**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



**Liêu Hà Phương Huy  
Lê Trọng Nhân**

**XÂY DỰNG WEBSITE BÁO ONLINE  
BẰNG REACTJS VÀ LARAVEL FRAMEWORK**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2023BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**



**Liêu Hà Phương Huy**

**Lê Trọng Nhân**

**XÂY DỰNG WEBSITE BÁO ONLINE  
BẰNG REACTJS VÀ LARAVEL FRAMEWORK**

**Mã số sinh viên:   
Liêu Hà Phương Huy - 2051012034**

**Lê Trọng Nhân - 2051010215**

**ĐỒ ÁN NGÀNH**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Giảng viên hướng dẫn: VÕ THỊ KIM ANH**

**TP. HỒ CHÍ MINH, 2023**

**LỜI CẢM ƠN**

Sự thành công của một sinh viên dù ít hay nhiều luôn đi cùng với một người giảng viên. Nhóm em xin được phép gửi lời cảm ơn chân thành này đến cô Võ Thị Kim Anh, người đã hỗ trợ nhóm em trong suốt quá trình chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp những lời khuyên giúp chúng em thực hiện đồ án tốt nhất. Nhờ có những chỉ dẫn của cô đã giúp nhóm em hiểu rõ về kiến thức để thực hiện, các cách trình bày cũng như thực hiện đồ án nên chúng em đã hoàn thành đúng tiến độ cùng với đó là rất nhiều kinh nghiệm được chúng em học hỏi và tiếp thu. Một lần nữa, nhóm em xin cảm ơn cô.

Đồ án được thực hiện trong vòng một tháng, khoảng thời gian cũng vừa đủ để nhóm em hoàn tất được đồ án. Tuy nhiên do nhiều kiến thức mới cũng như thời gian chúng em thực hiện qua tuần không phải là tối ưu nhất nên đồ án thực hiện sẽ có nhiều sai sót, đó là điều không thể tránh khỏi. Chúng em rất mong nhận được mọi ý kiến đóng góp quý báu của các thầy cô để giúp cho kiến thức của chúng em ngày càng vững chắc hơn. Chúng em xin chân thành cảm ơn.

Thành phố Hồ Chí Minh, ngày 12 tháng 10 năm 2023

Nhóm thực hiện

(ký và ghi họ tên)

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**TÓM TẮT ĐỒ ÁN NGÀNH**

Trong ứng dụng đọc báo online này với mục tiêu mang lại những thông tin có giá trị nhất, mới nhất và nhanh nhất cho đọc giả.

Vì vậy hệ thống sẽ liên tục cập nhật những thông tin mới nhất thông qua một trang web cực kì uy tín tại Việt Nam là VNExpress bằng công nghê RSS. Không những vậy để cùng đóng góp những thông tin tích cực nhanh chóng cho cộng đồng. Các người dùng có thể gửi yêu cầu đến hệ thống để trở thành tác giả. Khi trở thành tác giả, người dùng có thể tạo và tùy chỉnh các bài viết của chính mình. Và để có thể biết chắc rằng thông tin có hữu ích không, các bài báo sẽ được kiểm duyệt trước khi được đăng trên website.

Nếu không muốn trở thành tác giả, các người dùng cũng có thể hỗ trợ hệ thống tìm ra những bài báo độc hại bằng cách tố cáo bài báo đó về cho hệ thống để có thể ngăn chặn bài báo đó đến với nhiều người hơn.

Website sẽ có thể mất nhiều thời gian hơn trong lần đầu tải trang do được áp dụng công nghệ ReactJS với CSR (Client Side Rendering) cùng với Laravel Framework. Nhưng trải nghiệm tổng thể vẫn sẽ đủ để phục vụ tốt mục tiêu ban đầu mà ứng dụng này mang lại

**ABSTRACT**

**MỤC LỤC**

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 12](#_Toc17840)

[I. Giới thiệu về đề tài: 12](#_Toc7749)

[II. Đối tượng nghiên cứu: 12](#_Toc15210)

[III. Ý nghĩa thực tiễn: 12](#_Toc1208)

[Chương 2. CƠ SỞ LÝ THUYẾT 14](#_Toc17994)

[I. PHP Laravel Framework 14](#_Toc27154)

[II. MySQL 14](#_Toc17011)

[A. Ưu điểm: 14](#_Toc13977)

[B. Nhược điểm: 15](#_Toc18679)

[III. RESTful API 15](#_Toc20822)

[A. Ưu điểm: 15](#_Toc5629)

[B. Nhược điểm: 15](#_Toc6700)

[IV. REACT.JS 16](#_Toc5349)

[A. Ưu điểm: 16](#_Toc21979)

[B. Nhược điểm: 16](#_Toc27333)

[V. TAILWIND 17](#_Toc8978)

[VI. RSS FEED 17](#_Toc22975)

[Chương 3. CƠ SỞ DỮ LIỆU 18](#_Toc2285)

[I. Xác định các đối tượng của hệ thống: 18](#_Toc27636)

[Chức năng của User: 18](#_Toc2851)

[Chức năng của Author: 18](#_Toc18689)

[Chức năng của Admin: 18](#_Toc4664)

[II. Mô Hình ERD: 19](#_Toc8844)

[Chương 4. CÁC CHỨC NĂNG CHÍNH 20](#_Toc4181)

[I. Lấy dữ liệu thông qua RSS: 20](#_Toc667)

[II. Đăng ký, đăng nhập tài khoản: 21](#_Toc28953)

[III. Hiển thị trang chủ (Home): 24](#_Toc11552)

[B. Hiển thị chi tiết bài báo: 30](#_Toc7221)

[Chương 5. CÁC CHỨC NĂNG PHÂN QUYỀN 33](#_Toc30830)

[I. Cách phân quyền 33](#_Toc28299)

[II. Chức năng của User 33](#_Toc5708)

[A. Bình luận (Comment) 33](#_Toc16750)

[B. Đánh giá (Rating) 34](#_Toc22212)

[C. Báo cáo (Report) 36](#_Toc6335)

[III. Chức năng của Author: 37](#_Toc14919)

[A. Tạo, cập nhật và xóa bài báo: 37](#_Toc5873)

[IV. Chức năng của Admin: 41](#_Toc5664)

[A. Kiểm duyệt yêu cầu: 41](#_Toc3647)

[B. Quản lý user: 43](#_Toc25930)

[C. Quản lý bài báo: 46](#_Toc162)

[Chương 6. TÀI LIỆU THAM KHẢO 50](#_Toc28962)

[Chương 7. PHỤ LỤC 51](#_Toc11322)

[I. Cài đặt Composer 51](#_Toc6293)

[II. Cài đặt Xampp 51](#_Toc28752)

[III. Cài đặt NodeJs 52](#_Toc29100)

[IV. Hướng dẫn sử dụng App 52](#_Toc30492)

[V. Hướng dẫn cách khắc phục lỗi 53](#_Toc8370)

**DANH MỤC HÌNH VẼ**

[*Hình 1.* Ảnh mô hình ERD 19](#_Toc5244)

[*Hình 2.* Ảnh code minh họa lấy dữ liệu. 20](#_Toc12636)

[*Hình 3.* Ảnh code minh họa lưu trữ dữ liệu. 21](#_Toc5554)

[*Hình 4.* Ảnh giao diện đăng nhập. 21](#_Toc1346)

[*Hình 5.* Ảnh giao diện đăng ký. 22](#_Toc14473)

[*Hình 6.* Ảnh code phần login phía client 23](#_Toc20920)

[*Hình 7.* Ảnh code phần register phía client 23](#_Toc593)

[*Hình 8.* Ảnh code phương thức login phía server. 23](#_Toc31878)

[*Hình 9.* Ảnh code phương thức register phía server. 24](#_Toc10393)

[*Hình 10.* Ảnh route api của login, register. 24](#_Toc22589)

[*Hình 11.* Ảnh trang chủ. 25](#_Toc25061)

[*Hình 12.* Ảnh kết quả trả về của chức năng search. 26](#_Toc7847)

[*Hình 13.* Ảnh giao diện phân trang. 26](#_Toc12516)

[*Hình 14.* Ảnh code lấy dữ liệu báo từ phía client. 27](#_Toc24467)

[*Hình 15.* Ảnh code xử lý lấy dữa liệu các bài báo phía server 27](#_Toc9856)

[*Hình 16.* Ảnh code lấy chủ đề danh mục từ phía client. 27](#_Toc3897)

[*Hình 17.* Ảnh code xử lý chủ đề danh mục phía server 27](#_Toc14655)

[*Hình 18.* Ảnh code thực hiện tìm kiếm từ phía client. 28](#_Toc30858)

[*Hình 19.* Ảnh code xử lý tìm kiếm theo kw phía server 28](#_Toc1223)

[*Hình 20.* Ảnh code thực hiện lọc theo chủ đề từ phía client. 29](#_Toc15137)

[*Hình 21.* Ảnh code xử lý phân trang phía client. 30](#_Toc6802)

[*Hình 22.* Ảnh chi tiết bài báo 30](#_Toc31161)

[*Hình 23.* Ảnh phần bình luận trước khi được xác thực đăng nhập. 31](#_Toc5380)

[*Hình 24.* Ảnh phần bình luận sau khi đã xác thực đăng nhập. 31](#_Toc17746)

[*Hình 25.* Ảnh code phần lấy xử lý dữ liệu thông qua API. 32](#_Toc26763)

[*Hình 26.* Ảnh xử lý phân quyền theo role của CheckRole Middleware 33](#_Toc29285)

[*Hình 27.* Ảnh sử dụng middleware cho role 2 (Author) và role 3 (Admin) 33](#_Toc2678)

[*Hình 28.* Ảnh giao diện để xử lý bình luận. 33](#_Toc12506)

[*Hình 29.* Ảnh code thực hiện chức năng comment phía client. 34](#_Toc916)

[*Hình 30.* Ảnh code thực hiện chức năng comment phía server. 34](#_Toc17572)

[*Hình 31.* Ảnh route liên quan đến comment. 34](#_Toc29932)

[*Hình 32.* Ảnh giao diện tương tác đánh giá. 35](#_Toc27682)

[*Hình 33.* Ảnh code xử lý đánh giá từ phía client. 35](#_Toc1795)

[*Hình 34.* Ảnh code xử lý phần đánh giá phía server. 36](#_Toc10945)

[*Hình 35.* Ảnh code xử lý lấy dữ liệu rating để gửi lên frontend. 36](#_Toc16530)

[*Hình 36.* Ảnh code API phần rating ở phía server. 36](#_Toc8561)

[*Hình 37.* Ảnh giao diện thực hiện việc tố cáo. 36](#_Toc26513)

[*Hình 38.* Ảnh code xử lý tố cáo bài báo từ phía client. 37](#_Toc11235)

[*Hình 39.* Ảnh code xử lý thêm báo cáo bài báo vào database phía server. 37](#_Toc26634)

[*Hình 40.* Ảnh code API lấy thông tin report từ phía client về. 37](#_Toc28851)

[*Hình 41.* Ảnh giao diện tạo bài viết mới. 38](#_Toc22030)

[*Hình 42.* Ảnh giao diện hiển thị các bài viết của author. 39](#_Toc17386)

[*Hình 43.* Ảnh giao diện chỉnh sửa bài báo. 39](#_Toc3262)

[*Hình 44.* Ảnh code xử lý thêm bài báo mới phía client. 40](#_Toc30329)

[*Hình 45.* Ảnh code xử lý chỉnh sửa, cập nhật bài báo. 40](#_Toc14931)

[*Hình 46.* Ảnh code xử lý xóa bài báo. 40](#_Toc17050)

[*Hình 47.* Ảnh route api liên quan đến việc tạo, cập nhật, xóa bài báo. 41](#_Toc26997)

[*Hình 48.* Ảnh giao diện kiểm duyệt bài báo. 41](#_Toc17472)

[*Hình 49.* Ảnh giao diện kiểm duyệt yêu cầu nâng cấp role của user. 42](#_Toc23159)

[*Hình 50.* Ảnh code xử lý kiểm duyệt bài báo và user phía client. 42](#_Toc9723)

[*Hình 51.* Ảnh route dành cho việc kiểm duyệt bài báo 43](#_Toc21285)

[*Hình 52.* Ảnh route dành cho việc nâng cấp role (role-up, cancel). 43](#_Toc11766)

[*Hình 53.* Ảnh giao diện quản lý user. 43](#_Toc582)

[*Hình 54.* Ảnh code xử lý nâng cấp role phía client thông qua API. 44](#_Toc29695)

[*Hình 55.* Ảnh code xử lý đặt lại role phía client thông qua API. 44](#_Toc15368)

[*Hình 56.* Ảnh code xử lý xóa tài khoản phía client thông qua API. 44](#_Toc4282)

[*Hình 57.* Ảnh code hàm lưu thông tin và hàm lấy thông tin ra. 45](#_Toc16110)

[*Hình 58.* Ảnh code hàm xử lý nâng cấp và giảm role phía server. 45](#_Toc22374)

[*Hình 59.* Ảnh code hàm từ chối yêu cầu của user (hủy). 45](#_Toc27890)

[*Hình 60.* Ảnh code hàm xóa tài khoản user. 46](#_Toc30436)

[*Hình 61.* Ảnh route liên quan đến việc quản lý user. 46](#_Toc764)

[*Hình 62.* Ảnh giao diện của chức năng quản lý bài báo. 47](#_Toc991)

[*Hình 63.* Ảnh code xử lý lọc báo theo comment. 48](#_Toc26850)

[*Hình 64.* Ảnh code xử lý lọc theo tố cáo. 48](#_Toc15685)

[*Hình 65.* Ảnh code xử lý lọc theo lượt xem. 48](#_Toc18055)

[*Hình 66.* Ảnh code xử lý chấp nhận cho phép đăng bài báo. 49](#_Toc9584)

[*Hình 67.* Ảnh route liên quan đến chức năng quản lý bài báo. 49](#_Toc11991)

[*Hình 68.* Ảnh hướng dẫn 1 51](#_Toc7600)

[*Hình 69.* Ảnh hướng dẫn 2 51](#_Toc30635)

[*Hình 70.* Ảnh hướng dẫn 3 52](#_Toc24343)

[*Hình 71.* Ảnh hướng dẫn 4 53](#_Toc19941)

[*Hình 72.* Ảnh hướng dẫn 5 53](#_Toc17080)

[*Hình 73.* Ảnh hướng dẫn 6 53](#_Toc20282)

**MỞ ĐẦU**

Trong thời đại số hóa ngày nay, các trang web tin tức đóng vai trò vô cùng quan trọng trong cung cấp thông tin và kết nối mọi người với thế giới. Đây là nơi mà mọi người tìm kiếm thông tin mới nhất về các sự kiện, xu hướng và diễn biến xã hội. Khả năng tiếp cận thông tin 24/7 thông qua Internet đã biến những trang web tin tức thành nguồn thông tin hàng đầu cho hàng triệu người trên khắp thế giới.

Nhưng thay đổi đáng kể trong ngành truyền thông công nghệ thông tin đã biến cách chúng ta tiếp cận và tiêu thụ thông tin trở nên hoàn toàn mới mẻ. Trong bối cảnh thời đại công nghệ 4.0, việc xây dựng một website tin tức không chỉ là sự nhu cầu, mà còn là một nhiệm vụ hết sức quan trọng, đòi hỏi sự kết hợp hài hòa giữa kiến thức về công nghệ, sáng tạo trong thiết kế và trải nghiệm người dùng, cùng với khả năng hiểu biết về lĩnh vực cụ thể. Sự tạo dựng nên một nền tảng đáng tin cậy, mạnh mẽ và linh hoạt không chỉ góp phần vào việc lan truyền thông tin đúng đắn mà còn thúc đẩy sự nhận thức xã hội. Bên cạnh đó việc tạo dựng một website tin tức thú vị và dễ sử dụng có thể thu hút số lượng lớn đọc giả, giúp nguồn thu nhập và tiếp thị hiệu quả hơn. Website tin tức không chỉ đóng vai trò là nguồn thông tin đáng tinh cậy, mà còn là một nền tảng để tạo cầu nối giữa thông tin và xã hội, giữa người viết và người đọc.

Và để tiếp cận cũng như nâng cao sự phổ biến của việc cập nhật thông tin - tin tức, cũng như góp phần tạo ra thêm nhiều nguồn tiếp cận các thông tin - tin tức trên nhiều lĩnh vực khác nhau dành cho thị trường kiến thức ở Việt Nam, chúng em chọn đề tài:“Xây dựng website tin tức sử dụng PHP Laravel Framework, MySQL và ReactJS” để làm đề tài cho đồ án ngành của mình.

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu về đề tài:

Hiện nay, nhu cầu học hỏi cập nhật thông tin về xã hội, con người và thế giới xung quanh đối với mỗi người là vô cùng lớn. Nhất là trong xã hội điện tử thông minh, tự động hóa đang phát triển vô cùng mạnh mẽ.

Website đọc báo và tin tức nhanh là một nền tảng trực tuyến mang đến nhiều sự tiện lợi và linh hoạt cho người dùng trong việc tiếp cận thông tin và tin tức từ nhiều nguồn khác nhau. Với một giao diện thân thiện và linh hoạt, trang web này cho phép người dùng dễ dàng theo dõi, đọc các bài viết và cho phép đăng các bài đọc, tin tức và cập nhật mới nhất về các lĩnh vực như chính trị, kinh tế, khoa học, giải trí và nhiều lĩnh vực khác. Tận dụng công nghệ như: “Laravel framework và ReactJS” để cập nhật thông tin thời gian thực và tương tác xã hội, trang web đọc báo và tin tức nhanh mang đến một trải nghiệm đa chiều và đáp ứng nhu cầu ngày càng tăng về thông tin và kiến thức một cách linh hoạt của người dùng trên khắp mọi lĩnh vực.

## Đối tượng nghiên cứu:

Tìm hiểu cách vận hành và xây dựng mô hình web Client-Server, sử dụng ngôn ngữ PHP, PHP Laravel Framework và hệ quản trị cơ sở dự liệu MySQL ở phía Server, ReactJS ở phía Client đồng thời tìm hiểu cách xây dựng cũng như vận hành và quản lý một website cung cấp thông tin, tin tức nhanh cho tất cả mọi người.

## Ý nghĩa thực tiễn:

Trong thời đại công nghệ 4.0 ngày nay, các website tin tức trở nên ngày càng quan trọng và đa dạng hơn. Không chỉ đóng vai trò trong việc cung cấp thông tin mà còn tạo điều kiện cho sự kết nối xã hội, hỗ trợ quyết định thông mình và là nguồn tài liệu phong phú cho người dùng, với nhiều ý nghĩa thực tiễn chủ chốt:

* Cập nhật thông tin nhanh chóng: Website tin tức giúp cung cấp thông tin mới nhất và nhanh nhất về các sự kiện quan trọng trên toàn thế giới. Điều này quan trọng để người dùng có thể theo kịp nhưng thay đổi trong thời gian thực.
* Phân loại thông tin: Với lượng thông tin vô cùng lớn trên Internet, website tin tức giúp người đọc lọc và tìm kiếm thông tin cần thiết nhanh chóng, giúp họ tìm hiểu về các chủ đề mà họ quan tâm một cách linh hoạt.
* Kết nối xã hội: Website tin tức giúp tạo cầu nối giữa người dùng và những nguồn thông tin đáng tin cậy. Người dùng có thể tham gia vào cuộc thảo luận, đọc ý kiến của người khác và tương tác xã hội thông qua các bài viết và bình luận.

# CƠ SỞ LÝ THUYẾT

## PHP Laravel Framework

- PHP Laravel là một framework dùng để phát triển ứng dụng web được xây dựng bằng ngôn ngữ lập trình PHP. Laravel được tạo ra để giúp các nhà phát triển xây dựng ứng dụng web một cách nhanh chóng và dễ dàng bằng cách cung cấp các công cụ và thư viện phổ biến để xử lý các nhiệm vụ phức tạp như xử lý định tuyến (routing), xử lý HTTP request và responses, truy cập cơ sở dữ liệu, quản lý các phiên (session), xử lý người dùng và quyền truy cập và nhiều tính năng khác.

- Laravel được phát triển bới Taylor Otwell và được phát hành lần đầu vào năm 2011. Nó đã nhánh chóng trở thành một trong những framework PHP phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển web. Laravel cung cấp cú pháp dễ đọc và hiện đại, hỗ trợ tích hợp với các công nghệ như Composer (quản lý gói PHP), Blade (hệ thống giao diện mẫu) và Eloquent ORM (Object-Relational Mapping) cho việc làm việc với cơ sở dữ liệu.

- Laravel cũng đi kèm với nhiều tính năng tiện ích như quản lý tài nguyên (resource management), bảo mật, xác thực và nhiều gói mở rộng có sẵn để mở rộng khả năng của framework. Laravel giúp tăng hiệu suất phát triển ứng dụng web và giảm thời gian và công sức cần thiết để xây dựng các tính năng phức tạp.

## MySQL

- MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở phổ biến. Nó được lưu trữ và quản lý dữ liệu trong các ứng dụng web, ứng dụng di động và nhiều loại ứng dụng khác.

### Ưu điểm:

* Miễn phí và mã nguồn mở, được tối ưu hóa để xử lý truy vấn nhanh chóng và có khả năng làm việc với các tải công việc lớn.
* Có thể chạy trên nhiều hệ điều hành khác nhau, hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các API và thư viện cho nhiều ngôn ngữ khác nhau.
* Cung cấp các tính năng bảo mật và mã hóa dữ liệu. Có một cộng đồng lớn nhiều người dùng và nhiều tài liệu phong phú.

### Nhược điểm:

* So với các một số hệ quản trị cơ sở dữ liệu khác, MySQL có một só hạn chế về tính năng và không hỗ trợ một số tính tính năng phức tạp.
* Không thể quản lý tìa nguyên đồng thời với số lượng lớn người dùng hoặc tải công việc cao mà không có sự tối ưu hóa phù hợp.

## RESTful API

- RESTful (Representational State Transferful) là một kiến trúc thiết kế cho việc phát triển dịch vụ web dựa trẹn các giao thức HTTP. Nó dựa trên một số nguyên tắc cơ bản giúp tạo ra các dịch vụ web dễ hiểu, dễ mở rộng và dễ sử dụng.

- RESTful API là một dạng API được thiết kế để hoạt động như một cách tiêu chuẩn để truyền tải dữ liệu giữa các máy chủ. Nó bao gồm như HTTP method như GET, POST, PUT và DELETE để thực hiện các thao tác trên tài nguyên.

### Ưu điểm:

* Linh hoạt: RESTful API có thể được triển khai trên nhiều ngôn ngữ lập trình khác nhau và chạy trên nhiều nên tản khác nhau.
* Dễ sử dụng: sử dụng các phương thức HTTP chuẩn, giúp dễ hiểu và dễ sử dụng.
* Dễ bảo trì: RESTful API tuân theo các nguyên tắc thiết kế và có cấu trúc tốt, giúp cho việc bảo trì và mở rộng ứng dụng dễ dàng hơn.

### Nhược điểm:

* Khó kiểm soát: RESTful API dựa trên các phương thức HTTP chuẩn, do đó khó kiểm soát các lỗi trong quá trình truyền tải dữ liệu.
* Hiệu suất kém: Vì sử dụng các HTTP, do đó có thể hiệu suất kém hơn so với các giải pháp tối ưu hơn cho việc truyền tải dữ liệu.

## REACT.JS

- ReactJS (hay React) là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook. Nó đươc sử dụngđể xây dựng giao diện người dùng cho các ứng dụng web đơn trang (Single Page Applications - SPA) và ứng dụng di động.

### Ưu điểm:

* Thành phần (Component) tái sử dụng: ReactJS dựa trên mô hình phát triển dựa trên thành phần, cho phép xây dưng các thành phần giao diện độc lập và tái sử dụng chúng trong nhiều phần của ứng dụng.
* Virtual DOM: Sử dụng DOM ảo để tối ưu hóa cập nhật giao diện, giúp cải thiện hiệu suất và tốc độ cập nhật.
* JSX: JSX cho phép viết mã giao diện một cách trực quan và dễ đọc, giúp tạo ra mã nguồn dễ bảo trì.
* Hệ sinh thái công cụ và thư viện phong phú: React có nhiều công cụ và thư viện hỗ trợ, chẳng hạn React Router, Redux, Material-UI, giúp phát triển ứng dụng nhanh chóng.
* Cộng đồng lớn và hỗ trỡ đa dạng: React có mợt cộng đồng phát triển lớn , nhiều tài liệu, hướng dẫn và hỗ trợ trực tuyến.
* Hỗ trợ React Native: ReactJS là cơ sở cho React Native, cho phép phát triển ứng dụng di động đa nền tảng bằng JavaScript.

### Nhược điểm:

* Khả năng học và sử dụng ban đầu khó khăn, cần phải mất một thời gian để làm quen và cần có kiến thức về JavaScript cần thiết. Trong các ứng dụng lớn việc quản lý trạng thái có thể phức tạp và yêu cầu hiểu biết về Redux hoặc các thư viện quản lý trạng thái khác.
* Đôi khi sử dụng React cho các dự án nhỏ có thể tạo thêm độ phức tạp không cần thiết.

## TAILWIND

- Tailwind CSS là một framework CSS mã nguồn mở giúp bạn xây dựng giao diện web nhanh chóng và dễ dàng. Nó sử dụng một hệ thống class CSS đã được định sẵn để giúp bạn thiết kế và tại ra thành phần giao diện một cách hiệu quả. Thay vì phải viết CSS tùy chỉnh cho từng phần tử HTML, bạn có thể sử dụng các class đã có để áp dụng kiểu dáng và chức năng cụ thể.

- Một trong những ưu điểm lớn nhất của tailwind là khả năng tùy chỉnh linh hoạt, với cộng đồng lớn và nguồn tài liệu phong phú.

## RSS FEED

- RSS feed là một dạng của thông tin tổ chức dưới dạng tập tin XML (Extensile Markup Languaue) hoặc dạng khác dùng để cung cấp các cập nhật từ một trang web, blog hoặc nguồn trực tuyến khác. RSS cứa thông tin tiêu đề bài viết, dường dưỡng đến bài viết hoặc trang gốc và thời gian đăng.

- RSS feed cho phép người dùng đăng ký và theo dõi nội dung từ nhiều nguồn khác nhau một cách thuận tiện. Thay vì phải thủ công kiểm tra các trang web để xem có cập nhật hay không, người dùng có thể sử dụng các ứng dụng đọc RSS (hoặc trình duyệt web tích hợp hỗ trợ RSS) để tự động nhận thông báo khi có nội dung mới từ các nguồn đã đăng ký.

- Các RSS feed thường có các biểu đồ như sau:

* `<rss>`: Thẻ gốc định nghĩa dữa liệu RSS
* `<channel>`: Định nghĩa một kênh thông tin, chứa tổng quan các nguồn thông tin.
* `<title>`: Tiêu đề của kênh, trang web hoặc bài viết.
* `<description>`: Mô tả ngắn về kênh, trang web hoặc bài viết.
* `<link>`: Đường dẫn đến trang gốc của kênh hoặc trang web liên quan.
* `<item>`: Một phần tử đại diện cho mỗi bài viết hoặc thông tin cụ thể.

# CƠ SỞ DỮ LIỆU

## Xác định các đối tượng của hệ thống:

Hệ thống báo online sẽ bao gồm những chức năng và các đối tượng sau:

#### Các chức năng chính: đây là các chức năng không yêu cầu người dùng phải xác thực.

* Lấy dữ liệu cho Database thông qua RSS.
* Đăng ký, đăng nhập tài khoản.
* Hiển thị trang chủ chứa tất cả bài báo và trang xem chi tiết bài báo.
* Lọc tin tức theo chủ đề và tìm kiếm bài báo theo tiêu đề.

#### Các chức năng phân quyền:

### Chức năng của User:

* Bình luận trong các bài báo.
* Đánh giá các bài báo.
* Tố cáo các bài báo.

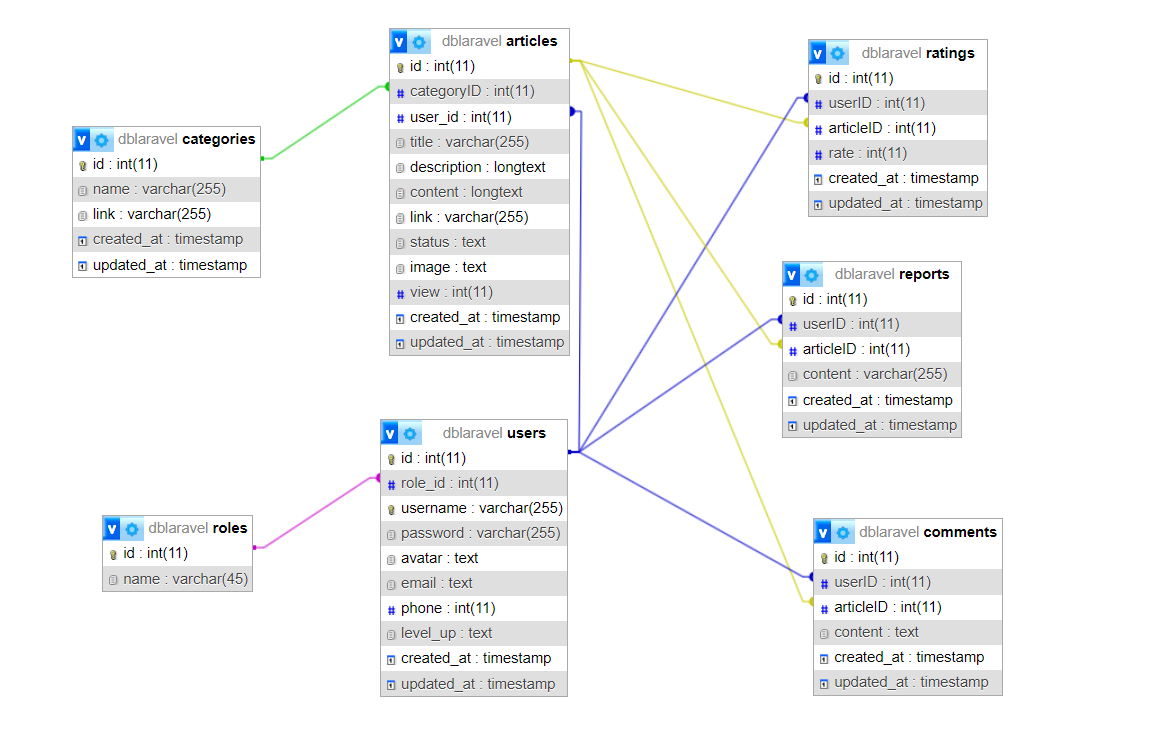
### Chức năng của Author:

* Thêm, xóa, sửa bài báo.
* Kế thừa các chức năng User.

### Chức năng của Admin:

* Kiểm duyệt yêu cầu.
* Quản lý User.
* Quản lý bài báo.
* Kế thừa các chức năng từ Author.

## Mô Hình ERD:

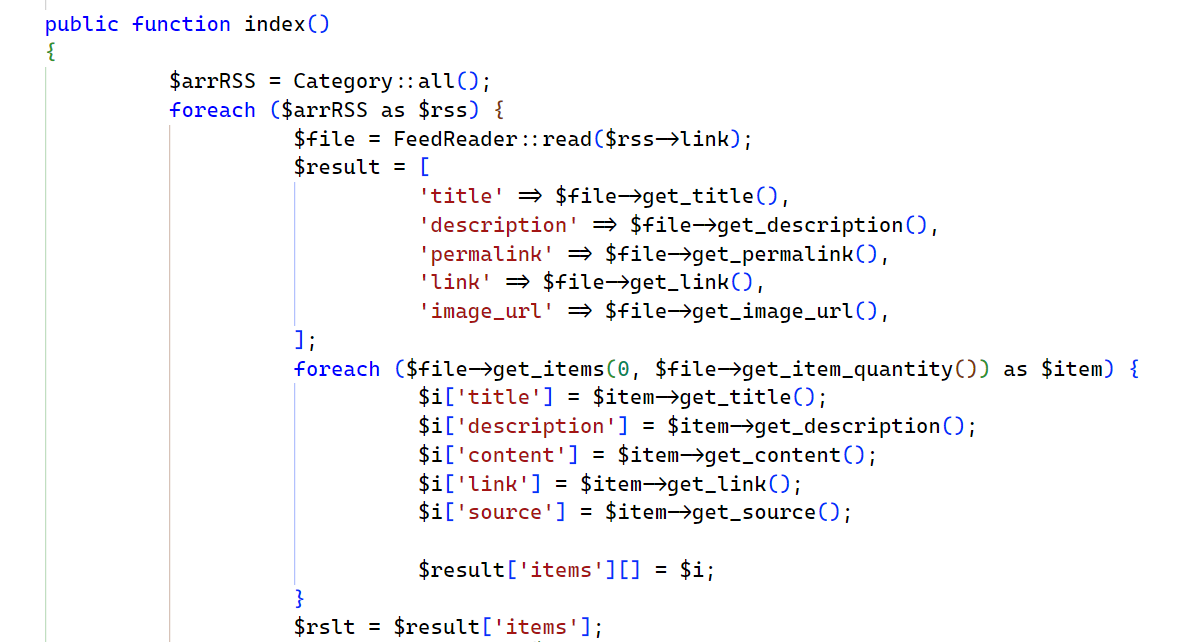


###### Ảnh mô hình ERD

# CÁC CHỨC NĂNG CHÍNH

## Lấy dữ liệu thông qua RSS:

* Để có thể fetch dữ liệu RSS (trong hệ thống này sẽ lấy RSS từ trang VNExpress) thì package laravel-feed-reader sẽ hỗ trợ ta lấy dễ dàng hơn.



###### Ảnh code minh họa lấy dữ liệu.

* Sau đó sẽ thực hiện lưu trữ vào database và trả response để phía client gọi ra giao diện.

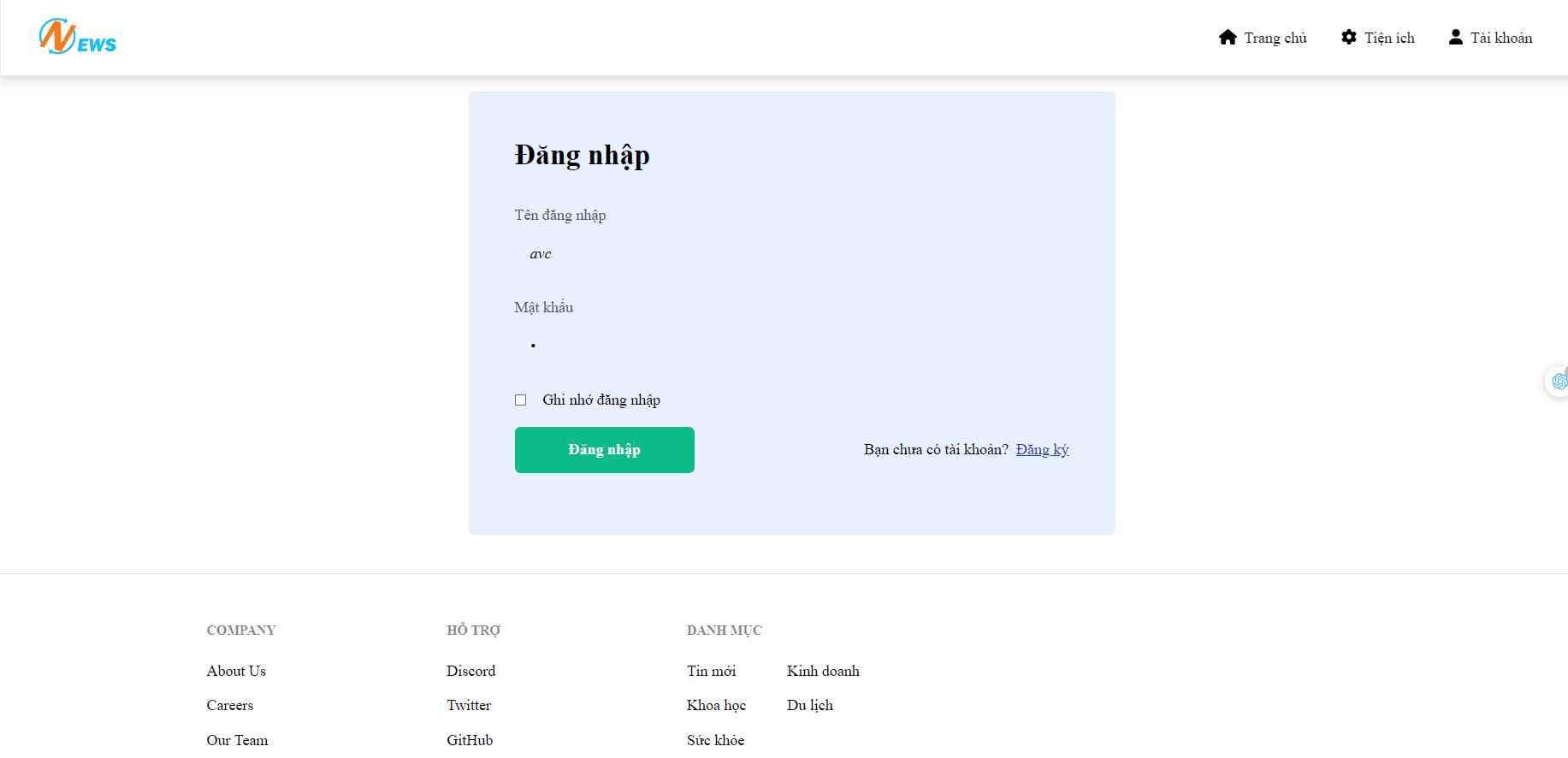


###### Ảnh code minh họa lưu trữ dữ liệu.

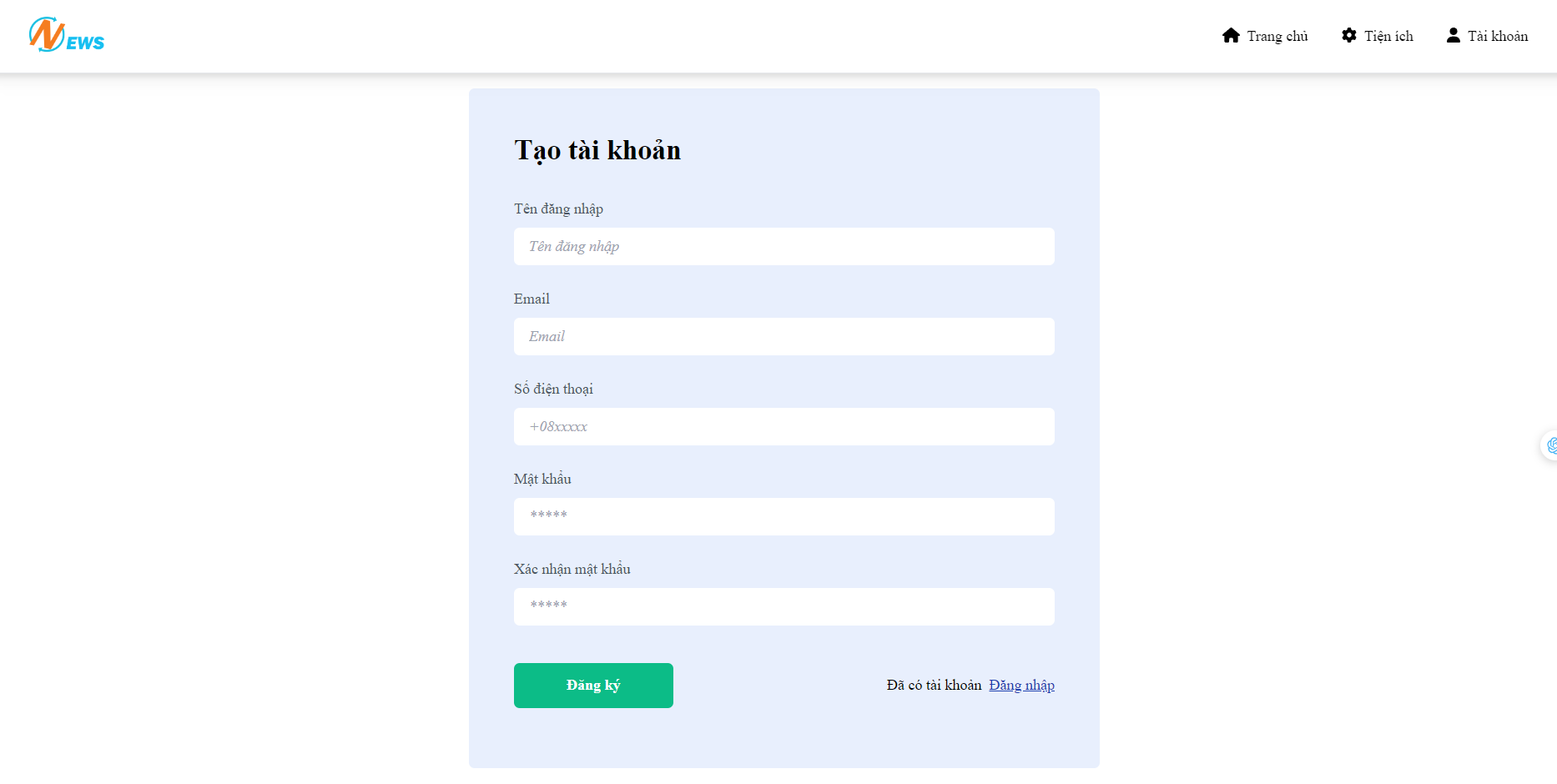
## Đăng ký, đăng nhập tài khoản:

* Hiện nay có rất nhiều cách thức để xác thực và trong hệ thống này sẽ sử dụng JWT để làm chức năng này.
* Trong Laravel để sử dụng JWT cần phải import vào một package, trong hệ thống này là jwt-auth.

#### Phần giao diện:



###### Ảnh giao diện đăng nhập.



###### Ảnh giao diện đăng ký.

#### Phần code:

- Tạo một controller để xử lý các chức năng liên quan đến các tài khoản ở đây là ApiUserController.

- Các chức năng phục vụ cho việc xác thực sẽ bao gồm login, register.

+ Phía client: client sẽ gửi username và password về server để kiểm tra xác thực, nếu đúng sẽ lưu cookie thông tin user đăng nhập đó trong suốt quá trình hoạt động ở phần login hoặc thêm thông tin user đó vào database ở phần register.

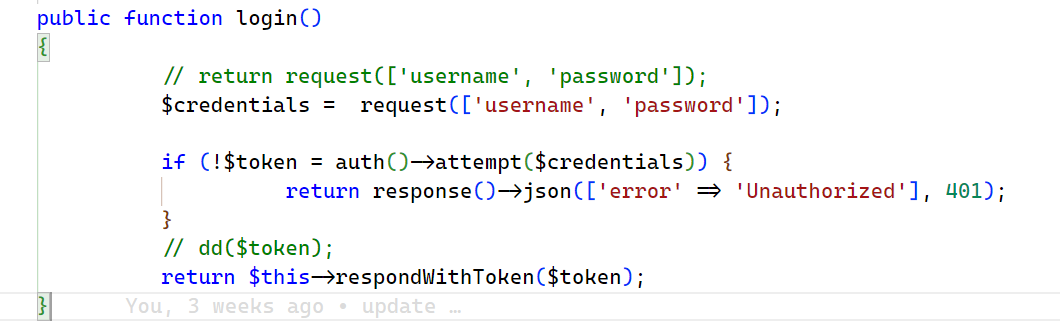


###### Ảnh code phần login phía client

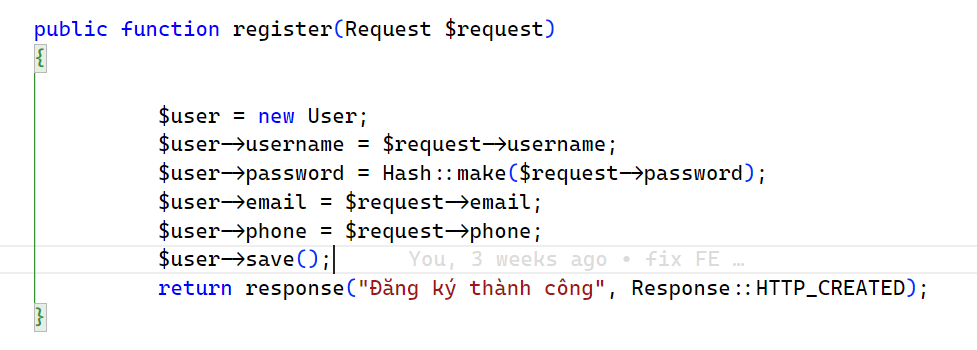


###### Ảnh code phần register phía client

+ Phía server: Kiểm tra thông tin username và password được gửi từ Client và so sánh theo database để xác thực thông tin, nếu đúng sẽ trả về client Cookie lưu thông tin user.



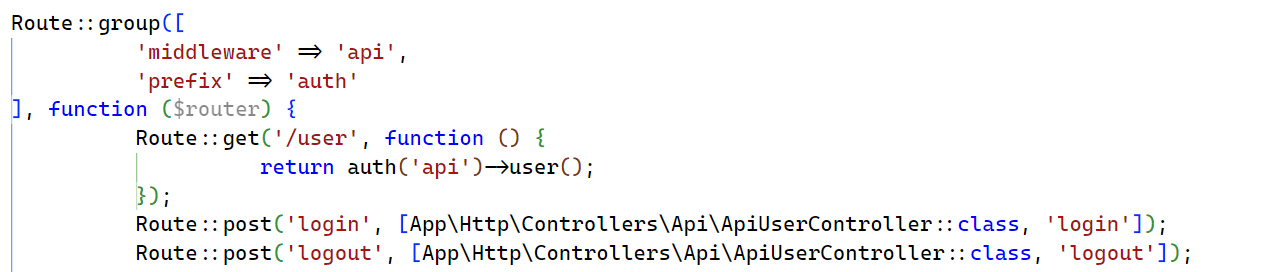
###### Ảnh code phương thức login phía server.



###### Ảnh code phương thức register phía server.

#### Phần API:

* Để có thể sử dụng ở phía client thì ta sẽ phải khai báo routes trong file api.php của Laravel



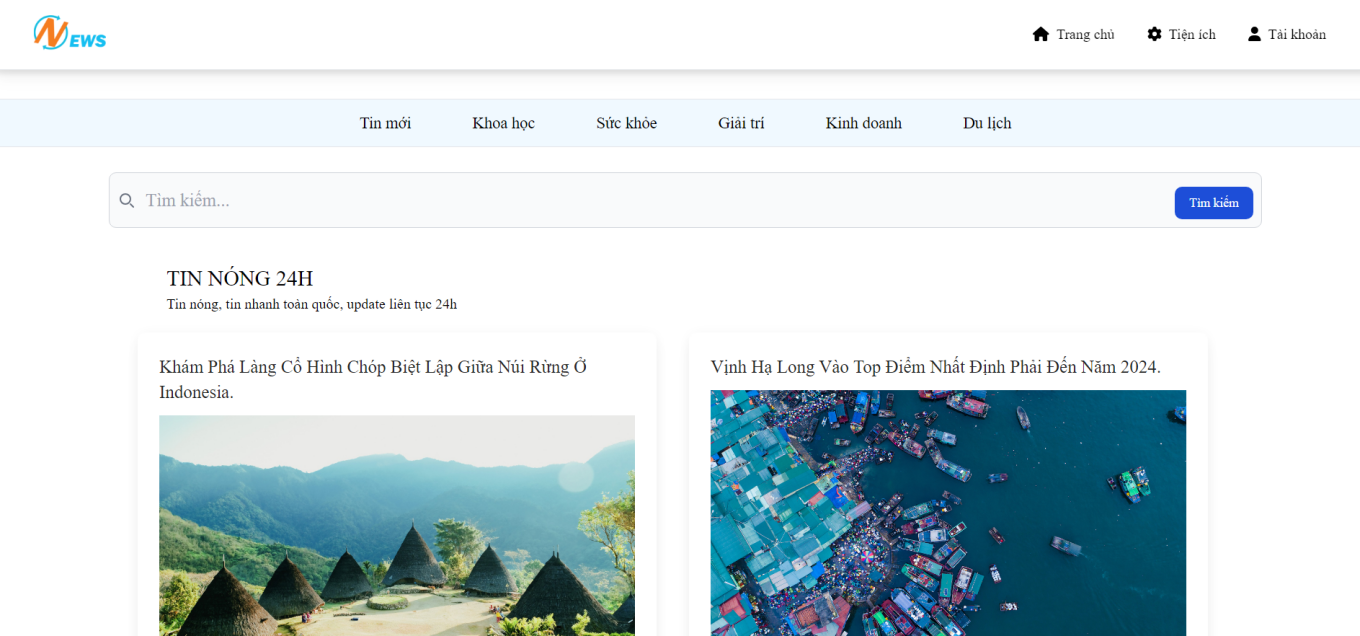
###### Ảnh route api của login, register.

## Hiển thị trang chủ (Home):

* Trang chủ sẽ hiển thị tất cả bài báo, tin tức, bao gồm cả search engine theo từ khóa bằng thanh tìm kiếm và search theo thể loại bằng danh mục và phần phân trang theo số lượng bài báo.

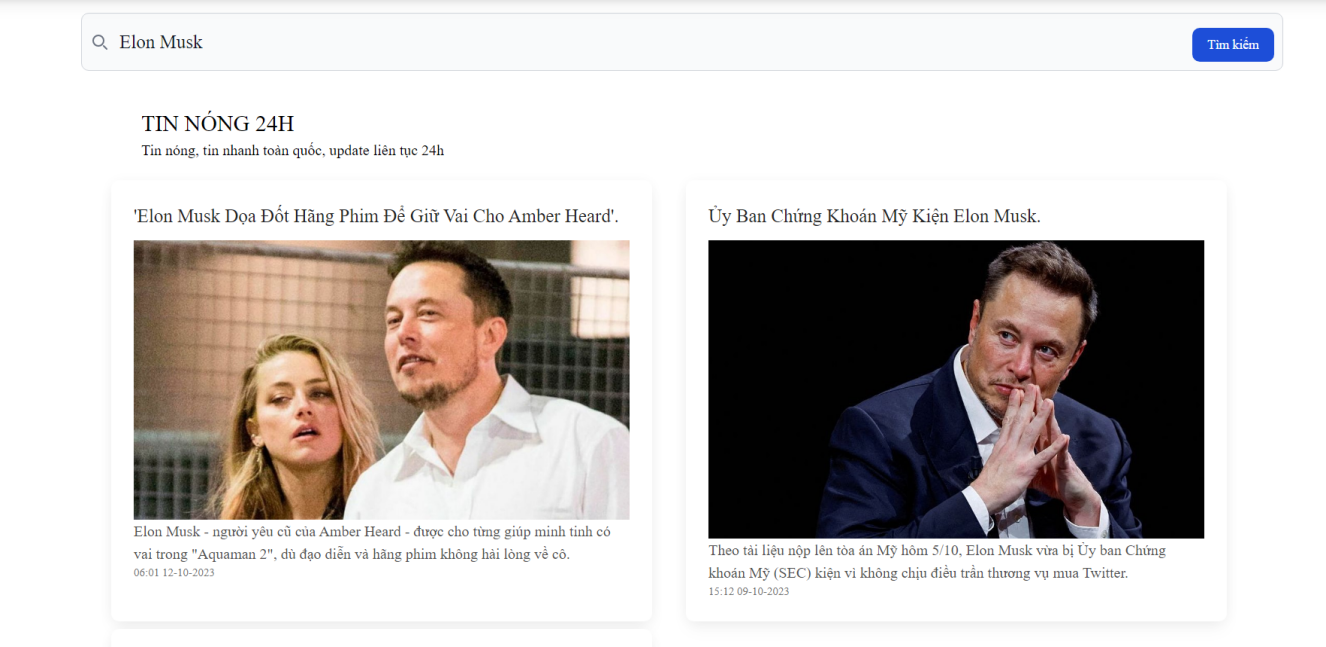
#### Phần giao diện:

* Ở trang chính sẽ chứa phần danh mục, khi click vào sẽ trả ra kết quả tìm kiếm theo thể loại danh mục đó.

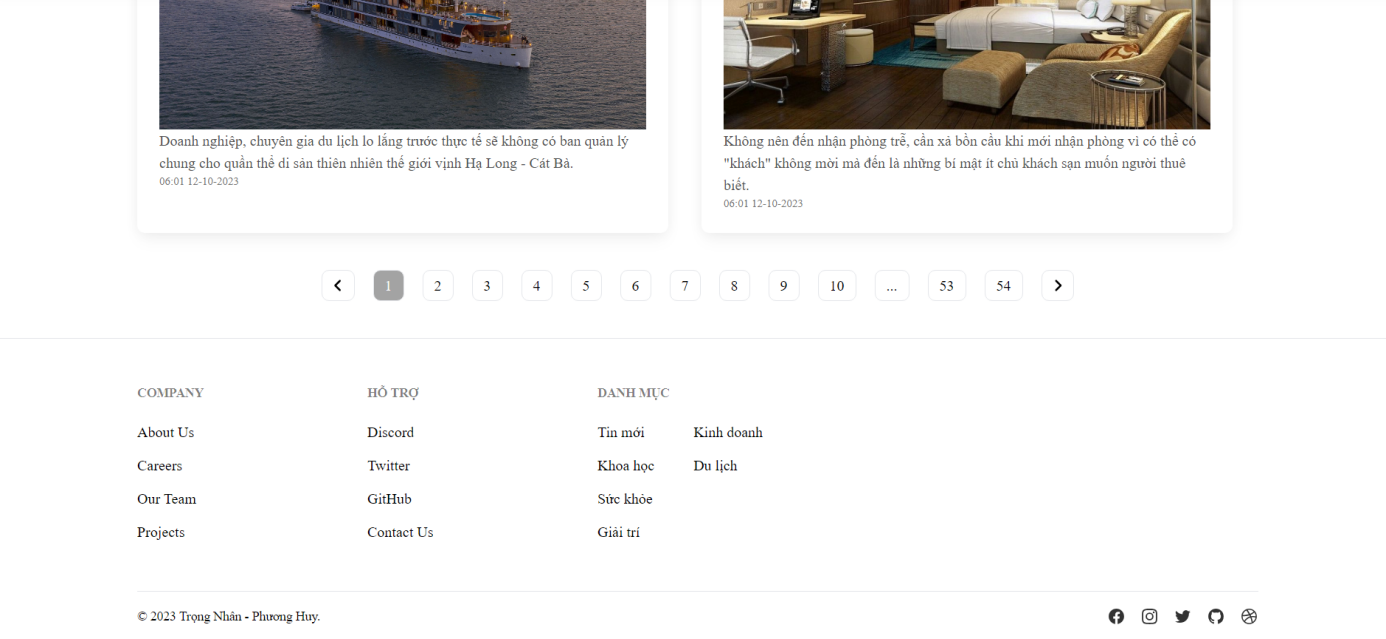


###### Ảnh trang chủ.

+ Sau đó là thanh tìm kiếm Search Engine cũng sẽ làm nhiệm vụ tìm kiếm, nhưng người dùng sẽ nhập từ khóa và tìm theo title (tiêu đề) của từng bài báo.



###### Ảnh kết quả trả về của chức năng search.

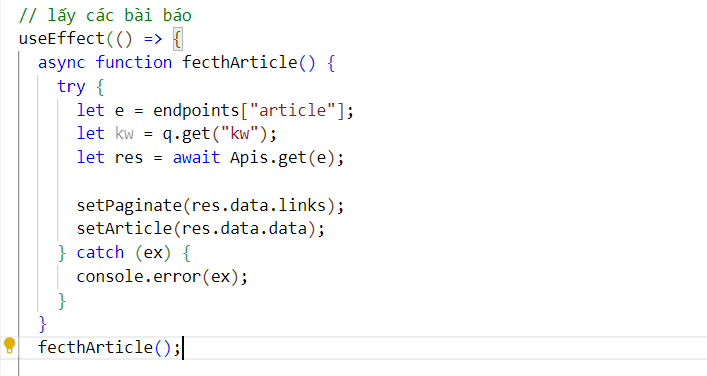
* Giao diện phân trang: 

###### Ảnh giao diện phân trang.

#### Phần code:

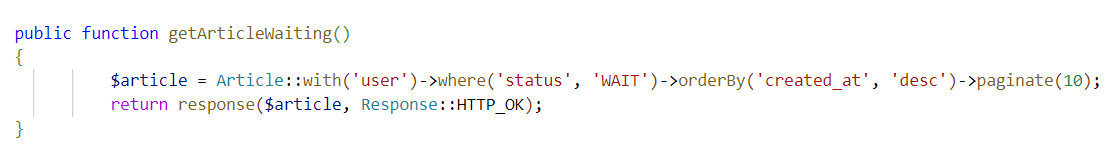
Data được lấy về từ Api luôn theo dạng Json, sau khi lấy về sẽ gán vào biến cần dùng.

* Hàm lấy dữ liệu phân trang và dữ liệu bài báo: gán dữ liệu link phân trang vào setPaginate (dữ liệu phân trang) và dữ liệu các bài báo vào setArticle (dữ liệu bài báo) đã khai báo trước.



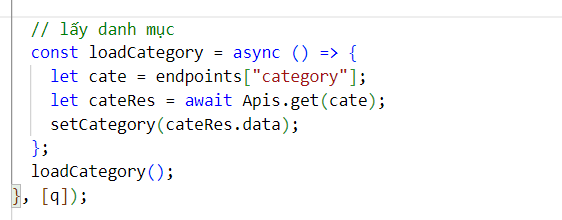
###### Ảnh code lấy dữ liệu báo từ phía client.

* Hàm xử lý dữ liệu các bài báo phía sever: xử lý lấy dữ liệu tất cả các bài báo trong databse và gửi lên phía client thông qua API.



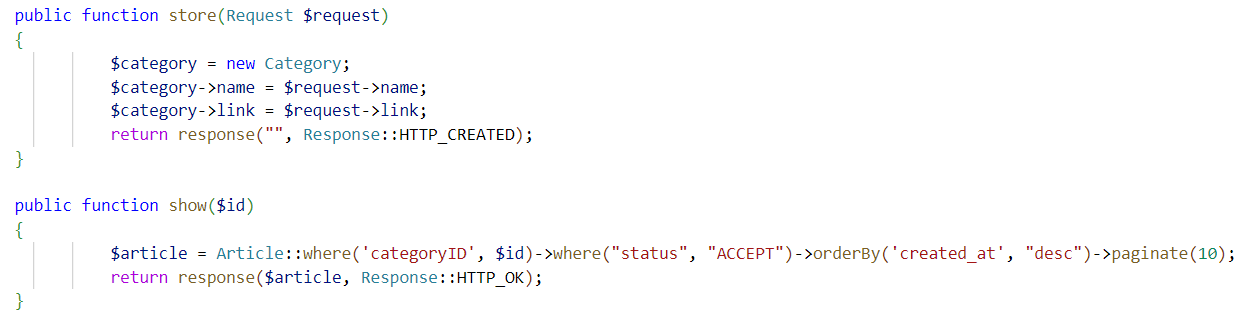
###### Ảnh code xử lý lấy dữa liệu các bài báo phía server

* Hàm lấy dữ liệu danh mục: xử lý gán dữ liệu các chủ đề danh mục vào setCategory để dùng.



###### Ảnh code lấy chủ đề danh mục từ phía client.

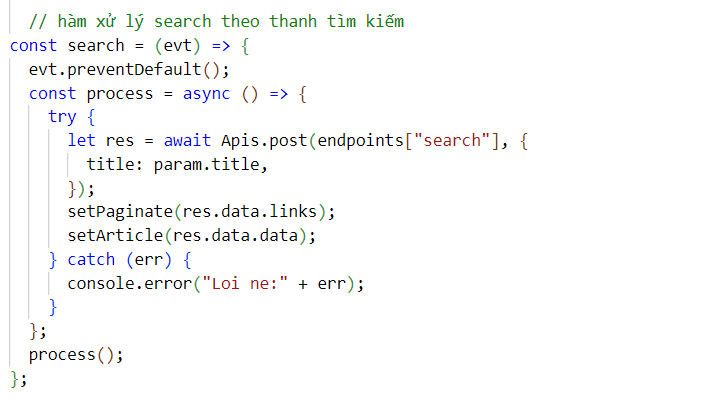
* Hàm xử lý danh mục phía server: lấy danh mục từ database để đưa lên phía client thông qua API:



###### Ảnh code xử lý chủ đề danh mục phía server

* Hàm xử lý search engine theo kw (keyword) ở thanh tìm kiếm phía client:

Lấy dữ liệu keyword gửi vào API để xử lý phía backend và response lại dữ liệu tương ứng rồi gán vào setArticle và setPaginate.



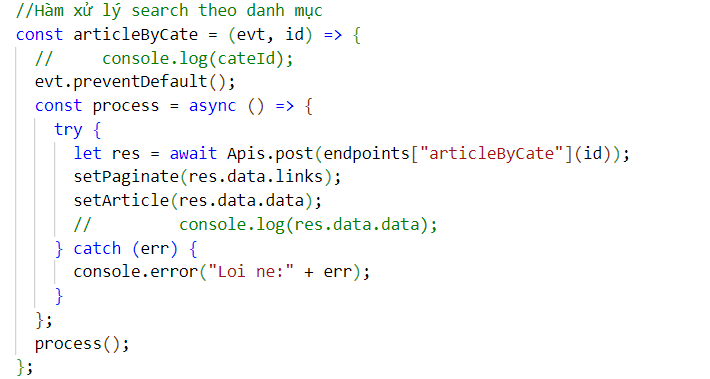
###### Ảnh code thực hiện tìm kiếm từ phía client.

* Hàm xử lý tìm kiếm theo kw (keyword) phía server: Nhận kw từ phía client sau đó tìm kiếm theo title từng bài báo rồi trả kết quả về client thông qua API.



###### Ảnh code xử lý tìm kiếm theo kw phía server

* Hàm xử lý tìm kiếm theo danh mục: gửi id của danh mục về backend để xử lý rồi response lại dữ liệu gán vào setPaginate và setArticle



###### Ảnh code thực hiện lọc theo chủ đề từ phía client.

* Hàm xử lý phân trang phía client: Xử lý hiện só lượng trang, xử lý event click button tương ứng theo từng trang.



###### Ảnh code xử lý phân trang phía client.

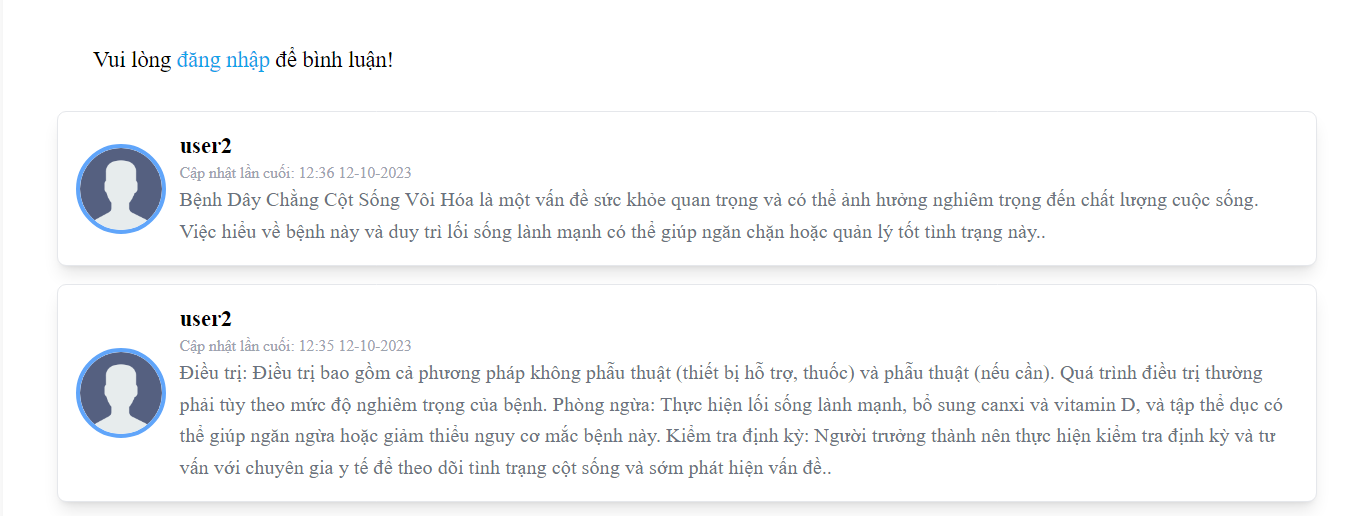
### Hiển thị chi tiết bài báo:

* Phần giao diện: gồm hai phần chính là hiển thị chi tiết bài báo và phần đánh giá bình luận của đọc giả. Tất cả đều có xử lý responsive theo kích thước màn hình:
* Giao diện phần hiển thị chi tiết bài báo:

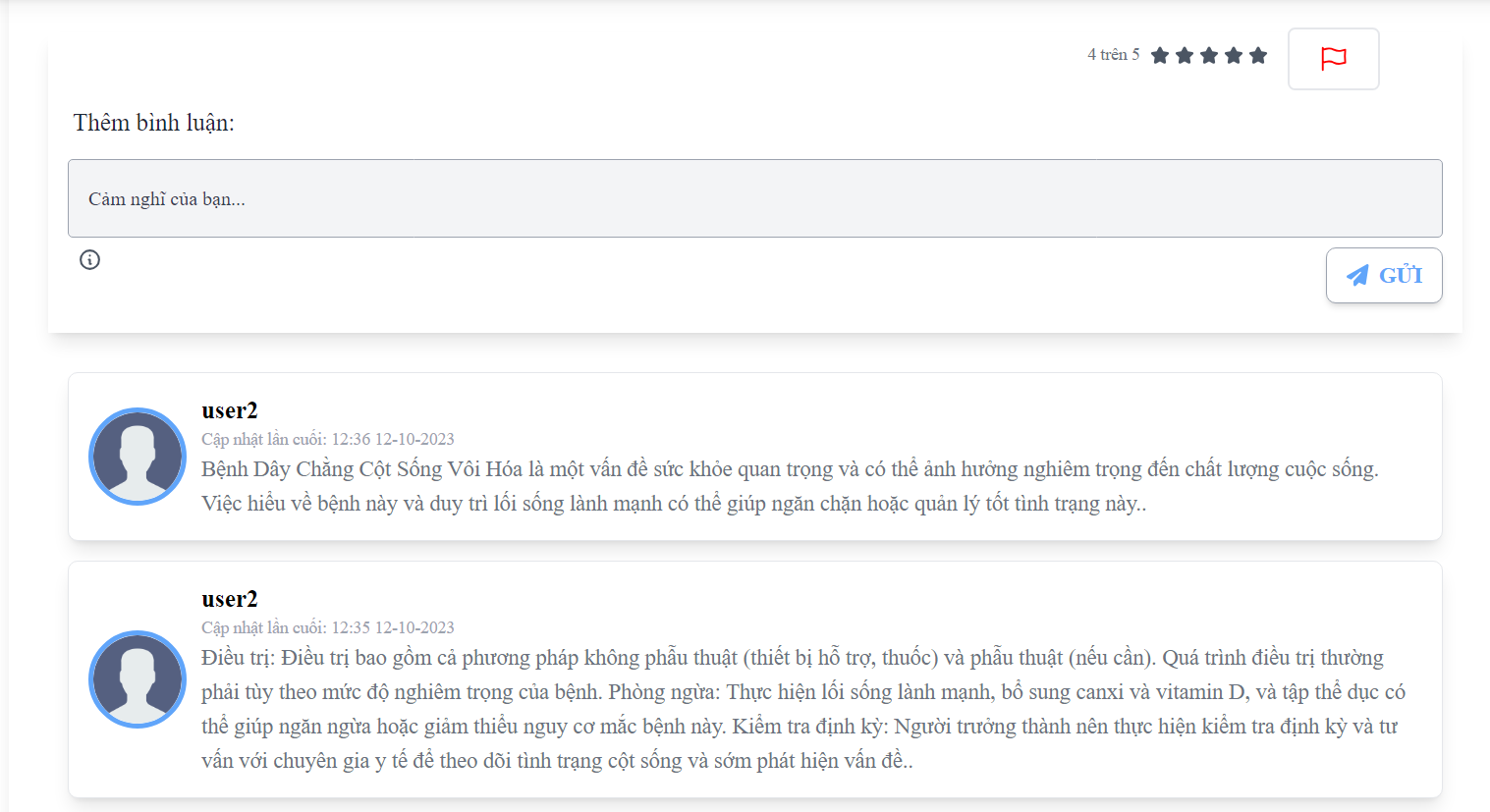


###### Ảnh chi tiết bài báo

* Giao diện phần bình luận, đánh giá: người dùng bắt buộc phải đăng nhập để được bình luận và đánh giá.



###### Ảnh phần bình luận trước khi được xác thực đăng nhập.



###### Ảnh phần bình luận sau khi đã xác thực đăng nhập.

#### Phần code:

* Hàm xử lý dữ liệu đầu vào và gán dữ liệu vào các biến setArticle, setComments, setRatings đã khai báo trước để sử dung.

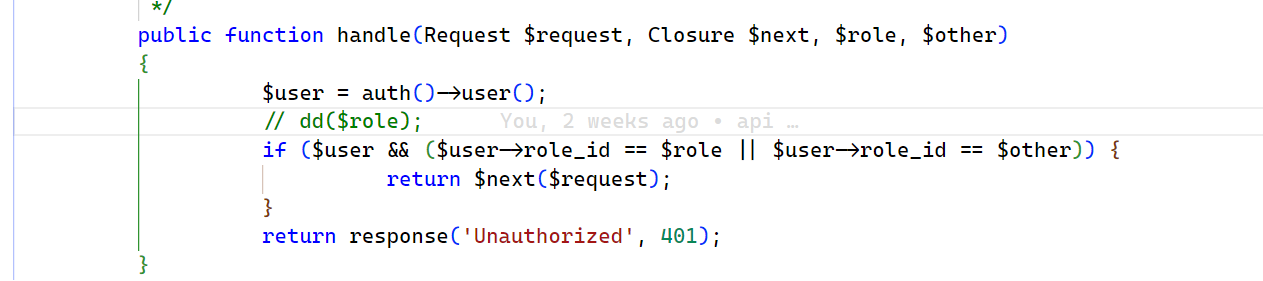


###### Ảnh code phần lấy xử lý dữ liệu thông qua API.

# CÁC CHỨC NĂNG PHÂN QUYỀN

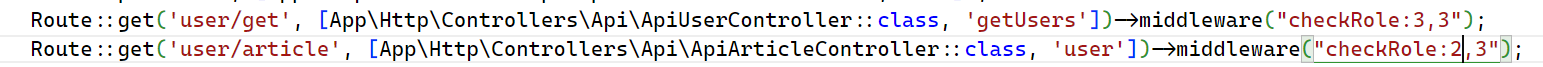
## Cách phân quyền

- Để phân quyền cần phải khai báo, cấu hình và xử lý một middleware.



###### Ảnh xử lý phân quyền theo role của CheckRole Middleware

- Sử dụng middleware lên route để áp dụng. Ở đây có nhiều route cần áp dụng 2 role nên có thể khai báo trùng nhau.



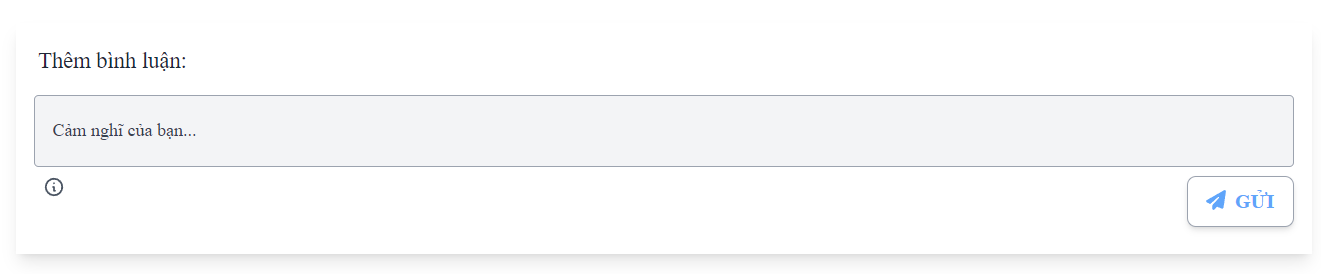
###### Ảnh sử dụng middleware cho role 2 (Author) và role 3 (Admin)

## Chức năng của User

### Bình luận (Comment)

* Để có thể bình luận, người dùng phải xác thực đăng nhập để có thể sử dụng API

#### Phần giao diện:



###### Ảnh giao diện để xử lý bình luận.

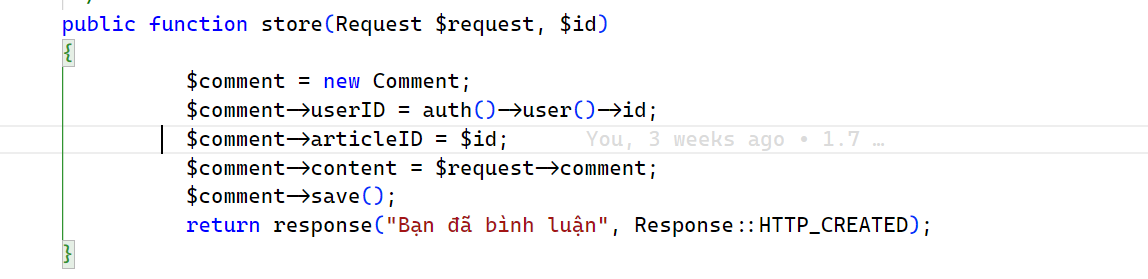
#### Phần code:

+ Phía Client: vì để có thể bình luận thì cần phải đăng nhập nên authApi sẽ được dùng để lấy chứng thực từ JWT và từ đó request comment sẽ được gửi về server



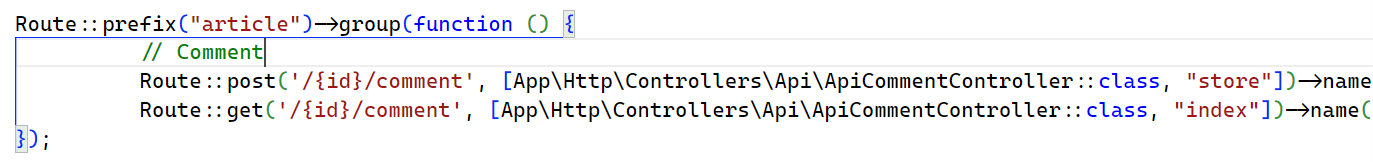
###### Ảnh code thực hiện chức năng comment phía client.

+ Phía Sever:



###### Ảnh code thực hiện chức năng comment phía server.

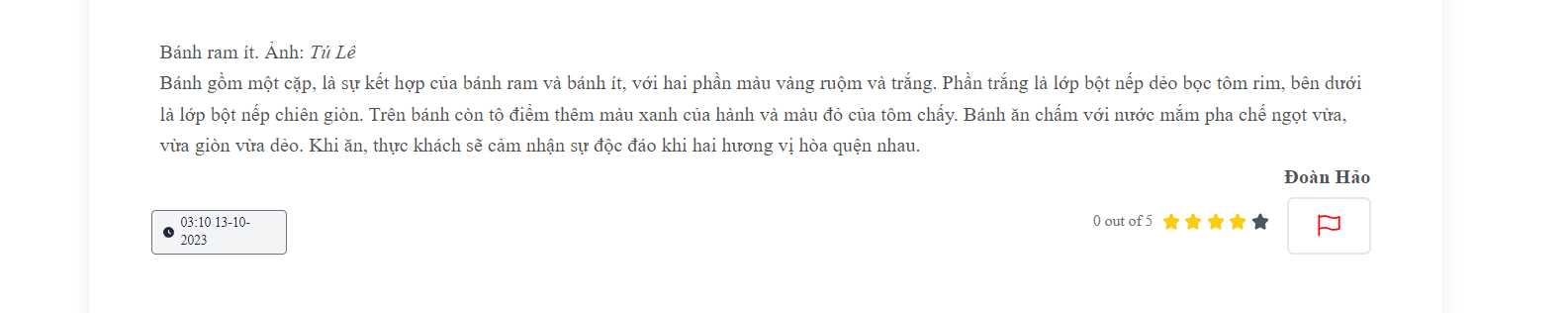
#### Phần API:



###### Ảnh route liên quan đến comment.

### Đánh giá (Rating)

* Người dùng cần đăng nhập để đánh giá bài báo, sau khi đánh giá hệ thống sẽ tính trung bình các đánh giá của người dùng về bài báo đó.
* Phần giao diện:



###### Ảnh giao diện tương tác đánh giá.

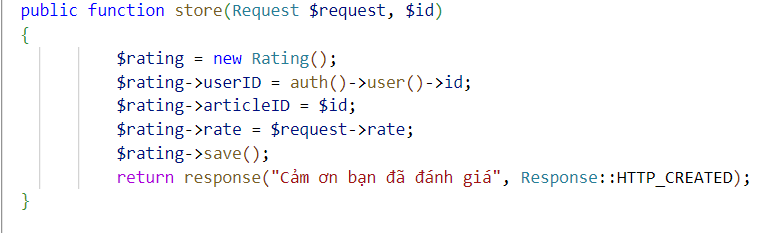
#### Phần code:

* Phía client:



###### Ảnh code xử lý đánh giá từ phía client.

* Phía server: server sẽ lưu thông tin báo về bài viết đó vào database.

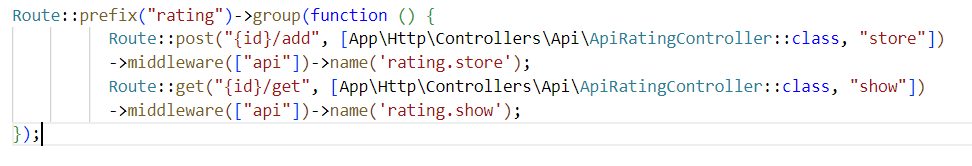


###### Ảnh code xử lý phần đánh giá phía server.

#### Phần API:



###### Ảnh code xử lý lấy dữ liệu rating để gửi lên frontend.

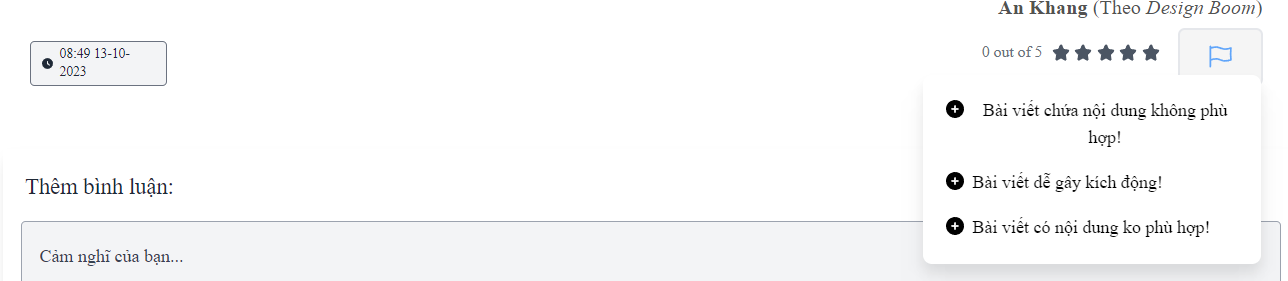


###### Ảnh code API phần rating ở phía server.

### Báo cáo (Report)

* Khi cảm thấy bài báo có một số vấn đề, người dùng có thể báo cáo lên hệ thống với điều kiện tài khoản cần xác thực.

#### Phần giao diện:



###### Ảnh giao diện thực hiện việc tố cáo.

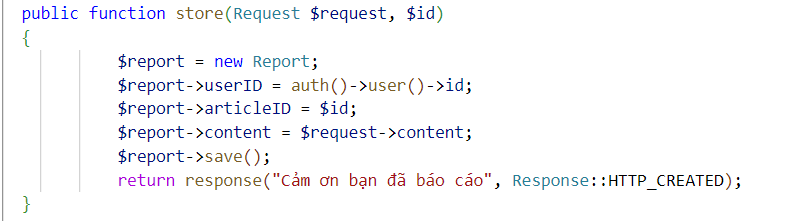
#### Phần code:

* Phía client:



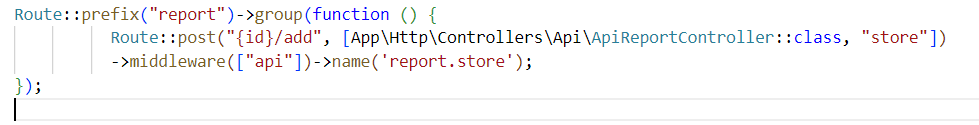
###### Ảnh code xử lý tố cáo bài báo từ phía client.

* Phía server:



###### Ảnh code xử lý thêm báo cáo bài báo vào database phía server.

#### Phần API:



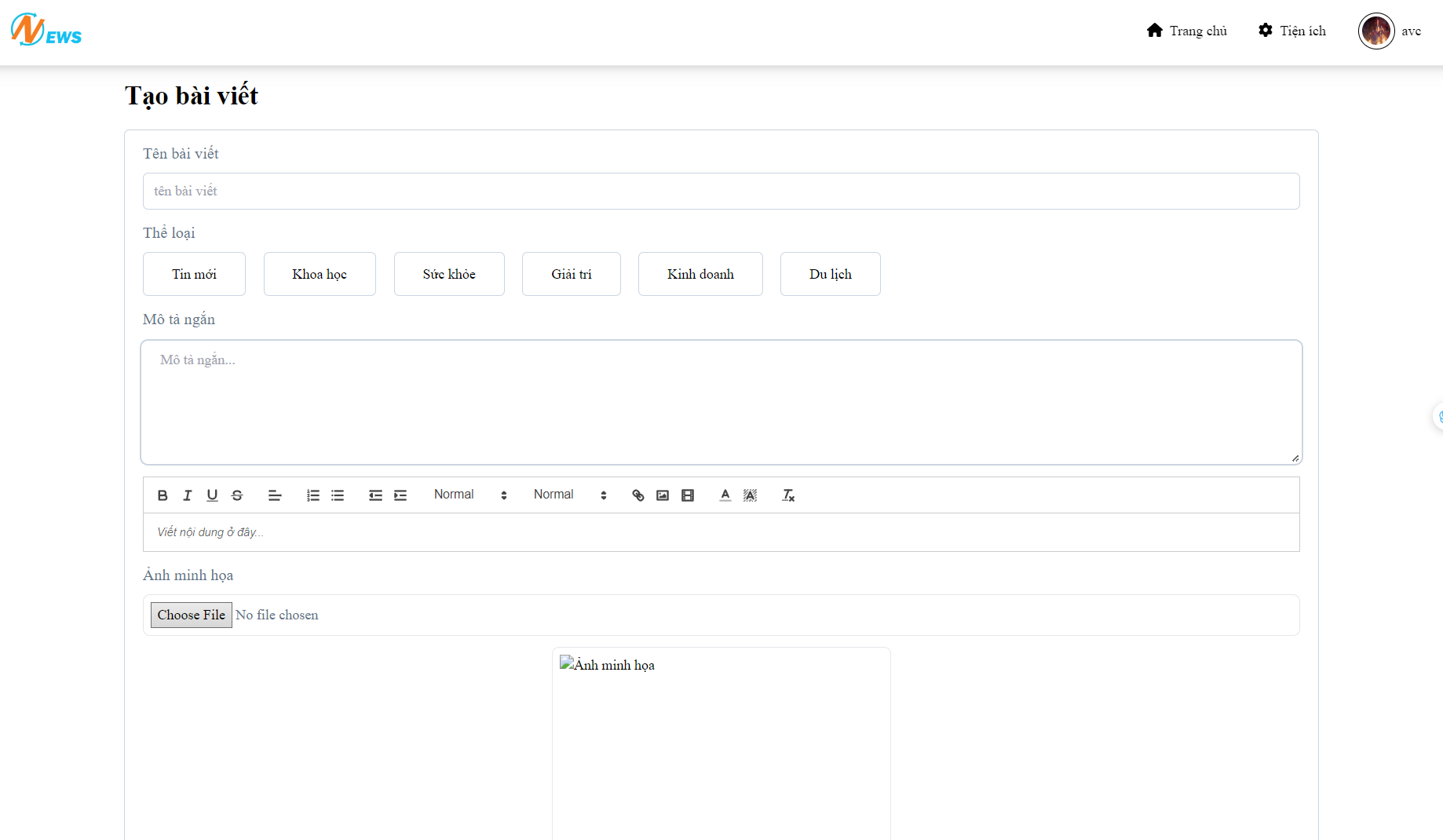
###### Ảnh code API lấy thông tin report từ phía client về.

## Chức năng của Author:

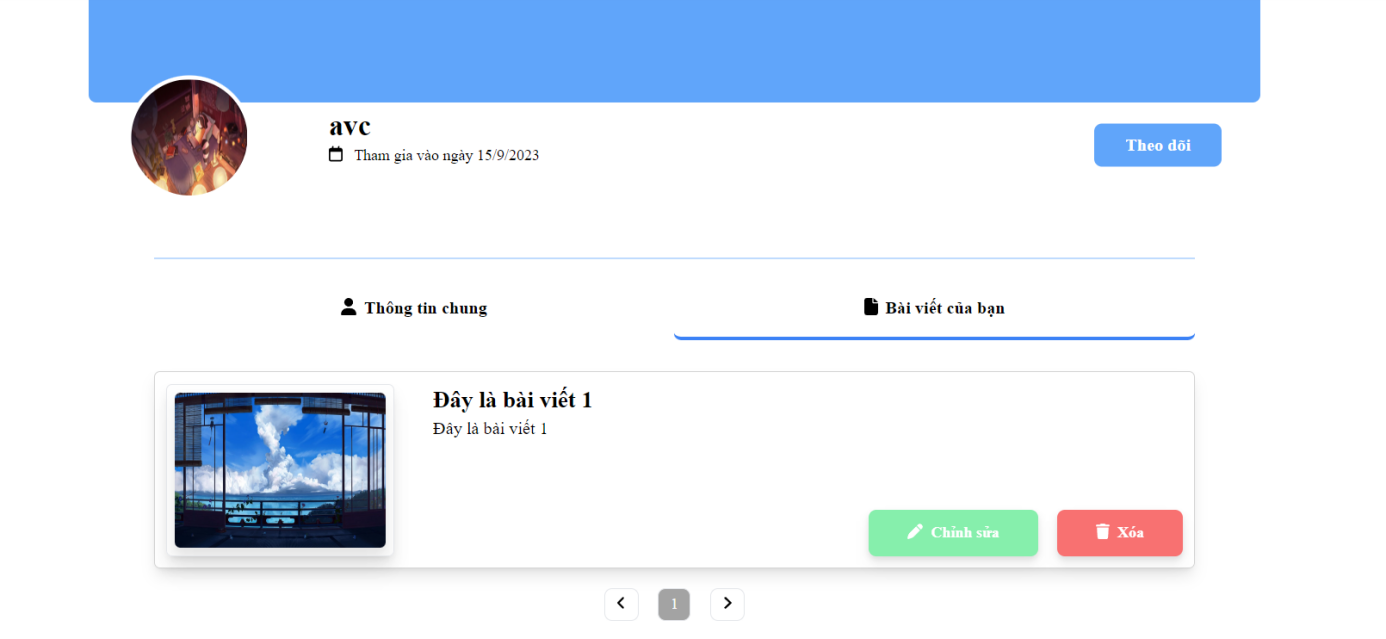
### Tạo, cập nhật và xóa bài báo:

* Các tác giả có thể tạo và tùy chỉnh các bài báo theo form đã cho sẵn nhưng để được đăng lên trang chủ thì các bài báo phải được Admin phê duyệt.

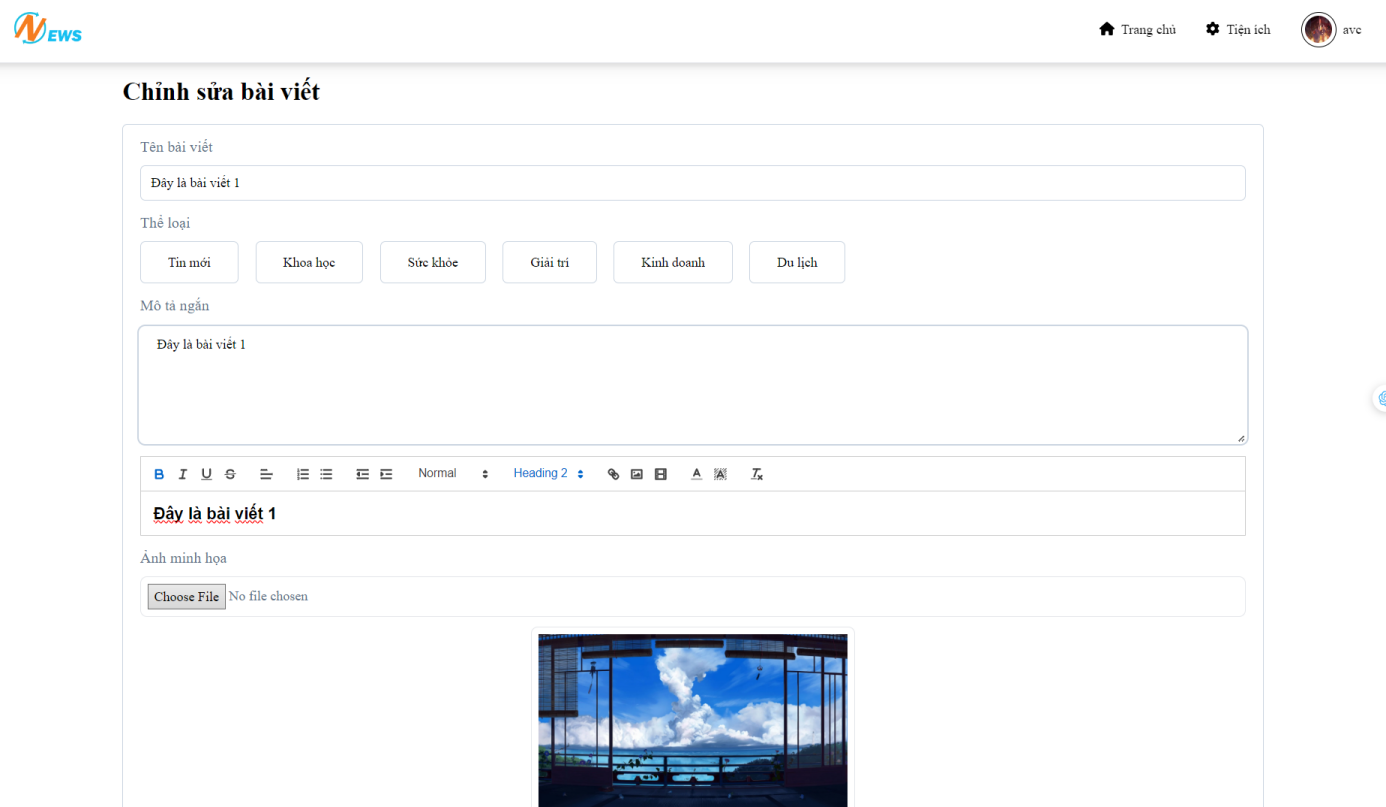
#### Phần giao diện:



###### Ảnh giao diện tạo bài viết mới.



###### Ảnh giao diện hiển thị các bài viết của author.



###### Ảnh giao diện chỉnh sửa bài báo.

#### Phần code:

* Phía server: xử lý thêm, sửa, xóa và cập nhật vào database.



###### Ảnh code xử lý thêm bài báo mới phía client.

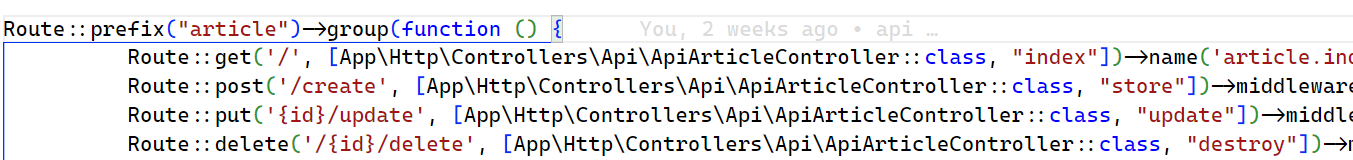


###### Ảnh code xử lý chỉnh sửa, cập nhật bài báo.

**

###### Ảnh code xử lý xóa bài báo.

#### Phần API:



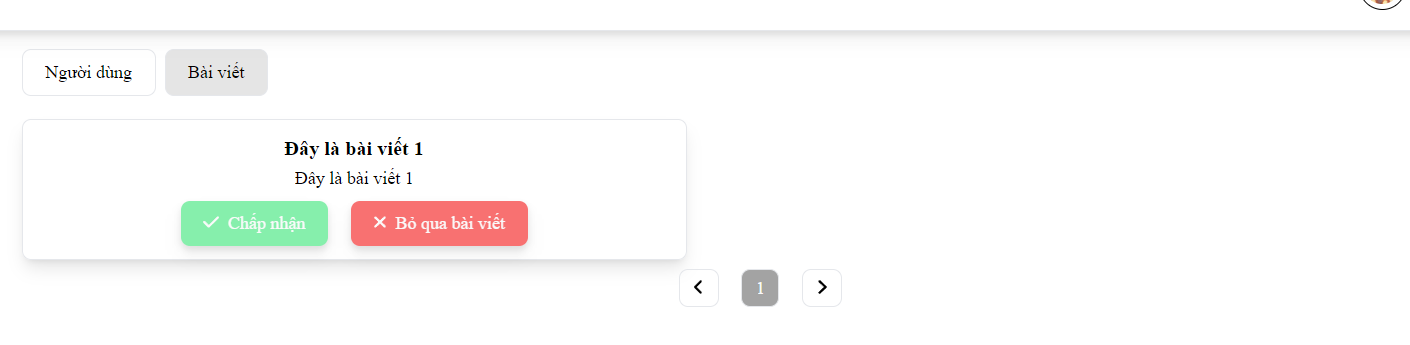
###### Ảnh route api liên quan đến việc tạo, cập nhật, xóa bài báo.

## Chức năng của Admin:

### Kiểm duyệt yêu cầu:

* Đây là chức năng sẽ kiểm duyệt các yêu cầu như cho phép đăng bài báo và nâng cấp role để được phép đăng bài báo.

#### Phần giao diện:



###### Ảnh giao diện kiểm duyệt bài báo.



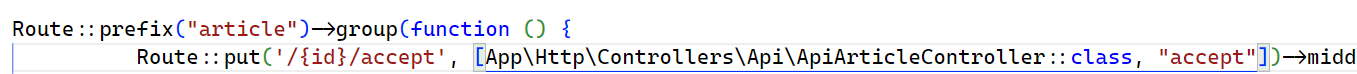
###### Ảnh giao diện kiểm duyệt yêu cầu nâng cấp role của user.

#### Phần code:



###### Ảnh code xử lý kiểm duyệt bài báo và user phía client.

#### Phần API:



###### Ảnh route dành cho việc kiểm duyệt bài báo

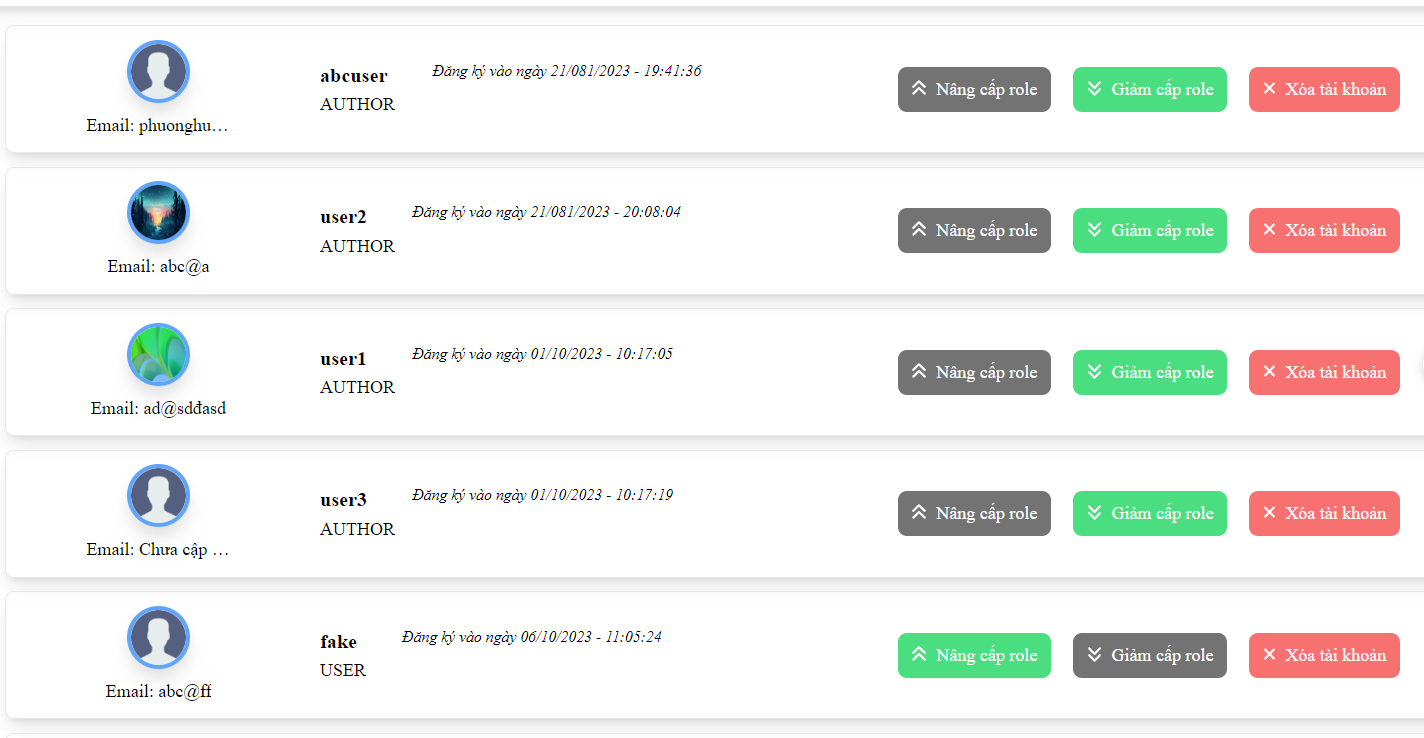


###### Ảnh route dành cho việc nâng cấp role (role-up, cancel).

### Quản lý user:

* Tại đây Admin có quyền nâng cấp hay hạ cấp role của bất kì người dùng nào và kể cả xóa tài khoản cũng vậy.

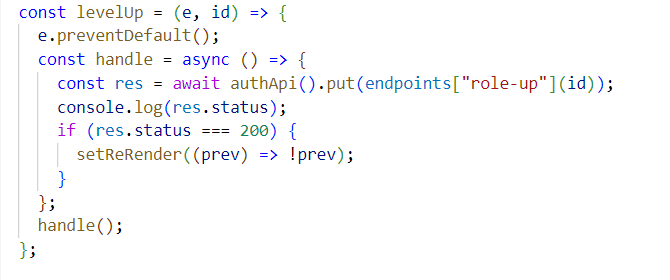
#### Phần giao diện:



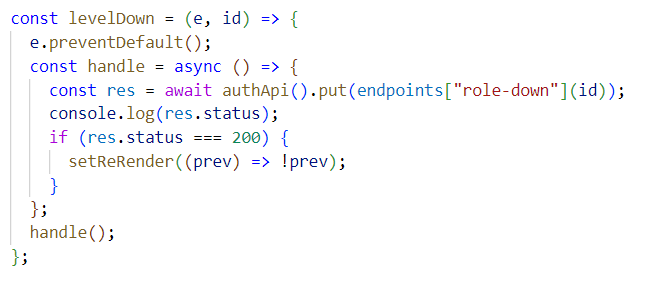
###### Ảnh giao diện quản lý user.

#### Phần code:

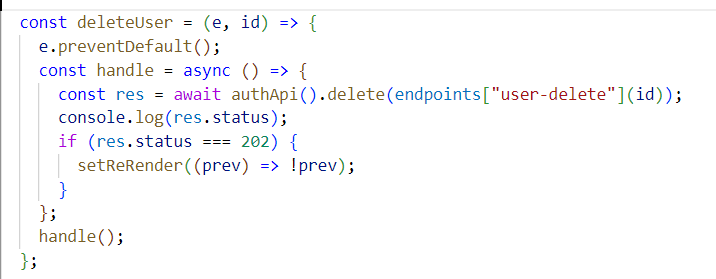
* Phía client:



###### Ảnh code xử lý nâng cấp role phía client thông qua API.

**

###### Ảnh code xử lý đặt lại role phía client thông qua API.

**

###### Ảnh code xử lý xóa tài khoản phía client thông qua API.

+ Phía server:

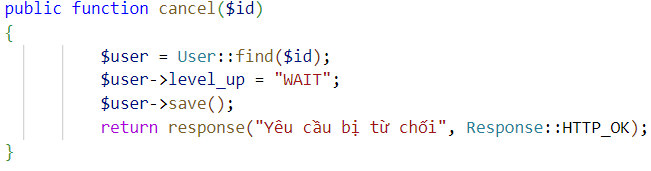
- Hàm lưu thông tin về yêu cầu nâng role cho user vào database và hàm lấy thông tin đó ra gửi cho phía client thông qua API.



###### Ảnh code hàm lưu thông tin và hàm lấy thông tin ra.

**

###### Ảnh code hàm xử lý nâng cấp và giảm role phía server.

**

###### Ảnh code hàm từ chối yêu cầu của user (hủy).

- Khi xóa một user sẽ xảy ra 2 trường hợp:

+ User không tương tác với hệ thống thì sẽ xóa trực tiếp

+ User có tương tác với hệ thống thì cần phỉa xóa từ những table đang có liên kết khóa ngoại.

**

###### Ảnh code hàm xóa tài khoản user.

#### Phần API:



###### Ảnh route liên quan đến việc quản lý user.

### Quản lý bài báo:

* Các bài báo tại đây là các bài được thống kê theo số lượng report, comment và lượt xem của các người dùng. Nếu có quá nhiều tố cáo Admin có thể xóa.

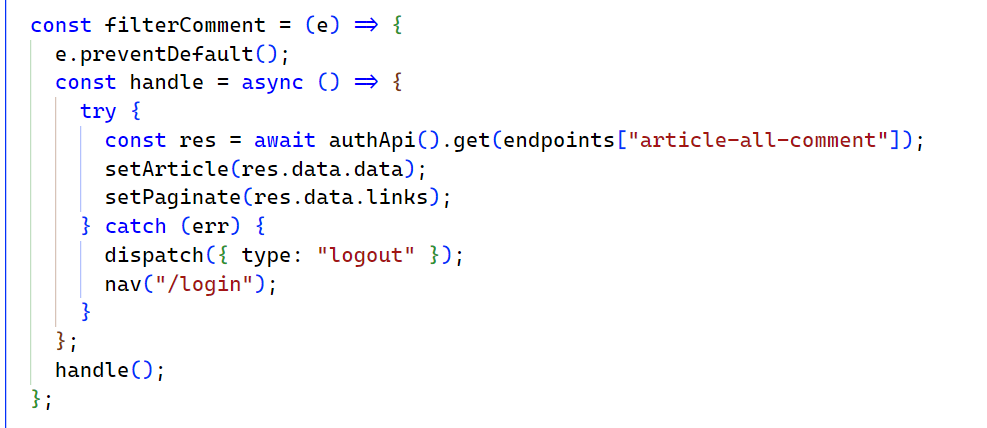
#### Phần giao diện:



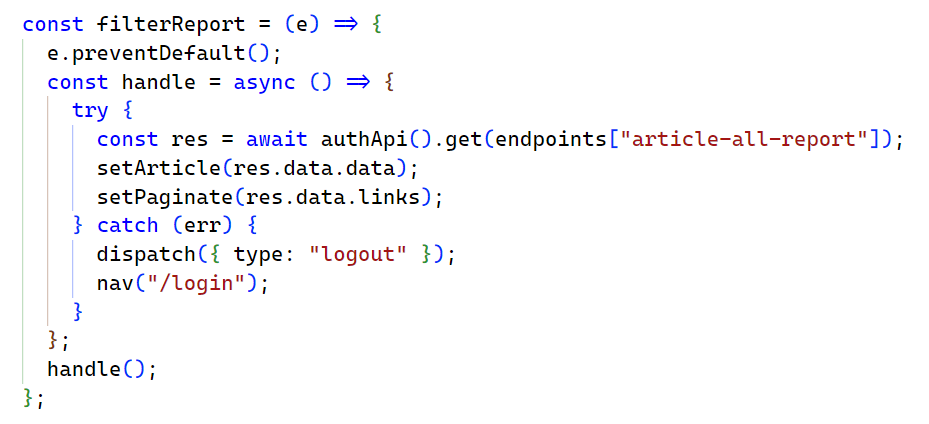
###### Ảnh giao diện của chức năng quản lý bài báo.

#### Phần code:

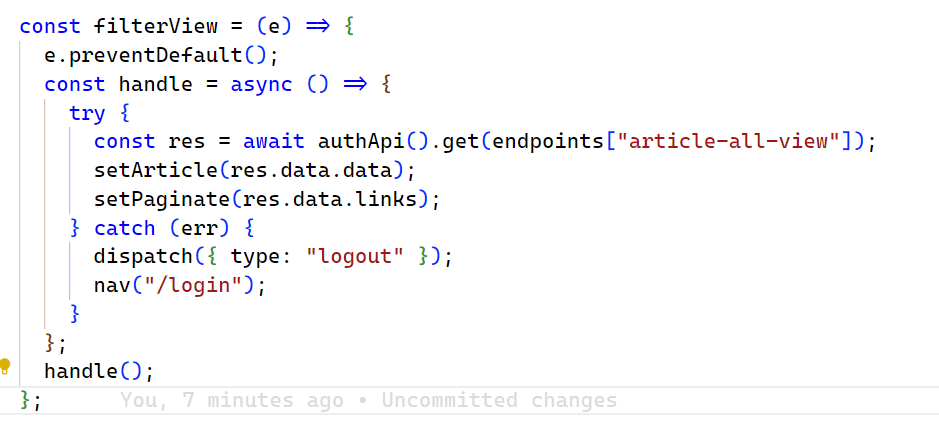
* Phía client:



###### Ảnh code xử lý lọc báo theo comment.

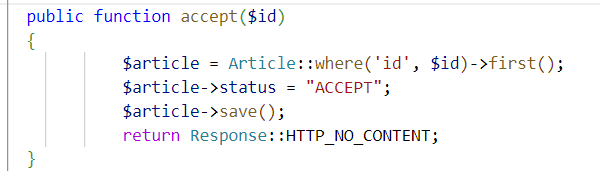


###### Ảnh code xử lý lọc theo tố cáo.



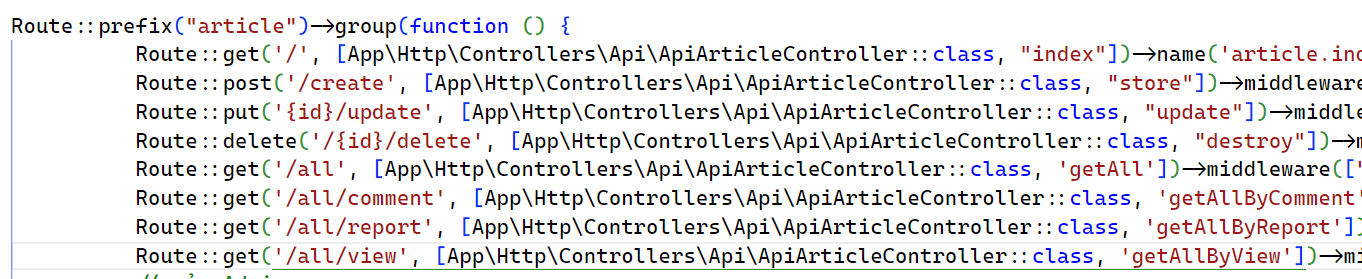
###### Ảnh code xử lý lọc theo lượt xem.

* Phía server:



###### Ảnh code xử lý chấp nhận cho phép đăng bài báo.

#### Phần API:



###### Ảnh route liên quan đến chức năng quản lý bài báo.

# TÀI LIỆU THAM KHẢO

Vedmant, “Laravel-feed-reader”, 10/07/2023. Địa chỉ: <https://github.com/vedmant/laravel-feed-reader>

Tymondesigns, “jwt-auth”, 01/08/2023. Địa chỉ: <https://github.com/tymondesigns/jwt-auth>

The Laravel Team, “Laravel Documentation”, 15/08/2023. Địa chỉ: <https://laravel.com/docs/9.x>

FaceBook, “React Documentation”, 14/06/2023. Địa chỉ: <https://react.dev/learn>

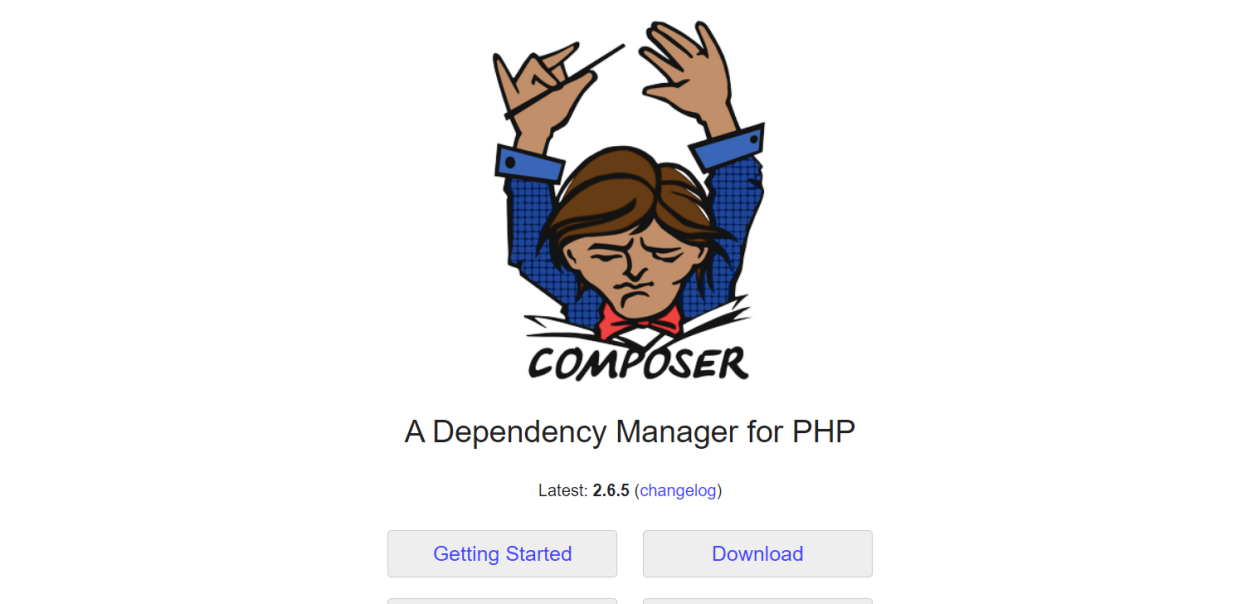
Adam Wathan, Jonathan Reinink, David Hamphill and Steve Schoger, “TailwindCSS Documentation”, 12/10/2023. Địa chỉ: <https://tailwindcss.com/docs/installation>

Creative-Tim, “Material Tailwind Documentation”, 08/10/2023. Địa chỉ: <https://www.material-tailwind.com/docs/react/installation>

# PHỤ LỤC

## Cài đặt Composer

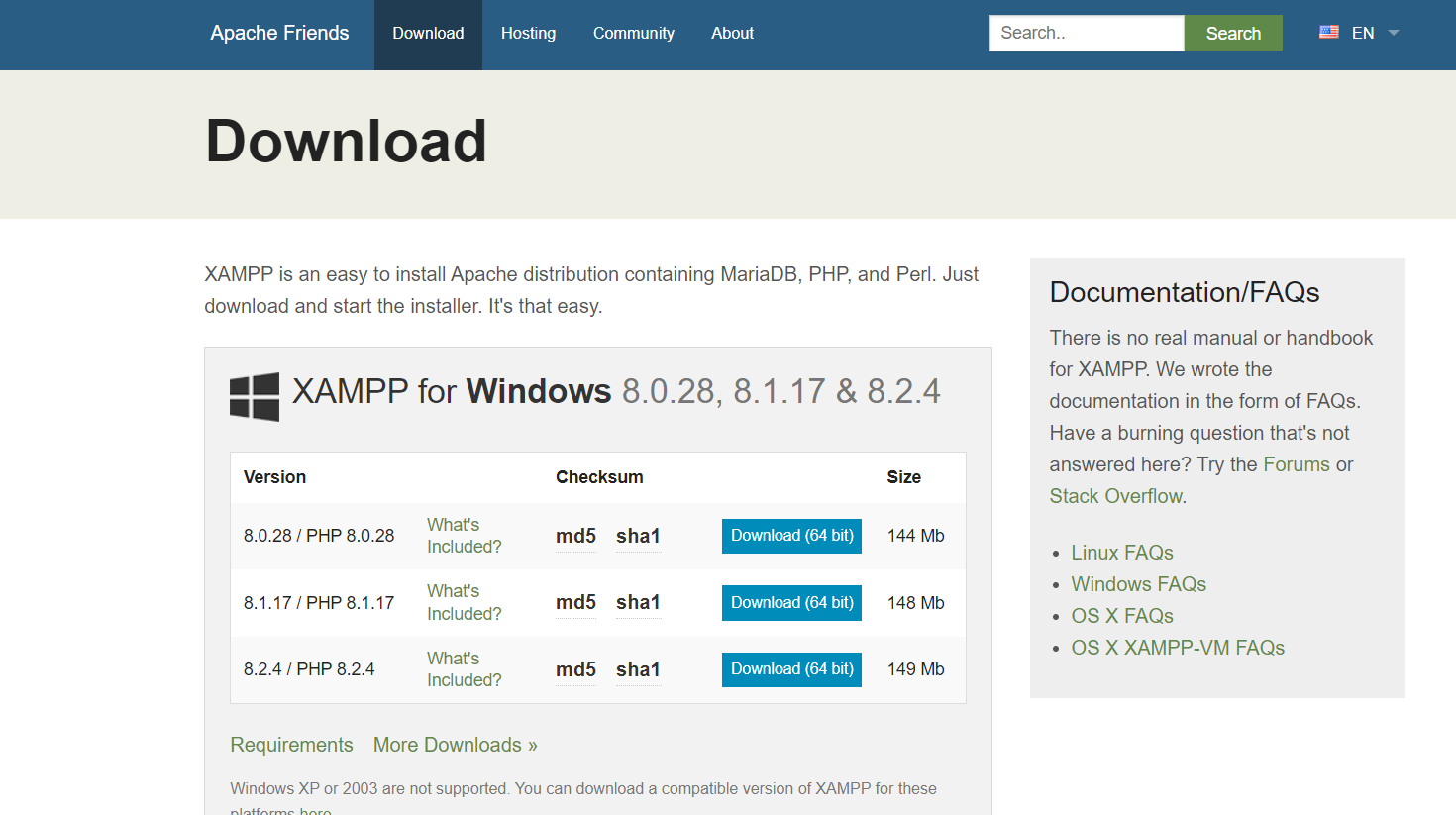
Tải file và cài đặt từ trang chủ Composer (getcomposer.org)



###### Ảnh hướng dẫn 1

## Cài đặt Xampp

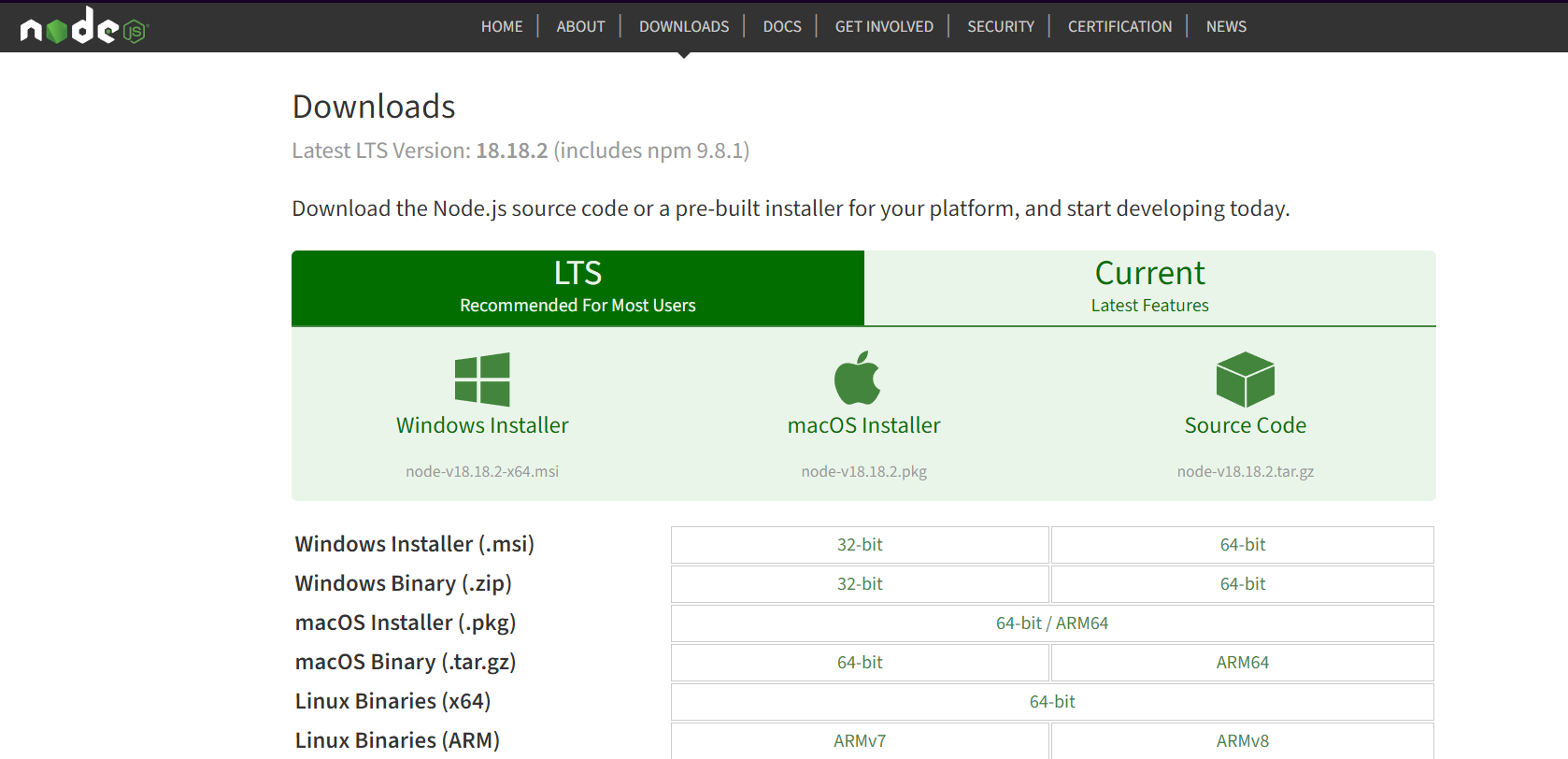
Tải file và cài đặt (nên dùng phiên bản 8.1.17) từ trang chủ Apache (apacherfriends.org)



###### Ảnh hướng dẫn 2

## Cài đặt NodeJs

Tải file và cài đặt (nên dùng phiên bản LTS) từ trang chủ NodeJs (nodejs.org)



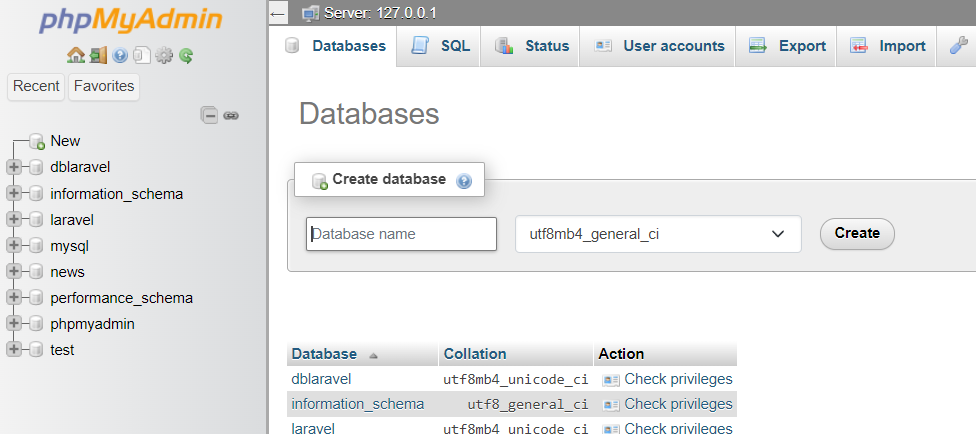
###### Ảnh hướng dẫn 3

## Hướng dẫn sử dụng App

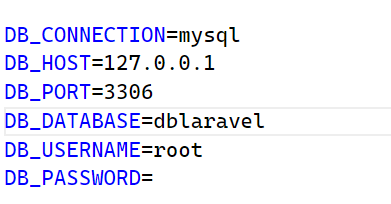
* Cài đặt các file cần thiết ở trên
* Chạy Xampp với quyền Admin (Run as administrator)
* Chọn Start ở Apache và MySQL
* Vào trình duyệt với domain localhost:80. Chọn phpMyAdmin  
  

###### Ảnh hướng dẫn 4

* Tạo một database mới với tên bất kì (nên đặt là dblaravel). Sau đó vào folder back-end và mở file .env sau đó đổi tên lạ.



###### Ảnh hướng dẫn 5



###### Ảnh hướng dẫn 6

* Sau khi tạo, import database đã có trong folder news-app
* Tiếp theo là về chạy app, vào folder news-app
* Mở Terminal của folder back-end
* Chạy câu lệnh ***composer i (hoặc composer install)***. Sau khi hoàn thành việc cài đặt, chạy lệnh ***php artisan serve*** để chạy server.
* Mở Terminal của folder front-end
* Chạy câu lệnh ***npm i (hoặc npm install)***. Sau khi hoàn thành việc cài đặt chạy lệnh ***npm start*** để chạy client.

## Hướng dẫn cách khắc phục lỗi

- Nếu phía client chạy nhưng lại xuất hiện Network Error. Hãy kiểm tra thật kỹ rằng MySQL trong xampp đã bật hay chưa.

- Nếu bạn thay đổi tên database với tên khác (dblaravel) và phía client chạy nhưng lại xuất hiện error 500. Hãy chạy lệnh sau:

**Php artisan config:cache**