

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LỜI CẢM ƠN

Kính gửi: thầy Đoàn Phước Miền

Trước hết, cho phép em được gửi lời cảm ơn chân thành nhất đến Đoàn Phước Miền đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình thực hiện đồ án chuyên ngành. Thầy đã dành thời gian quý báu của mình để chia sẻ những kiến thức bổ ích, giúp em hoàn thành đồ án một cách tốt nhất. Thầy cũng đã luôn động viên, khích lệ em trong suốt quá trình thực hiện đồ án, giúp em vượt qua những khó khăn và hoàn thành đồ án đúng tiến độ.

Em cũng xin gửi lời cảm ơn đến tất cả các bạn đã giúp đỡ và hỗ trợ em khi gặp những trở ngại và khó khăn trong quá trình thực hiện đồ án. Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên đồ án thực hiện không tránh khỏi những sai sót. Em rất mong nhận được những ý kiến góp ý từ thầy cô để em có thêm kinh nghiệm và kỹ năng để tiếp tục hoàn thiện đồ án hơn.

Trân trọng

Lê Trực Tín

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	2
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	3
2.1 HTML (Ngôn Ngữ Đánh Dấu Siêu Văn Bản)	3
2.1.1 Khái niệm cơ bản	3
2.1.2 Cấu trúc cơ bản của HTML.....	3
2.1.3 Ưu và Nhược điểm HTML.....	4
2.1.4 Vai trò của HTML trong website	5
2.2 Ngôn ngữ lập trình PHP	6
2.2.1 Khái niệm PHP.....	6
2.2.2 Tính năng cơ bản của PHP	6
2.2.3 Cấu trúc cơ bản của PHP.....	7
2.2.4 Ứng dụng của PHP	14
2.2.5 Ưu điểm và nhược điểm của ngôn ngữ PHP.....	14
2.3 Hệ quản trị cơ sở dữ liệu MySQL.....	16
2.3.1 Ngôn ngữ truy vấn SQL	16
2.3.2 Tầm quan trọng của SQL	16
2.3.3 Những thành phần của hệ thống SQL	17
2.3.4 Khái niệm về MySQL	18
2.3.5 Các ngôn ngữ truy vấn dữ liệu	19
2.3.6 Các câu lệnh SQL thường được sử dụng	20
2.4 Composer	23
2.4.1 Khái niệm Composer.....	23
2.4.2 Những lợi ích của Composer	23
2.5 Laravel	24

2.5.1	Laravel Framework	24
2.5.2	Ngôn ngữ lập trình phía back-end.....	24
2.5.3	Những tính năng nổi bật của Laravel.....	25
2.5.4	Cấu trúc thư mục của Laravel	27
2.5.5	Vòng đời của Laravel.....	28
2.5.6	Ưu điểm và nhược điểm của Laravel	29
2.6	Laragon	31
2.6.1	Sơ lược về WAMP, LAMP và MAMP	31
2.6.2	Laragon.....	32
2.6.3	Những tính năng nổi bật của Laragon.....	32
2.6.4	Những chức năng của Laragon	33
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU		34
3.1	Mô tả bài toán	34
3.2	Phân tích chức năng	35
3.2.1	Đối với người dùng	35
3.2.2	Đối với người quản trị.....	36
3.3	Thiết kế CSDL	37
3.3.1	Lược đồ quan hệ thực thể (ERD)	37
3.3.2	Mô tả các bảng CSDL.....	37
3.4	Mô hình USE CASE tổng quan hệ thống	42
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU		43
4.1	Người dùng	43
4.1.1	Trang đăng ký	43
4.1.2	Trang đăng nhập.....	43

4.1.3	Trang chủ.....	44
4.1.4	Trang danh mục.....	44
4.1.5	Tìm kiếm và xem bài viết.....	44
4.1.6	Tạo bài viết mới	45
4.1.7	Chỉnh sửa và ẩn bài viết	46
4.1.8	Chức năng bình luận	46
4.1.9	Chức năng tương tác với bài viết	47
4.1.10	Quản lý hồ sơ cá nhân	47
4.1.11	Chức năng Quên mật khẩu	48
4.2	Quản trị	49
4.2.1	Trang quản trị.....	49
4.2.2	Chức năng quản lý danh mục.....	49
4.2.3	Chức năng quản lý bài viết.....	50
4.2.4	Chức năng quản lý người dùng	50
4.2.5	Thống kê trang web.....	51
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		52
5.1	Kết luận	52
5.1.1	Về mặt tổng quan	52
5.1.2	Về mặt chức năng.....	52
5.2	Hướng phát triển	52
DANH MỤC TÀI LIỆU THAM KHẢO		53

DANH MỤC HÌNH ẢNH

Hình 2.2.1 Khái niệm PHP	6
Hình 2.2.5 Ưu điểm & nhược điểm PHP	15
Hình 2.3.1 Khái niệm SQL	16
Hình 2.3.4 Khái niệm MySQL	18
Hình 2.3.5 Các ngôn ngữ truy vấn dữ liệu	19
Hình 2.4.1 Khái niệm Composer	23
Hình 2.5.1 Laravel	24
Hình 2.6.1 Sơ lược LAMP, WAMP, MAMP	31
Hình 2.6.2 Laragon	32
Hình 3.3.1 Lược đồ quan hệ thực thể	37
Hình 3.4 Biểu đồ Use Case tổng quát	42
Hình 4.1.1 Giao diện đăng ký	43
Hình 4.1.2 Giao diện đăng nhập	43
Hình 4.1.3 Giao diện trang chủ	44
Hình 4.1.4 Giao diện danh mục	44
Hình 4.1.5 Giao diện tìm kiếm	44
Hình 4.1.6 Giao diện thêm bài đăng	45
Hình 4.1.7 Giao diện chỉnh sửa bài viết	46
Hình 4.1.8 Giao diện bình luận	46
Hình 4.1.9 Giao diện chức năng tương tác	47
Hình 4.1.10 Giao diện quản lý hồ sơ cá nhân	47
Hình 4.1.11 Chức năng Quên mật khẩu	48
Hình 4.1.12 Email reset mật khẩu	48
Hình 4.2.1 Giao diện trang quản trị	49

Hình 4.2.2 Giao diện trang Quản lý danh mục.....	49
Hình 4.2.3 Giao diện trang Quản lý bài viết	50
Hình 4.2.4 Giao diện trang Quản lý người dùng	50
Hình 4.2.5 Giao diện trang Thống kê trang web	51

DANH MỤC BẢNG BIỂU

Bảng 3.3.2.1 users.....	37
Bảng 3.3.2.2 roles	38
Bảng 3.3.2.3 posts	38
Bảng 3.3.2.4 categories	39
Bảng 3.3.2.5 post_details.....	39
Bảng 3.3.2.6 post_comments.....	40
Bảng 3.3.2.7 post_reacts.....	40
Bảng 3.3.2.8 sessions.....	41
Bảng 3.3.2.9 password_reset_tokens.....	41

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Với sự phát triển nhanh chóng của Internet, nhu cầu xây dựng một nền tảng diễn đàn chuyên nghiệp, dễ sử dụng và quản lý hiệu quả ngày càng được chú trọng. Dự án này tập trung vào việc thiết kế và phát triển một hệ thống diễn đàn trực tuyến có đầy đủ các chức năng cơ bản, đảm bảo tính tương tác giữa người dùng, đồng thời cung cấp công cụ quản lý mạnh mẽ cho quản trị viên.

Vấn đề đặt ra là làm thế nào để tạo ra một nền tảng diễn đàn đáp ứng các yêu cầu của người dùng như đăng ký tài khoản, đăng bài viết, bình luận, tương tác với các bài viết khác và quản lý hồ sơ cá nhân. Đồng thời, hệ thống cũng cần có cơ chế duyệt bài, phân quyền và quản lý danh mục nội dung dành cho quản trị viên, nhằm đảm bảo chất lượng bài viết và duy trì tính trật tự trong diễn đàn.

Hướng tiếp cận của dự án là xây dựng hệ thống theo mô hình client-server với cơ sở dữ liệu tập trung. Các công nghệ phổ biến như PHP (Laravel) cho backend và MySQL cho cơ sở dữ liệu đã được lựa chọn để đảm bảo sự ổn định và bảo mật. Giao diện người dùng được thiết kế bằng Blade Template, tối ưu cho cả thiết bị di động và máy tính để bàn, tạo sự thân thiện và dễ sử dụng.

Giải pháp được đưa ra là phát triển các module chức năng bao gồm:

- Chức năng dành cho người dùng: Đăng ký, đăng nhập, quản lý hồ sơ cá nhân, tìm kiếm, đăng bài, bình luận và tương tác với các bài viết.
- Chức năng dành cho quản trị viên: Quản lý bài viết, duyệt bài, ẩn bài vi phạm, quản lý người dùng và phân quyền theo vai trò.
- Cơ chế quản lý danh mục: Giúp tổ chức bài viết khoa học, dễ tìm kiếm và theo dõi.

Kết quả đạt được là một hệ thống diễn đàn trực tuyến hoàn chỉnh với đầy đủ các chức năng cơ bản và mở rộng, đáp ứng nhu cầu sử dụng thực tế. Hệ thống hoạt động ổn định, bảo mật tốt và có khả năng phát triển thêm các tính năng trong tương lai như bình luận lồng nhau, thông báo đẩy và tìm kiếm nâng cao. Dự án đã thành công trong việc tạo ra một nền tảng giao tiếp trực tuyến, nơi người dùng có thể chia sẻ kiến thức, tương tác và xây dựng cộng đồng tích cực.

MỞ ĐẦU

Với sự phát triển không ngừng của công nghệ, nhu cầu trao đổi và chia sẻ kiến thức trong lĩnh vực tin học ngày càng gia tăng. Tuy nhiên, nhiều diễn đàn hiện tại vẫn chưa đáp ứng được yêu cầu về bảo mật, hiệu suất, và trải nghiệm người dùng. Do đó, việc xây dựng một diễn đàn tin học sử dụng Laravel framework là cần thiết nhằm mang đến một nền tảng hiện đại, an toàn, và dễ sử dụng.

Mục đích nghiên cứu

Mục đích của đề tài là phát triển một diễn đàn tin học sử dụng Laravel framework, tập trung vào tối ưu về hiệu suất, bảo mật và trải nghiệm người dùng, phục vụ nhu cầu trao đổi kiến thức và gắn kết cộng đồng công nghệ.

Đối tượng nghiên cứu

Đối tượng nghiên cứu bao gồm các khía cạnh kỹ thuật của framework Laravel trong việc xây dựng ứng dụng web, cụ thể là các thành phần liên quan đến hệ thống quản lý người dùng, bảo mật, và quản lý nội dung. Đồng thời, nghiên cứu cũng tập trung vào hành vi người dùng trên các diễn đàn trực tuyến để tối ưu hóa trải nghiệm người dùng và nâng cao mức độ tương tác của cộng đồng.

Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài bao gồm:

- Phát triển và triển khai hệ thống diễn đàn tin học: Sử dụng các công cụ và công nghệ được hỗ trợ bởi Laravel, bao gồm hệ thống xác thực, định tuyến, quản lý cơ sở dữ liệu, và giao diện người dùng.
- Thử nghiệm và đánh giá: Kiểm thử hiệu suất và bảo mật của diễn đàn trong môi trường phát triển, đồng thời thu thập ý kiến từ người dùng để đánh giá tính khả dụng và hiệu quả.
- Giới hạn: Nghiên cứu chủ yếu tập trung vào việc phát triển chức năng cốt lõi của diễn đàn mà không đi sâu vào các công nghệ hoặc framework khác ngoài Laravel.

CHƯƠNG 1: TỔNG QUAN

Vấn đề đặt ra trong bối cảnh này là sự thiếu hụt các nền tảng diễn đàn tin học hiện đại, an toàn, và dễ sử dụng để tạo điều kiện thuận lợi cho việc trao đổi kiến thức và kinh nghiệm. Các diễn đàn hiện có có thể gặp phải các hạn chế về mặt giao diện người dùng, tốc độ truy cập, khả năng mở rộng, hoặc các yếu tố bảo mật.

Việc nghiên cứu và phát triển một diễn đàn tin học bằng framework Laravel nhằm giải quyết những vấn đề này, tập trung vào việc cung cấp một nền tảng:

- Tối ưu hóa hiệu suất và bảo mật: Sử dụng các tính năng mạnh mẽ và hiệu quả được tích hợp trong Laravel để bảo vệ dữ liệu người dùng và quản lý thông tin một cách an toàn.
- Trải nghiệm người dùng: Cải thiện giao diện người dùng (UI/UX) để đảm bảo dễ sử dụng và thu hút sự tham gia của cộng đồng.
- Quản lý và tổ chức nội dung: Xây dựng các chức năng quản lý bài viết, phân loại chủ đề, và hệ thống phản hồi hiệu quả.

Sản phẩm dự kiến sẽ cung cấp cho người dùng một diễn đàn hiện đại, nơi các cá nhân và nhóm có thể dễ dàng tạo chủ đề thảo luận, đặt câu hỏi, và chia sẻ kiến thức một cách nhanh chóng, đồng thời đảm bảo rằng nền tảng đáp ứng tốt các yêu cầu kỹ thuật và bảo mật cao.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 HTML (Ngôn Ngữ Đánh Dấu Siêu Văn Bản)

2.1.1 Khái niệm cơ bản

HTML (Hypertext Markup Language) là một ngôn ngữ đánh dấu, không phải là ngôn ngữ lập trình. Nó được sử dụng để tạo và trình bày các trang web. Đây là một phần quan trọng của cơ sở hạ tầng của World Wide Web (WWW). HTML rất cần thiết trong việc xây dựng một trang web, vì những gì chúng ta có thể nhìn được và tương tác được trên web đều được tạo nên nhờ HTML.

2.1.2 Cấu trúc cơ bản của HTML

HTML sử dụng các thẻ để đánh dấu các phần tử trong trang web. Mỗi thẻ bao gồm một tên có ý nghĩa duy nhất. Trang HTML thường bắt đầu với thẻ **<!DOCTYPE html>** để xác định phiên bản HTML sử dụng.

Thẻ **<html>** là thẻ gốc của mọi trang HTML.

Thẻ **<head>** chứa thông tin về trang web như tiêu đề, tập tin CSS và các meta tags.

Thẻ **<body>** chứa nội dung chính của trang web.

```
#Ví dụ về cú pháp cơ bản của HTML
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<!--Các thẻ khác có thể được thêm vào đây-->
</head>
<body>
<!-- Nội dung trang web sẽ được đặt ở đây -->
</body>
</html>
```

Các thẻ cơ bản thường được sử dụng

Các thẻ định dạng văn bản: Các thẻ như **<p>** (đoạn văn bản), **<h1>** đến **<h6>** (tiêu đề), **** (đậm), **** (in nghiêng), và **
** (ngắt dòng) được sử dụng để định dạng văn bản.

Thẻ hình ảnh và đa phương tiện: Thẻ **<a>** được sử dụng để tạo liên kết. Thuộc tính **href** của nó chỉ định đường dẫn đến trang hoặc tài nguyên khác.

Biểu mẫu và nhập liệu: Thẻ **<form>** được sử dụng để tạo biểu mẫu. Các phần tử như **<input>**, **<select>**, và **<textarea>** được sử dụng để tạo các trường nhập liệu.

Ngoài ra còn có những thẻ phổ biến khác chẳng hạn như thẻ **<div>** trong HTML là một thẻ phổ biến và quan trọng, được sử dụng để nhóm và tổ chức các phần tử HTML trong một khối. Tên "div" là viết tắt của "division" (phân khúc), và thường được sử dụng để tạo ra các khu vực hay khối trong trang web để dễ dàng quản lý và kiểm soát bố cục.

2.1.3 Ưu và Nhược điểm HTML

Ưu điểm:

- Được sử dụng rộng rãi, có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng lớn.
- Học đơn giản và dễ hiểu.
- Mã nguồn mở và hoàn toàn miễn phí.
- Markup gọn gàng và đồng nhất.
- Tiêu chuẩn thế giới được vận hành bởi World Wide Web Consortium (W3C).
- Dễ dàng tích hợp với các ngôn ngữ backend như PHP, Python...

Nhược điểm:

- Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động như update hay realtime thời gian thực, chúng ta cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.

- Một số trình duyệt chậm hỗ trợ tính năng mới.

2.1.4 Vai trò của HTML trong website

Với những ưu và nhược điểm trên, điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website. Một website chuẩn sẽ được hình thành bởi:

- **HTML** – Xây dựng cấu trúc và định dạng các siêu văn bản.
- **CSS** – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền,....
- **Javascript** – Tạo ra các sự kiện tương tác với hành động của người dùng (ví dụ như là chat, update nội dung, hiệu ứng slide).
- **PHP** – Ngôn ngữ lập trình để xử lý và trao đổi dữ liệu giữa máy chủ đến trình duyệt.
- **MySQL** – Hệ quản trị cơ sở dữ liệu truy vấn có cấu trúc.

Nếu website là một cơ thể hoàn chỉnh thì HTML chính là bộ xương của cơ thể đó. Dù website thuộc thể loại nào, giao tiếp với ngôn ngữ lập trình nào để xử lý dữ liệu thì vẫn phải cần HTML để hiển thị nội dung ra cho người dùng xem.

2.2 Ngôn ngữ lập trình PHP

2.2.1 Khái niệm PHP

PHP là viết tắt của ‘PHP: Hypertext Preprocessor’, ban đầu có nghĩa là ‘Personal Home Page’. Từ viết tắt này đã thay đổi theo thời gian kể từ khi ngôn ngữ được ra mắt vào năm 1994 để phản ánh chính xác hơn chức năng của nó.



Hình 2.2.1 Khái niệm PHP

PHP là một ngôn ngữ lập trình mã nguồn mở chạy trên máy chủ, được sử dụng để tạo ra các trang web, ứng dụng, hệ thống quản lý quan hệ khách hàng và nhiều hơn nữa. Đây là một ngôn ngữ đa dụng được sử dụng rộng rãi và có thể nhúng vào HTML. Nhờ khả năng tích hợp với HTML, PHP vẫn được các nhà phát triển ưa chuộng vì giúp đơn giản hóa mã HTML. [1]

Một cách giải thích đơn giản hơn thì HTML dùng để tạo nên giao diện để sử dụng và PHP sẽ tương tác với dữ liệu.

2.2.2 Tính năng cơ bản của PHP

Tích hợp dễ dàng: PHP được thiết kế để giúp các nhà phát triển dễ dàng nhúng mã PHP vào HTML. Điều này giúp tạo ra các trang web động mà không cần cấu trúc quá phức tạp. Ngoài ra, PHP cũng có khả năng tích hợp mạnh mẽ với các cơ sở dữ liệu như MySQL, PostgreSQL, SQLite, và nhiều hệ quản trị cơ sở dữ liệu khác.

Dễ học: PHP được biết đến với cú pháp đơn giản và rõ ràng, làm cho nó trở thành một ngôn ngữ lý tưởng cho người mới học lập trình web. Ví dụ như các biến trong PHP không cần khai báo kiểu trước, và cú pháp của nó cũng rất dễ hiểu, nhất là

đối với những ai đã quen với ngôn ngữ lập trình như C hoặc JavaScript. Hơn nữa, với tài liệu phong phú và cộng đồng người dùng lớn, người học có thể dễ dàng tìm thấy các tài nguyên và lời giải đáp khi gặp khó khăn.

2.2.3 Cấu trúc cơ bản của PHP

2.2.3.1 PHP Tags

Mã PHP được bắt đầu bằng `<?php` và kết thúc bằng `?>`. Nó tương tự như tất cả các thẻ HTML vì tất cả chúng đều bắt đầu bằng biểu tượng (`<`) và kết thúc bằng biểu tượng (`>`). Các ký hiệu `<?php` và `?>` được gọi là thẻ PHP.

```
<html>
<body>
<?php
echo "Nội dung động từ PHP!";
?>
</body>
</html>
```

Các thẻ PHP giúp cho máy chủ web hiểu nơi mã PHP bắt đầu và kết thúc, văn bản giữa các thẻ này được hiểu là PHP. Còn lại bất kỳ văn bản nào bên ngoài các thẻ này đều được coi là HTML bình thường.

2.2.3.2 PHP Keywords

Trong PHP, keywords (từ khóa) là những từ được định nghĩa sẵn và có ý nghĩa đặc biệt trong ngôn ngữ lập trình. Chúng không thể được sử dụng làm tên biến, hàm, hoặc lớp. Dưới đây là danh sách các từ khóa phổ biến trong PHP cùng với mô tả ngắn gọn về chức năng của chúng:

abstract: Được sử dụng để khai báo một lớp hoặc phương thức trừu tượng.	function: Khai báo một hàm.
and: Tương đương với toán tử logic <code>&&</code> .	global: Để sử dụng biến toàn cục trong một hàm.
array: Được sử dụng để định nghĩa mảng.	if: Bắt đầu một cấu trúc điều kiện.
as: Sử dụng trong vòng lặp <code>foreach</code> để chỉ định tên biến cho mỗi phần tử.	for: Bắt đầu một vòng lặp <code>for</code> .
	implements: Sử dụng để thực hiện một giao diện.
	include, require: Để chèn một file PHP khác vào

break: Dùng để thoát khỏi vòng lặp hoặc switch. const: Khai báo một hằng số trong lớp. class: Khai báo một lớp trong PHP. echo: Được sử dụng để in ra một chuỗi. else: Sử dụng trong câu lệnh điều kiện if để chỉ định mã sẽ thực thi nếu điều kiện không đúng. elseif: Kết hợp với if để kiểm tra nhiều điều kiện. empty: Kiểm tra xem một biến có rỗng hay không. exit: Kết thúc một kịch bản PHP. extends: Sử dụng để kế thừa từ một lớp khác.	file hiện tại. interface: Khai báo một giao diện. isset: Kiểm tra xem một biến có được khai báo và khác NULL hay không. list: Dùng để gán giá trị của mảng vào nhiều biến. new: Tạo một thể hiện mới của lớp. private, protected, public: Định nghĩa mức độ truy cập cho các thuộc tính và phương thức trong lớp. unset: Xóa một biến.
--	--

2.2.3.3 Biến trong PHP

Biến là các đối tượng được sử dụng để lưu trữ dữ liệu. Biến có thể chứa các loại dữ liệu khác nhau, như số nguyên, số thực, chuỗi, mảng, đối tượng, và nhiều hơn nữa.

Khai báo Biến

Biến trong PHP được khai báo bằng cách sử dụng ký tự “\$” trước tên biến. Tên biến có thể chứa chữ cái, số và dấu gạch dưới, nhưng không thể bắt đầu bằng số.

```
#Ví dụ:  
$tenBien = giaTri;  
//Exp  
$ten = "Nguyễn Văn A";  
$tuoi = 25;
```

Quy tắc đặt tên biến

- Tên biến phải bắt đầu bằng ký tự chữ cái hoặc dấu gạch dưới (_).
- Tên biến không được bắt đầu bằng số.
- Tên biến phân biệt chữ hoa và chữ thường (ví dụ: \$Ten và \$ten là hai biến khác nhau).
- Không sử dụng từ khóa hoặc ký tự đặc biệt (ngoại trừ dấu gạch dưới).

Biến toàn cục và biến cục bộ

Biến cục bộ là biến được khai báo bên trong một hàm. Nó chỉ có thể được truy cập và sử dụng trong hàm đó. Khi hàm kết thúc, biến cục bộ sẽ không còn tồn tại và không thể truy cập từ bên ngoài.

```
function tinhTong($a, $b) {  
    $tong = $a + $b; // biến cục bộ  
    return $tong;  
}  
echo tinhTong(5, 10); // In ra: 15  
// echo $tong; // Lỗi: biến cục bộ không thể truy cập bên ngoài hàm
```

Biến toàn cục là biến được khai báo bên ngoài bất kỳ hàm nào. Nó có thể được truy cập và sử dụng từ bất kỳ đâu trong mã, bao gồm cả bên trong các hàm. Để sử dụng biến toàn cục trong một hàm, chúng ta cần khai báo từ khóa **global**.

```
$bienToanCuc = "Hello";  
  
function ham() {  
    global $bienToanCuc; // Sử dụng biến toàn cục  
    echo $bienToanCuc;  
}  
ham(); // In ra: Hello
```

Nối chuỗi

Trong PHP chúng ta có thể nối (kết hợp) hai hoặc nhiều chuỗi thành một chuỗi duy nhất bằng cách sử dụng dấu "." giữa các chuỗi cần nối.

```
<?php
$firstName = "Ho";
$lastName = "Ten";
// Nối chuỗi
$fullName = $firstName . " " . $lastName;
echo $fullName; // Kết quả: Ho Ten
?>
```

2.2.3.4 Kiểu dữ liệu

PHP là một ngôn ngữ dynamically typed (kiểu dữ liệu động), có nghĩa là chúng ta không cần chỉ định kiểu dữ liệu khi khai báo biến. PHP sẽ tự động xác định kiểu dữ liệu dựa trên giá trị mà biến nhận.

Chuỗi (String): Dùng để lưu trữ chuỗi ký tự.

```
$ten = "Nguyễn Văn A";
```

Số nguyên (Integer): Dùng để lưu trữ số nguyên.

```
$tuoi = 25;
```

Số thực (Float): Dùng để lưu trữ số thực (số thập phân).

```
$diem = 8.5;
```

Boolean: Dùng để lưu trữ giá trị đúng hoặc sai.

```
$duyet = true;
```

Mảng (Array): Dùng để lưu trữ một tập hợp các giá trị.

```
$mang = array(1, 2, 3);
```

Đối tượng (Object): Dùng để lưu trữ một thể hiện của một lớp.

```
class Người {
    public $ten;
    public $tuoi;
}
$nguoi = new Người();
$nguoi->ten = "Nguyễn Văn A";
```

2.2.3.5 Câu lệnh điều kiện

Câu lệnh điều kiện được sử dụng để thực hiện các thao tác dựa trên một điều kiện nào đó. PHP cung cấp nhiều loại câu lệnh điều kiện để kiểm tra các điều kiện và điều khiển luồng chương trình. Dưới đây là các loại câu lệnh điều kiện phổ biến trong PHP:

Câu lệnh if

Câu lệnh if kiểm tra một điều kiện và chỉ thực thi khối lệnh nếu điều kiện đó đúng (true).

```
//Cú pháp
if (điều_kiện) {
    // Khối lệnh được thực thi nếu điều kiện đúng
}
//ví dụ
$tuoi = 20;
if ($tuoi >= 18) {
    echo "Bạn đã đủ tuổi trưởng thành.";
}
```

Câu lệnh if...else

Câu lệnh if...else cho phép chúng ta thực thi một khối lệnh nếu điều kiện đúng, và một khối lệnh khác nếu điều kiện sai.

```
//Cú pháp
if (điều_kiện) {
    // Khối lệnh nếu điều kiện đúng
} else {
    // Khối lệnh nếu điều kiện sai
}
//ví dụ
$diem = 7;
if ($diem >= 5) {
    echo "Bạn đã đậu.";
} else {
    echo "Bạn đã rớt.";
}
```

Câu Lệnh if...elseif...else

Câu lệnh này cho phép kiểm tra nhiều điều kiện khác nhau. Nếu điều kiện đầu tiên sai, nó sẽ kiểm tra các điều kiện elseif tiếp theo, và nếu tất cả đều sai, nó sẽ thực hiện khối lệnh else.

```
//cú pháp
if (điều_kiện_1) {
    // Khối lệnh nếu điều kiện_1 đúng
} elseif (điều_kiện_2) {
    // Khối lệnh nếu điều kiện_2 đúng
} else {
    // Khối lệnh nếu tất cả điều kiện đều sai
}

//ví dụ
$diem = 8;
if ($diem >= 9) {
    echo "Xuất sắc";
} elseif ($diem >= 7) {
    echo "Khá";
} else {
    echo "Trung bình";
}
```

Câu Lệnh switch

Câu lệnh switch được sử dụng để so sánh một biến hoặc biểu thức với nhiều giá trị khác nhau và thực thi khối lệnh tương ứng.

```
//cú pháp                                //ví dụ
switch (biểu_thức) {                      $thang = 5;
    case giá_trị_1:                        switch ($thang) {
        // Khối lệnh nếu biểu_thức ==    case 1:
        giá_trị_1                         echo "Tháng Một";
        break;                            break;
    case giá_trị_2:                       case 2:
        // Khối lệnh nếu biểu_thức ==    echo "Tháng Hai";
        giá_trị_2                         break;
        break;                            case 5:
    default:                              echo "Tháng Năm";
        // Khối lệnh nếu không có giá    break;
        trị nào khớp                     default:
    }                                     echo  "Không    tìm    thấy
                                         tháng.";
                                         }
}
```

2.2.4 Ứng dụng của PHP

Ngôn ngữ lập trình PHP thường tập trung vào việc thiết lập chương trình cho máy chủ, tạo các cơ sở dữ liệu, xây dựng nội dung website, nhận dữ liệu cookie. Chưa hết, chúng ta còn có thể thực hiện được nhiều thao tác, công năng khác khi sử dụng ngôn ngữ này.

Một số ứng dụng phổ biến của PHP [1]

- *Thiết lập chương trình cho hệ thống máy chủ:* Đây là một ứng dụng chủ yếu nhất của PHP. Các PHP Developer sẽ phải thực hiện các thao tác như phân tích ngôn ngữ lập trình PHP, xây dựng máy chủ web và trình duyệt web.
- *Tạo các dòng tập lệnh:* Các lập trình viên sẽ tạo ra một dòng tập lệnh để vận hành chương trình PHP mà không cần đến máy chủ. Kiểu lập trình này được sử dụng trên các hệ điều hành phổ biến như Linux hay Windows.
- *Xây dựng các ứng dụng làm việc:* Chúng ta có thể ứng dụng những điểm mạnh vốn có của PHP để xây dựng ứng dụng phần mềm. Các lập trình viên thường dùng PHP – GTK làm nền tảng xây dựng phần mềm vì đây là nhánh mở rộng của ngôn ngữ lập trình này và không có sẵn trong các bản phân phối chính thức hiện nay.
- *Hỗ trợ cho mọi loại cơ sở dữ liệu khác nhau:* Khi một website có hỗ trợ cơ sở dữ liệu tốt sẽ giúp ích cho việc vận hành, sao lưu và đặc biệt là backup dữ liệu đề phòng trường hợp xảy ra an ninh mạng.

2.2.5 Ưu điểm và nhược điểm của ngôn ngữ PHP

Ưu điểm:

PHP được sử dụng phổ biến bởi nhiều lợi ích mà nó mang lại. Dưới đây là một số ưu điểm cơ bản của ngôn ngữ PHP:

- *Mã nguồn mở và miễn phí:* PHP sử dụng miễn phí nên giúp tiết kiệm đáng kể ngân sách dự án. Việc cài đặt và sử dụng ngôn ngữ này cũng rất dễ dàng, chúng ta chỉ cần học chăm chỉ trong 3 – 6 tháng là đã có thể sử dụng thuần thục.

- Tính linh hoạt: PHP là một ngôn ngữ đa nền tảng, có thể hoạt động trên bất kỳ hệ điều hành nào (Windows, Linux, macOS,...). Hơn nữa, PHP còn có thể kết hợp với nhiều ngôn ngữ lập trình khác để xây dựng các tính năng công nghệ một cách hiệu quả nhất.
- Hệ thống thư viện phong phú, tính cộng đồng cao: Do sự phổ biến của ngôn ngữ PHP nên việc tìm các thư viện code hay hàm liên quan đến PHP sẽ cực kỳ đơn giản. Chưa kể, chúng ta sẽ nhận được sự trợ giúp từ các diễn đàn, đội nhóm chuyên sâu của PHP giúp việc học tập hay làm việc trở nên dễ dàng.
- Cơ sở dữ liệu đa dạng: PHP cho phép kết nối với hầu hết các cơ sở dữ liệu khác như MySQL, SQLite, PostgreSQL, MS-SQL,...



Hình 2.2.5 Ưu điểm & nhược điểm PHP

Nhược điểm:

Mặc dù sở hữu nhiều lợi ích nhưng ngôn ngữ PHP vẫn có một số hạn chế nhất định, trong đó vấn đề bảo mật được nhiều người quan tâm nhất. Bởi bản chất của PHP có mã nguồn mở nên các lỗ hổng của mã nguồn sẽ bị công khai ngay sau khi chúng được tìm thấy. Và các lỗ hổng này có thể bị khai thác cho các mục đích xấu trước khi chúng ta kịp sửa chữa.

Bên cạnh đó, ngôn ngữ lập trình PHP chỉ hoạt động được trên các website và giao diện không được gọn gàng, đẹp mắt. Độ bảo mật và hiệu suất của ngôn ngữ này cũng chưa tốt.

2.3 Hệ quản trị cơ sở dữ liệu MySQL

2.3.1 Ngôn ngữ truy vấn SQL

Từ SQL là viết tắt của cụm Structured Query Language, có nghĩa là loại ngôn ngữ truy vấn có cấu trúc. [2] Đây là một loại ngôn ngữ máy tính vô cùng đặc thù, giúp hỗ trợ các thao tác lưu trữ, truy xuất dữ liệu cùng một cơ sở dữ liệu quan hệ một cách vô cùng nhanh chóng.



Hình 2.3.1 Khái niệm SQL

Cơ sở dữ liệu quan hệ lưu trữ thông tin dưới dạng bảng có các hàng và cột đại diện cho những thuộc tính dữ liệu và nhiều mối quan hệ khác nhau giữa các giá trị dữ liệu. Chúng ta có thể sử dụng các câu lệnh SQL để lưu trữ, cập nhật, loại bỏ, tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu. Chúng ta cũng có thể sử dụng SQL để duy trì và tối ưu hóa hiệu suất cơ sở dữ liệu.

2.3.2 Tầm quan trọng của SQL

Ngôn ngữ truy vấn có cấu trúc (SQL) là một ngôn ngữ truy vấn phổ biến thường được sử dụng trong tất cả các loại ứng dụng. Các nhà phân tích và phát triển dữ liệu tìm hiểu và sử dụng SQL do ngôn ngữ này tích hợp hiệu quả với nhiều ngôn ngữ lập trình khác nhau. Ví dụ: họ có thể nhúng truy vấn SQL với ngôn ngữ lập trình Java để xây dựng ứng dụng xử lý dữ liệu hiệu năng cao có các hệ thống cơ sở dữ liệu SQL chính như Oracle hoặc MS SQL Server. SQL cũng khá dễ tìm hiểu do nó sử dụng những từ khóa tiếng Anh phổ biến trong các câu lệnh. [3]

2.3.3 Những thành phần của hệ thống SQL

Hệ thống quản lý cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để lưu trữ và quản lý dữ liệu. Hệ thống này lưu trữ nhiều bảng cơ sở dữ liệu có liên quan đến nhau. MS SQL Server, MySQL hoặc MS Access là những ví dụ về hệ thống quản lý cơ sở dữ liệu quan hệ. Một hệ thống như vậy có các thành phần sau.

Bảng SQL

Bảng SQL là phần tử cơ bản của cơ sở dữ liệu quan hệ. Bảng cơ sở dữ liệu SQL bao gồm các hàng và cột. Nhà phát triển dữ liệu tạo ra các mối quan hệ giữa nhiều bảng cơ sở dữ liệu để tối ưu hóa không gian lưu trữ dữ liệu.

Giả sử tạo một bảng SQL cho các sản phẩm trong cửa hàng:

ID sản phẩm	Tên sản phẩm	ID màu
001	Bàn phím	Màu 1
002	Chuột	Màu 2

Sau đó liên kết bảng sản phẩm với bảng màu có *ID màu*:

ID màu	Tên màu
Màu 1	Màu đỏ
Màu 2	Màu xanh lục

Câu lệnh SQL

Câu lệnh SQL, hoặc truy vấn SQL, là các lệnh hướng dẫn hợp lệ mà hệ thống quản lý cơ sở dữ liệu quan hệ hiểu được. Nhà phát triển phần mềm xây dựng các câu lệnh SQL bằng nhiều phần tử ngôn ngữ SQL khác nhau. Phần tử ngôn ngữ SQL là các thành phần như mã định danh, biến và điều kiện tìm kiếm tạo thành một câu lệnh SQL đúng.

Giả sử câu lệnh SQL sau sử dụng lệnh SQL INSERT để lưu trữ *Sản phẩm Máy tính có ID màu 2* vào bảng có tên *Sanpham* với các cột tên *TenSP* (*Tên sản phẩm*) và *IDcolor* (*ID màu*):

```
INSERT INTO Sanpham(TenSP, IDcolor)
VALUES('Máy tính','Màu 2');
```

2.3.4 Khái niệm về MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ nguồn mở do Oracle cung cấp. Nhà phát triển có thể tải xuống và sử dụng MySQL mà không phải trả phí cấp phép. Họ có thể cài đặt MySQL trên nhiều hệ điều hành hoặc máy chủ đám mây khác nhau. MySQL là một hệ thống cơ sở dữ liệu phổ biến cho các ứng dụng web.



Hình 2.3.4 Khái niệm MySQL

Sự khác nhau giữa SQL và MySQL

Ngôn ngữ truy vấn có cấu trúc (SQL) là một ngôn ngữ tiêu chuẩn để tạo và thao tác với cơ sở dữ liệu. MySQL là một chương trình cơ sở dữ liệu quan hệ sử dụng các truy vấn SQL. Các lệnh SQL được quy định theo tiêu chuẩn quốc tế, còn phần mềm MySQL thì được nâng cấp và cải tiến liên tục.

2.3.5 Các ngôn ngữ truy vấn dữ liệu

Lệnh ngôn ngữ truy vấn có cấu trúc (SQL) là những từ khóa hoặc câu lệnh SQL cụ thể được các nhà phát triển sử dụng để thao tác với dữ liệu được lưu trữ trong cơ sở dữ liệu quan hệ. Chúng ta có thể phân loại các lệnh SQL như sau.

DDL	DML	DQL	DCL	TCL
Create Alter Drop Truncate	Insert Update Delete	Select	Grant Revoke	Commit Rollback Savepoint

Hình 2.3.5 Các ngôn ngữ truy vấn dữ liệu

Ngôn ngữ định nghĩa dữ liệu

Ngôn ngữ định nghĩa dữ liệu (DDL) là các lệnh SQL thiết kế cấu trúc cơ sở dữ liệu. Nhà phát triển sử dụng DDL để tạo và điều chỉnh các đối tượng cơ sở dữ liệu dựa trên các yêu cầu nghiệp vụ. Ví dụ: sử dụng lệnh CREATE để tạo các đối tượng cơ sở dữ liệu như bảng, chế độ xem và chỉ mục.

Ngôn ngữ thao tác dữ liệu

Các câu lệnh ngôn ngữ thao tác dữ liệu (DML) viết thông tin mới hoặc điều chỉnh các bản ghi hiện có trong cơ sở dữ liệu quan hệ. Ví dụ: một ứng dụng sử dụng lệnh INSERT để lưu trữ một bản ghi mới trong cơ sở dữ liệu.

Ngôn ngữ truy vấn dữ liệu

Ngôn ngữ truy vấn dữ liệu (DQL) bao gồm các lệnh hướng dẫn để truy xuất dữ liệu được lưu trữ trong cơ sở dữ liệu quan hệ. Các ứng dụng phần mềm sử dụng lệnh SELECT để lọc và trả về kết quả cụ thể từ một bảng SQL.

Ngôn ngữ kiểm soát dữ liệu

Quản trị viên cơ sở dữ liệu sử dụng ngôn ngữ kiểm soát dữ liệu (DCL) để quản lý hoặc cấp quyền truy cập cơ sở dữ liệu cho người dùng khác. Ví dụ: họ có thể sử dụng lệnh GRANT để cho phép các ứng dụng nhất định thao tác với một hoặc nhiều bảng.

Ngôn ngữ kiểm soát giao dịch

Công cụ quan hệ sử dụng ngôn ngữ kiểm soát giao dịch (TCL) để tự động thực hiện các thay đổi đối với cơ sở dữ liệu. Ví dụ: cơ sở dữ liệu sử dụng lệnh ROLLBACK để hoàn tác một giao dịch bị lỗi.

2.3.6 Các câu lệnh SQL thường được sử dụng

2.3.6.1 CREATE

Lệnh Create thường dùng để tạo mới một CSDL hay một bảng cụ thể.

```
## Cú pháp  
#Tạo CSDL mới  
CREATE DATABASE TenCosodulieu;  
#Tạo bảng mới  
CREATE TABLE tenBang (  
    cot1 kieuDulieu,  
    cot2 kieuDulieu,  
    ...  
);
```

2.3.6.2 SELECT

Lệnh Select dùng để lấy dữ liệu từ bảng

```
## Cú pháp  
#Lấy toàn bộ dữ liệu của bảng bằng kí tự *  
SELECT * FROM tenBang;  
#Lấy dữ liệu cụ thể của cột cot1, cot2  
SELECT cot1, cot2, ...  
FROM tenBang;
```

2.3.6.3 WHERE

Lệnh Where dùng để *lọc điều kiện* khi thực hiện truy vấn dữ liệu như Select, Update, Delete,...

```
## Cú pháp  
SELECT * FROM tenBang WHERE dieuKien;
```

2.3.6.4 INSERT INTO

Lệnh Insert dùng để chèn dữ liệu vào một bảng.

Cú pháp

#Chèn dữ liệu giaTri1, giaTri2 vào bảng tenBang với trường dữ liệu cot1,cot2

```
INSERT INTO tenBang (cot1, cot2) VALUES (giaTri1, giaTri2);
```

2.3.6.5 UPDATE

Lệnh Update dùng để cập nhật dữ liệu trong một bảng.

Cú pháp

#Cập nhật dữ liệu của cot1 là giaTri1, cot2 là giaTri2 với điềuKien trong bảng tenBang

```
UPDATE tenBang
```

```
SET cot1 = giaTri1, cot2 = giaTri2, ...
```

```
WHERE dieuKien;
```

Chú ý: Hãy cẩn thận khi cập nhật các records trong bảng. Câu lệnh **Where** chỉ định record(s) nào thì record(s) đó sẽ được cập nhật. Nếu bỏ qua câu lệnh **Where** thì sẽ cập nhật tất cả các records trong bảng

2.3.6.6 DELETE

Lệnh Delete dùng để xóa bảng ghi trong một bảng.

Cú pháp

#Xóa dữ liệu với điềuKien trong bảng tenBang

```
DELETE FROM tenBang WHERE dieuKien;
```

Chú ý: Tương tự câu lệnh **Update**, hãy cẩn thận khi xóa các records trong bảng. Câu lệnh **Delete** chỉ định record(s) nào thì record(s) đó sẽ được xóa. Nếu bỏ qua câu lệnh **Where** thì sẽ xóa tất cả các records trong bảng.

2.3.6.7 JOIN

Lệnh Join dùng để kết hợp dữ liệu từ hai bảng.

Giả sử ta có 2 bảng như sau:

Bảng **sinhvien**

maSV	tenSV	maLop
1	AA	101
2	BB	102
3	CC	NULL

Bảng **lophoc**

maLop	tenLop
101	Toán
102	Ngoại ngữ
103	Ngữ Văn

INNER JOIN

Inner Join kết hợp các bản ghi có điều kiện khớp nhau trong cả hai bảng.

Ví dụ: Lấy tên của sinh viên và tên lớp học của họ.

```
SELECT sinhvien.tenSV, lophoc.tenLop  
FROM sinhvien  
INNER JOIN lophoc ON sinhvien.maLop = lophoc.maLop;
```

Kết quả: Chỉ những sinh viên có maLop khớp với maLop trong bảng lophoc sẽ được trả về.

tenSV	tenLop
AA	Toán
BB	Ngoại ngữ

2.4 Composer

2.4.1 Khái niệm Composer



Hình 2.4.1 Khái niệm Composer

Composer, lần đầu ra mắt vào ngày 01/03/2012, là một trình quản lý phụ thuộc trong hệ sinh thái PHP. Đây là công cụ quản lý các gói thư viện mà các dự án PHP hoặc Laravel của chúng ta sử dụng. Composer giúp quản lý sự phụ thuộc của các tài nguyên trong một dự án, cho phép chúng ta khai báo những thư viện cần thiết.

Sau đó, Composer sẽ tự động tải mã của các thư viện và tạo ra các tệp cần thiết, đưa chúng vào dự án. Khi có phiên bản mới, Composer sẽ tự động cập nhật các thư viện.

Lưu ý: Để Laravel hoạt động ổn định, nên cài đặt Composer trước khi sử dụng, vì nó quản lý toàn bộ thư viện cần thiết của PHP.

2.4.2 Những lợi ích của Composer

Trước khi Composer ra đời, nhiều lập trình viên đã cố gắng phát triển các công cụ để hỗ trợ công việc của họ trở nên thuận tiện hơn. Composer được lấy cảm hứng từ những công cụ trước đó, như npm của Node.js. Cách thức hoạt động của Composer tương tự như APT trên Ubuntu hoặc Yum trên CentOS. Tuy nhiên, không giống như APT hay Yum có thể quản lý toàn bộ hệ điều hành, Composer chỉ hoạt động trong phạm vi của PHP. [4]

2.5 Laravel

2.5.1 Laravel Framework

Laravel là một framework PHP mã nguồn mở miễn phí được phát hành lần đầu tiên vào năm 2011. Taylor Otwell đã tạo ra Laravel nhằm giúp các Developer xây dựng các ứng dụng web bằng cách sử dụng mẫu kiến trúc Model-View-Controller (MVC).



Hình 2.5.1 Laravel

Laravel tận dụng tối đa các tính năng tiên tiến của PHP, chẳng hạn như lập trình hướng đối tượng (OOP), cung cấp cú pháp rõ ràng và có khả năng truyền đạt tốt, giúp dễ dàng viết mã code duy trì và mở rộng. Nếu đã quen thuộc với Core PHP và PHP nâng cao, Laravel sẽ giúp công việc của chúng ta trở nên dễ dàng hơn, tiết kiệm nhiều thời gian, đặc biệt nếu dự định phát triển một trang web từ đầu.

2.5.2 Ngôn ngữ lập trình phía back-end

Laravel là một framework back-end, được viết bằng ngôn ngữ lập trình PHP. Laravel cung cấp các công cụ và cấu trúc để xây dựng phần logic của ứng dụng web ví dụ như: Xử lý yêu cầu và phản hồi HTTP, kết nối và thao tác với cơ sở dữ liệu, xác thực và phân quyền người dùng, quản lý cache, gửi email và nhiều tính năng khác.

2.5.3 Những tính năng nổi bật của Laravel

2.5.3.1 Kiến trúc MVC

Laravel khuyến khích việc sử dụng kiến trúc MVC (Model-View-Controller) trong phát triển ứng dụng web. Framework này cung cấp một tập hợp các quy tắc nhằm hỗ trợ việc xây dựng ứng dụng web có khả năng mở rộng và dễ dàng bảo trì. Kiến trúc MVC trong Laravel giúp các nhà phát triển tổ chức mã nguồn một cách gọn gàng và nhất quán, đặc biệt có ích khi làm việc với các ứng dụng web có quy mô lớn.

2.5.3.2 Artisan CLI

Artisan là công cụ dòng lệnh (CLI) trong Laravel, hỗ trợ các nhà phát triển web trong nhiều tác vụ, chẳng hạn như:

- Di chuyển dữ liệu.
- Quản lý cơ sở dữ liệu.
- Tạo sẵn mã cho các thành phần như bộ điều khiển, models, và các thành phần khác.

Artisan CLI đơn giản hóa quy trình phát triển web bằng cách tự động hóa việc tạo mã và quản lý cơ sở dữ liệu thông qua một số lệnh đơn giản. Nhờ công cụ này, các lập trình viên có thể tập trung nhiều hơn vào xây dựng logic ứng dụng thay vì phải viết mã thủ công.

2.5.3.3 Authentication

Laravel cung cấp các giải pháp tích hợp sẵn cho việc xác thực và ủy quyền. Chúng ta có thể dễ dàng thiết lập hệ thống xác thực đăng tin cậy cho ứng dụng web chỉ với vài lệnh Artisan. Thêm phần mềm trung gian xác thực vào các tuyến đường của Laravel giúp đơn giản hóa quy trình xác thực người dùng. Khi quá trình xác thực hoàn tất, chúng ta có thể sử dụng Auth Facade để truy cập thông tin người dùng.

2.5.3.4 Eloquent ORM

Eloquent ORM trong Laravel giúp đơn giản hóa việc tương tác với dữ liệu trong ứng dụng. Với Eloquent, chúng ta có thể nhanh chóng tạo các model, thực hiện di chuyển dữ liệu, và thiết lập mối quan hệ giữa các bảng trong cơ sở dữ liệu. Mỗi bảng dữ liệu đều có một model tương ứng, giúp chúng ta dễ dàng thao tác với bảng.

Ngoài việc truy xuất các bản ghi, Eloquent còn cung cấp khả năng chèn, cập nhật, và xóa dữ liệu một cách linh hoạt.

2.5.3.5 Blade Template

Blade Template Engine, một công cụ mạnh mẽ đi kèm với Laravel, hỗ trợ:

- Kết nối với các mô hình dữ liệu;
- Xử lý mã ứng dụng bên trong các thẻ mẫu;
- Chuyển đầu ra sang tệp văn bản hoặc các luồng khác.

Với những tính năng này, Blade giúp tối ưu hóa và đơn giản hóa quá trình phát triển web, mang lại sự nhanh chóng và tiện lợi cho lập trình viên.

2.5.3.6 Event & Websockets

Laravel hỗ trợ tính năng sự kiện, giúp chúng ta gửi và theo dõi các sự kiện trong ứng dụng một cách dễ dàng. Các listener có thể được đưa vào hàng đợi nền để xử lý hiệu quả hơn. Bên cạnh đó, chúng ta có thể sử dụng Laravel Echo và WebSockets để kết nối ứng dụng front-end, cho phép tạo ra các ứng dụng web theo thời gian thực.

2.5.4 Cấu trúc thư mục của Laravel

Thư mục	Chức năng
/app	Chứa các controller, model, views, và assets của ứng dụng, bao gồm toàn bộ mã nguồn chính.
/public	Lưu trữ các tệp CSS, JavaScript, hình ảnh, và các tệp khác, đồng thời chứa tệp khởi động index.php.
/vendor	Nơi lưu giữ toàn bộ mã nguồn của bên thứ ba, bao gồm các plugin được cài đặt cho ứng dụng.
/app/config/	Chứa các tệp cấu hình như cơ sở dữ liệu, session, và các thiết lập quan trọng khác của ứng dụng.
/app/config/app.php	Tệp cấu hình chung, quản lý cài đặt như múi giờ, ngôn ngữ, chế độ debug, và khóa mã hóa duy nhất.
/app/config/auth.php	Cấu hình xác thực của ứng dụng.
/app/config/cache.php	Cấu hình bộ nhớ đệm, giúp cải thiện hiệu suất phản hồi của ứng dụng.
/app/database/migrations/	Lưu các lớp PHP dùng để cập nhật Schema của cơ sở dữ liệu trong khi vẫn giữ được các phiên bản dữ liệu đã đồng bộ, với các tệp migration được tạo qua công cụ Artisan.
/app/database/seeds/	Chứa các tệp PHP để Artisan thêm dữ liệu mẫu vào các bảng cơ sở dữ liệu.
/app/models/	Nơi lưu trữ các tệp model của ứng dụng.
/app/views/	Thư mục chứa các tệp HTML được sử dụng bởi các controller hoặc route.

/app/lang/	Chứa các tệp ngôn ngữ mặc định, sử dụng cho việc phân trang và xác thực form với ngôn ngữ gốc là tiếng Anh.
/app/start/	Lưu các thiết lập tùy chỉnh liên quan đến Artisan và bối cảnh local hoặc global.
/app/storage	Dùng để lưu trữ các tệp tạm thời cho các dịch vụ như session, cache, và template views đã biên dịch. Thư mục này có quyền ghi và được quản lý tự động bởi Laravel.
/app/routes.php	Tệp định tuyến chính, chứa tất cả các route và cách kết nối ứng dụng khi có yêu cầu. Tệp này cũng định nghĩa các sự kiện như trang lỗi và view composers.
/app/filter.php	Sử dụng để hạn chế quyền truy cập vào một số khu vực nhất định của trang web.

2.5.5 Vòng đời của Laravel

Gọi là vòng đời như thể là sinh vật có sự sống, nhưng ở đây vòng đời có nghĩa là luồng xử lý dữ liệu khi một ứng dụng Laravel nhận được yêu cầu → xử lý → trả về kết quả.

1. Yêu cầu HTTP được định tuyến đến một Controller (được khai báo trong tệp routing ở thư mục /app/routes.php).
2. Controller sẽ thực hiện các hành động cần thiết và chuyển kết quả đến view (trong thư mục /app/controllers).
3. Sau đó, view sẽ hiển thị dữ liệu ở định dạng phù hợp và gửi lại phản hồi HTTP (thư mục /app/views).

2.5.6 Ưu điểm và nhược điểm của Laravel

2.5.6.1 Ưu điểm

Cú pháp PHP rõ ràng

Cú pháp của Laravel rất rõ ràng và dễ hiểu, giúp các nhà phát triển viết mã một cách dễ dàng. Nhờ đó, việc theo dõi logic của mã nguồn trở nên đơn giản hơn, góp phần giảm bớt thời gian viết mã và khắc phục lỗi. Laravel còn tích hợp nhiều tính năng hữu ích, giúp đơn giản hóa các tác vụ phát triển web phổ biến như định tuyến, xác thực, và quản lý cơ sở dữ liệu.

Thân thiện với nhà phát triển

Laravel được thiết kế thân thiện với nhà phát triển, cung cấp nhiều công cụ và tính năng giúp đơn giản hóa quá trình phát triển. Điển hình là giao diện dòng lệnh trực quan (CLI) và các công cụ kiểm tra tích hợp như unit tests, giúp đảm bảo rằng các đoạn mã nhỏ hoạt động như mong đợi.

Giao diện dòng lệnh Artisan mạnh mẽ mang lại hiệu suất cao trong môi trường phát triển. Nó cho phép nhà phát triển thực hiện các tác vụ như xuất bản nội dung gói, quản lý di chuyển cơ sở dữ liệu, và tạo mã tự động. Artisan cũng giúp dễ dàng tạo các tệp MVC và cung cấp khả năng tùy chỉnh lệnh để phù hợp với nhu cầu cụ thể của từng dự án.

Sử dụng kiến trúc MVC

Laravel áp dụng mẫu kiến trúc Model-View-Controller (MVC), giúp tách biệt logic nghiệp vụ và chức năng phần mềm khỏi logic giao diện hoặc trình bày. Kiến trúc MVC hỗ trợ việc duy trì và mở rộng ứng dụng dễ dàng hơn, đặc biệt khi thêm các thư viện mã theo thời gian. Mô hình này giúp sắp xếp và nhất quán mã nguồn, đồng thời quản lý dữ liệu một cách hiệu quả, thay vì xử lý mã không có cấu trúc.

Các gói sử dụng và thư viện đa dạng

Laravel cung cấp rất nhiều gói (packages) và thư viện có sẵn thông qua trình quản lý Composer. Những gói và thư viện này giúp mở rộng chức năng của framework, làm cho việc xây dựng các ứng dụng Laravel phức tạp trở nên đơn giản và hiệu quả hơn.

2.5.6.2 Nhược điểm

Lộ trình học khó khăn

Laravel được xây dựng trên nền tảng PHP, vì vậy việc học Laravel sẽ không hiệu quả nếu chưa nắm vững PHP. Nếu không hiểu rõ PHP, sẽ khó có thể tạo các chức năng bổ sung và sẽ phụ thuộc hoàn toàn vào những gì Laravel cung cấp sẵn. Để sử dụng Laravel hiệu quả và khai thác tối đa tiềm năng của framework nên cần nắm vững các khái niệm về PHP và lập trình hướng đối tượng (OOP).

Laravel có một lộ trình học tập khá khó khăn đối với các lập trình viên mới làm quen. Nguyên nhân là framework này bao gồm nhiều tính năng và khái niệm phức tạp, đòi hỏi thời gian để hiểu và sử dụng thành thạo.

Hiệu suất chậm

Hiệu suất của ứng dụng web Laravel có thể kém hơn so với một số framework PHP khác, đặc biệt khi xử lý khối lượng dữ liệu lớn hoặc lưu lượng truy cập cao. Tuy nhiên, các nhà phát triển có thể áp dụng một số kỹ thuật tối ưu hóa để cải thiện hiệu suất của framework.

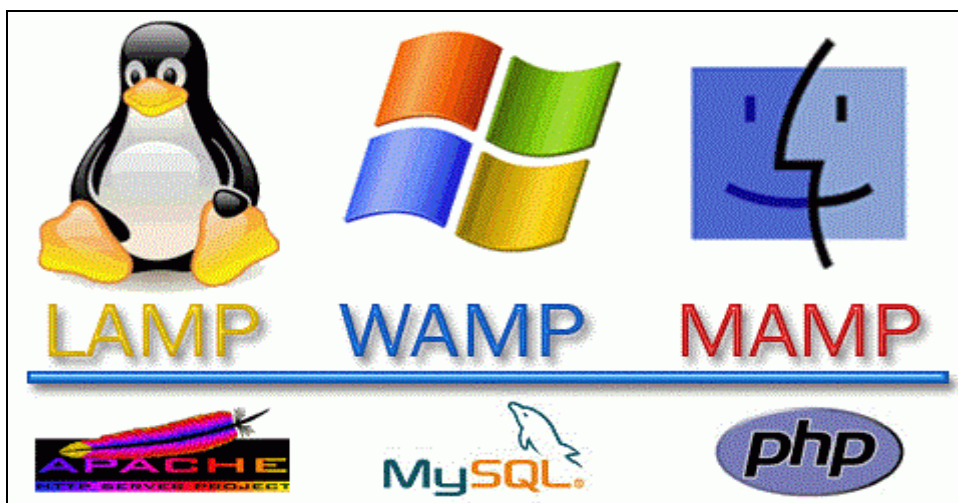
Khó khăn khi nâng cấp

Quá trình nâng cấp lên phiên bản mới của Laravel có thể gặp thách thức, đặc biệt khi ứng dụng phụ thuộc vào nhiều gói và thư viện của bên thứ ba. Nguyên nhân là do các phiên bản mới của Laravel có thể yêu cầu cập nhật mã nguồn để phù hợp với những thay đổi.

2.6 Laragon

2.6.1 Sơ lược về WAMP, LAMP và MAMP

Trước khi tìm hiểu về các phần mềm như Laragon, XAMPP, hay những công cụ tương tự, chúng ta cần hiểu về WAMP, LAMP và MAMP. AMP là viết tắt của Apache, MySQL và PHP, một bộ công cụ vô cùng quan trọng đối với các nhà phát triển web.



Hình 2.6.1 Sơ lược LAMP, WAMP, MAMP

W, L và M đại diện cho ba hệ điều hành chính: Windows, Linux và Mac. Tùy vào hệ điều hành sử dụng, sẽ phải cài đặt các bộ công cụ lập trình khác nhau. Ví dụ, nếu sử dụng Linux thì phải cài đặt LAMP để có thể thiết lập một server ngay trên máy tính của mình.

Với ba công cụ này, chúng ta không cần phải đăng ký một dịch vụ web hosting tốn kém chỉ để phát triển mã nguồn, giúp tiết kiệm chi phí khi xây dựng ứng dụng web.

2.6.2 Laragon

Laragon là một phần mềm cung cấp môi trường phát triển trên Windows, cho phép các công cụ như MySQL, PHP, Redis, và Apache (WAMP Stack) chạy dễ dàng và nhanh chóng sau khi cài đặt. Ngoài ra, Laragon còn tương thích với các ngôn ngữ khác như Python, Java, và Ruby. Chúng ta cũng có thể tích hợp các cơ sở dữ liệu như MongoDB và PostgreSQL một cách đơn giản chỉ với vài thao tác.



Hình 2.6.2 Laragon

Laragon mang đến một môi trường WAMP mạnh mẽ và hoàn toàn miễn phí. Phần mềm này cho phép thiết lập một web server ngay trên máy tính mà không cần đăng ký dịch vụ nào khác. Laragon nhẹ và dễ cài đặt, chỉ chiếm khoảng 4MB RAM khi hoạt động. Đặc biệt, Laragon không sử dụng bất kỳ dịch vụ nào của Windows, vì nó đã tích hợp các dịch vụ của riêng mình.

2.6.3 Những tính năng nổi bật của Laragon

- **Pretty URLs:** Cho phép tạo tên miền tùy chỉnh (ví dụ: duanlaravel.local thay vì localhost).
- **Portable:** Dễ dàng di chuyển thư mục Laragon sang vị trí khác, chẳng hạn như máy tính khác hoặc USB, mà không gặp lỗi hoặc ảnh hưởng đến hệ thống.
- **Isolated:** Tạo môi trường hoạt động riêng biệt để giữ cho hệ điều hành sạch sẽ, không ảnh hưởng bởi các cấu hình khác.
- **Easy Operation:** Laragon tự động cấu hình mọi thứ, giúp dễ dàng thêm hoặc thay đổi phiên bản của Java, Python, PHP, Go, Apache, và nhiều công cụ khác.

- **Hiện đại và mạnh mẽ:** Laragon hỗ trợ chạy đồng thời Apache và Nginx, giúp tiết kiệm thời gian quản lý. Ngoài ra, Laragon cung cấp các tính năng như cài đặt SSL, dùng dòng lệnh (cmd), và thao tác trực tiếp với cơ sở dữ liệu. Và có thể triển khai dự án và thông báo cho khách hàng hoặc gửi email ngay từ môi trường cục bộ.

2.6.4 Những chức năng của Laragon

Laragon là một bộ công cụ được phát triển bởi một lập trình viên người Việt Nam (anh Khoa Le) và rất quen thuộc với cộng đồng những người làm việc với Laravel. Nhiều lập trình viên trên thế giới cũng đã tải về và sử dụng Laragon, nhận xét rằng phần mềm này hoạt động mượt mà và hiệu quả. Laragon cung cấp nhiều chức năng hữu ích cho lập trình viên, bao gồm: [5]

- Tối ưu hóa cho phát triển web
- Cài đặt nhanh chóng các nền tảng như WordPress, Drupal, Laravel Framework,...
- Quản lý cơ sở dữ liệu dễ dàng
- Hỗ trợ Terminal
- Nhiều phiên bản PHP để lựa chọn
- Cài đặt SSL đơn giản, không yêu cầu nhiều bước phức tạp như XAMPP
- Tạo tên miền tùy chỉnh
- Tích hợp gửi email từ localhost qua giao thức SMTP đã được cài sẵn
- Tích hợp công cụ Git, Memcached, và Redis

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán

Trong thời đại công nghệ thông tin phát triển mạnh mẽ hiện nay, việc trao đổi kiến thức, kinh nghiệm và giải quyết các vấn đề liên quan đến tin học trở nên ngày càng quan trọng đối với cả cá nhân và tổ chức. Các diễn đàn trực tuyến đóng vai trò thiết yếu trong việc tạo ra không gian trao đổi, học hỏi và hợp tác giữa các thành viên trong cộng đồng. Tuy nhiên, nhiều diễn đàn hiện tại còn tồn tại một số hạn chế như giao diện người dùng không thân thiện, tính năng quản lý nội dung chưa hiệu quả, khả năng mở rộng hạn chế và thiếu các chức năng tương tác hiện đại.

Bài toán xây dựng trang web diễn đàn tin học sử dụng framework Laravel nhằm giải quyết các vấn đề trên bằng cách phát triển một nền tảng diễn đàn hiện đại, dễ sử dụng và linh hoạt. Cụ thể, dự án đặt ra các mục tiêu sau:

1. **Thiết kế giao diện người dùng thân thiện:** Tạo ra một giao diện trực quan, dễ dàng điều hướng, phù hợp với nhiều đối tượng người dùng từ người mới bắt đầu đến các chuyên gia trong lĩnh vực tin học.
2. **Quản lý nội dung hiệu quả:** Phát triển các công cụ quản lý bài viết, chủ đề, và người dùng mạnh mẽ, giúp quản trị viên dễ dàng kiểm soát và duy trì chất lượng nội dung trên diễn đàn.
3. **Tính năng tương tác hiện đại:** Tích hợp các chức năng như bình luận, đánh giá, thông báo để tăng cường sự tương tác giữa các thành viên.
4. **Khả năng mở rộng và bảo trì:** Sử dụng framework Laravel để đảm bảo mã nguồn có cấu trúc rõ ràng, dễ bảo trì và dễ mở rộng khi cần thêm các tính năng mới hoặc cải tiến hiện có.
5. **Bảo mật và hiệu suất:** Đảm bảo trang web diễn đàn hoạt động ổn định, bảo mật thông tin người dùng và dữ liệu, đồng thời tối ưu hóa hiệu suất để đáp ứng nhu cầu sử dụng cao.

3.2 Phân tích chức năng

3.2.1 Đối với người dùng

Khi người dùng chưa đăng nhập, họ có thể truy cập các chức năng như **tìm kiếm bài viết**, **xem danh mục** và **đọc các bài viết có sẵn**. Việc này giúp người dùng dễ dàng tiếp cận thông tin, khám phá nội dung mà không cần tạo tài khoản ngay lập tức. Điều này góp phần gia tăng tính phổ biến và khả năng lan tỏa của diễn đàn.

Tuy nhiên, để thực hiện các hành động mang tính tương tác cao hơn như **đăng bài viết mới**, **bình luận**, hoặc **thích/không thích các bài viết**, hệ thống yêu cầu người dùng phải **đăng ký tài khoản** và **đăng nhập**. Khi đăng ký, người dùng cần cung cấp các thông tin cơ bản như tên, email, mật khẩu và số điện thoại. Quá trình đăng ký cũng có thể bao gồm việc xác thực email để đảm bảo tính bảo mật và tránh việc tạo tài khoản giả mạo.

Sau khi đăng nhập, người dùng có thể quản lý hồ sơ cá nhân của mình thông qua chức năng **chỉnh sửa thông tin cá nhân**. Các thông tin như tên, địa chỉ, số điện thoại và hình đại diện có thể được cập nhật dễ dàng. Điều này giúp cá nhân hóa tài khoản, tạo sự chuyên nghiệp và gắn kết giữa người dùng và diễn đàn.

Một trong những chức năng chính mà người dùng có thể thực hiện là **tạo bài viết mới**. Khi tạo bài viết, người dùng có thể chọn danh mục phù hợp, thêm hình ảnh, viết mô tả chi tiết và trình bày nội dung chính của bài viết. Hệ thống cũng hỗ trợ tính năng xem trước bài viết trước khi đăng lên diễn đàn. Sau khi gửi bài, bài viết sẽ được chuyển vào trạng thái chờ duyệt và chỉ hiển thị công khai sau khi được quản trị viên phê duyệt. Điều này giúp kiểm soát chất lượng nội dung trên diễn đàn, tránh tình trạng spam hoặc nội dung không phù hợp.

Bên cạnh đó, người dùng có thể **chỉnh sửa hoặc ẩn bài viết** của mình sau khi bài viết đã được đăng. Hệ thống không cho phép xóa hoàn toàn bài viết nhằm bảo vệ nội dung và tránh mất mát dữ liệu. Trong trường hợp người dùng nhận thấy bài viết không còn phù hợp, họ có thể chuyển bài viết sang trạng thái ẩn, điều này giúp bài viết không còn hiển thị công khai mà không ảnh hưởng đến cấu trúc của diễn đàn.

Ngoài việc tạo bài viết, người dùng cũng có thể **bình luận và tương tác** với các bài viết khác. Tính năng bình luận cho phép người dùng đưa ra ý kiến, góp ý hoặc trao đổi trực tiếp với tác giả của bài viết. Các bình luận có thể bao gồm văn bản, hình ảnh và liên kết, tạo ra môi trường thảo luận sôi nổi. Đồng thời, người dùng có thể **thích hoặc không thích bài viết** thông qua nút tương tác nhanh, từ đó tạo ra các bài viết nổi bật và thúc đẩy chất lượng nội dung.

Hệ thống cũng cung cấp **chức năng tìm kiếm** bài viết theo từ khóa, giúp người dùng dễ dàng tìm thấy các nội dung mà họ quan tâm. Các bài viết có thể được lọc theo danh mục, lượt thích hoặc thời gian đăng, mang lại trải nghiệm tìm kiếm hiệu quả và nhanh chóng.

3.2.2 Đối với người quản trị

Quản trị viên có quyền **duyet bài viết mới** từ người dùng. Mỗi bài viết sau khi được tạo bởi người dùng sẽ được chuyển vào danh sách chờ duyệt. Admin có thể kiểm tra nội dung bài viết, đánh giá mức độ phù hợp và đưa ra quyết định phê duyệt hoặc ẩn bài viết nếu vi phạm các quy định của diễn đàn. Tính năng này giúp hệ thống kiểm soát chặt chẽ chất lượng và tính chính xác của các thông tin được đăng tải, tránh tình trạng bài viết không lành mạnh hoặc có nội dung gây tranh cãi.

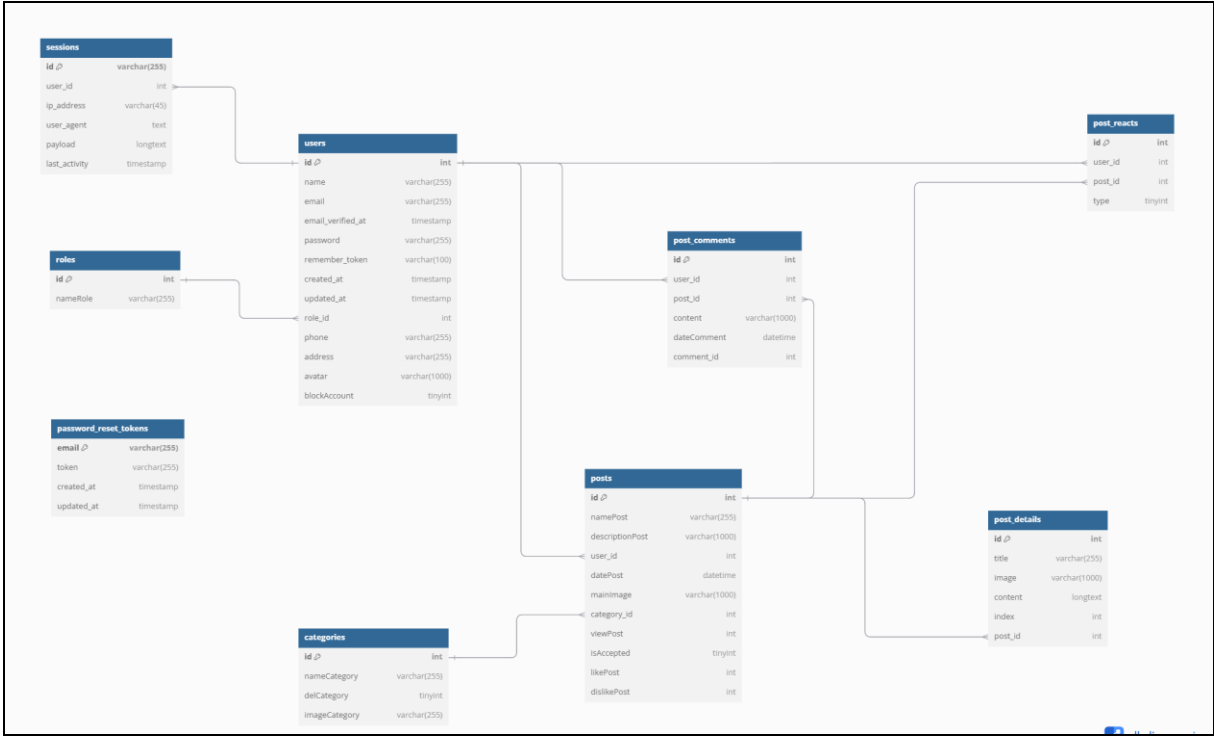
Ngoài ra, admin còn có thể **ẩn bài viết vi phạm** trực tiếp ngay trên giao diện quản lý. Nếu một bài viết được nhiều người báo cáo hoặc vi phạm quy tắc cộng đồng, admin có thể nhanh chóng ẩn bài viết, đảm bảo môi trường diễn đàn lành mạnh. Hệ thống không cho phép xóa bài viết hoàn toàn mà chỉ hỗ trợ ẩn để tránh xung đột hoặc mất mát dữ liệu.

Bên cạnh việc quản lý bài viết, admin còn có quyền **quản lý người dùng**. Các chức năng bao gồm việc **xem danh sách người dùng, chỉnh sửa thông tin người dùng** hoặc **khóa tài khoản** nếu phát hiện hành vi vi phạm.

Quản trị viên cũng có quyền **quản lý danh mục bài viết**, thêm mới hoặc chỉnh sửa các danh mục sẵn có. Việc này giúp tổ chức nội dung bài viết rõ ràng, tạo ra sự phân loại hợp lý cho các chủ đề khác nhau. Hệ thống cho phép admin thêm ảnh đại diện cho danh mục, tạo sự sinh động và dễ nhận diện hơn khi người dùng tìm kiếm.

3.3 Thiết kế CSDL

3.3.1 Lược đồ quan hệ thực thể (ERD)



Hình 3.3.1 Lược đồ quan hệ thực thể

3.3.2 Mô tả các bảng CSDL

Bảng 3.3.2.1 users

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính (Primary Key), tự động tăng (Auto Increment)
name	varchar(255)	Tên người dùng
email	varchar(255)	Địa chỉ email, duy nhất
email_verified_at	timestamp	Thời điểm xác thực email
password	varchar(255)	Mật khẩu đã mã hóa

remember_token	varchar(100)	Token ghi nhớ đăng nhập
created_at	timestamp	Thời điểm tạo tài khoản
updated_at	timestamp	Thời điểm cập nhật tài khoản
role_id	int	Khóa ngoại liên kết bảng roles
phone	varchar(255)	Số điện thoại người dùng
address	varchar(255)	Địa chỉ người dùng
avatar	varchar(1000)	URL ảnh đại diện
blockaccount	tinyint	Trạng thái khóa tài khoản (0: hoạt động, 1: khóa)

Bảng 3.3.2.1 roles

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính
nameRole	varchar(255)	Tên vai trò (Admin, Member)

Bảng 3.3.2.2 posts

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính
namePost	varchar(255)	Tiêu đề bài viết
descriptionPost	varchar(1000)	Mô tả ngắn

used_id	int	Khóa ngoại liên kết với bảng users
datePost	datetime	Thời gian tạo bài viết
mainImage	varchar(1000)	Ảnh bài viết
category_id	int	Khóa ngoại liên kết với bảng categories
viewPost	int	Số lượt xem bài viết
isAccepted	tinyint	Trạng thái duyệt bài (0: chờ duyệt, 1: Đã duyệt)
likePost	int	Số lượt thích
dislikePost	int	Số lượt không thích

Bảng 3.3.2.3 categories

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính
nameCategory	varchar(255)	Tên danh mục
delCategory	tinyint	Trạng thái (0: Hoạt động, 1: đã xóa)
imageCategory	varchar(255)	Ảnh đại diện cho danh mục

Bảng 3.3.2.4 post_details

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính

title	varchar(255)	Tiêu đề phần chi tiết
image	varchar(1000)	Ảnh bổ sung
content	longtext	Nội dung chi tiết của bài viết
index	int	Thứ tự hiển thị phần chi tiết
post_id	int	Khóa ngoại liên kết với bảng posts

Bảng 3.3.2.5 post_comments

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính
user_id	int	Khóa ngoại liên kết đến bảng users
post_id	int	Khóa ngoại liên kết với bảng posts
content	varchar(1000)	Nội dung bình luận
dateComment	datetime	Ngày tạo bình luận
Comment_id	int	ID của bình luận cha

Bảng 3.3.2.6 post_reacts

Tên cột	Kiểu dữ liệu	Mô tả
id	int	Khóa chính
user_id	int	Khóa ngoại liên kết với bảng users
post_id	int	Khóa ngoại liên kết với bảng posts

type	tinyint	Loại phản hồi (1: Like, 0: Dislike)
------	---------	-------------------------------------

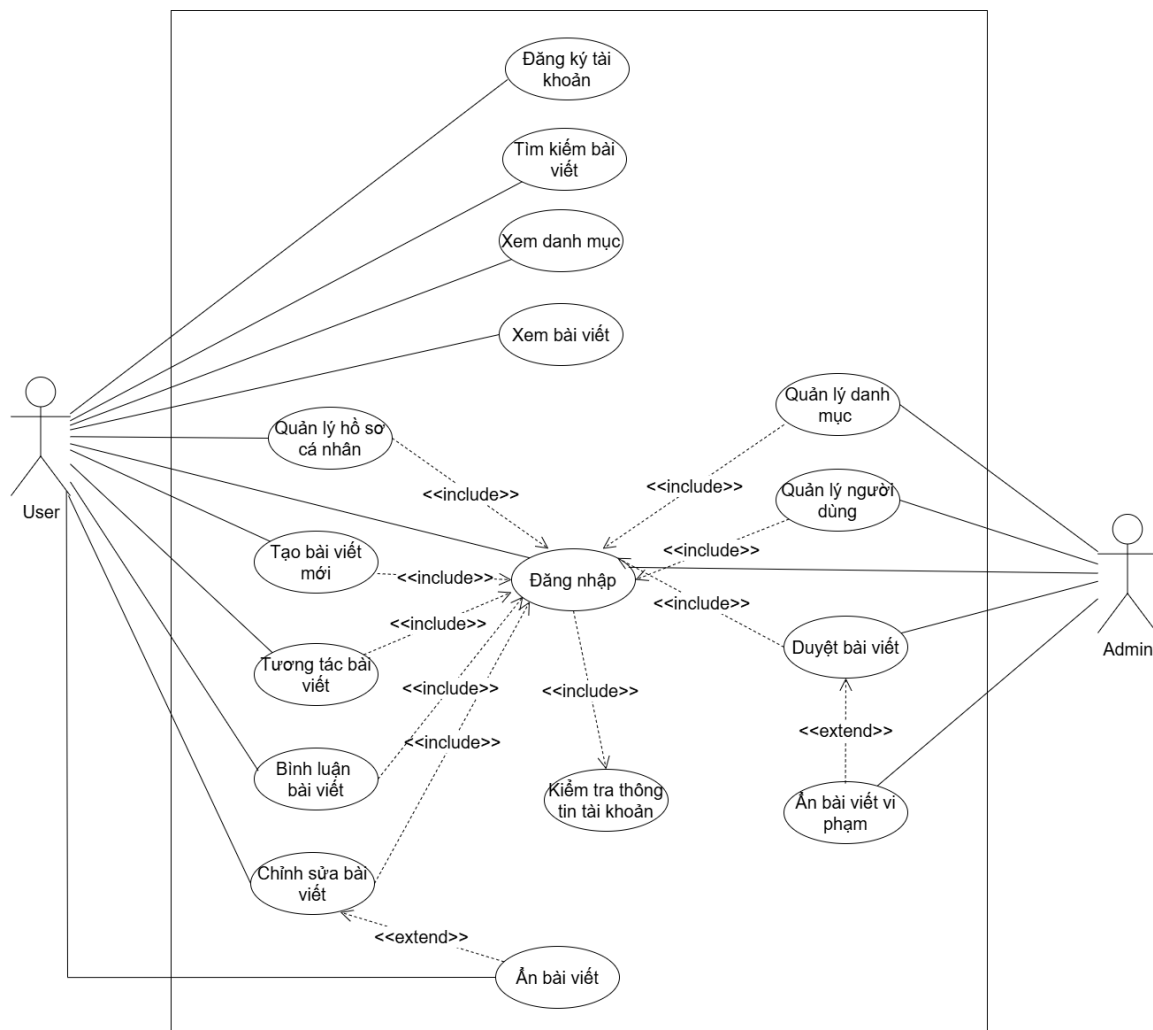
Bảng 3.3.2.7 sessions

Tên cột	Kiểu dữ liệu	Mô tả
id	varchar(255)	Khóa chính
user_id	int	Khóa ngoại liên kết đến bảng users
ip_address	varchar(45)	Địa chỉ ip đăng nhập
user_agent	text	Thông tin trình duyệt
payload	longtext	Dữ liệu phiên đăng nhập
last_activity	timestamp	Thời điểm hoạt động lần cuối

Bảng 3.3.2.8 password_reset_tokens

Tên cột	Kiểu dữ liệu	Mô tả
email	varchar(255)	Địa chỉ email (Khóa chính)
token	varchar(255)	Token reset mật khẩu
created_at	timestamp	Thời điểm tạo token
updated_at	timestamp	Thời điểm cập nhật token

3.4 Mô hình USE CASE tổng quan hệ thống

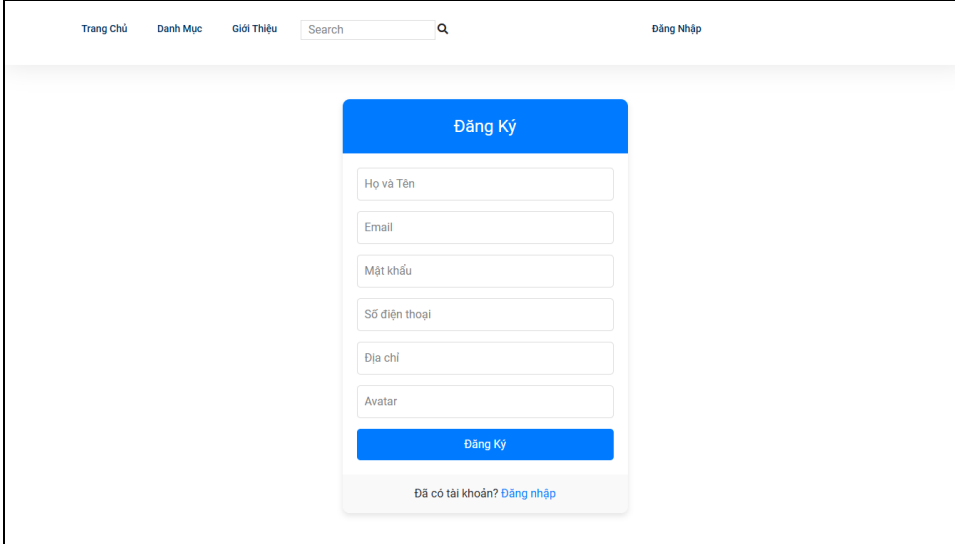


Hình 3.4 Biểu đồ Use Case tổng quát

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Người dùng

4.1.1 Trang đăng ký

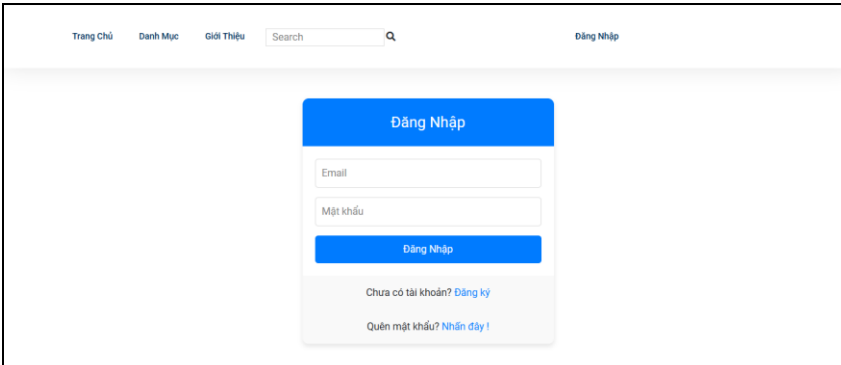


The screenshot shows a web interface for a forum. At the top, there is a navigation bar with links: Trang Chủ, Danh Mục, Giới Thiệu, a search bar, and Đăng Nhập. The main content area features a registration form titled 'Đăng Ký'. The form includes input fields for 'Họ và Tên', 'Email', 'Mật khẩu', 'Số điện thoại', 'Địa chỉ', and 'Avatar'. Below these fields is a blue 'Đăng Ký' button. At the bottom of the form, there is a link: 'Đã có tài khoản? Đăng nhập'.

Hình 4.4.1 Giao diện đăng ký

Người dùng có thể dễ dàng đăng ký tài khoản thông qua việc cung cấp các thông tin cơ bản như tên, email, số điện thoại và mật khẩu. Quá trình đăng ký được hỗ trợ bởi tính năng xác thực email, đảm bảo người dùng là thực và tránh tình trạng spam hoặc tạo tài khoản ảo. Sau khi đăng ký thành công, người dùng có thể đăng nhập và bắt đầu sử dụng hệ thống với đầy đủ các quyền hạn phù hợp.

4.1.2 Trang đăng nhập

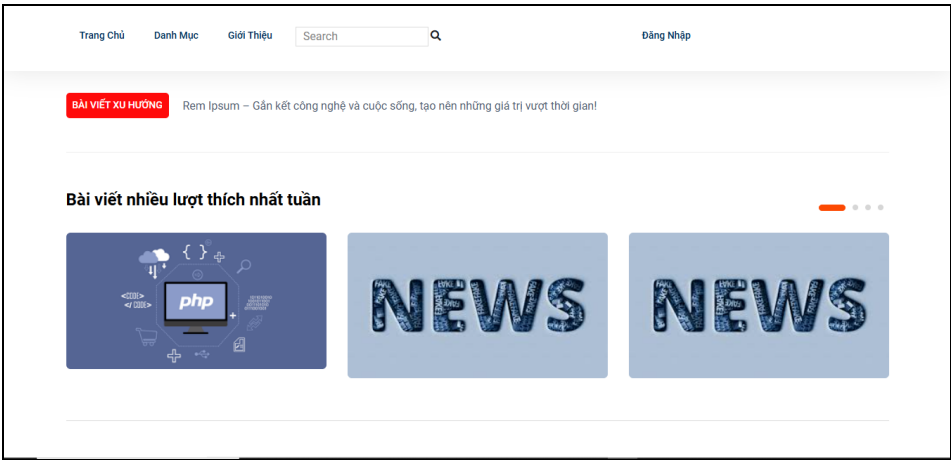


The screenshot shows a web interface for a forum. At the top, there is a navigation bar with links: Trang Chủ, Danh Mục, Giới Thiệu, a search bar, and Đăng Nhập. The main content area features a login form titled 'Đăng Nhập'. The form includes input fields for 'Email' and 'Mật khẩu'. Below these fields is a blue 'Đăng Nhập' button. At the bottom of the form, there are two links: 'Chưa có tài khoản? Đăng ký' and 'Quên mật khẩu? Nhấn đây!'.

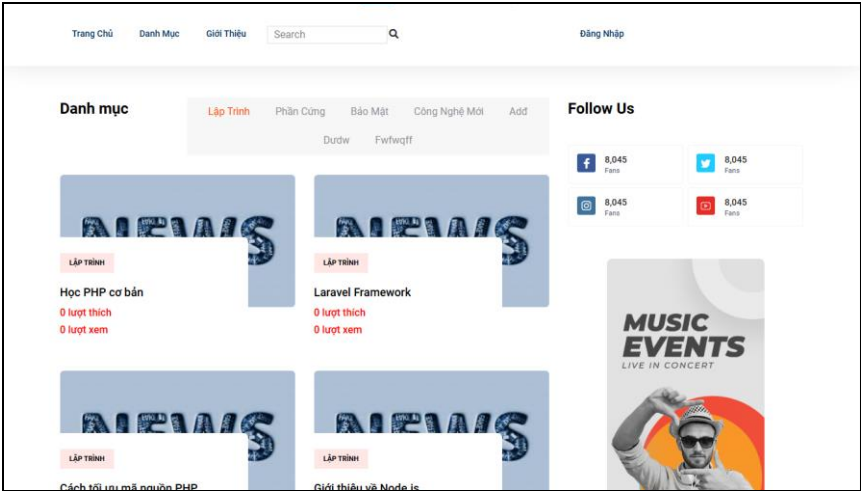
Hình 4.1.2 Giao diện đăng nhập

Hệ thống hỗ trợ đăng nhập nhanh chóng và ghi nhớ tài khoản bằng token. Người dùng có thể đăng xuất khỏi tài khoản bất cứ lúc nào.

4.1.3 Trang chủ

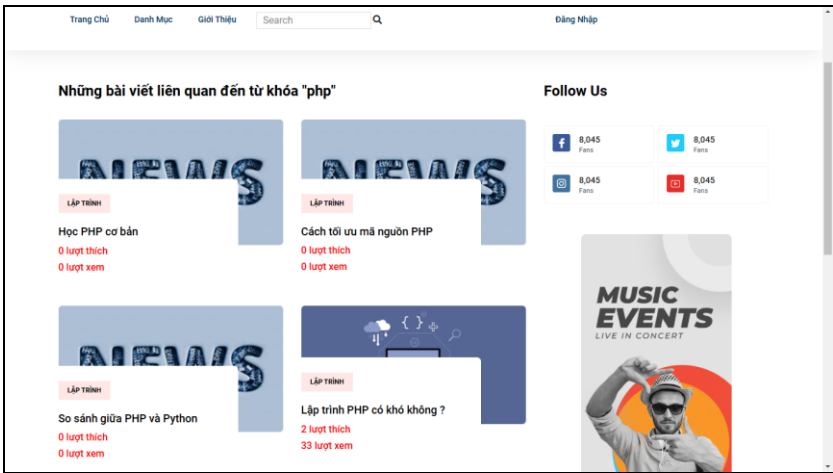


4.1.4 Trang danh mục



Hình 4.1.4 Giao diện danh mục

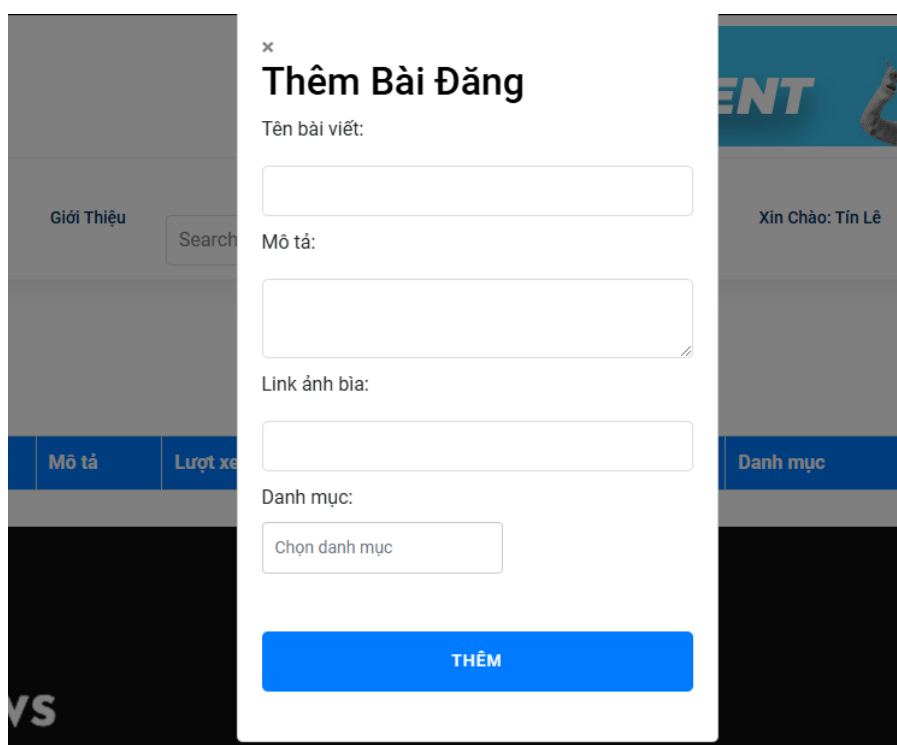
4.1.5 Tìm kiếm và xem bài viết



Hình 4.1.5 Giao diện tìm kiếm

Người dùng có thể tìm kiếm bài viết theo từ khóa hoặc duyệt qua các danh mục. Việc xem bài viết không yêu cầu đăng nhập, giúp hệ thống mở rộng khả năng tiếp cận nội dung. Các bài viết được phân loại rõ ràng theo danh mục, và người dùng có thể lọc bài viết theo từ khóa, danh mục hoặc thời gian đăng. Điều này giúp họ dễ dàng tìm kiếm các bài viết liên quan hoặc theo dõi những chủ đề mà mình quan tâm.

4.1.6 Tạo bài viết mới



Hình 4.1.6 Giao diện thêm bài đăng

Sau khi đăng nhập, người dùng có thể tạo bài viết mới với đầy đủ nội dung, hình ảnh và chọn danh mục phù hợp. Trước khi bài viết được công khai, hệ thống cho phép người dùng xem trước, từ đó chỉnh sửa và hoàn thiện nội dung trước khi gửi duyệt. Điều này giúp đảm bảo bài viết chất lượng, đáp ứng tiêu chí của diễn đàn. Nếu bài viết chưa được duyệt, người dùng vẫn có thể quay lại để chỉnh sửa hoặc bổ sung thông tin.

4.1.7 Chỉnh sửa và ẩn bài viết

Trang Chủ

Danh Mục

Giới Thiệu

Search

Xin Chào: Nguyễn Vũ Đại Nam

Đăng Xuất

Quản Trị Bài Đăng

THÊM BÀI ĐĂNG

#	Ảnh bìa	Tiêu Đề	Mô tả	Lượt xem	Lượt like	Lượt dislike	Danh mục	Trạng thái	Sửa	Ẩn
1		Học PHP cơ bản	Bài viết hướng dẫn cách học PHP cơ bản cho người mới bắt đầu.	0	0	0	Lập trình	Đã được phê duyệt	Sửa	Ẩn
2		Laravel Framework	Tìm hiểu về Laravel Framework và cách sử dụng.	0	0	0	Lập trình	Đã được phê duyệt	Sửa	Ẩn
		Cách tối ưu mã	Những mẹo tối ưu mã nguồn PHP để					Đã được phê		

Hình 4.1.7 Giao diện chỉnh sửa bài viết

Người dùng có quyền chỉnh sửa bài viết của mình sau khi đăng tải để cập nhật hoặc chỉnh sửa nội dung. Nếu bài viết không còn phù hợp, người dùng có thể ẩn bài viết thay vì xóa hoàn toàn. Điều này giúp tránh mất mát nội dung trong trường hợp cần khôi phục lại.

4.1.8 Chức năng bình luận

Share.

Danh sách bình luận

Hãy là người đầu tiên đánh giá !

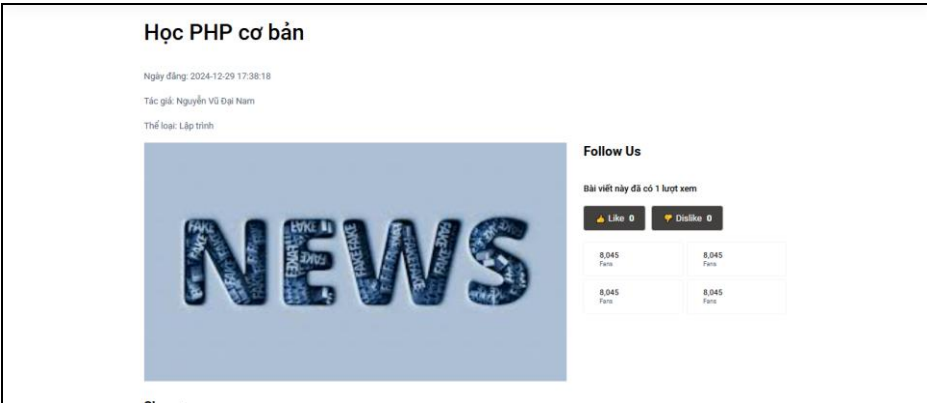
Enter Message

SEND

Hình 4.1.8 Giao diện bình luận

Người dùng có thể tham gia bình luận, trao đổi trực tiếp dưới các bài viết để tạo sự tương tác giữa các thành viên trong cộng đồng. Hệ thống hỗ trợ bình luận nhiều cấp (trả lời bình luận), tạo ra luồng thảo luận mạch lạc và logic.

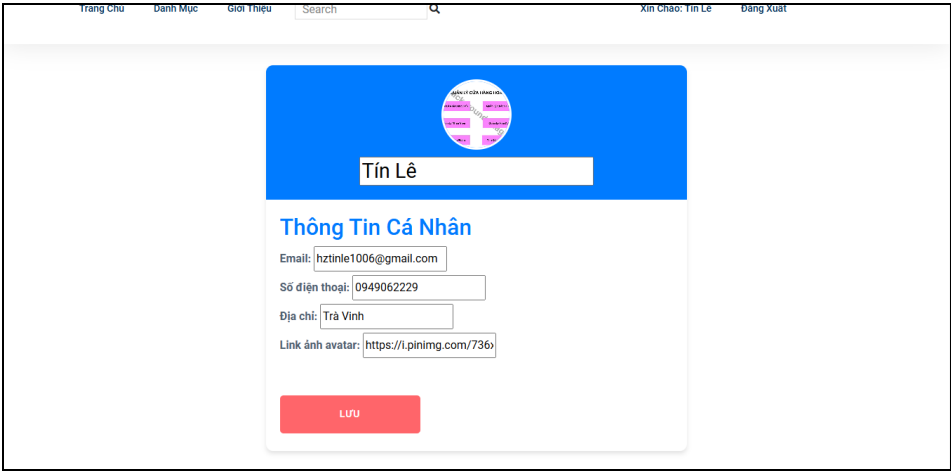
4.1.9 Chức năng tương tác với bài viết



Hình 4.1.9 Giao diện tương tác

Người dùng có thể bày tỏ cảm xúc bằng cách thích (like) hoặc không thích (dislike) bài viết. Chức năng này giúp đánh giá mức độ hữu ích của bài viết và góp phần nâng cao chất lượng nội dung trên diễn đàn.

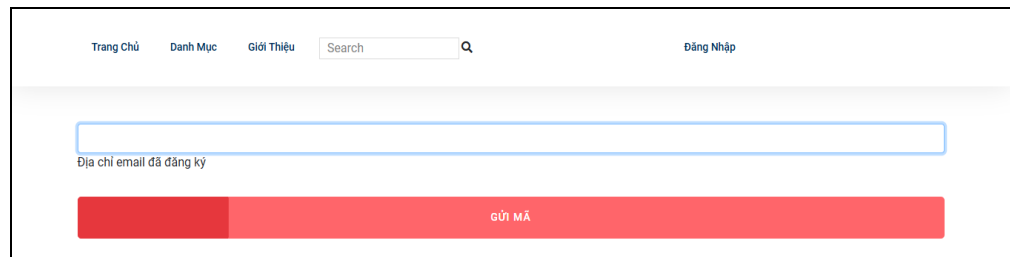
4.1.10 Quản lý hồ sơ cá nhân



Hình 6 Giao diện quản lý hồ sơ cá nhân

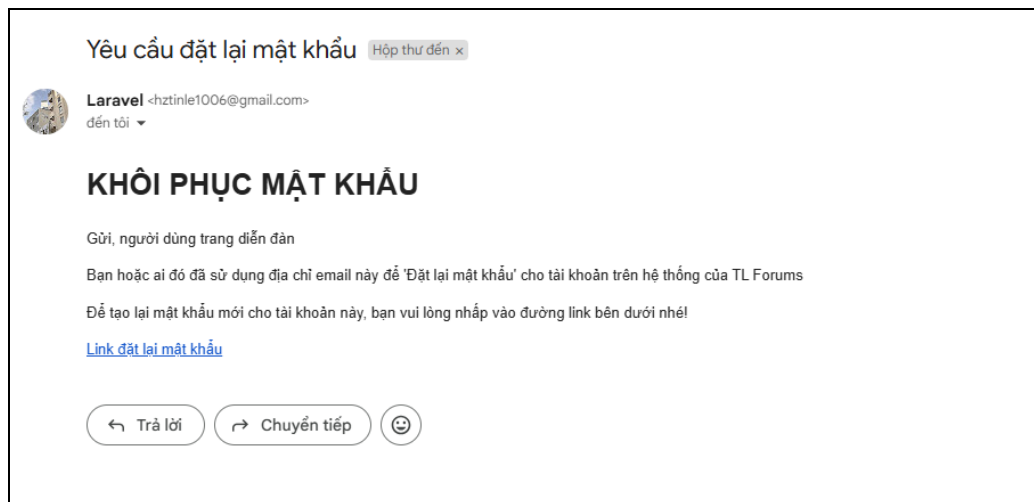
Người dùng có thể cập nhật thông tin như tên, địa chỉ, số điện thoại và ảnh đại diện. Điều này giúp cá nhân hóa tài khoản, đồng thời tạo ra sự chuyên nghiệp khi tham gia thảo luận trên diễn đàn.

4.1.11 Chức năng Quên mật khẩu



The screenshot shows a web interface for password reset. At the top, there are navigation links: 'Trang Chủ', 'Danh Mục', 'Giới Thiệu', a search bar with the text 'Search' and a magnifying glass icon, and a 'Đăng Nhập' button. Below these is a large, empty text input field. Underneath the input field is the label 'Địa chỉ email đã đăng ký'. At the bottom of the form is a red button with the text 'GỬI MÃ'.

Hình 4.1.11 Chức năng Quên mật khẩu



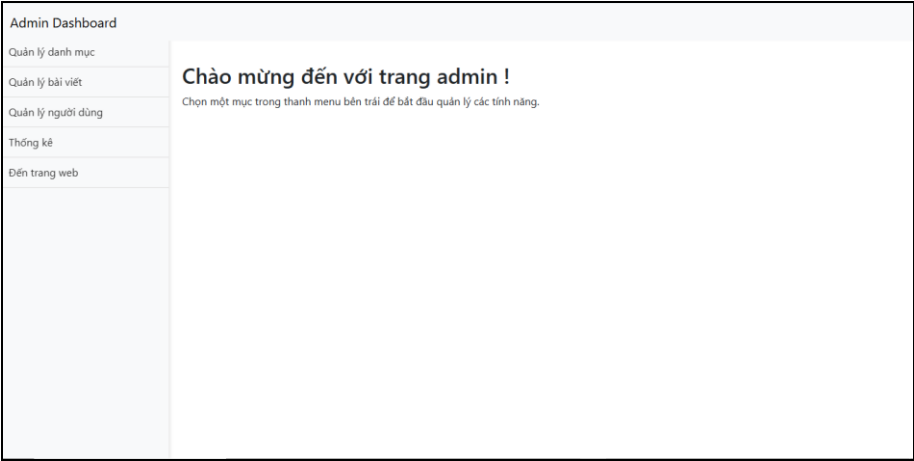
The screenshot shows an email interface. At the top, it says 'Yêu cầu đặt lại mật khẩu' followed by a 'Hộp thư đến x' button. Below this is a profile picture of a person and the name 'Laravel' with the email address '<hztinle1006@gmail.com>' and a 'đến tôi' dropdown menu. The main heading is 'KHÔI PHỤC MẬT KHẨU'. The body of the email contains the following text: 'Gửi, người dùng trang diễn đàn', 'Bạn hoặc ai đó đã sử dụng địa chỉ email này để "Đặt lại mật khẩu" cho tài khoản trên hệ thống của TL Forums', and 'Để tạo lại mật khẩu mới cho tài khoản này, bạn vui lòng nhấp vào đường link bên dưới nhé!'. There is a blue link labeled 'Link đặt lại mật khẩu'. At the bottom, there are three buttons: '← Trả lời', '→ Chuyển tiếp', and a smiley face icon.

Hình 4.1.12 Email reset mật khẩu

Hệ thống cũng tích hợp chức năng đặt lại mật khẩu thông qua email. Khi người dùng quên mật khẩu, họ có thể dễ dàng yêu cầu một liên kết đặt lại mật khẩu và nhanh chóng khôi phục tài khoản của mình. Điều này đảm bảo người dùng luôn có quyền kiểm soát tài khoản một cách an toàn và thuận tiện.

4.2 Quản trị

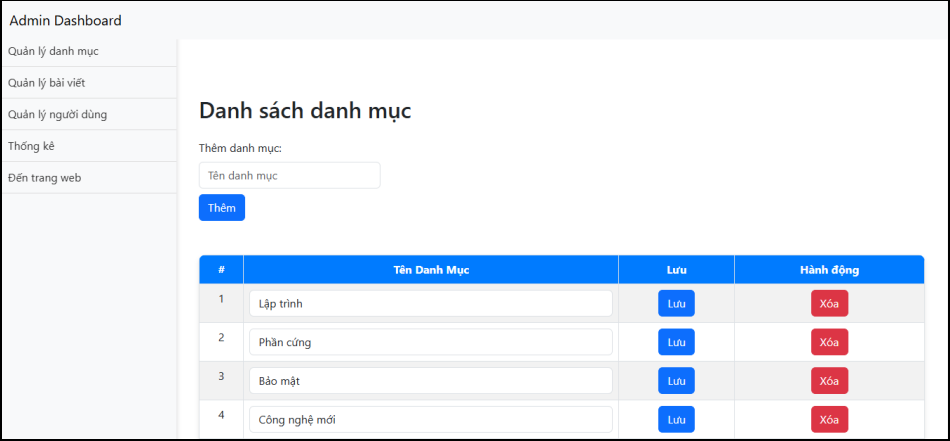
4.2.1 Trang quản trị



Hình 4.2.1 Giao diện trang quản trị

Trang quản trị cung cấp các chức năng chính để quản lý cũng như quan sát hệ thống diễn đàn.

4.2.2 Chức năng quản lý danh mục



Hình 4.2.2 Giao diện trang Quản lý danh mục

Admin có thể tạo, chỉnh sửa hoặc xóa danh mục bài viết. Việc này giúp tổ chức nội dung khoa học, phân loại rõ ràng và tạo ra sự thuận tiện cho người dùng khi tìm kiếm bài viết theo chủ đề.

4.2.3 Chức năng quản lý bài viết

Admin Dashboard									
Quản lý danh mục									
Quản lý bài viết									
Quản lý người dùng									
Thống kê									
Đến trang web									
Quản lý bài viết									
Được phê duyệt Chờ phê duyệt									
Được phê duyệt									
Ảnh bài viết	Tên bài viết	Mô tả	Danh mục	Tác giả	Lượt xem	Lượt like	Lượt dislike	Xem bài viết	Hành động
	Lập trình PHP có khó không ?	Đây là 1 bài viết rất đáng đọc cho các bạn chuẩn bị học php	Lập trình	Nguyễn Vũ Đại Nam	34	2	0	Xem	Bỏ phê duyệt
	Học PHP cơ bản	Bài viết hướng dẫn cách học PHP cơ bản cho người mới bắt đầu.	Lập trình	Nguyễn Vũ Đại Nam	1	0	0	Xem	Bỏ phê duyệt
	Laravel Framework	Tìm hiểu về Laravel Framework và cách sử dụng.	Lập trình	Nguyễn Vũ Đại Nam	0	0	0	Xem	Bỏ phê duyệt
	Cách tối ưu mã nguồn PHP	Những mẹo tối ưu mã nguồn PHP để cải thiện hiệu suất.	Lập trình	Nguyễn Vũ Đại Nam	0	0	0	Xem	Bỏ phê duyệt
	Giới thiệu về Node.js	Node.js là gì? Ưu điểm và nhược điểm của Node.js	Lập trình	Nguyễn Vũ Đại Nam	0	0	0	Xem	Bỏ phê duyệt

Hình 4.2.3 Giao diện trang Quản lý bài viết

Quản trị viên có thể xem xét các bài viết do người dùng gửi lên. Chỉ những bài viết được admin duyệt mới được phép hiển thị công khai trên diễn đàn. Việc này giúp kiểm soát nội dung, tránh các bài viết không phù hợp hoặc vi phạm chính sách cộng đồng.

Admin có quyền ẩn các bài viết nếu chúng vi phạm quy tắc. Việc ẩn bài viết đảm bảo môi trường diễn đàn trong sạch và tránh nội dung gây tranh cãi.

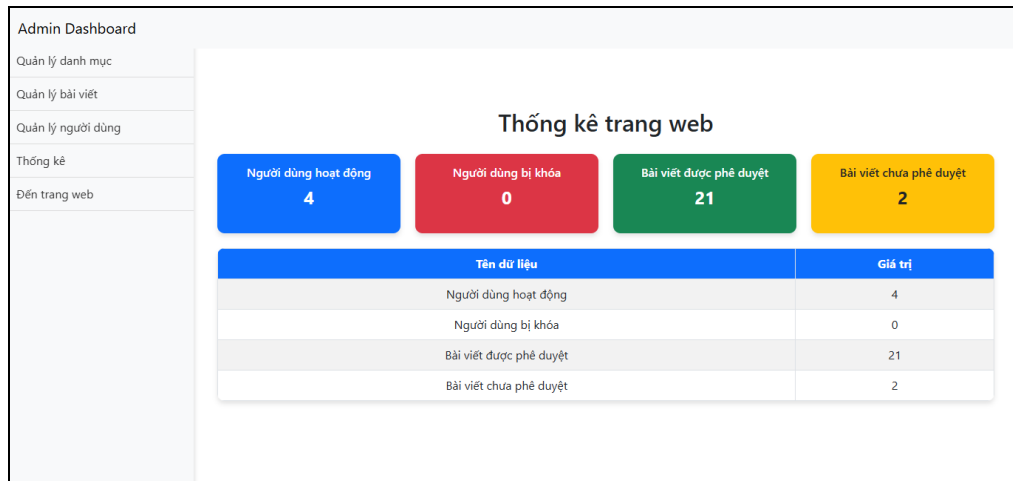
4.2.4 Chức năng quản lý người dùng

Admin Dashboard							
Quản lý danh mục							
Quản lý bài viết							
Quản lý người dùng							
Thống kê							
Đến trang web							
Danh sách người dùng							
#	Tên Người Dùng	Email	Address	Phone	Ngày đăng ký	Khóa tài khoản	Khóa quyền hoạt động
1	Nguyễn Vũ Đại Nam	nguyenvudaianm113@gmail.com	Trà Vinh	0365245602	2024-12-29 09:36:13	Khóa	Khóa
2	Nguyễn Ngọc Thạch	thach32314@gmail.com	Trà Vinh	0365245602	2024-12-30 07:07:21	Khóa	Khóa
3	Tín Lê	hztinle1006@gmail.com	Trà Vinh	0949062229	2025-01-01 18:28:11	Khóa	Khóa
4	Tín Lê	hztinle1005@gmail.com	Hẻm 87, lè lợi khóm 4 phường 1 tp.trà vinh	0949062229	2025-01-06 17:33:12	Khóa	Khóa

Hình 4.2.4 Giao diện trang Quản lý người dùng

Admin có thể xem danh sách tất cả người dùng, chỉnh sửa thông tin tài khoản hoặc khóa tài khoản. Tính năng này cho phép admin giữ sự ổn định và ngăn chặn các hành vi xấu.

4.2.5 Thống kê trang web



Hình 4.2.5 Giao diện trang Thống kê trang web

Hệ thống cũng cung cấp chức năng thống kê giúp admin theo dõi số lượng bài viết, tài khoản. Từ đó, admin có thể đưa ra các chính sách phù hợp nhằm thúc đẩy sự phát triển của diễn đàn.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 Về mặt tổng quan

Hệ thống diễn đàn đã được xây dựng và hoàn thiện với đầy đủ các chức năng cơ bản, đáp ứng nhu cầu của người dùng và quản trị viên. Quá trình phát triển bám sát yêu cầu thực tế, đảm bảo tính ổn định, bảo mật và khả năng mở rộng trong tương lai.

5.1.2 Về mặt chức năng

Người dùng có thể thực hiện các chức năng như đăng ký, đăng nhập, quản lý hồ sơ cá nhân, đăng bài viết, bình luận và tương tác với các nội dung trên diễn đàn.

Quản trị viên (Admin) có khả năng duyệt bài viết, quản lý người dùng, ẩn bài vi phạm, quản lý danh mục.

Hệ thống vận hành trơn tru, đảm bảo quản lý dữ liệu tập trung và phân cấp quyền hạn rõ ràng giữa người dùng và quản trị viên.

5.2 Hướng phát triển

Trong tương lai, hệ thống diễn đàn sẽ được nâng cấp để đáp ứng tốt hơn nhu cầu người dùng và cải thiện trải nghiệm. Các tính năng như bình luận lồng nhau cho phép phản hồi trực tiếp vào từng bình luận, và hệ thống thông báo đẩy giúp người dùng nhận cập nhật về tương tác mới hoặc bài viết được duyệt sẽ được phát triển.

Hệ thống sẽ tích hợp tìm kiếm nâng cao theo từ khóa, tác giả và bình luận, hỗ trợ sắp xếp theo lượt thích hoặc thời gian đăng. Bảo mật sẽ được tăng cường thông qua xác thực hai lớp (2FA), đặc biệt là cho quản trị viên.

Ngoài ra, tính năng báo cáo bài viết vi phạm sẽ giúp duy trì môi trường an toàn, trong khi API chia sẻ bài viết lên mạng xã hội sẽ mở rộng phạm vi tiếp cận và tăng tương tác.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] ITviec, "PHP là gì: Chi tiết và đầy đủ cách viết cú pháp PHP cơ bản," 07 July 2024. [Online]. Available: <https://itviec.com/blog/php-la-gi/>. [Đã truy cập 30 12 2024]
- [2] Kim Khánh, "Ngôn ngữ SQL là gì? Giới thiệu một số lệnh cơ bản của SQL," TEKY, 7 October 2023. [Online]. Available: https://teky.edu.vn/blog/ngon-ngu-sql/#Ngon_ngu_SQL_la_gi. [Đã truy cập 30 12 2024].
- [3] Le Thi Ngoc Anh, "Tìm hiểu về Framwork Laravel," VIBLO, 29 May 2016. [Online]. Available: <https://viblo.asia/p/tim-hieu-ve-framwork-laravel-p1-amoG8191vz8P>. [Đã truy cập 30 12 2024].
- [4] Hồ Hữu Hiền, "Composer Là Gì và Nó Giúp Ích Như Thế Nào?," HocLaravel, 6 October 2022. [Online]. Available: <https://hoclaravel.vn/chi-tiet/composer-la-gi-va-no-giup-ich-nhu-the-nao>. [Đã truy cập 30 12 2024].
- [5] Hồ Hữu Hiền, "Laragon Là Gì? Cách Cài Laravel Trên Laragon," HocLaravel, 30 July 2022. [Online]. Available: <https://hoclaravel.vn/chi-tiet/laragon-la-gi-cach-cai-laravel-tren-laragon>. [Đã truy cập 30 12 2024].