh viên.

Đồng thời, tôi xin gửi lời cảm ơn sâu sắc đến thầy giáo Nguyễn Hồng Phương đã tận tình giúp đỡ và hướng dẫn tôi trong suốt quá trình nghiên cứu và thực hiện khóa luận tốt nghiệp. Thầy cô đã nhiệt tình giảng giải, hỗ trợ và định hướng đưa ra những đóng góp quý báu cho tôi.

Để có được ngày hôm nay, tôi cũng xin gửi lời cảm ơn tới gia đình của tôi, lời cảm ơn bố mẹ, chị gái - những người đã lao động vất vả để nuôi tôi ăn học và là nguồn động lực để tôi phấn đấu trưởng thành hơn. Cuối cùng, tôi xin cảm ơn tới các bạn sinh viên của trường nói chung và tập thể lớp CNTT2.1 nói riêng. Cảm ơn các bạn đã đồng hành với tôi trong suốt 5 năm học qua. Các bạn là những người bạn chân thành tốt bụng năng động, và giúp đỡ tôi trong học tập cũng nhưng trong cuộc sống.

**TÓM TẮT**

**Tóm tắt:** Ngày nay, các kiến thức rất đa dạng phong phú trong học tập và đời sống, vì vậy việc chia sẻ các khái niệm, các kiến thức mới luôn luôn cần thiết. Dựa trên ý tưởng của trang web Viblo tôi đã phát triển một ứng dụng web dựa trên công nghệ Laravel và Vuejs. Tôi đã sử dụng Laravel và một chút Nodejs để phát triển ứng dụng phần máy chủ và sử dụng Vuejs để xây dựng giao diện người dùng.

Khóa luận gồm 6 chương có cấu trúc như sau:

Chương 1: Lời mở đầu - Trình bày tính cần thiết, động lực phát triển của đề tài.

Chương 2: Cơ sở lý thuyết và công nghệ - Trình cày cơ sở lý thuyết, giới thiệu các công nghệ sử dụng để xây dựng ứng dụng.

Chương 3: Phân tích yêu cầu bài toán - Trình bày các yêu cầu của hệ thống, mô tả và phân tích yêu cầu các chức năng của bài toán.

Chương 4: Giải quyết bài tóan với Laravel và Vuejs- Trình bày cách áp dụng công nghệ vào bài toán để triển khai và xây dựng ứng dụng.

Chương 5: Thực nghiệm - Trình bày thực nghiệm của ngừoi dùng thử và mô tả cách tiếp cận ứng dụng.

Chương 6: Kết luận – Trình bày kết quả mà khóa luận đạt được, nhưng điều bản thân đã học được khi làm khóa luận và đưa ra định hướng tiếp theo của ứng dụng.

**LỜI CAM ĐOAN**

Tôi xin cam đoan mọi kết quả trong khóa luận này đều do tôi thực hiện dưới sự hướng dẫn của thầy Nguyễn Hồng Phương. Tất cả các tham khảo nghiên cứu liên quan đều được nêu rõ nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong khóa luận.

Khóa luận không sao chép tài liệu, công trình nghiên cứu từ người khác mà không chỉ rõ về mặt tài liệu tham khảo. Các thống kê, các kết quả trình bày trong khóa luận đều được lấy từ thực nghiệm khi chạy chương trình. Nếu tôi sai tôi hoàn toàn chịu trách nhiệm theo quy định của trường Đại học Bách khoa Hà Nội.

Hà Nội, ngày 02 tháng 05 năm 2019

Sinh viên

**MỤC LỤC**

[Chương 1. Lời mở đầu 1](#_Toc512896068)

[1.1. Lý do chọn đề tài 1](#_Toc512896069)

[1.2. Phạm vi và đối tượng nghiên cứu 2](#_Toc512896070)

[1.3. Mục tiêu khóa luận 2](#_Toc512896071)

[1.4. Khó khăn và giải pháp 2](#_Toc512896072)

[1.5. Nội dung khóa luận 3](#_Toc512896073)

[Chương 2. Cơ sở lý thuyết và công nghệ 4](#_Toc512896074)

[2.1. Tổng quan về lập trình web và phát triển ứng dụng web 4](#_Toc512896075)

[2.2. Angular 2 5](#_Toc512896076)

[2.2.1. Single-page Application (SPA) 5](#_Toc512896077)

[2.2.2. Giới thiệu về Angular 2 5](#_Toc512896078)

[2.2.3. Những thành phần chính của Angular 2 6](#_Toc512896079)

[2.3. Ruby và Ruby on Rails Framework 7](#_Toc512896080)

[2.3.1. Ruby là gì? 7](#_Toc512896081)

[2.3.2. Ruby on Rails framework 8](#_Toc512896082)

[2.4. MySQL 10](#_Toc512896083)

[2.4.1. Hệ quản trị cơ sở dữ liệu 10](#_Toc512896084)

[2.4.2. MySQL là gì 10](#_Toc512896085)

[Chương 3. Phân tích yêu cầu bài toán 12](#_Toc512896086)

[3.1. Mô tả bài toán 12](#_Toc512896087)

[3.2. Thu thập yêu cầu 12](#_Toc512896088)

[3.3. Phân tích tác nhân chức năng của hệ thống 12](#_Toc512896089)

[3.4. Phân tích yêu cầu 15](#_Toc512896090)

[3.4.1. Thiết kế ca sử dụng 15](#_Toc512896091)

[3.4.2. Phân tích biểu đồ ca sử dụng 17](#_Toc512896092)

[Chương 4. Giải quyết bài toán với Ruby On Rails, Angular 2 29](#_Toc512896093)

[4.1. Thiết kế phần mềm 29](#_Toc512896094)

[4.1.1. Thiết kế giao diện 29](#_Toc512896095)

[4.1.2. Thiết kế cơ sở dữ liệu 29](#_Toc512896096)

[4.2. Triển khai phần mềm: 32](#_Toc512896097)

[4.2.1. Ứng dụng Angular 2 để xây dựng phía máy khách 32](#_Toc512896098)

[4.2.2. Xây dựng backend với ruby on rails 34](#_Toc512896099)

[4.3. Kiểm thử tự động 35](#_Toc512896100)

[Chương 5. Thực nghiệm 37](#_Toc512896101)

[5.1. Môi trường thực nghiệm 37](#_Toc512896102)

[5.2. Dữ liệu thực nghiệm 37](#_Toc512896103)

[5.3. Kết quả thực nghiệm 38](#_Toc512896104)

[5.3.1. Xác thực và ủy quyền ngừoi dùng 38](#_Toc512896105)

[5.3.2. Dành cho người dùng 40](#_Toc512896106)

[5.3.3. Khu vực quản lý của quản trị viên 47](#_Toc512896107)

[5.4. Nhận xét, đánh giá ứng dụng 49](#_Toc512896108)

[Chương 6. Kết luận 51](#_Toc512896109)

[6.1. Kết quả đạt được 51](#_Toc512896110)

[6.2. Định hướng phát triển 51](#_Toc512896111)

[TÀI LIỆU THAM KHẢO 52](#_Toc512896112)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Mô hình MVC 9](#_Toc512417602)

[Hình 3.1: Mô hình ca sử dụng của người dùng 15](#_Toc512417603)

[Hình 3.2: Mô hình ca sử dụng của quản trị viên 15](#_Toc512417604)

[Hình 3.3 Ràng buộc giữa các use-case 16](#_Toc512417605)

[Hình 4.1: Mô hình dữ liệu trong cơ sở dữ liệu MySQL 29](#_Toc512417606)

[Hình 4.2: Kết quả chạy kiểm thử tự động 36](#_Toc512417607)

[Hình 5.1: Dữ liệu dưới dạng excel của google sheets 37](#_Toc512417608)

[Hình 5.2: Giao diện khi chưa đăng nhập 38](#_Toc512417609)

[Hình 5.3: Giao diện đăng ký tài khoản của ngừoi dùng 39](#_Toc512417610)

[Hình 5.4: Giao diện trang chủ dành cho quản trị viên 39](#_Toc512417611)

[Hình 5.5: Giao diện trang chủ dành cho người dùng 40](#_Toc512417612)

[Hình 5.6: Giao diện trang cá nhân của người dùng 40](#_Toc512417613)

[Hình 5.7: Giao diện trang cá nhân của người dùng 41](#_Toc512417614)

[Hình 5.8: Giao diện xem và chỉnh sửa thông tin người dùng 41](#_Toc512417615)

[Hình 5.9: Giao diện trang chủ của người dùng 42](#_Toc512417616)

[Hình 5.10: Giao diện của một bộ thẻ với các bài học 43](#_Toc512417617)

[Hình 5.11: Giao diện của một bộ thẻ với các thẻ 43](#_Toc512417618)

[Hình 5.12: Giao diện bài học với Dạng câu hỏi học từ 44](#_Toc512417619)

[Hình 5.13: Giao diện bài học với Dạng câu hỏi viết 44](#_Toc512417620)

[Hình 5.14: Giao diện bài học với Dạng câu hỏi nhớ các mặt 45](#_Toc512417621)

[Hình 5.15: Tạo bộ thẻ 45](#_Toc512417622)

[Hình 5.16: Nhập dữ liệu màn tạo bộ thẻ 46](#_Toc512417623)

[Hình 5.17: Giao diện sau khi đăng nhập của quản trị viên 47](#_Toc512417624)

[Hình 5.18: Giao diện tất cả người dùng 47](#_Toc512417625)

[Hình 5.19: Giao diện tất cả bộ thẻ 48](#_Toc512417626)

[Hình 5.20: Giao diện tất cả phản hồi 48](#_Toc512417627)

**DANH MỤC BẢNG BIỂU**

[Bảng 3.1: Mô tả ca sử dụng đăng nhập 17](#_Toc512216527)

[Bảng 3.2: Mô tả ca sử dụng đăng ký 18](#_Toc512216528)

[Bảng 3.3: Mô tả ca sử dụng đăng xuất 19](#_Toc512216529)

[Bảng 3.4: Mô tả ca sử dụng tạo bộ thẻ 20](#_Toc512216530)

[Bảng 3.5: Mô tả ca sử dụng học theo bài học 21](#_Toc512216531)

[Bảng 3.6: Mô tả ca sử dụng chỉnh sửa bộ thẻ 22](#_Toc512216532)

[Bảng 3.7: Mô tả ca sử dụng tìm kiếm bộ thẻ 23](#_Toc512216533)

[Bảng 3.8: Mô tả ca sử dụng chỉnh sửa trang cá nhân 24](#_Toc512216534)

[Bảng 3.9: Mô tả ca sử dụng xem tất cả bộ thẻ 26](#_Toc512216535)

[Bảng 3.10: Mô tả ca sử dụng xem tất cả người dùng 27](#_Toc512216536)

[Bảng 3.11: Mô tả ca sử dụng phản hồi của người dùng 25](#_Toc512216537)

[Bảng 3.12: Mô tả ca sử dụng xem tất cả phản hồi của bộ thẻ 28](#_Toc512216538)

[Bảng 4.1: Mô tả bảng User 30](#_Toc512216539)

[Bảng 4.2: Mô tả bảng Decks 30](#_Toc512216540)

[Bảng 4.3: Mô tả bảng Cards 30](#_Toc512216541)

[Bảng 4.4: Mô tả bảng Lessons 31](#_Toc512216542)

[Bảng 4.5: Mô tả bảng User\_Lessons 31](#_Toc512216543)

[Bảng 4.6: Mô tả bảng Feedbacks 31](#_Toc512216544)

[Bảng 4.7: Mô tả bảng Favorites 31](#_Toc512216545)

[Bảng 4.8: Mô tả bảng Relationships 32](#_Toc512216546)

[Bảng 4.9: Mô tả bảng Admins 32](#_Toc512216547)

[Bảng 4.10: Bảng định nghĩa các định tuyến của ứng dụng 32](#_Toc512216548)

[Bảng 4.11: Bảng mô tả API 34](#_Toc512216549)

[Bảng 5.1: Phản hồi của người dùng 49](#_Toc512216550)

**DANH SÁCH KÝ HIỆU, CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| ***Ký hiệu*** | ***Diễn giải*** | ***Tiếng Việt*** |
| *HTML* | *HyperText Markup Language* | *Ngôn ngữ đánh dấu siêu văn bản* |
| *CSS* | *Cascading Style Sheets* | *Ngôn ngữ định dạng các ngôn ngữ tạo ra bởi ngôn ngữ đánh dấu* |
| *API* | *Application Programming Interface* | *Giao diện lập trình ứng dụng* |
| *ASP* | *Active Server Pages* | *Trang máy chủ hoạt động* |
| *CGI* | *Computer-generated imagery* | *Máy tính tạo ra hình ảnh* |
| *JSP* | *JavaServer Pages* | *Trang máy chủ java* |
| *JS* | *Javascript* |  |
| *AI* | *Artificial intelligence* | *Trí tuệ nhân tạo* |
| *UI* | *User Interface* | *Giao diện người dùng* |
| *OOP* | *Object-oriented programming* | *Lập trình hướng đối tượng* |
| *REST* | *Representational state transfer* | *Chuyển trạng thái biểu hiện* |
| *ORM* | *Object-relational mapping* | *Bản đồ quan hệ giữa các đối tượng* |
| *SQL* | *Structured Query Language* | *Ngôn ngữ truy vấn cấu trúc* |
| *TCP* | *Transmission Control Protocol* | *Giao thức điều khiển truyền* |
| *BDD* | *Behavior-driven Development* | *Phát triển hành vi* |
| *MMORPG* | *Massively Multiplayer Online Role-Playing Game* | *Trò chơi nhập vai trực tuyến nhiều người chơi* |

# Lời mở đầu

* 1. Lý do chọn đề tài

**Trong những năm gần đây, con số các công cụ được thiết kế để hỗ trợ cho các kĩ sư phát triển phần mềm thực sự bùng nổ. Những frameworks giúp bạn kết nối các ứng dụng lại nhanh hơn, các thư viện mã nguồn mở giúp bạn tránh phải dựng và tái dựng những phần nhỏ nhặt thông thường. Đồng thời, các dịch vụ đám mây giúp cho việc thêm các đặc tính dễ dàng và tiết kiệm hơn rất nhiều so với khi bạn tự dựng.**

Nhưng nếu bạn muốn học cách làm thế nào để sử dụng tất cả các công nghệ đó thì nó đồng nghĩa với việc bạn phải đọc sách rất nhiều. Đọc hàng tá các tập tài liệu có thể là một việc quá khó khăn và dễ gây chán nản, đặc biệt là nếu tài liệu không cho bạn biết những gì bạn cần biết. Nhiều devs cuối cùng đành phải tìm đến các trang web để được giúp đỡ. Một trong số đó là trang web Viblo, một trang web có tiếng nơi mà các kĩ sư có thể chia sẻ kinh nghiệm, kiến thức về một lĩnh vực công nghệ nào đó hay đặt câu hỏi và tìm lời giải đáp cho một vấn đề thắc mắc.

Dựa trên mục đích đó tôi đã xây dựng một ứng dụng chia sẻ kiến thức với giao diện dễ dàng sử dụng và giúp cho mọi người có thể tiếp cận, trao đổi nhanh nhất với một lượng dữ liệu lớn.

Để xây dựng ứng dụng tốt nhất,tôi đã sử dụng công nghệ Laravel kết hợp với Vuejs. Tôi sẽ trình bày chi tiết về các công nghệ ở các chương sau của khóa luận.

* 1. Phạm vi và đối tượng nghiên cứu
* Phạm vi địa lý của khóa luận: Trên lãnh thổ Việt Nam.
* Đối tượng nghiên cứu: Những người có nhu cầu tìm hiểu hay chia sẻ kiến thức về lập trình.
* Các công nghệ sử dụng trong khóa luận:
* Laravel
* Vuejs
* Socketio
* Nodejs
* Redis
* Scrapy
* Elasticsearch
  1. Mục tiêu khóa luận

Bài toán đặt ra yêu cầu một giải pháp để giải quyết yêu cầu bài toán, do đó việc xây dựng và phát triển hoàn thiện ứng dụng là một vấn đề không nhỏ cần nhiều thời gian và công sức. Nội dung và kết quả của khóa luận là tập trung phân tích và thiết kế các chức năng cơ bản, cần thiết nhất đáp ứng yêu cầu cơ bản được đặt ra của người dùng và giải quyết các vấn đề mà các ứng dụng tương tự còn gặp phải. Đầu ra của ứng dụng là một ứng dụng web đáp ứng yêu cầu của bài toán.

* 1. Khó khăn và giải pháp

Hiện tại, có nhiều ứng dụng web để chia sẻ kiến thức. Tôi cần phải xây dựng một hệ thống vừa giúp người dùng dễ dàng sử dụng vừa đáp ứng được đầy đủ chức năng. Do đó cần xây dựng trang web với một giao diện thân thiện, tăng trải nghiệm cho người dùng.

Ngoài ra, số lượng bài viết ban đầu cho trang web là vô cùng quan trọng, chính vì vậy cần xây dựng một hệ thống để thu thập dữ liệu một cách nhanh nhất phù hợp với dữ liệu của trang web.

* 1. Nội dung khóa luận

Khóa luận gồm 6 chương có cấu trúc như sau:

Chương 1: Lời mở đầu - Trình bày tính cần thiết, động lực phát triển của đề tài.

Chương 2: Cơ sở lý thuyết và công nghệ - Trình cày cơ sở lý thuyết, giới thiệu các công nghệ sử dụng để xây dựng ứng dụng.

Chương 3: Phân tích yêu cầu bài toán - Trình bày các yêu cầu của hệ thống, mô tả và phân tích yêu cầu các chức năng của bài toán.

Chương 4: Giải quyết bài tóan với Vuejs - Trình bày cách áp dụng công nghệ vào bài toán để triển khai và xây dựng ứng dụng.

Chương 5: Thực nghiệm - Trình bày thực nghiệm của ngừoi dùng thử và mô tả cách tiếp cận ứng dụng.

Chương 6: Kết luận – Trình bày kết quả mà khóa luận đạt được, nhưng điều bản thân đã học được khi làm khóa luận và đưa ra định hướng tiếp theo của ứng dụng.

# Cơ sở lý thuyết và công nghệ

Ứng dụng chia sẻ kiến thức được xây dựng dựa trên các công nghệ Laravel và Vuejs cùng với cơ sở dữ liệu MySQL. Chương 2 sẽ tập trung giới thiệu và phân tích những công nghệ này.

* 1. Tổng quan về lập trình web và phát triển ứng dụng web

Một ứng dụng web là một trình ứng dụng mà có thể tiếp cận qua web thông qua mạng như Internet hay intranet. Chẳng hạn, ứng dụng web được dùng để hiện thực Webmail, bán hàng trực tuyến, đấu giá trực tuyến, wiki, diễn đàn thảo luận, Weblog, MMORPG, hệ quản trị nội dung, phần mềm quản lý nguồn nhân lực và nhiều chức năng khác.

Dù có nhiều biến thể, một ứng dụng Web thông thường được cấu trúc như một ứng dụng ba lớp. Ở dạng phổ biến nhất, một trình duyệt Web là lớp thứ nhất, một bộ máy sử dụng một vài công nghệ nội dung Web động (như ASP, ASP.NET, CGI, ColdFusion, JSP/Java, PHP, Python, hoặc Ruby On Rails) là lớp giữa, và một cơ sở dữ liệu là lớp thứ ba. Trình duyệt sẽ gửi yêu cầu đến lớp giữa, lớp giữa sẽ phục vụ bằng cách tạo ra truy vấn và cập nhật cơ sở dữ liệu và tạo ra giao diện người dùng.

Ranh giới giữa một ứng dụng web và một website thực ra rất mong manh, và giữa 2 khái niệm có nhiều sự giao thoa dễ gây ra sự nhầm lầm. Để phân biệt 2 khái niệm này, chúng ta điểm lại một chút về khái niệm website trước.

Một website được viết lên dựa trên khung là HTML và các tài nguyên như hình ảnh, âm thanh, video,…, nó bao gồm nhiều trang web khác nhau. Tập hợp các webpage đơn lẻ như vậy được liên kết với nhau qua các đường dẫn HTML và hợp thành một website lớn và sử dụng chung một tên miền được gọi là Website. Website có tính tương tác thấp, ít chức năng, chỉ dùng để lưu giữ và hiển thị thông tin cho người truy cập chứ không có khả năng tương tác giữa người dùng với hệ thống hoặc giữa người dùng với nhau.

Trong khi đó, một ứng dụng web có thể hiểu là một ứng dụng chạy trên nền Web. Mà ứng dụng là một loại chương trình có khả năng làm cho máy tính thực hiện trực tiếp một công việc nào đó người dùng muốn thực hiện. Từ đó ta có thể hiểu ứng dụng web được tạo ra với mục đích cung cấp khả năng thực hiện những điều người dùng mong muốn, tăng khả năng tương tác giữa người dùng với hệ thống cũng như giữa người dùng với nhau. Webapp được tạo ra gồm cả 2 phần: phần Backend xây dựng bằng các ngôn ngữ lập trình web như PHP, Python, Java, .NET… còn phía người dùng sử dụng HTML/CSS/JS…

Lấy một ví dụ cho dễ hiểu, một trang báo mạng như Dân trí chẳng hạn. Người đọc báo khi truy cập vào dantri.com.vn thì họ được quyền xem thông tin các bài báo và bình luận về nội dung… cho nên đối với họ thì Dân trí là một website. Tuy nhiên admin hoặc biên tập viên của báo Dân trí thì sẽ coi nó là ứng dụng web. Bởi vì họ có quyền quản trị và được thực hiện nhiều thao tác trên hệ thống cũng như khả năng truy cập và sử dụng dữ liệu của dantri.com.vn.

Lập trình web là công việc tiếp nhận kết quả giao diện web của bên phía thiết kế Web kết hợp với yêu cầu của khách hàng để chuyển thành một hệ thống Website hoàn chỉnh. Website này có thể là Web tĩnh (chỉ có HTML, CSS, JS, không có người dùng hay là tương tác với cơ sở dữ liệu) hoặc là có sự tương tác với cơ sở dữ liệu cũng như khả năng tương tác với người dùng (hay còn gọi là web động – sử dụng các ngôn ngữ lập trình web như PHP, .NET, Ruby…).

Khác với thiết kế Web là công việc của người thiết kế – những người có khả năng sử dụng các phần mềm đồ họa và thiết kế như là Photoshop, AI, Flash… để xây dựng nên giao diện cho website dưới dạng ảnh hoặc một dạng tường minh chi tiết theo ý tưởng và yêu cầu của khách hàng.

Như vậy, chúng ta có thể hình dung luồng quy trình trong lập trình Web sẽ như sau: Những người thiết kế làm việc với khách hàng để cho ra giao diện Web bằng ảnh hoặc tệp photoshop, sau đó người lập trình viên sẽ biến giao diện này thành một website hoàn chỉnh theo yêu cầu khách hàng.

* 1. Vuejs

Các ứng dụng web ngày càng phong phú, đa dạng và luôn đưa trải nghiệm người dùng lên tiêu chí hàng đầu, nhờ vậy Javascript được ưu chuộng trong việc kết hợp xây dựng giao diện kể đến như React, Angular...Gần đây, Vue.js nổi lên như một hiện tượng mới, được Laravel phiên bản từ 5.3 gợi ý là một Javascript framework mặc định.

2.2.1. Vuejs là gì?

Theo như định nghĩa trên [trang chủ](https://vuejs.org/) thì Vue là một framework dùng để xây dựng giao diện người dùng (user interfaces).

Cụ thể, Vue là một progressive framework (framework linh động), cho phép và khuyến khích phát triển ứng dụng theo từng bước. Phần lõi của VueJS chủ yếu là vào phần View, ngoài ra có thể cài thêm các thành phần, thư viện hỗ trợ để đáp ứng nhu cầu xây dựng những ứng dụng SPA (Single-Page-Applications) .Khác với các monolithic framework (framework nguyên khối) cung cấp tất cả mọi thứ cần có để xây dựng app trong một framework duy nhất.

Ngoài ra cần chú ý một vấn đề nữa, VueJS sử dụng mô hình MVVM (Model- View- ViewModel), nghe có vẻ hơi lạ đúng không, hẹn các bạn trong bài viết khác sẽ tìm hiểu về nó và mô hình MVC.

2.2.2. Single-page Application (SPA)

SPA là một ứng dụng web hoặc trang web tương tác với người dùng bằng cách tự động thay đổi một phần nội dung của trang hiện tại thay vì tải toàn bộ trang mới từ máy chủ. Cách tiếp cận này tránh gián đoạn trải nghiệm người dùng giữa các trang tiếp theo, làm cho ứng dụng hoạt động giống như một ứng dụng máy tính để bàn. Trong một SPA, hoặc là tất cả các mã cần thiết - HTML, JavaScript, và CSS - được lấy ra với một trang tải hoặc tài nguyên thích hợp được tự động nạp và thêm vào trang khi cần thiết, thường là để phản hồi với hành động của người dùng. Trang không tải lại ở bất kỳ thời điểm nào trong quy trình, cũng như không kiểm soát chuyển sang trang khác.

Có rất nhiều bộ khung (framework) của javascript có thể triển khai một ứng dụng SPA như là VueJs, Angular, React,... , ở khóa luận này tôi đã sử dụng Angular 2.

2.2.3. Vì sao lại chọn Vuejs?

Hiện nay có rất nhiều các thư viện, framework Javascript mạnh mẽ, kể đến có React (hậu thuẫn bởi Facebook), Angular (hậu thuẫn bởi Google). Tuy nhiên Vue.js đang là sự lựa chọn lý tưởng cho các ứng dụng web ở mức vừa. Có một vài lí do như sau:

* Hiệu năng VueJS là thực sự đáng nể so với các đối thủ khác.
* VueJS có dung lượng tải thấp do chỉ giữ lại phần core, từ đó tăng đốc độ tải của toàn trang.
* VueJS đơn giản, dễ học, dễ áp dụng hơn, đặc biệt là với những người chưa có nhiều kiến thức nền.

2.2.4. Cài đặt Vuejs

Cách 1: CDN

<script src="https://cdn.jsdelivr.net/npm/vue@2.5.16/dist/vue.js"></script>

Cách 2: Tải file js và nhúng vào trang web

<script src="your-app/your-folder/vue.js"></script>

Cách 3: NPM

npm install vue

* 1. PHP và Laravel Framework
     1. PHP là gì?

PHP là viết tắt của chữ "Hypertext Preprocessor", đây là một ngôn ngữ lập trình được sử dụng để viết ở phía máy chủ (lập trình web). Hiện nay có rất nhiều ngôn ngữ có thể viết ở máy chủ như C#, Java, Python và PHP chính là một trong những ngôn ngữ của nhóm này.

PHP có cú pháp đơn giản, tốc độ nhanh và nhỏ gọn, dễ học và thời gian để tiếp cập PHP ngắn hơn so với các ngôn ngữ lập trình khác nên hiện nay PHP đang chiếm số lương lập trình viên tương đối lớn, có thể gọi là lớn nhất. Hiện PHP có rất nhiều các [CMS](https://freetuts.net/cms-la-gi-cac-cms-pho-bien-hien-nay-342.html) mã nguồn mở như WordPress, OpenCart, NukeViet nên nó rất đa dạng và đáp ứng hầu hết các website thông thường từ blog cá nhân cho đến website giới thiệu công ty và bán hàng.

* + 1. Laravel framework

https://viblo.asia/p/tim-hieu-ve-framwork-laravel-p1-amoG8191vz8P

* + - 1. Khái niệm

Laravel là một framework rõ ràng và ưu việt cho việc phát triển web PHP. Giải thoát bạn khỏi mã spaghetti, nó giúp bạn tạo ra những ứng dụng tuyệt vời, sử dụng syntax đơn giản. Phát triển thích thú những trải nghiệm sáng tạo chứ không phải là sự khó chịu.

Laravel được miêu tả như ‘một framework cho web artisan’. Theo tác giả Taylor Otwell, Laravel mang lại niềm vui cho việc lập trình bởi nó đơn giản, súc tích và đặc biệt là trình bày hợp lý.

* + - 1. Các tính năng chính

Laravel có 3 đặc tính nổi trội:

+ Đơn giản: các chức năng của Laravel rất dễ hiểu và thực hiện.

+ Ngắn gọn: hầu hết các chức năng của Laravel hoạt động liên tục với cấu hình rất nhỏ, dựa vào các quy tắc chuẩn để giảm bớt code-bloat

+ Trình bày hợp lý: hướng dẫn sử dụng Laravel rất đầy đủ và luôn cập nhật. Nhà lập trình, người tạo ra framework luôn cập nhật tài liệu trước khi cho ra một phiên bản mới, đảm bảo những người học lập trình luôn luôn có những tài liệu mới nhất.

* + - 1. Điều khiến Laravel trở nên khác biệt

**Bundles:**\_ là dành cho Laravel cũng như PEAR là cho PHP. Ngoài ra laravel đi kèm với công cụ command-line gọi là Artisan, giúp việc cài đặt các gói dễ dàng.

**Eloquent ORM:** là file thực thi PHP Active Record tiên tiến nhất hiện có.

**Migrations:**\_ trog laravel, migrations được xây dựng trong framework, chúng có thể thực hiện thông quan Artisan command-line.

**Unit-testing:** laravel là 1 framework tuyệt vời để tích hợp PHP Unit.

**Redis**: Laravel hỗ trợ Redis rất ngắn gọn.

2.3.2.4. Vòng đời Laravel

Yêu cầu HTTP từ Routed tới một Controller (routing nằm trong thư mục app/routes.php)

Controller sẽ thực hiện những action và gửi kết quả tới view (app/controllers)

View sẽ hiển thị những kiểu dữ liệu phù hợp và gửi lại HTTP Response (app/views)

**Chú ý:**

* Route có thể trả trực tiếp View hay Response, bỏ qua Controller
* Filters (app/filters.php) có thể xảy ra trước hoặc sau route
  1. MySQL
     1. Hệ quản trị cơ sở dữ liệu

Cơ sở dữ liệu là tập hợp các dữ liệu được tổ chức theo một cấu trúc nhất định để có thể dễ dàng quản lý. Hệ quản trị cơ sở dữ liệu là một chuơng trình phần mềm hay hệ thống được thiết kế để quản trị một cơ sở dữ liệu. Hệ quản trị cơ sở dữ liệu hỗ trợ khả năng lưu trữ, sửa chữa, xóa và tìm kiếm thông tin trong một cơ sở dữ liệu. Có nhiều loại hệ quản trị cơ sở dữ liệu khác nhau như MySQL, Oracle, PostgreSQL, SQL Server,…

* + 1. MySQL là gì

MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet...

Đặc điểm:

* Là một mã nguồn mở.
* Là một chương trình rất mạnh mẽ
* Sử dụng form chuẩn của ngôn ngữ dữ liệu SQL
* Làm việc trên nhiều hệ điều hành với nhiều ngôn ngữ
* Làm việc nhanh và khỏe với các tập dữ liệu lớn.
* Rất thân thiện, dễ dàng tìm hiểu để phát triển web
* Hỗ trợ các cơ sở dữ liệu lớn.

2.5 Nodejs

<https://viblo.asia/p/mot-cai-nhin-tong-quan-nhat-ve-nodejs-Ljy5VeJ3lra>

Nodejs là một **nền tảng** (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

2.6 Socketio

<https://viblo.asia/p/co-ban-ve-socketio-bJzKm0kY59N>

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

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

Phía server Đây là nơi sẽ cài đặt socket io. Ngôn ngữ để dựng server có thể là php, [asp.net](http://asp.net/), nodejs,... Tuy nhiên, tùy vào ngôn ngữ lựa chọn mà cách cấu trúc server khác nhau. Khuyến khích sử dụng nodejs để dựng server, vì như vậy có thể cài trực tiếp socketio vào cùng một server. Nếu sử dụng php thì phải cài thêm những package khác, hoặc phải chuẩn bị riêng server để chạy socketio.

Phía client: Ở phía client sẽ xây dựng giao diện người dùng. Ở đây có thể sử dụng js, hoặc các thư viện của js như jquery,...

2.7 Python

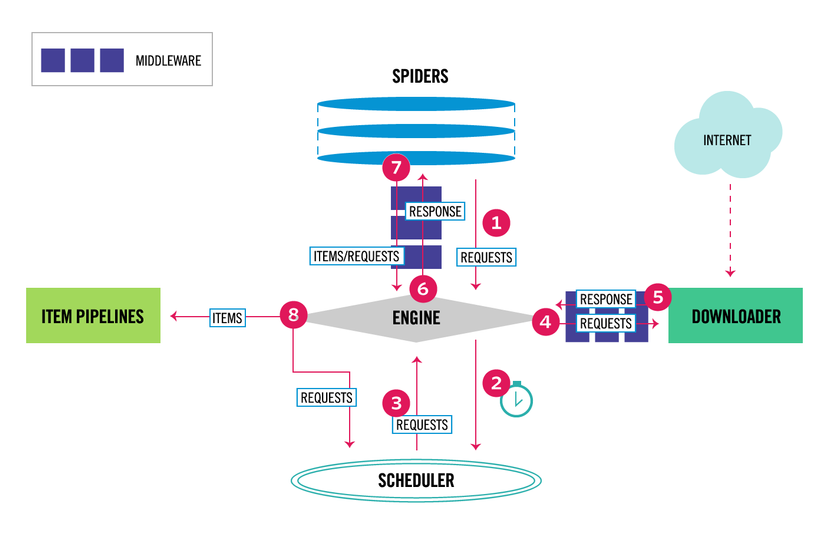
*Python là một ngôn ngữ lập trình mã nguồn mở đa mục đích, có hỗ trợ các phong cách lập trình cốt lõi khác nhau (chẳng hạn như các mô hình hướng đối tượng và thủ tục) mà triết lý thiết kế của nó rất thuận lợi cho việc*[đọc hiểu code](https://vinacode.net/2015/01/09/lap-trinh-vien-doc-hieu-code/)*, đơn giản và rõ ràng.*

*2.8 Scrapy*

[*https://en.wikipedia.org/wiki/Scrapy*](https://en.wikipedia.org/wiki/Scrapy)

[*https://viblo.asia/p/gioi-thieuhuong-dan-ve-crawler-voi-scrapy-framework-ByEZkWoEZQ0*](https://viblo.asia/p/gioi-thieuhuong-dan-ve-crawler-voi-scrapy-framework-ByEZkWoEZQ0)

*Scrapy là một khung thu thập dữ liệu web miễn phí và nguồn mở được viết bằng Python. Được thiết kế ban đầu để khai phá web, nó cũng có thể được sử dụng để trích xuất dữ liệu bằng API hoặc làm trình thu thập dữ liệu web có mục đích chung. Nó hiện đang được duy trì bởi Scrapinghub Ltd., một công ty dịch vụ và phát triển web-scraping.*



## **1. Thành phần**

### **Scrapy Engine**

Scrapy Engine có trách nhiệm kiểm soát luồng dữ liệu giữa tất cả các thành phần của hệ thống và kích hoạt các sự kiện khi một số hành động xảy ra

### **Scheduler**

Giống như một hàng đợi (queue), scheduler sắp xếp thứ tự các URL cần download

### **Dowloader**

Thực hiện dowload trang web và cung cấp cho engine

### **Spiders**

Spiders là class được viết bởi người dùng, chúng có trách nhiệm bóc tách dữ liệu cần thiết và tạo các url mới để nạp lại cho scheduler qua engine.

### **Item Pipeline**

Những dữ liệu được bóc tách từ spiders sẽ đưa tới đây, Item pipeline có nhiệm vụ xử lý chúng và lưu vào cơ sở dữ liệu

### **Các Middlewares**

Là các thành phần nằm giữa Engine với các thành phần khác, chúng đều có mục địch là giúp người dùng có thể tùy biến, mở rổng khả năng xử lý cho các thành phần. VD: sau khi dowload xong url, bạn muốn tracking, gửi thông tin ngay lúc đó thì bạn có thể viết phần mở rộng và sửa lại cấu hình để sau khi Dowloader tải xong trang thì sẽ thực hiện việc tracking.

#### **a. Spider middlewares**

Là thành phần nằm giữa Eninge và Spiders, chúng xử lý các response đầu vào của Spiders và đầu ra (item và các url mới).

#### **b. Dowloader middlewares**

Nằm giữa Engine và Dowloader, chúng xử lý các request được đẩy vào từ Engine và các response được tạo ra từ Dowloader

#### **c. Scheduler middlewares**

Nằm giữa Engine và Scheduler để xử lý những requests giữa hai thành phần

## **2. Luồng dữ liệu**

1. Khi bắt đầu crawl một website, Engine sẽ xác định tên miền và tìm vị trí của spider đó và yêu cầu spider đó tìm các urls đầu tiên để crawl
2. Engine nhận danh sách các urls đầu tiên từ spider, gửi cho Scheduler để sắp xếp
3. Engine yêu cầu danh sách cách urls tiếp theo từ Scheduler
4. Engine nhận danh sách các url tiếp theo từ Scheduler vào gửi đến Dowloader (requests)
5. Downloader nhận request và thực hiện việc tải trang, sau khi tải xong sẽ tạo một response và gửi lại Engine
6. Respone từ Dowloader sẽ được Engine đẩy qua Spiders để xử lý
7. Tại Spiders, khi nhận được response, chúng bóc tách thông tin từ response (tilte, content, author, date publish...) và những url có khả năng để crawl và đẩy lại cho Engine (requests)
8. Ở bước này, Engine nhận được kết quả từ Spiders sẽ thực hiện 2 công việc: đẩy những dữ liệu đã được bóc tách tới Item Pipeline để xử lý và lưu vào Databases, đẩy những url mới (requests) mới về Scheduler và quay về bước 3

2.9 Redis

<https://kipalog.com/posts/Tim-hieu-Redis--Phan-1>

[Redis](https://kipalog.com/posts/www.redis.io) là hệ thống lưu trữ key-value với rất nhiều tính năng và được [sử dụng rộng rãi](http://redis.io/topics/whos-using-redis). Redis nổi bật bởi việc hỗ trợ nhiều cấu trúc dữ liệu cơ bản (hash, list, set, sorted set, string), đồng thời cho phép scripting bằng ngôn ngữ lua. Bên cạnh lưu trữ key-value trên RAM với hiệu năng cao, redis còn hỗ trợ lưu trữ dữ liệu trên đĩa cứng (persistent redis) cho phép phục hồi dữ liệu khi gặp sự cố.

2.10 Elasticsearch

<https://viblo.asia/p/elasticsearch-la-gi-1Je5E8RmlnL>

Elasticsearch là một search engine.

* Elasticsearch được kế thừa từ Lucene Apache
* Elasticsearch thực chất hoặt động như 1 web server, có khả năng tìm kiếm nhanh chóng (near realtime) thông qua giao thức RESTful
* Elasticsearch có khả năng phân tích và thống kê dữ liệu
* Elasticsearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nên nó không phụ thuộc vào client viết bằng gì hay hệ thống hiện tại của bạn viết bằng gì. Nên việc tích hợp nó vào hệ thống bạn là dễ dàng, bạn chỉ cần gửi request http lên là nó trả về kết quả.
* Elasticsearch là 1 hệ thống phân tán và có khả năng mở rộng tuyệt vời (horizontal scalability). Lắp thêm node cho nó là nó tự động auto mở rộng cho bạn.
* Elasticsearch là 1 open source được phát triển bằng Java

Một số khái niệm cần biết:

1. Document

Document là một JSON object với một số dữ liệu. Đây là basic information unit trong ES. Hiểu 1 cách cơ bản thì đây là đơn vị nhỏ nhất để lưu trữ dữ liệu trong Elasticsearch.

2. Index

Trong Elasticsearch , sử dụng một cấu trúc được gọi là inverted index . Nó được thiết kế để cho phép tìm kiếm full-text search. Cách thức của nó khá đơn giản, các văn bản được phân tách ra thành từng từ có nghĩa sau đó sẽ đk map xem thuộc văn bản nào. Khi search tùy thuộc vào loại search sẽ đưa ra kết quả cụ thể.

## **3, Shard**

* Shard là đối tượng của Lucene , là tập con các documents của 1 Index. Một Index có thể được chia thành nhiều shard.
* Mỗi node bao gồm nhiều Shard . Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu.
* Chúng ta gần như không bao giờ làm việc trực tiếp với các Shard vì Elasticsearch đã support toàn bộ việc giao tiếp cũng như tự động thay đổi các Shard khi cần thiết.
* Có 2 loại Shard là : primary shard và replica shard.

## **4, Node**

* Là trung tâm hoạt động của Elasticsearch. Là nơi lưu trữ dữ liễu ,tham gia thực hiện đánh index cúa cluster cũng như thực hiện các thao tác tìm kiếm
* Mỗi node được định danh bằng 1 **unique name**

## **5, Cluster**

Chức năng chính của Cluster đó chính là quyết định xem shards nào được phân bổ cho node nào và khi nào thì di chuyển các Cluster để cân bằng lại Cluster

## **Ưu điểm**

* Tìm kiếm dữ liệu rất nhanh chóng, mạnh mẽ dựa trên Apache Lucene ( near-realtime searching)
* Có khả năng phân tích dữ liệu (Analysis data)
* Khả năng mở rộng theo chiều ngang tuyệt “vòi”
* Hỗ trợ tìm kiếm mờ (fuzzy), tức là từ khóa tìm kiếm có thể bị sai lỗi chính tả hay không đúng cú pháp thì vẫn có khả năng elasticsearch trả về kết quả tốt.
* Hỗ trợ Structured Query DSL (Domain-Specific Language ), cung cấp việc đặc tả những câu truy vấn phức tạp một cách cụ thể và rõ ràng bằng JSON.Các bạn có thể tìm hiểu thêm tại [đây](https://viblo.asia/p/query-dsl-trong-elasticsearch-Eb85oJq2l2G)
* Hỗ trợ nhiều Elasticsearc client như Java, PhP, Javascript, Ruby, .NET, Python

## **Nhược điểm**

* Elasticsearch được thiết kế cho mục đích search, do vậy với những nhiệm vụ khác ngoài search như CRUD thì elastic kém thế hơn so với những database khác như Mongodb, Mysql …. Do vậy người ta ít khi dùng elasticsearch làm database chính, mà thường kết hợp nó với 1 database khác.
* Trong elasticsearch không có khái niệm database transaction , tức là nó sẽ không đảm bảo được toàn vẹn dữ liệu trong các hoạt độngInsert, Update, Delete.Tức khi chúng ta thực hiện thay đổi nhiều bản ghi nếu xảy ra lỗi thì sẽ làm cho logic của mình bị sai hay dẫn tới mất mát dữ liệu. Đây cũng là 1 phần khiến elasticsearch không nên là database chính.
* Không thích hợp với những hệ thống thường xuyên cập nhật dữ liệu. Sẽ rất tốn kém cho việc đánh index dữ liệu.

# Phân tích yêu cầu bài toán

Chương này mô tả tổng quan về bài toán, thu thập các yêu cầu của người dùng về phần mềm, thiết kế các ca sử dụng và phân tích chi tiết ca sử dụng.

* 1. Mô tả bài toán

Bài toàn đặt ra là cần xây dựng một ứng dụng dựa trên ý tưởng của trang web Viblo giúp người dùng có thể đăng các bài viết để chia sẻ cũng như đặt ra các câu hỏi thắc mắc để mọi người giải đáp. Ứng dụng có người dùng là những người thích về công nghệ thông tin muốn trau dồi kiến thức và chia sẻ với cộng đồng những kiến thức của bản thân.

* 1. Thu thập yêu cầu

Yêu cầu người dùng (client)

* Là người dùng, tôi muốn đăng ký tài khoàn vào hệ thống để có thể sử dụng hệ thống
* Là người dùng, tôi muốn đăng nhập vào hệ thống để có thể sử dụng hệ thống
* Là người dùng tôi muốn xem được thông tin về tài khoản, họ tên, email của mình đề đảm bảo thông tin chính xác của tôi
* Là người dùng tôi muốn chỉnh sửa thông tin về tên, email, ảnh đại diện và mật khẩu của tài khoản của tôi.
* Là người dùng, tôi muốn tạo ra các bài viết cho tôi và mọi người cùng đọc trên hệ thống
* Là người dùng tôi muốn chỉnh sửa bài viết chính tôi tạo ra.
* Là người dùng, tôi muốn tạo ra các câu hỏi mà tôi thắc mắc để mọi người giải đáp.
* Là người dùng tôi muốn chỉnh sửa câu hỏi mà chính tôi tạo ra.
* Là người dùng tôi muốn bình chọn “thích” hoặc “không thích” một bài viết hay một câu hỏi.
* Là người dùng tôi muốn bình luận các bài viết mà tôi quan tâm.
* Là người dùng tôi muốn chỉnh sửa bình luận của tôi.
* Là người dùng tôi muốn xóa bình luận của tôi.
* Là người dùng tôi trả lời bình luận của người khác hoặc của tôi.
* Là người dùng tôi muốn theo dõi một người dùng khác.
* Là người dùng tôi muốn bỏ theo dõi một người dùng khác.
* Là người dùng tôi muốn đăng xuất ra khỏi ứng dụng.
* Là người dùng tôi muốn tìm kiếm các bài viết theo chủ đề mà tôi quan tâm.

Yêu cầu người quản trị (admin)

* Thêm, sửa, xóa người dùng
* Thêm, sửa xóa bài viết
* Thêm, sửa xóa câu hỏi
* Thêm, sửa xóa bình luận
* Thêm, sửa, xóa chủ đề bài viết
* Thêm, sửa, xóa các thẻ của bài viết
* Thêm, sửa, xóa các chức danh người dùng (role)
* Thêm, sửa, xóa các quyền của người dùng ( permission )
  1. Phân tích tác nhân chức năng của hệ thống

Sau khi thu thập yêu cầu của hệ thống, tôi đã xác định được các tác nhân của hệ thống gồm: Người dùng(client) và người quản trị(admin).

\* Yêu cầu chức năng

Để diễn tả yêu cầu chức năng của hệ thống, tôi chia chia chức năng theo từng vai trò của người sử dụng cho hợp lý.

Phân quyền người dùng:

Admin, Client có tài khoản và Client không có tài khoản

Admin sẽ được phân quyền theo từng vị trí (Super Admin, Editor, Author,…) và mỗi một vị trí sẽ có những quyền riêng biệt.

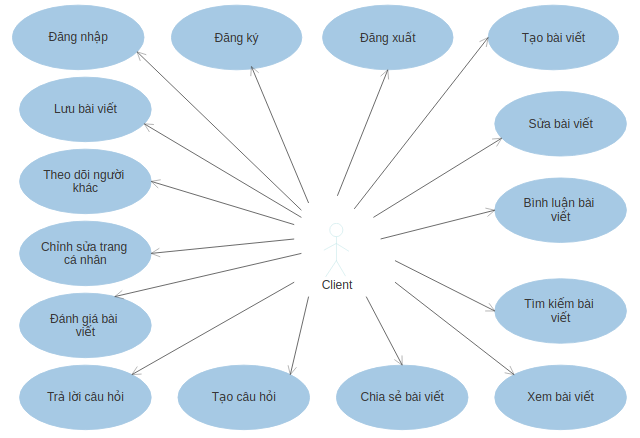
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | manager site | remove user | Edit post | read | comment | create post | publish post | manager post |
| Super Admin |  | x | x | x | x | x | x | x | x |
| Administrator |  |  | x | x | x | x | x | x | x |
| Editor |  |  |  | x | x | x |  |  |  |
| Author |  |  |  |  | x | x | x | x |  |
| Contributer |  |  |  |  |  | x | x |  |  |
| Subcriber |  |  |  |  | x | x |  |  |  |

Người quản trị: ( Giả sử có tất cả các quyền )

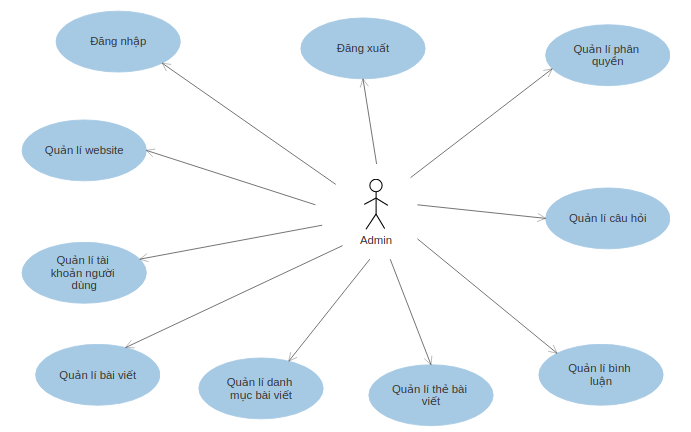
* Đăng nhập: Hệ thống mặc đinh sẽ tạo sẵn một tài khoản cho người quản trị, có thể tạo nhiều hơn một tài khoản người quản trị.
* Quản lý tài khoản: Người quản trị có trách nhiệm quản lý các tài khoản.
* Quản lý bài viết: Người quản trị có trách nhiệm quản lý các bài viết.
* Quản lý quyền: Người quản trị có trách nhiệm quản lý các quyền.
* Quản lý chủ đề: Người quản trị có trách nhiệm quản lý chủ đề bài viết.
* Quản lý thẻ: Người quản trị có trách nhiệm quản lý thẻ bài viết.
* Quản lý bình luận: Người quản trị có trách nhiệm quản lý bình luận.
* Quản lý câu hỏi: Người quản trị có trách nhiệm quản lý câu hỏi.

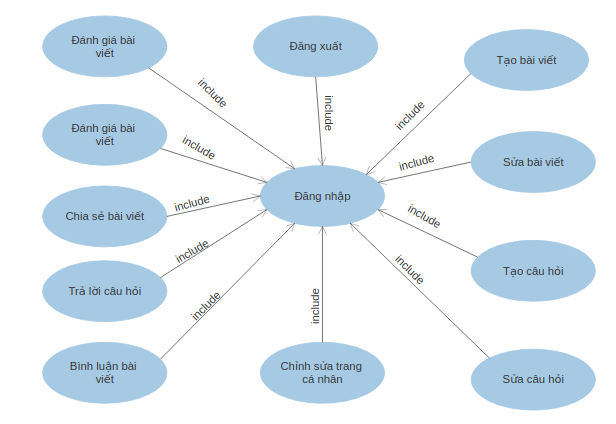
Người dùng là khách:

* Đăng nhập: Đăng nhập khi đã có tài khoản trên hệ thống.
* Đăng ký: Đăng ký tài khoản để sử dụng hệ thống.
* Đăng xuất: Đăng xuất tài khoản ra khỏi hệ thống.
* Tạo bài viết hoặc câu hỏi: Người dùng đã có tài khoản và đã đăng nhập có khả năng tạo bài viết hoặc tạo câu hỏi.
* Sửa bài viết hoặc câu hỏi: người dùng có khả năng sửa bài viết hoặc câu hỏi của chính mình.
* Cập nhật thông tin của người dùng: Khi người dùng muốn xem hoặc thay đổi thông tin cá nhân, hoặc người dùng có thể thay đổi mật khẩu cá nhân. Người dùng muốn cập nhật thông tin chỉ cần vào phần cài đặt để tiến hành xem và chỉnh sửa thông tin.
* Tìm kiếm bài viết: người dùng có khả năng tìm kiếm bài viết dựa vào chủ đề quan tâm.
* Theo dõi người dùng: Khi quan tâm đến một người khác, người dùng có thể theo dõi(follow) họ để có thể cập nhật được các bài viết mới nhất của họ.
* \* Yêu cầu phi chức năng
* Hệ thống cần được xây dựng với các yêu cầu phi chức năng như sau:
* Hệ thống được xây dựng trên nên tảng framework Laravel với phiên bản 5.7, Vuejs phiên bản 2.5 và Nodejs phiên bản 11.13
* Xác thực và phần quyền: Hệ thống cần có sự bảo mật với cái vai trò nhất định. Mỗi một chức năng trong hệ thống đề được phân quyền và người dùng chỉ có thê thực hiện các chức năng mà mình được phép.
* Triển khai hệ thống: Mã nguồn của hệ thống được lưu giữu trên github.
* Hiệu năng hoạt động của hệ thống: Hệ thống cần hoạt động ổn định với dữ liệu lên tới hàng nghìn người cùng truy cập và sử dụng, thời gian tương tác nhanh, tài nguyên sử dụng hợp lý.
* Tương thích: Tường thích trên trình duyệt của các thiết bị di dộng mà máy tính để bàn.
  1. Phân tích yêu cầu
     1. Thiết kế ca sử dụng



Hình 3.1: Mô hình ca sử dụng của người dùng

Hình 3.2: Mô hình ca sử dụng của quản trị viên



Hình 3.3 Ràng buộc giữa các use-case

* + 1. Phân tích biểu đồ ca sử dụng

Ca sử dụng đăng nhập

Bảng 3.1: Mô tả ca sử dụng đăng nhập

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng nhập vào ứng dụng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng nhập vào ứng dụng với tài khoản đã đăng ký | |
| **Điều kiện tiên quyết** | Thông tin của người dùng đã có trong cơ sở dữ liệu | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng nhấn vào đăng nhập (Login) ở trên thanh menu của màn hình chính. |
| 2 | Người dùng nhập thông tin email và mật khẩu đã đăng ký vào biểu mẫu có sẵn. |
| 3 | Người dùng bấm vào nút đăng nhập (Login) trên màn hình |
| 4 | Hệ thống trả về đăng nhập thành công |
| 5 | Hệ thống lưu sesson cho người dùng |
| **Luồng thay thế** | Tại bước 3 nếu thông tin người dùng nhập không chính xác hệ thống sẽ trả về thông báo lỗi. | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng đăng ký

Bảng 3.2: Mô tả ca sử dụng đăng ký

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng ký tài khoản vào ứng dụng để có quyền thực hiện tất cả các chức năng của người dùng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng ký tài khoản vào ứng dụng | |
| **Điều kiện tiên quyết** | Thông tin người dùng chưa có trong cơ sở dữ liệu | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng vào phần đăng ký tài khoản (Sign up) |
| 2 | Người dùng nhập thông tin về tên, email, mật khẩu theo đúng định dạng trên biểu mẫu ở màn hình |
| 3 | Người dùng bấm vào nút đăng ký (Sign Up) |
| 4 | Hệ thống trả về đăng ký thành công |
| **Luồng thay thế** | Tại bước 3 nếu người dùng đăng ký tài khoản theo email đã được sử dụng để đăng ký rồi thì hệ thống sẽ thông báo lỗi “email đã tồn tại” | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Đăng xuất

Bảng 3.3: Mô tả ca sử dụng đăng xuất

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Đăng xuất khỏi ứng dụng | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng đăng xuất khỏi ứng dụng | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng chọn Đăng xuất (Logout) |
| 2 | Hệ thống trả về đăng xuất thành công và trả về giao diện trang chủ cho người dùng |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Hệ thống gửi các client đã tải thông tin về người dùng trước đó thông điệp là người dùng này đã thoát khỏi hệ thống | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Tạo bài viết

Bảng 3.4: Mô tả ca sử dụng tạo bài viết

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Tạo ra một viết để học, lưu trữ và chia sẻ cho mọi người | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể tự tạo bài viết những chủ đề(category) và những thẻ(tag) | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập vào hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng chọn Tạo bài viết (Create Post) ở trên màn hình Home |
| 2 | Hệ thống trả về giao diện Tạo bài v |
| 3 | Người dùng nhập thông tin của bài viết bao gồm chủ đề(title), slug, nội dung bài viết theo đúng định dạng và gắn bài viết vào các chủ đề hoặc các thẻ. |
| 4 | Người dùng bấm vào nút Tạo bài viết(Create Post) trên giao diện |
| 5 | Hệ thống trả về bài viết thành công và trả về giao diện của bài viết vừa mới tạo |
| **Luồng thay thế** | Tại bước 4 nếu thông tin của bài viết chưa đầy đủ hoặc có lỗi xảy ra thì hệ thống giữ nguyên giao diện và thông báo lỗi cho người dùng | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Chỉnh sửa bài viết

Bảng 3.5: Mô tả ca sử dụng chỉnh sửa bài viết

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể chỉnh sửa những bài viết do chính người dùng tạo ra | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể chỉnh sửa các bài viết để thêm các chủ đề, thẻ(tag), chỉnh sửa publish hoặc chỉnh sửa nội dùng của bài viết | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập hệ thống và có quyền sửa bài viết | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của một bài viết, người dùng bấm vào icon option, sau đó chọn Chỉnh sửa(Edit) |
| 2 | Hệ thống trả về giao diện của trang chỉnh sửa bài viết |
| 3 | Người dùng chỉnh sử nội dung của bài viết là tiêu đề, slug, nội dung bài viết, gắn chủ đề, thẻ bài viết sau đó nhấn nút Cập nhật (Update Post) |
| 4 | Hệ thống cập nhật lại thông tin của bài viết, thông báo thành công và chuyển đến trang bài viết vừa chỉnh sửa |
| **Luồng thay thế** | Tại bước 3 nếu xảy ra lỗi, hệ thống sẽ thông báo lỗi cho người dùng và thông tin đang chỉnh sửa được giữ nguyên. | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Tìm kiếm bài viết

Bảng 3.6: Mô tả ca sử dụng tìm kiếm bài viết

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể tìm kiếm bài viết bằng các từ khóa | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể tìm kiếm các bài viết bằng từ khóa là tiêu để của bài viết | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện trang web, người dùng kích chọn vào thanh tìm kiếm. |
| 2 | Hệ thống tìm kiếm và trả về tất cả các bài viết một cách tự động với mỗi chữ cái người dùng điền vào |
| 3 | Hệ thống sẽ liệt kê tất cả kết quả tìm kiếm ngay dưới thanh tìm kiếm dưới dạng scroll bar |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng Chỉnh sửa trang cá nhân

Bảng 3.7: Mô tả ca sử dụng chỉnh sửa trang cá nhân

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người dùng có thể chỉnh sửa những thông tin trang cá nhân của mình | |
| **Tác nhân** | Người dùng | |
| **Mô tả** | Người dùng có thể chỉnh sửa thông tin về email, tên, mật khẩu và mô tả người dùng | |
| **Điều kiện tiên quyết** | Người dùng đã đăng nhập vào hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Người dùng kích chọn vào biểu tượng cá nhân góc phải trên màn hình, sau đó chọn phần cài đặt. |
| 2 | Hệ thống trả về giao diện của trang chỉnh sửa thông tin cá nhân của người dùng |
| 3 | Người dùng chỉnh sửa thông tin cá nhân gồm tên người dùng, mật khẩu, ảnh đại diện và nhấn vào nút Cập nhật(Update). |
| 4 | Hệ thống cập nhật thông tin và chuyển đến trang cá nhân của người dùng |
| **Luồng thay thế** | Nếu có bất cứ lỗi nào tại bước 3, hệ thống sẽ thông báo lỗi cho người dùng và mọi thông tin đang chỉnh sửa vẫn được giữ nguyên | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả bài viết

Bảng 3.8: Mô tả ca sử dụng xem tất cả bài viết

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Người quản trị viên có thể xem thông tin của tất cả các bài viết có trong hệ thống | |
| **Tác nhân** | Người quản trị viên | |
| **Mô tả** | Để nắm rõ hệ thống và quản lý tốt hơn, quản trị viên có thể xem tất cả các bài viết của hệ thống chứa các thông tin về chủ đề, mô tả,... | |
| **Điều kiện tiên quyết** | Không có | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của quản trị viên, quản trị viên truy cập vào mục tất cả bài viết (Post Manager) |
| 2 | Hệ thống trả về giao diện của trang chứa tất cả các bài viếtcủa hệ thống. |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả người dùng

Bảng 3.9: Mô tả ca sử dụng xem tất cả người dùng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Quản trị viên có thể xem thông tin của tất cả người dùng trong hệ thống | |
| **Tác nhân** | Quản trị viên | |
| **Mô tả** | Quản trị viên có thể xem thông tin chi tiết của người dùng để có thể quản lý | |
| **Điều kiện tiên quyết** | Quản trị viên có quyền xem tất cả người dùng của hệ thống | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện của quản trị viên, quản trị viên truy cập vào người dùng (User Manager) |
| 2 | Hệ thống trả về giao diện của trang tất cả người dùng |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

Ca sử dụng xem tất cả phản bình luận

Bảng 3.10: Mô tả ca sử dụng xem tất cả bình luận của trang web

|  |  |  |
| --- | --- | --- |
| **Mục đích** | Quản trị viên sẽ xem tất cả các phản hồi của người dùng về bài viết | |
| **Tác nhân** | Quản trị viên | |
| **Mô tả** | Quản trị viên xem tất cả các phản hồi để xem xét có xóa các bộ thẻ không | |
| **Điều kiện tiên quyết** | Quản trị viên có quyền xem bình luận bài viết | |
| **Luồng sự kiện** | STT | Sự kiện |
| 1 | Trên giao diện trang web của người quản trị, người dùng vào trang tất cả phản hồi (List Reports) |
| 2 | Hệ thống trả về giai diện tất cả phản hồi của người dùng chứa các thông tin về người phản hồi, bộ thẻ bị phản hồi. |
| **Luồng thay thế** | Không có | |
| **Điều kiện sau** | Không có | |
| **Điểm mở rộng** | Không có | |

# Giải quyết bài toán với Laravel, Vuejs

Để thực hiện các chức năng, yêu cầu như trên, tôi đã sử dụng các công nghệ mới như Laravel và Vuejs để giải quyết. Ứng dụng gồm có hai phần là frontend và backend. Trước tiền, phần frontend sử dụng Vuejs sẽ giúp hệ thống cải thiện được tốc độ và hiệu năng sự dụng, tăng trải nghiệm người dùng. Đối với backend, tôi sử dụng PHP, cụ thể là Laravel framework kết nối cơ sở dữ liệu và tạo ra các api chứa các thông tin cần thiết cho phần front end. Ngoài ra tôi còn sử dụng socketio trong nodejs và redis để thực hiện chức năng realtime. Thêm vào đó là kĩ thuật thu thập dữ liệu sử dụng Scrapy framework kết hợp với Elasticsearch để giúp cải thiện phần tìm kiếm gần như là realtime.

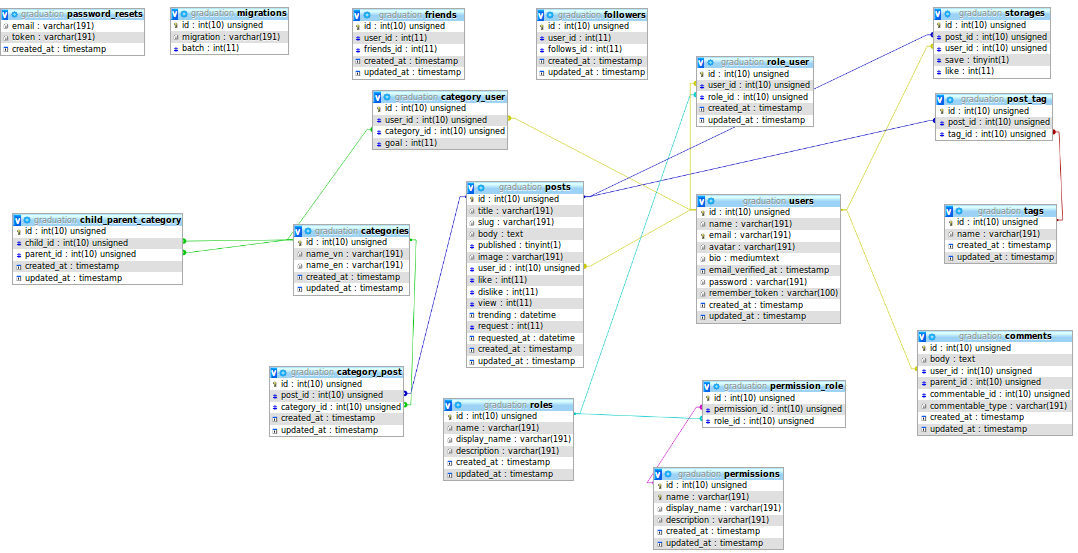
* 1. Thiết kế phần mềm
     1. Thiết kế giao diện

Giao diện được thiết kế sao cho đơn giản, hiệu quả và dễ sử dụng thân thiện với người dùng.

Các màn hình được thiết kế sẽ được giới thiệu ở phần Kết quả thực nghiệm ở chương 5.

* + 1. Thiết kế cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu được sử dụng là MySQL. Nó rất linh động, dễ mở rộng. Mô hình cơ sở dữ liệu được giải thích trực quan như dưới đây:



Hình 4.4: Mô hình dữ liệu trong cơ sở dữ liệu MySQL

Users

Bảng này lưu các thông tin về người dùng.

Bảng 4.11: Mô tả bảng Users

|  |  |
| --- | --- |
| Từ khóa | Ý Nghĩa |
| **Id** | Định danh người dùng |
| **Name** | Tên người dùng |
| **Email** | Địa chỉ email người người dùng |
| **Avatar** | Ảnh đại diện của người dùng |
| **Bio** | Miêu tả những thông tin, giới thiệu khái quát về bản thân người dùng |
| Password | Mật khẩu người dùng |
| Created at | Thời gian tạo tài khoản |
| Updated at | Thời gian cập nhật tài khoản |

Posts

Bảng này lưu các thông tin về bài viết. Mỗi bộ thẻ phải thuộc về một người dùng.

Bảng 4.12: Mô tả bảng Posts

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh bài viết |
| **user\_id** | Định danh người dùng tạo ra bài viết |
| **image** | Ảnh của bài viết |
| **slug** | url để định một trang |
| **body** | Nội dung bài viết |
| Published | Trạng thái công khai của bài viết |
| Like | Số lượng thích của bài viết |
| Dislike | Số lượng không thích của bài viết |
| **View** | Lượt xem của bài viết |
| Trending | Xác định bài viết có hot không |
| Request | Trạng thái yêu cầu published bài viết |
| Requested\_at | Thời gian yêu cầu |
| Created\_at | Thời gian tạo bài viết |
| Updated\_at | Thời gian cập nhật bài viết |

**Category**

Bảng này lưu thông tin về các chủ đề của bài viết

Bảng 4.13: Mô tả bảng Category

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của chủ đề |
| **Name\_vn** | Tên của chủ đề theo ngôn ngữ tiếng việt |
| **Name\_en** | Tên của chủ đề theo ngôn ngữ tiếng anh |
| **Created\_at** | Thời gian tạo ra chủ đề |
| **Updated\_at** | Thời gian cập nhật chủ đề |

Tags

Bảng này lưu thông tin về thẻ của bài viết

Bảng 4.14: Mô tả bảng Tags

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của thẻ |
| **Name** | Tên của thẻ |
| Slug | url để định dạng |
| created\_at | Thời gian tạo thẻ |
| Updated\_at | Thời gian cập nhật thẻ |

Question

Bảng này lưu các thông tin về các câu hỏi

Bảng 4.15: Mô tả bảng Questions

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh bài viết |
| **user\_id** | Định danh người dùng tạo ra bài viết |
| **image** | Ảnh của bài viết |
| **slug** | url để định một trang |
| **body** | Nội dung bài viết |
| Published | Trạng thái công khai của bài viết |
| Like | Số lượng thích của bài viết |
| Dislike | Số lượng không thích của bài viết |
| **View** | Lượt xem của bài viết |
| Trending | Xác định bài viết có hot không |
| Request | Trạng thái yêu cầu published bài viết |
| Requested\_at | Thời gian yêu cầu |
| Created\_at | Thời gian tạo bài viết |
| Updated\_at | Thời gian cập nhật bài viết |

Category\_Post

Bảng này lưu các thông tin về mối quan hệ giữa bài viết và chủ đề.

Một bài viết có thể thuộc nhiều chủ đề.

Một chủ đề có thể thuộc nhiều bài viết

Bảng 4.16: Mô tả bảng Category\_Post

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của mối quan hệ chủ đề-bài viết |
| **Post\_id** | Định danh bài viết |
| **Category\_id** | Định danh của chủ đề |
| **Created\_at** | Thời gian tạo |
| Updated\_at | Thời gian cập nhật |

Category\_Question

Bảng này lưu các thông tin về mối quan hệ giữa câu hỏi và chủ đề.

Một câu hỏi có thể thuộc nhiều chủ đề.

Một chủ đề có thể thuộc nhiều câu hỏi

Bảng 4.17: Mô tả bảng Category\_Question

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của mối quan hệ chủ đề-câu hỏi |
| **Question\_id** | Định danh câu hỏi |
| **Category\_id** | Định danh của chủ đề |
| **Created\_at** | Thời gian tạo |
| Updated\_at | Thời gian cập nhật |

Post\_Tag

Bảng này lưu các thông tin về mối quan hệ giữa bài viết và thẻ.

Một bài viết có thể thuộc nhiều thẻ.

Một thẻ có thể thuộc nhiều câu hỏi

Bảng 4.18: Mô tả bảng Post\_Tag

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của mối quan hệ bài viết-thẻ |
| **Post\_id** | Định danh bài viết |
| **Tag\_id** | Định danh của thẻ |

Question\_Tag

Bảng này lưu các thông tin về mối quan hệ giữa câu hỏi và thẻ.

Một câu hỏi có thể thuộc nhiều thẻ.

Một thẻ có thể thuộc nhiều câu hỏi

Bảng 4.19: Mô tả bảng Question\_Tag

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của mối quan hệ bài viết-thẻ |
| **Question\_id** | Định danh câu hỏi |
| **Tag\_id** | Định danh của thẻ |

Followers

Bảng này lưu các thông tin về mối quan hệ giữa người dùng và người dùng.

Một người dùng có thể theo dõi nhiều người dùng khác.

Một người dùng có thể được theo dõi bởi nhiều người dùng khác.

Bảng 4.20: Mô tả bảng Followers

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của mối quan hệ người dùng và người dùng |
| **User\_id** | Định danh người dùng theo dõi |
| **Follow\_id** | Định danh người dùng được theo dõi |

Permissions

Bảng này lưu các thông tin về các quyền của từng một chức năng trong website

Bảng 4.21: Mô tả bảng Permission

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của chức năng |
| **Name** | Tên của chức năng |
| display\_name | Tên hiển thị của chức năng |
| Description | Giới thiệu về chức năng |
| created\_at | Thời gian tạo chức năng |
| Updated\_at | Thời gian cập nhật chức năng |

Roles

Bảng này lưu các thông tin về các chức vụ của những người dùng trong website

Bảng 4.22: Mô tả bảng Roles

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của chức vụ |
| **Name** | Tên của chức vụ |
| display\_name | Tên hiển thị của chức vụ |
| Description\_name | Giới thiệu về chức vụ |
| created\_at | Thời gian tạo chức vụ |
| Updated\_at | Thời gian cập nhật chức vụ |

Permission\_Role

Bảng này lưu các thông tin về mối quan hệ giữa chức vụ và chức năng.

Một chức vụ có thể thực hiện nhiều chức năng khác nhau.

Một chức năng có thể thuộc nhiều chức vụ.

Bảng 4.23: Mô tả bảng Permission\_Role

|  |  |
| --- | --- |
| Từ khóa | Ý nghĩa |
| **Id** | Định danh của chức vụ |
| **Permission\_id** | Định danh của chức năng |
| Role\_id | Định danh của chức vụ |

* 1. Triển khai phần mềm:
     1. Ứng dụng Vuejs để xây dựng phía máy khách

Xây dựng cấu trúc ứng dụng

Vuejs chịu trách nhiệm lấy dữ liệu từ máy chủ sau đó hiển thị ra giao diện cho người dùng.

Router

Các router định tuyến ứng dụng theo các thao tác của người dùng. Chúng được khai báo trong tập tin router.js . Các router của ứng dụng được liệt kê dưới bảng sau:

Bảng 4.24: Bảng định nghĩa các định tuyến của ứng dụng

|  |  |  |
| --- | --- | --- |
| **STT** | **Đường dẫn** | **Chức năng** |
| 1 | /register | Đăng ký tài khoản của người dùng |
| 2 | /login | Đăng nhập vào hệ thống |
| 4 | /u/{email} | Thông tin cá nhân của người dùng |
| 5 | /profile | Chỉnh sửa thông tin cá nhân của người dùng |
| 6 | /publish/post | Tạo mới một bài viết |
| 7 | / | Trang chủ của website |
| 8 | /questions | Trang danh sách câu hỏi |
| 9 | /authors | Trang danh sách người dùng |
| 10 | /p/{slug} | Trang một bài viết cụ thể |
| **11** | /p/{slug}/edit | Trang chỉnh sửa một bài viết cụ thể |
| **12** | /trending | Danh sách các bài viết hot |
| **13** | /t/{slug} | Trang thông tin một thẻ(tag) cụ thể |
| **14** | /t/{slug}/questions | Trang thông tin các câu hỏi có gắn thẻ(tag) đó |
|  |  |  |
| **14** | /u/{email} | Trang thông tin một người dùng cụ thể |
|  | /u/{email}/questions | Liệt kê các câu hỏi do người dùng tạo ra |
|  | /admin | Trang chủ Admin |
|  | /m/users | Trang quản lí người dùng |
|  | /m/posts | Trang quản lí bài viết |
|  | /m/tags | Trang quản lí thẻ bài viết |
|  | /m/categories | Trang quản lí chủ đề bài viết |
|  | /m/roles | Trang quản lí chức danh trong website |
|  | /m/comments | Trang quản lí bình luận của website |
|  |  |  |
|  | /m/questioins | Trang quản lí câu hỏi |
|  | /q/{slug}/edit | Trang chỉnh sửa câu hỏi |
| 11 | /q/{slug} | Trang một câu hỏi cụ thể |
| 12 | /logout | Đăng xuất ra khỏi hệ thống |

Component

Vuejs có cấu trúc hướng thành phần với các thành phần trong ứng dụng. Ứng dụng của tôi có các component sau:

Phần Client:

* App.vue: Hiển thị giao diện trang chủ với các thanh tiêu đề để người dùng dễ dàng tuơng tác.
* Notification.vue: Hiện thị thông báo cho người dùng
* Auth
  + auth/Login.vue: Hiển thị giao diện đăng nhập
  + auth/Register.vue: Hiển thị giao diện đăng kí
* user/User.vue: Hiển thị giao diện chứa dữ liệu của người dùng

user/Post.vue: Hiện thị danh sách bài viết của người dùng

* + user/Question.vue: Hiển thị danh sách câu hỏi của người dùng
  + user/Follower: Hiển thị danh sách những người đang theo dõi mình
  + user/Follow: HIển thị danh sách những người mình đang theo dõi
  + user/Draft: Hiển thị các bài viết chưa công khai (private)
  + user/Save: Hiển thị các bài viết đã lưu

\*\*\* post/Post.vue: Hiển thị giao diện chứa thông tin chung cho bài viết

post/Create.vue: Hiển thị giao diện tạo bài viết mới

post/EditSinglePost: Hiển thị giao diện chỉnh sửa bài viết-thẻ

post/NewList : Hiển thị giao diện danh sách bài viết mới nhất

post/Following: Hiển thị giao diện chứa các bài viết thuộc sở hữu của những người mà mình theo dõi

post/Trending: Hiển thị giao diện chứa các bài viết được gắn mác “Trending”

post/Comment: Hiển thị giao diện bình luận

post/CommentForm: Hiển thị giao diện điền thông tin bình luận

post/CommentReplies: Hiển thị giao diện bình luận con

post/NewestQuestion: Hiển thị giao diện chứa 5 câu hỏi mới nhất

post/SinglePost: Hiển thị giao diện chứa nội dung một bài viết cụ thể

\*\*\* question/Question: Hiển thị giao diện chứa các thông tin chung của phần câu hỏi

question/Newest: Hiển thị danh sách câu hỏi

question/NewestPost: Hiển thị danh sách 5 bài viết mới nhất

question/Single: Hiển thị giao diện chứa nội dung một câu hỏi cụ thể

question/Create: Hiển thị giao diện tạo câu hỏi mới

question/EditQuestionSingle: Hiển thị giao diện chỉnh sửa câu hỏi

question/Comment: Hiển thị giao diện bình luận của câu hỏi

question/CommentReplies: Hiển thị giao diện bình luận con

question/CommentForm: Hiển thị giao diện để người dùng điền nội dung bình luận

\*\*\* author/Author: Hiển thị danh sách các tác giả

\*\*\* tag/Single: Hiển thị giao diện chứa thông tin chung

tag/PostBaseTag: Hiển thị giao diện các bài viết được gán thẻ(tag) đó

tag/QuestionBaseTag: Hiển thị giao diện các câu hỏi được gán thẻ(tag) đó

Phần Admin:

App: HIển thị giao diện dùng chung của trang quản lí admin

Post: Hiển thị giao diện quản lí bài viết-thẻ

Question: Hiển thị giao diện quản lí câu hỏi

Tag: Hiển thị giao diện quản lí thẻ

Category: Hiển thị giao diện quản lí chủ đề

Role: Hiển thị giao diện quản lí chức danh

User: Hiển thị giao diện quản lí người dùng

Comment: Hiển thị giao diện quản lí bình luận

Profile: Hiển thị giao diện quản lí trang cá nhân

PostProfile: Quản lí các bài viết cá nhân

SettingProfile: Quản lí thông tin cá nhân

* + 1. Xây dựng backend với Laravel

Máy chủ sử dụng php để kết nối phía client với MySQL. Với php tôi đã sử dụng Laravel là framework để xây dựng các router, và các mô hình dữ liệu theo đúng chuẩn, sau đây là bảng mô tả API của hệ thống.

Bảng 4.25: Bảng mô tả API

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **API** | **Phương thức** | **Giải thích** |
| 1 | /m/users | GET | Lấy dữ liệu tất cả người dùng |
| 2 | /m/user | POST | Thêm người dùng |
| 3 | /m/user/{user} | DELETE | Xóa người dùng |
| 4 | /m/user/{user} | PUT | Chỉnh sửa người dùng |
| 5 | /user/current | GET | Lấy thông tin người dùng đăng nhập |
| 6 | /user/follows | GET | Lấy thông tin những người dùng đăng nhập theo dõi |
| 7 | /user/follow/{user} | DELETE | Xóa một người dùng theo dõi |
| 8 | /user/posts | POST | Lấy tất cả bài viết của người dùng đã đăng nhập |
| 9 | /user/countFollow | GET | Đếm số người mà mình đã theo dõi |
| 10 | /user/countFollower | GET | Đếm số người đã theo dõi mình |
| 11 | /user/countPost | GET | Đếm số bài viết của người dùng đã đăng nhập |
| 12 | /m/roles/all | GET | Lấy tất cả chức danh |
| 13 | /user/userPermission | GET | Lấy thông tin các quyền của người dùng đã đăng nhập |
| 14 | /profile | GET | Lấy thông tin cá nhân |
| 15 | /profile | PUT | Chỉnh sửa thông tin cá nhân |
| 16 | /m/tags | GET | Lấy toàn bộ thông tin các thẻ (có phân trang) |
| 17 | /m/tags/all | GET | Lấy toàn bộ thông tin thẻ(không phân trang) |
| 18 | /m/tag | POST | Thêm mới một thẻ |
|  | /m/tag/{tag} | PUT | Chỉnh sửa thẻ |
|  | /m/tag/{tag} | DELETE | Xóa một thẻ |
|  | m/categories | GET | Lấy toàn bộ thông tin chủ đề(có phân trang) |
|  | m/categories/all | GET | Lấy toàn bộ thông tin chủ đề(không phân trang) |
|  | m/category | POST | Tạo mới một chủ đề |
|  | m/category/{category} | PUT | Chỉnh sửa một chủ đề |
|  | m/category/{category} | DELETE | Xóa một chủ đề |
|  | m/roles | GET | Lấy toàn bộ thông tin chức danh(có phân trang) |
|  | m/roles | POST | Tạo mới một chức danh |
|  | m/role/{role} | PUT | Chỉnh sửa một chức danh |
|  | m/role/{role} | DELETE | Xóa một chức danh |
|  | m/posts | GET | Lấy toàn bộ bài viết |
|  | m/posts | POST | Thêm mới một bài viết |
|  | m/post/{post} | PUT | Chinh sửa một bài viết |
|  | m/post/{post} | DELETE | Xóa một bài viết |
|  | m/questions | GET | Lấy toàn bộ câu hỏi |
|  | m/questions | POST | Thêm mới một câu hỏi |
|  | m/question/{question} | PUT | Chỉnh sửa một câu hỏi |
|  | m/question/{question} | DELETE | Xóa một câu hỏi |
|  | m/comments | GET | Lấy toàn bộ bình luận |
|  | m/comments | POST | Thêm một bình luận |
|  | m/comment/{comment} | PUT | Chỉnh sửa một bình luận |
|  | m/comment/{comment} | DELETE | Xóa một bình luận |
|  | u/checkFollow/{email} | GET | Kiểm tra trạng thái theo dõi |
|  | u/removeFollow/{email} | POST | Xóa trạng thái theo dõi |
|  | u/addFollow/{email} | POST | Thêm trạng thái theo dõi |
|  | p/{post}/checkVote | GET | Kiểm tra trạng thái với một bài viết cụ thể (upvote, downvote,…) |
|  | p/{post}/upvote | POST | Tăng lượng vote cho bài viết |
|  | p/{post}/removeUpvote | POST | Bỏ tăng vote bài viết |
|  | p/{post}/downvote | POST | Giảm lượng vote bài viết |
|  | p/{post}/removeDownvote | POST | Bỏ giảm lượng vote cho bài viết |
|  | p/{post}/comment | POST | Bình luận một bài viết |
|  | p/{post}/makeTrending | POST | Gán “trending” cho một bài viết |
|  | p/following | GET | Lấy danh sách bài viết được tạo bởi những người mình đang theo dõi |
|  | u/checkSavePost | POST | Kiểm tra xem bài viết đã được lưu chưa |
|  | u/savePost | POST | Lưu bài viết |
|  | u/getStorages | POST | Lấy thông tin trang thái lưu bài viết |
|  | c/newestposts | GET | Lấy tất cả bài viết mới nhất |
|  | p/getPostBaseId | GET | Lấy bài viết dựa vào id |
|  | p/trending | GET | Lấy tất cả bài viết được gán “trending” |
|  | p/slidebar | GET | Lấy bài viết cho phần bên phải màn hình |
|  | p/{post} | GET | Lấy bài viết dựa vào slug |
|  | p/{post}/comments | GET | Lấy tất cả bình luận của bài viết |
|  | p/{post}/countVote | GET | Đếm lượt vote của bài viết |
|  | t/{tag} | GET | Lấy thẻ dựa vào slug |
|  | t/{tag}/posts | GET | Lấy các bài viết dựa vào thẻ |
|  | t/{tag}/questions | GET | Lấy các câu hỏi dựa vào thẻ |
|  | u/{email}/follows | GET | Trả về danh sách những người mà mình theo dõi |
|  | u/{email}/followers | GET | Trả về danh sách những người theo dõi mình |
|  | u/{email} | GET | Trả về dữ liệu người dùng dựa vào email |
|  | u/{email}/posts | GET | Trả về tất cả bài viết của người dùng dựa vào email |
|  | u/{email}/drafts | GET | Trả về tất cả bài viết nháp của người dùng dựa vào email |
|  | u/{email}/questions | GET | Trả về tất cả câu hỏi của người dùng dựa vào email |
|  | Authors | GET | Trả về dữ liệu của tất cả người dùng |
|  | u/{user}/getNotifications | GET | Trả về thông tin thông báo của người dùng |

**Lấy dữ liệu từ việc crawl trang web Viblo**

Tôi đã lấy khoảng 3000 bản ghi bài viết và 500 bản ghi câu hỏi từ trang web Viblo.

Công nghệ tôi đã sử dụng đó là Scrapy.

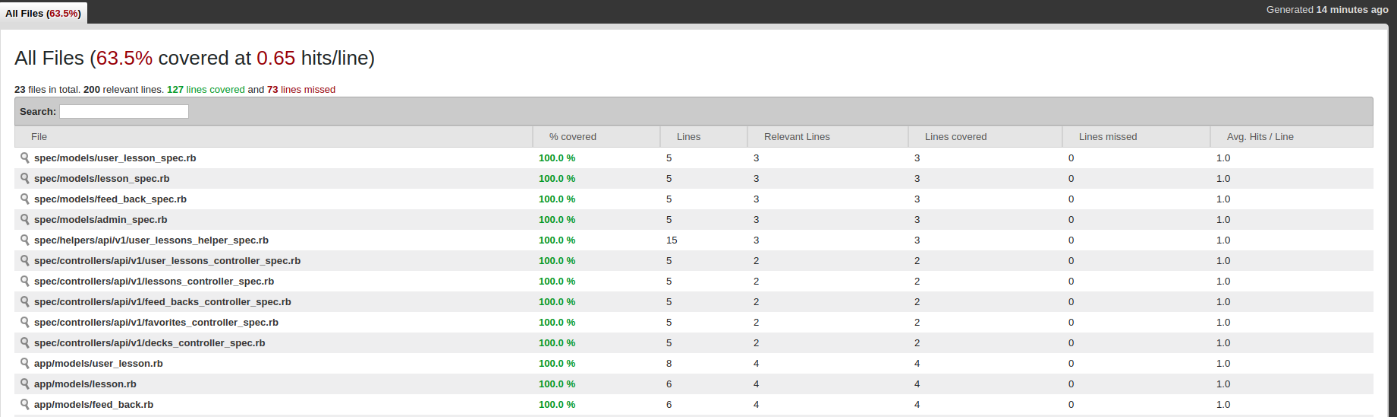
Chức năng thông báo cho người dùng:

Thông báo cho người dùng bằng việc gửi mail khi người dùng đăng kí một tài khoản mới.

Sử dụng websocket để xây dựng ứng dụng thười gian thực. Đầu tiên, websockets là giao thức được xây dựng dựa trên TCP protocol. Nó sẽ thiết lập kết nối giữa máy khách và máy chủ, duy trì kết nối đó liên tục để khi có một người dùng bình luận bài viết sẽ được thông báo ngay lập tức tới tác giả của bài viết đó.

* 1. Kiểm thử tự động

Trong qúa trình xây dựng và phát triển phần mềm, kiểm thử có một ý nghĩa và vai trò vô cùng quan trọng. Môi trường phát triển hiện nay đang hướng tới: "Quá trình phát triển ngắn", "kinh phí thấp", "chất lượng cao". Hiện nay, quá trình test đã chiểm đến gần 40% thời gian phát triển phần mềm Nhằm khắc phục thực trạng này, kiểm thử tự động càng được chú ý đến, đặc biệt là phương pháp kiểm thử "unit test". Tôi đã sử dụng Rspec để kiểm thử các chức năng của ứng dụng của tôi dựa theo mô hình BDD (Behavior Driven Development) để làm quy trình phát triển phần mềm. BDD được hiểu đơn giản là quá trình vòng lặp với 3 quá trình: viết kiểm thử lỗi bởi vì lúc này viết mã kiểm thử nhưng chưa có mã nguồn nên sẽ bị lỗi, sửa mã nguồn để kiểm thử thành công và chỉnh sửa lại mã nguồn để tối ưu hóa và loại bỏ mã dư thừa. Trong quá trình phát triển phần mềm, tôi nhận thấy càng đầu tư nhiều thời gian để viết kiểm thử thì việc chỉnh sửa mã nguồn và quản lí toàn bộ dự án càng trở nên dễ dàng, giảm gánh nặng khi cần chỉnh sửa code, đồng thời giam thiểu được lỗi của ứng dụng. Rspec là một công cụ kiểm thử dành cho Ruby, tuy nó rất mạnh nhưng lại có thể sử dụng đơn giản và hiệu quả. Tôi đã áp dụng gem simplecov (một thư viện chuyên dùng để đánh giá chất lượng kiểm thử trong dự án) để đánh giá hiệu quả của việc kiểm thử ứng dụng của tôi. Nó thống kê số lần mỗi dòng của mã ứng dụng được chạy trong quá trình viết kiểm thử. Kết quả đạt được thể hiện trong hình dưới đây.



Hình 4.5: Kết quả chạy kiểm thử tự động

# Thực nghiệm

Chương 5 mô tả môi trường thực nghiệm, dữ liệu thực nghiệm và kết quả thực nghiệm trên các môi trường cùng với sự phản hồi của những người dùng thử ứng dụng.

* 1. Môi trường thực nghiệm

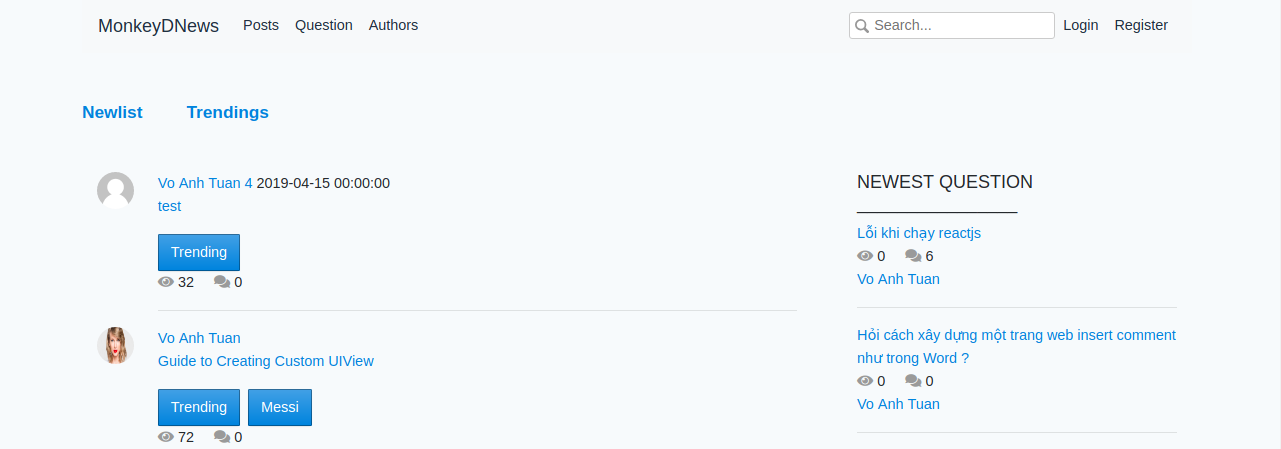
Môi trường thử nghiệm bao gồm như sau:

* Laptop hệ điều hành ubuntu với các trình duyệt chrome, firefox.
  1. Dữ liệu thực nghiệm

2500 bài viết và 500 câu hỏi được crawl từ trang trang web Viblo

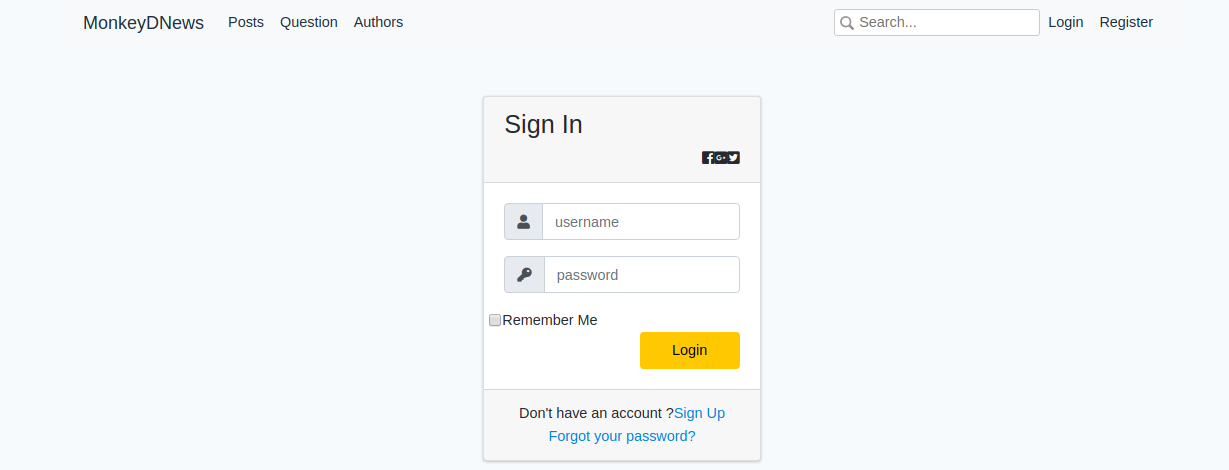
* 1. Kết quả thực nghiệm

Trang chủ của hệ thống



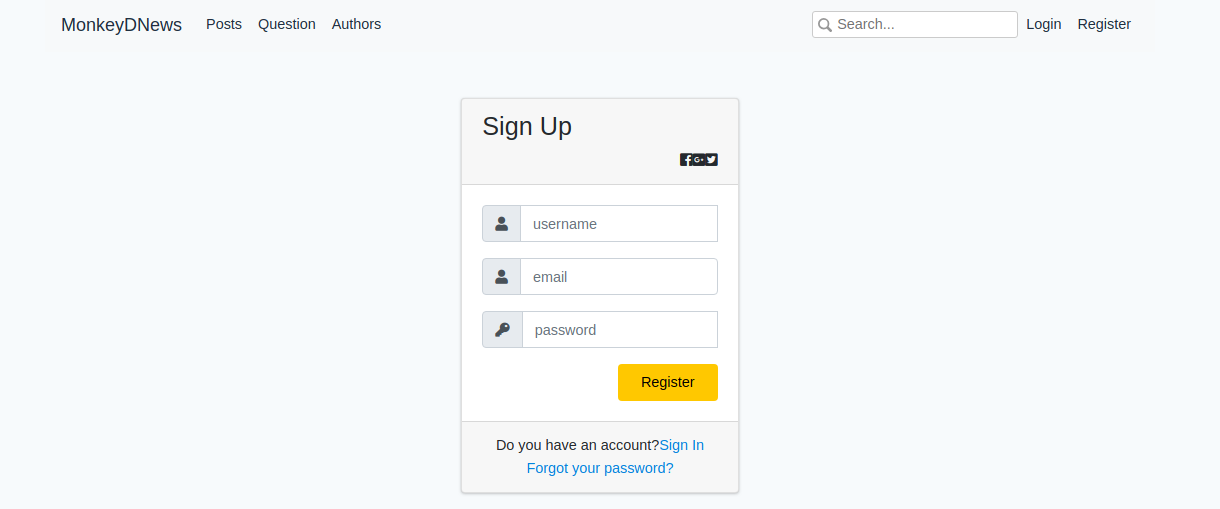
* + 1. Xác thực và ủy quyền ngừoi dùng

Giao diện đăng nhập vào hệ thống



Hình 5.6: Giao diện đăng nhập

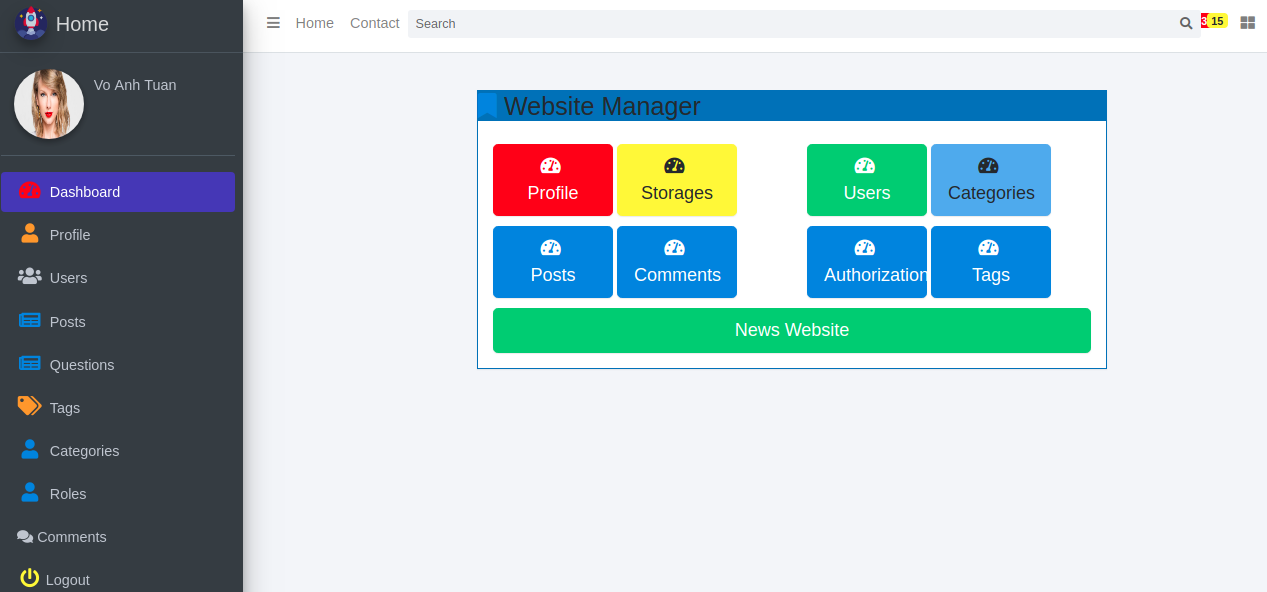
Nếu bạn chưa có tài khoản trong hệ thống, bạn sẽ nhấn vào đăng ký để tạo tài khoản. Giao diện đăng ký tài khoản như sau:



Hình 5.7: Giao diện đăng ký tài khoản của ngừoi dùng

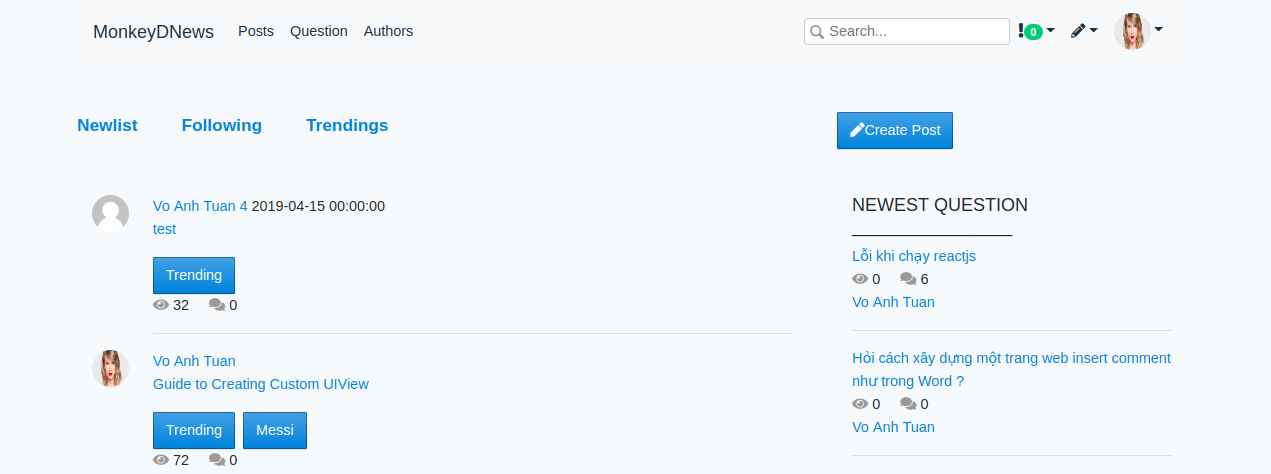
Hệ thống có thể xác mình xem người dùng có những quyền gì và chỉ cho phép họ truy cập vào những thao tác được phép. Nếu không có quyền truy cập vào một hành động nào đó hệ thống sẽ thông báo cho người dùng biết.

Giao diện sau khi đăng nhập dành cho quản trị viên



Hình 5.8: Giao diện trang chủ dành cho quản trị viên

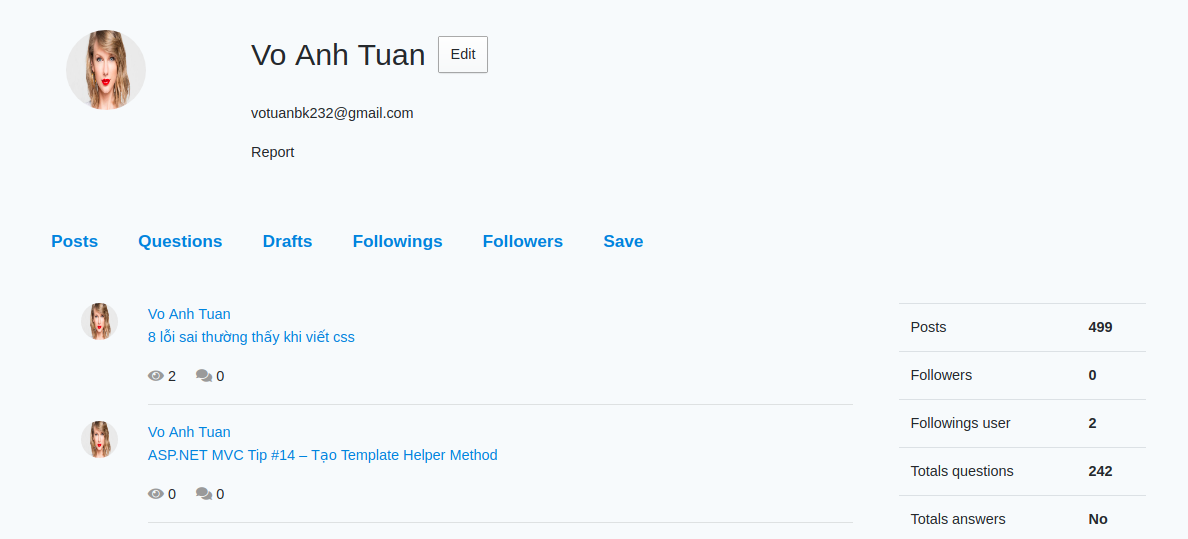
Giao diện sau khi đăng nhập của người dùng:



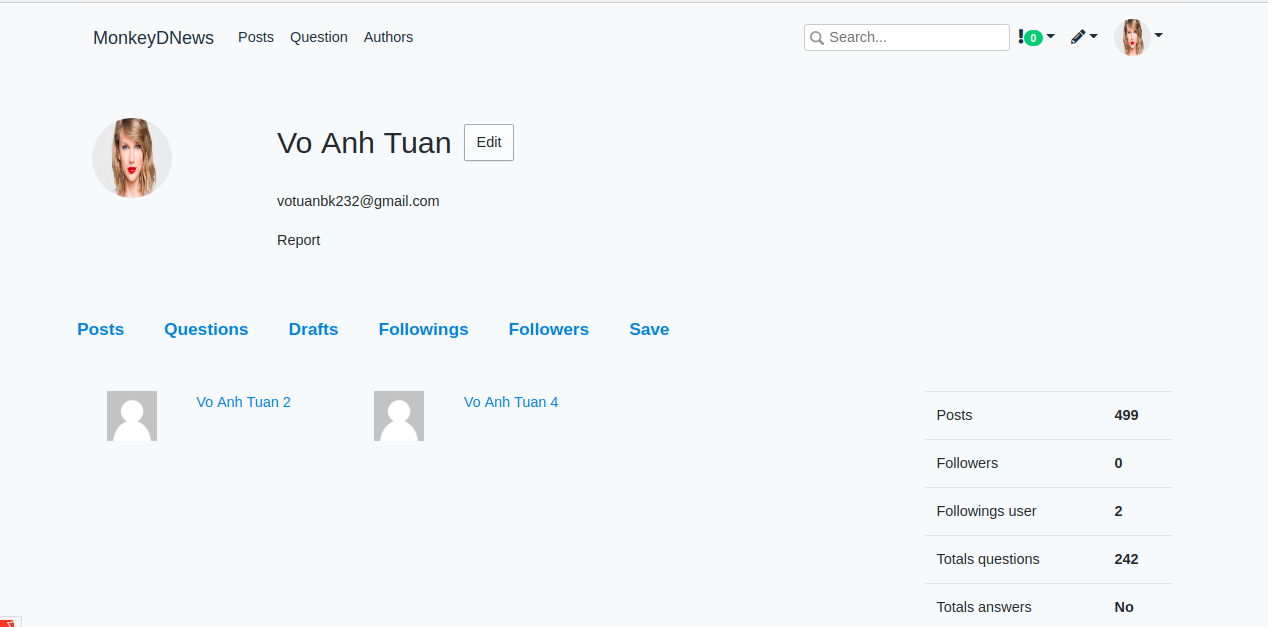
Hình 5.9: Giao diện trang chủ dành cho người dùng

* + 1. Dành cho người dùng

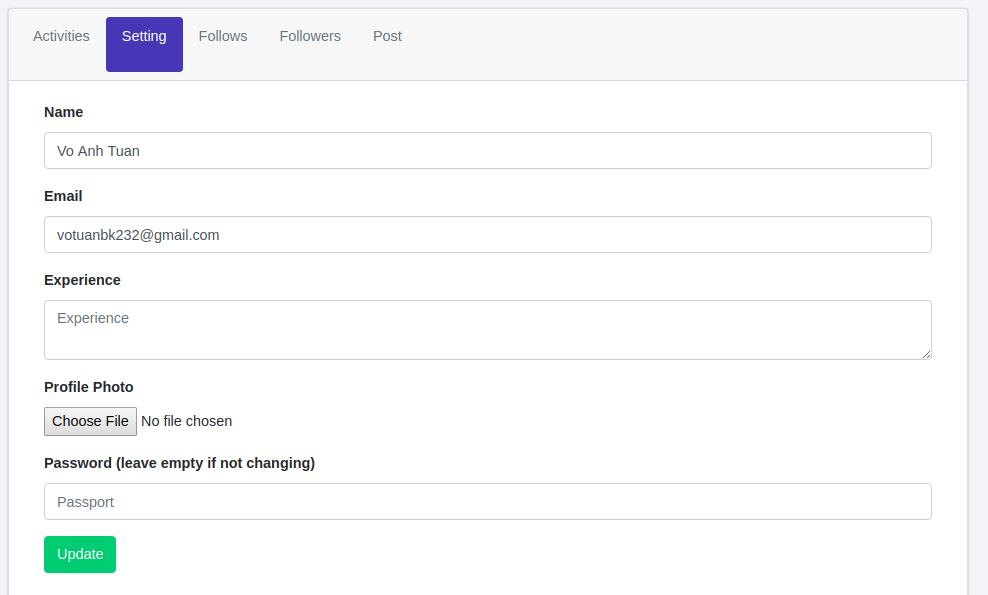
Sau khi đăng nhập, người dùng có thể chọn thông tin cá nhân để xem các thông tin cơ bản của mình trên hệ thống. Các thông tin về những bài viết của người dùng, những người mà mình đang theo dõi và những người đang theo dõi mình. Tại đây với danh sách những người đang theo dõi, người dùng có thể hủy theo dõi bất cứ ai.



Hình 5.10: Giao diện trang cá nhân của người dùng



Hình 5.11: Giao diện trang cá nhân của người dùng



Hình 5.12: Giao diện xem và chỉnh sửa thông tin người dùng

Giao diện trang chủ sau khi đăng nhập gồm một thanh công cụ chứa:

Posts: Danh sách các bài viết

Questions: Danh sách câu hỏi

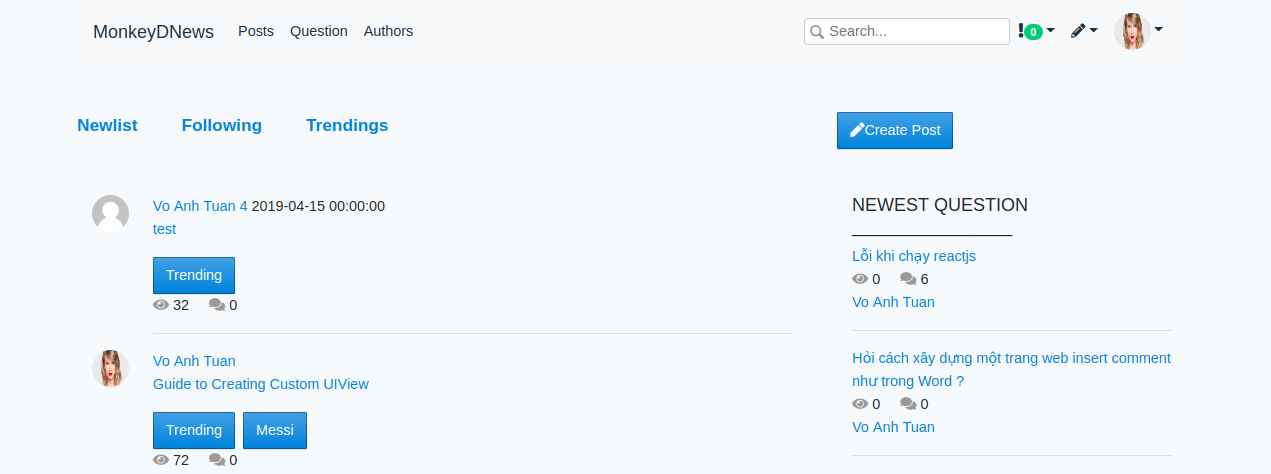
Author: Danh sách tác giả

Với mục bài viết, ta có các mục con:

Danh sách bài viết mới nhất

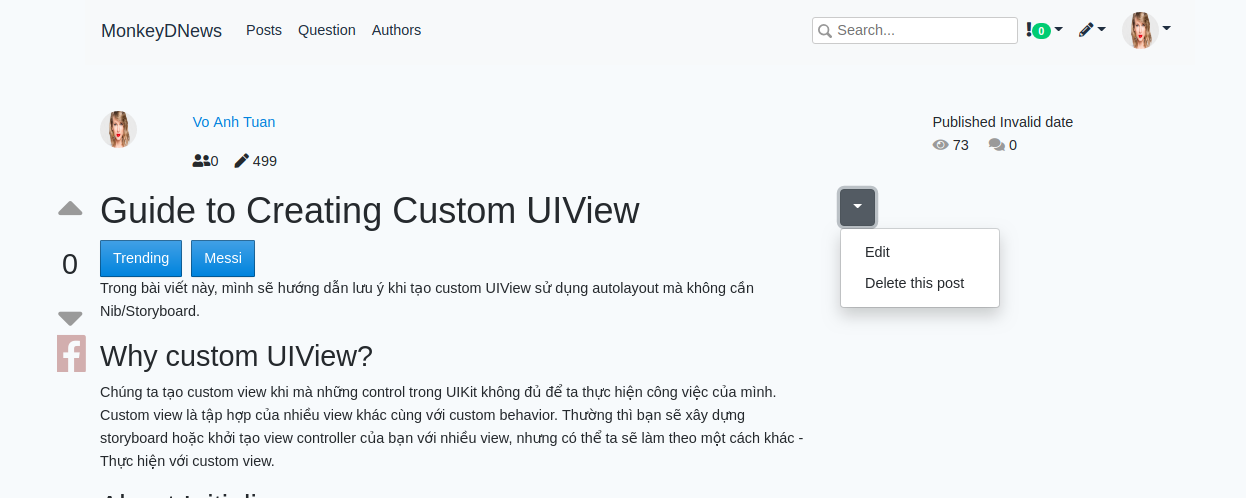
Danh sách bài viết theo dõi: Các bài viết thuộc quyền sở hữu của những tác giả mà người dùng theo dõi.

Danh sách bài viết được gán nhãn “trending”



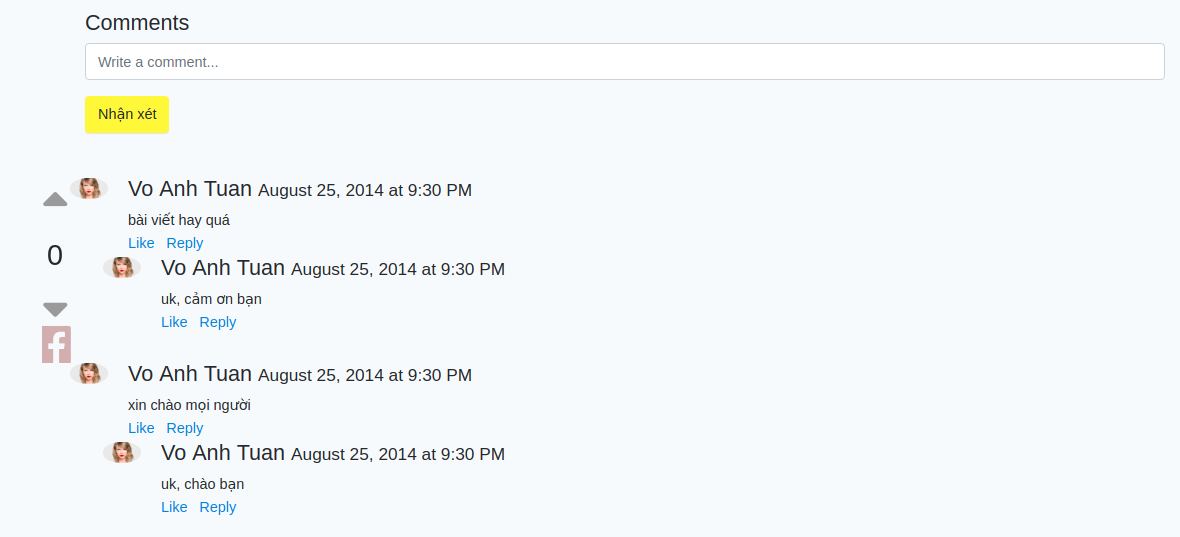
Hình 5.13: Giao diện danh sách bài viết

Khi chọn vào một bài viết, hệ thống sẽ chuyển hướng tới trang nội dung bài viết đó.



Hình 5.14: Giao diện của một bài viết

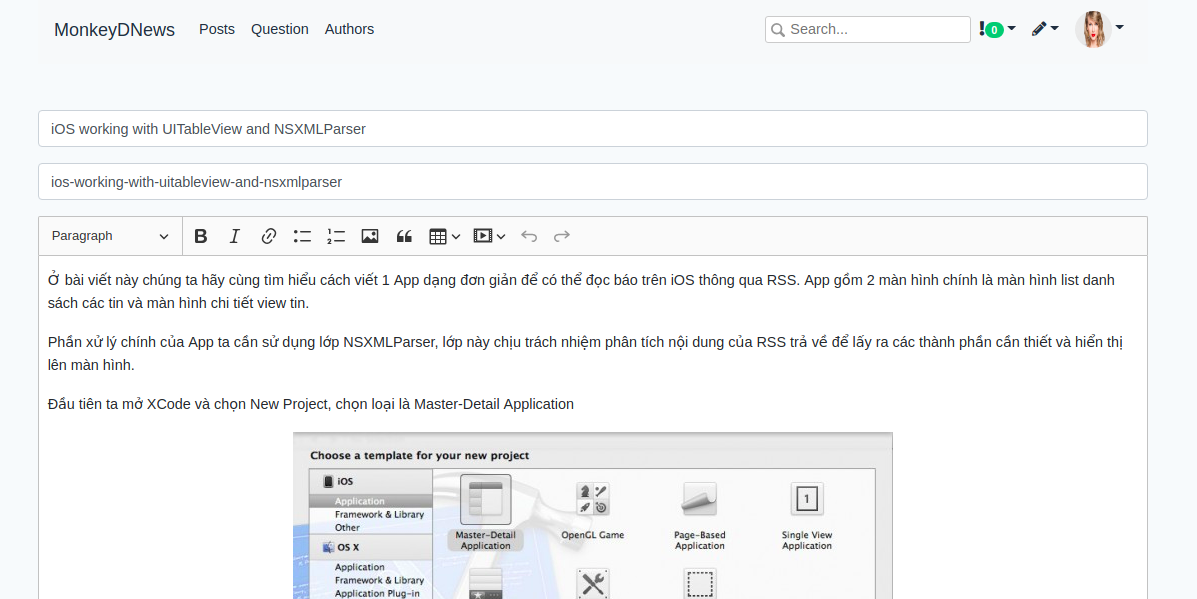
Trong bài viết, người dùng có thể chỉnh sửa hoặc xóa bài viết (nếu có quyền), upvote, downvote, chia sẻ bài viết hoặc bình luận về bài viết ở phía dưới.



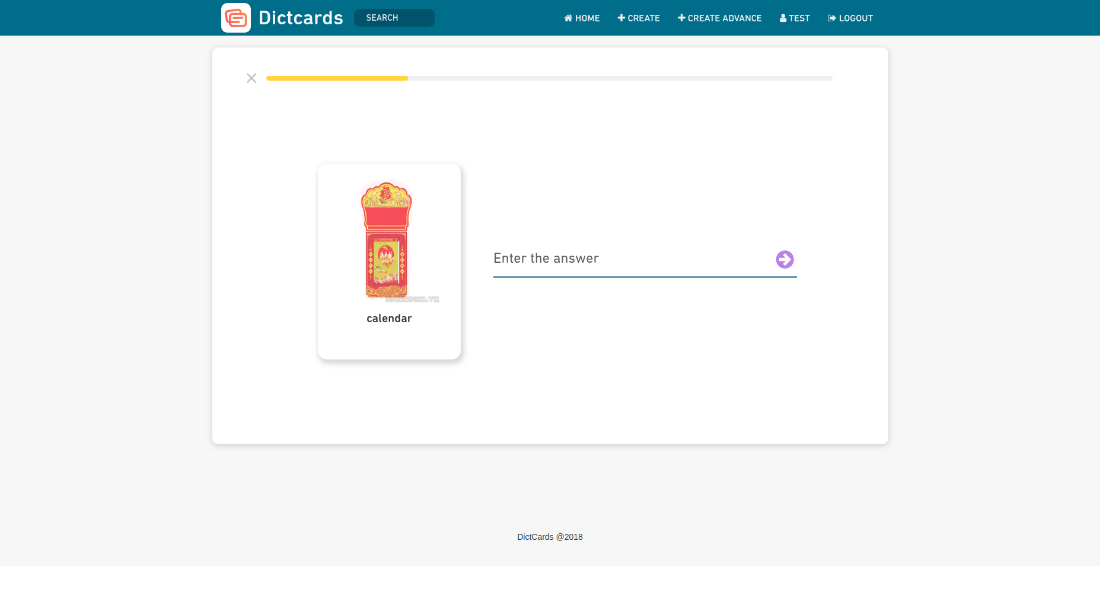
Hình 5.15: Giao diện bình luận bài viết

Trong trang bài viết, nếu người dùng có thể “make trending” một bài viết nếu có quyền.

Trong trang bài viết, người dùng có thể chỉnh sửa bài viết

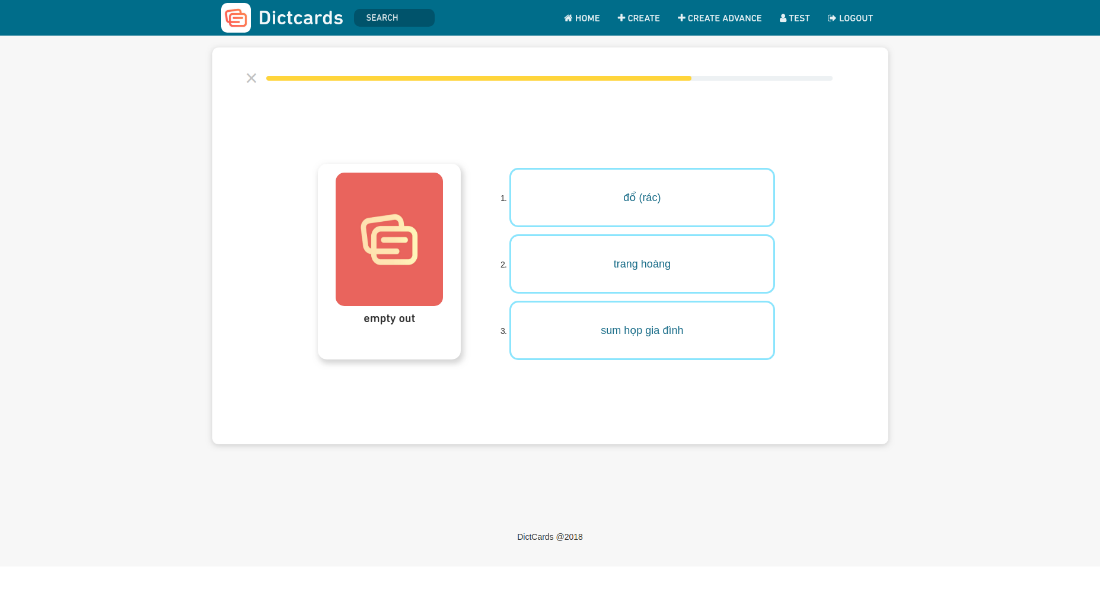


Hình 5.16: Giao diện chỉnh sửa bài viết



Hình 5.17: Giao diện bài học với Dạng câu hỏi viết

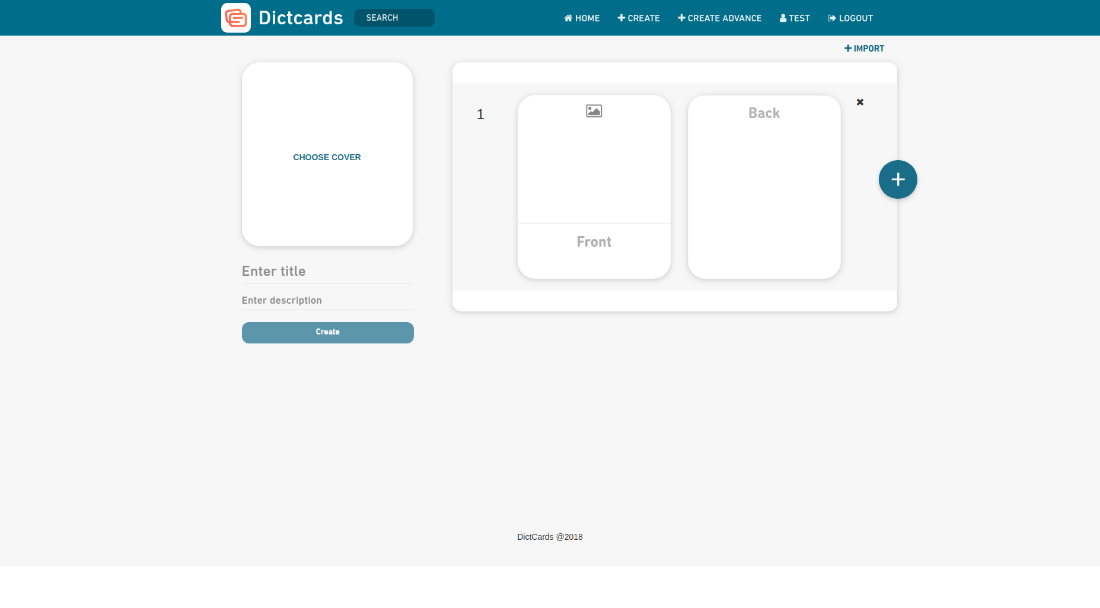
Thứ 3 là dạng câu hỏi chọn thẻ để nhớ các mặt. Hệ thống sẽ hiển thị một mặt của 1 thẻ và có 3 đáp án để người dùng chọn một trong số đó khớp với mặt kia.



Hình 5.18: Giao diện bài học với Dạng câu hỏi nhớ các mặt

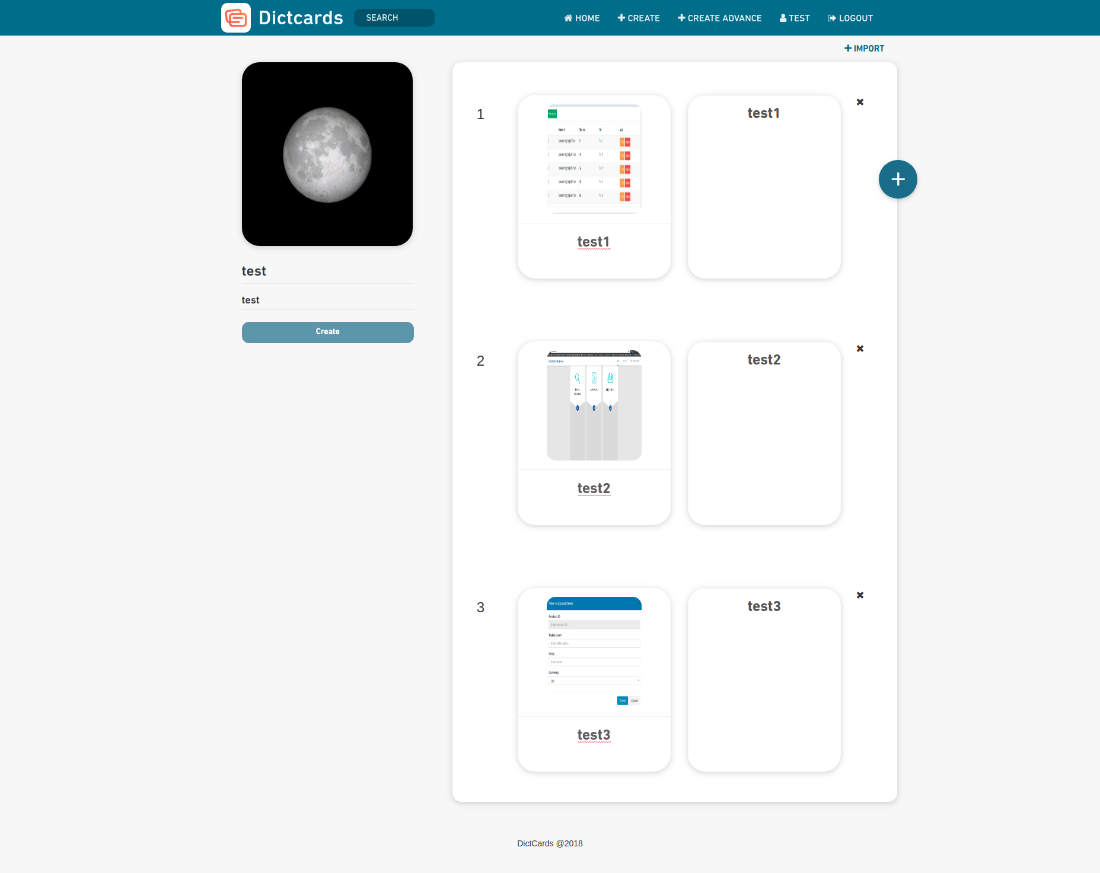
Tạo một bộ thẻ mới

Trên thanh công cụ người dùng bấm vào "Create", hệ thống sẽ chuyển sang giao diện tạo bộ thẻ như hình..



Hình 5.19: Giao diện Tạo bộ thẻ

Người dùng sẽ điền đầy đủ thông tin của bộ thẻ và thêm dữ liệu của các thẻ. Sau đó bấm vào "Create".

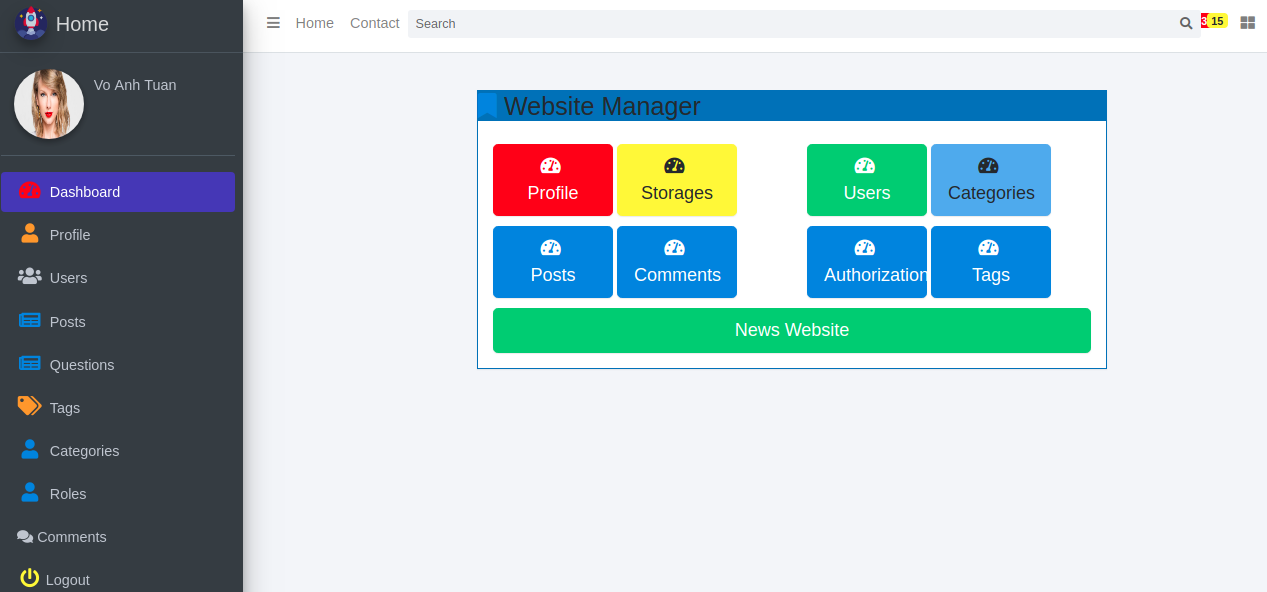


Hình 5.20: Giao diện Nhập dữ liệu màn tạo bộ thẻ

Sau khi tạo hệ thống sẽ chuyển đến trang của bộ thẻ vừa mới tạo và thông báo thành công.

* + 1. Khu vực quản lý của quản trị viên

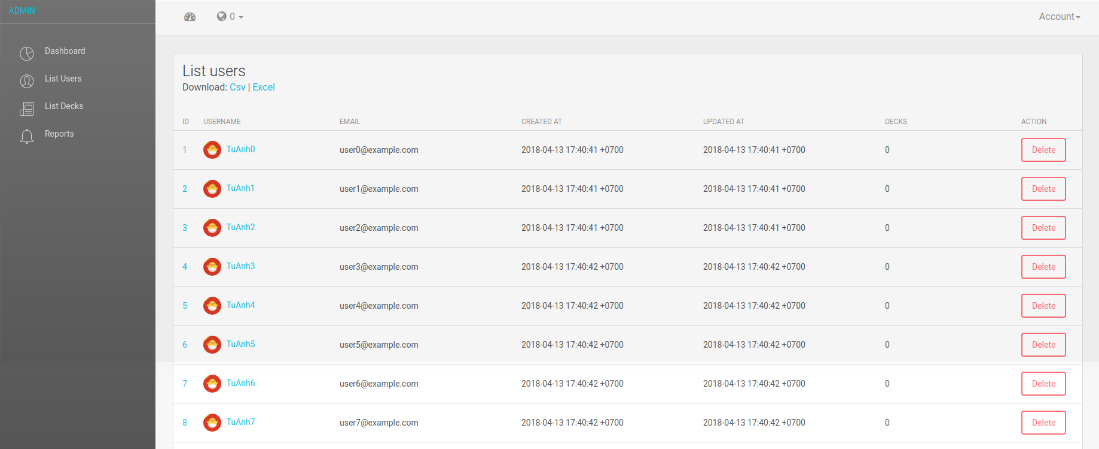
Sau khi người quản trị đăng nhập hệ thống hệ thống sẽ chuyển đến trang admin của hệ thống



Hình 5.21: Giao diện sau khi đăng nhập của quản trị viên

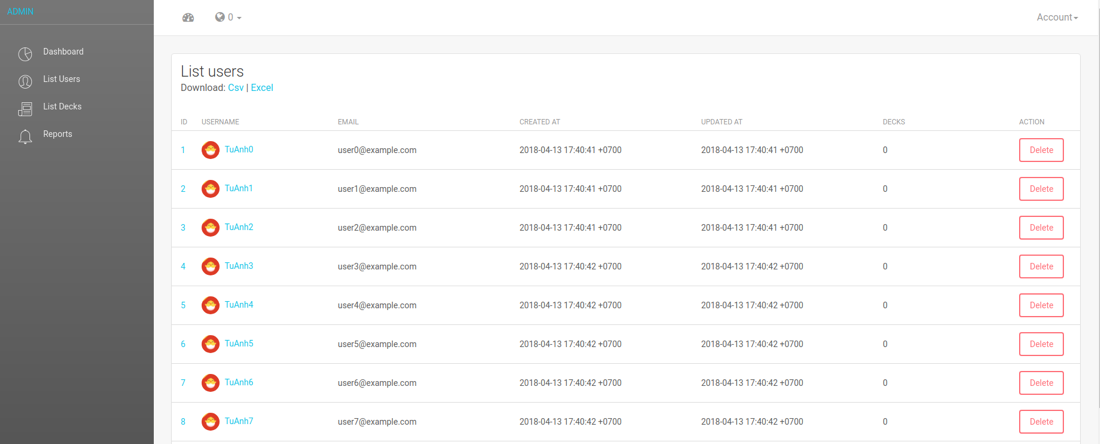
Dữ liệu về người dùng, số bộ thẻ, số thẻ sẽ đựoc thống kê theo ngày và hiện thị trong một tuần hoặc một tháng.

Trên thanh dọc công cụ sẽ có các mục để người quản trị nhấp vào. Về người dùng, sẽ hiển thị tất cả ngừoi dùng với đây đủ các thống tin về tên, email, số lựong bộ thẻ.



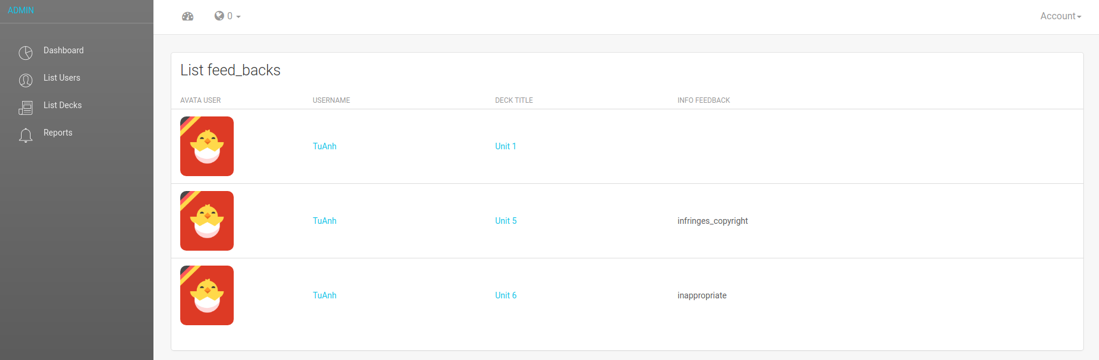
Hình 5.22: Giao diện tất cả người dùng

Về bộ thẻ, sẽ hiện thị toàn bộ thẻ với thông tin về ảnh của bộ thẻ, tiêu đề và số lượng thẻ cũng nhưng người đã tạo ra bộ thẻ đó.



Hình 5.23: Giao diện tất cả bộ thẻ

Về phản hồi bộ thẻ của người dùng, ở thanh công cụ nếu người quản trị bấm vào danh sách phản hồi (Reports) sẽ hiển thị người tạo phản hồi và nội dung phản hồi và bộ thẻ được phản hồi đề người quản lý kiểm tra để xem xét có xóa bộ thẻ hay không.



Hình 5.24: Giao diện tất cả phản hồi

* 1. Nhận xét, đánh giá ứng dụng

Hiện tại, ứng dụng cơ bản đã hoàn thành. Chức năng và giao diện được kiểm thử thông qua một số ngừoi dùng và đã nhận được các nhận xét, phần hồi khách quan. Đa số những người dùng thủ đều hài lòng với chức năng và giao diện của ứng dụng. Kết quả ở bảng sau:

Bảng 5.26: Phản hồi của người dùng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nghề nghiệp** | **Bạn thấy ứng dụng có hữu ích không?** | **Bạn thấy ứng dụng có dễ dàng sử dụng không?** | **Bạn thấy ứng dụng có chạy ổn định không?** | **Giao diện của ứng dụng như thế nào?** | **Ý kiến đóng góp của bạn** |
| 1 | Nguyễn Anh Tú | Lập trình viên | Có | Có | Ứng dụng chạy mượt mà, Tốc độ load trang nhanh | Đẹp, hiệu ứng tốt | Chưa có nhiều người dùng nên số lượng thẻ ít, chưa phong phú nội dung |
| 2 | Võ Mỹ Phương | Học sinh | Có | Có | Có | Mình thích những ứng dụng có giao diện đơn giản như thế này | Bổ sung tính năng thiết lập mục tiêu hàng ngày, nhắc nhở người dùng khi chưa đạt mục tiêu, hiển thị tiến độ so với mục tiêu đã đề ra. Ngoài ra nếu có thêm chức năng điểm số, xem được điểm số của bạn bè sẽ tạo ra tính ganh đua hiệu quả hơn. |
| 3 | Nguyễn Thị Hồng Lam | Sinh viên | Có | Có | Có | Giao diện đẹp, dễ dùng |  |
| 4 | Vũ Thanh Tùng | Lập trình viên | Có | Có | Có, chạy mượt, không bị giật | Giao diện đơn giản, đẹp | Ứng dụng tốt, dễ sử dụng, thích hợp với mọi lứa tuổi |
| 5 | Nguyễn Thị Hải | Giáo viên | Có | Có | Có | Khá đẹp | Ứng dụng còn ít dữ liệu |
| 6 | Trần Đức Anh | Sinh viên | Có | Có | Có | Đơn giản, dễ sử dụng |  |

# Kết luận

* 1. Kết quả đạt được

Xây dựng ứng dụng công đồng học nhanh qua thẻ là một ứng dụng áp dụng ý tưởng từ một ứng dụng có sẵn và có phần sáng tạo bổ sung thêm các chức năng mới giúp người dùng dễ sử dụng và thuận tiện hơn trong việc học và ghi nhớ các khái niệm. Việc sử dụng Angular giúp cho hệ thống có tốc độ tải nhanh và thân thiện với ngừoi dùng. Đa số ngừoi dùng thử nghiệm đều hài lòng với giao diện và chức năng của ứng dụng. Vì ứng dụng mở cho tất cả mọi người dùng có thể tạo bộ thẻ nên mọi người có thể cùng học và chia sẻ những kiến thức bổ ích với cộng đồng. Cách tạo ra các bài học để luyện kiến thức có khoa học giúp người dùng dễ nhớ và cảm thấy hấp dẫn thú vị hơn khi học. Bên cạnh các ưu điểm thì ứng dụng cũng có một số nhược điểm như các dạng câu hỏi chưa đa dạng để ôn luyện, số lượng người dùng chưa nhiều nên số lượng bộ thẻ còn ít.

Trong quá trình thực hiện khóa luận này, tôi đã tích lũy, học hỏi được những kiến thức về các công nghệ Ruby on Rails, Angular 2... Bên cạnh đó việc làm khóa luận cũng giúp tôi hiểu được quy trình phát triển phần mềm, các cách thức làm việc nhóm để giúp tôi tích lũy thêm nhiều kinh nghiệp phục vụ có công việc sau này.

* 1. Định hướng phát triển

Trong thời gian tới ứng dụng sẽ được phát triển theo các hướng sau:

* Người dùng có thể tự tạo ra các chủ đề của riêng mình.
* Thiết kế editor cho việc tạo bài viết một cách chuyên nghiệp và đẹp mắt hơn.
* Bảo mật website an toàn hơn.

TÀI LIỆU THAM KHẢO

**Tiếng Anh**

[1] "Angular 2," Google, [Online]. Available: https://angular.io/. [Accessed 12 February 2018].

[2] "Ruby on Rails," Google, [Online]. Available: rubyonrails.org. [Accessed 12 February 2018].

[3] “Single page application,” Google, [Online]. Available: https://en.wikipedia.org/wiki/Single-page\_application. [Accessed 12 February 2018].

[4] "Tinycards," [Online], Available: https://tinycards.duolingo.com/. [Accessed 12 February 2018]

[5] "Angular 2 - Service" Google, [Online]. Available: https://www.tutorialspoint.com/angular2/angular2\_services.html. [Accessed 12 February 2018]

**Tiếng Việt**

[6] "Ứng dụng web," Google, [Online]. Available: <https://vi.wikipedia.org/wiki/Ứng_dụng_web>. [Truy cập lần cuối 12 tháng 2 năm 2018]

[7] "Học lập trình web bằng ruby on rails," Google, [Online]. Available: <https://techmaster.vn/posts/7951/hoc-lap-trinh-web-bang-ruby-on-rails-2>. [Truy cập lần cuối 12 tháng 2 năm 2018]

[8] "[A2] Giới thiệu về Angular 2," Google, [Online]. Available: https://github.com/truonganhhoang/int3507-2016/wiki/%5BA2%5D-Giới-thiệu-về-Angular-2. [Truy cập lần cuối 12 tháng 2 năm 2018]

[9] "Hệ quản trị cơ sở dữ liệu," Google, [Online]. Available: https://vi.wikipedia.org/wiki/Hệ\_quản\_trị\_cơ\_sở\_dữ\_liệu. [Truy cập lần cuối 12 tháng 2 năm 2018]

[10] "ActionCable trong Rails 5," Google, [Online]. Available:

https://viblo.asia/p/actioncable-trong-rails-5-nwmkyZbbvoW. [Truy cập lần cuối 12 tháng 2 năm 2018]

[11] "Tìm hiểu về kiểm thử tự động," Google, [Online]. Available: https://viblo.asia/p/tim-hieu-ve-kiem-thu-tu-dong-dich-1l0rvmxDGyqA. [Truy cập lần cuối 12 tháng 2 năm 2018]