

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH HỆ THỐNG THÔNG TIN**

**XÂY DỰNG WEBSITE CHIA SẺ ĐỒ DÙNG HỌC TẬP
THÔNG MINH SỬ DỤNG PHP VÀ FRAMEWORK LARAVEL**

CBHD:	Ths. Trần Việt Hà
Sinh viên:	Nguyễn Thanh Mai
Mã số sinh viên:	2021607585

Hà Nội – 2025

NGUYỄN THANH MAI

HỆ THỐNG THÔNG TIN

LỜI CẢM ƠN

Trong hành trình hoàn thành đồ án tốt nghiệp, em đã nhận được sự hỗ trợ và giúp đỡ vô cùng quý giá từ nhiều cá nhân, cơ quan và tổ chức. Với tất cả lòng biết ơn, em xin gửi lời tri ân chân thành đến những người đã góp phần làm nên thành công của đề tài này.

Trước hết, em xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc đến các thầy, cô trường Công nghệ thông tin và truyền thông của Đại học Công nghiệp Hà Nội. Chính sự tận tình giảng dạy và những điều kiện tốt nhất mà các thầy, cô mang lại đã nâng bước em qua bốn năm học tập và cả chặng đường thực hiện đồ án đầy ý nghĩa.

Đặc biệt, em xin dành lời cảm ơn trọn vẹn nhất đến giảng viên hướng dẫn – Ths. Trần Việt Hà. Những lời khuyên chân tình, sự chỉ bảo tận tụy của thầy không chỉ là nguồn động lực lớn lao, mà còn là kim chỉ nam giúp em vượt qua những trở ngại trong quá trình nghiên cứu và hoàn thiện đề tài.

Em cũng xin gửi lời tri ân sâu sắc đến gia đình, bạn bè và đồng nghiệp – những người luôn là chỗ dựa vững chắc, không ngừng động viên, khích lệ và hỗ trợ em trên suốt chặng đường học tập cũng như trong thời gian thực hiện đồ án.

Trong suốt quá trình hoàn thành đồ án tốt nghiệp không thể tránh khỏi thiết sót, kính mong nhận được sự góp ý từ thầy cô.

Chúc các thầy, cô luôn dồi dào sức khỏe, thành công trong sự nghiệp giảng dạy và nghiên cứu, đồng thời tiếp tục thắp sáng ngọn lửa đam mê cho các thế hệ sinh viên mai sau.

Em xin chân thành cảm ơn!

Nguyễn Thanh Mai

MỞ ĐẦU

1. Tên đề tài

Xây dựng website chia sẻ đồ dùng học tập thông minh sử dụng PHP và Framework Laravel

2. Lý do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ thông tin đã và đang thay đổi sâu sắc mọi khía cạnh của đời sống, đặc biệt là trong lĩnh vực giáo dục và học tập. Các nền tảng trực tuyến không chỉ đơn thuần là công cụ hỗ trợ mà còn trở thành cầu nối quan trọng, giúp con người tiếp cận tri thức và tài nguyên một cách nhanh chóng, hiệu quả. Một trong những nhu cầu thiết thực của cộng đồng học tập hiện nay là việc kết nối giữa các cá nhân để chia sẻ và trao đổi đồ dùng học tập – từ sách vở, tài liệu đến các vật dụng hỗ trợ khác. Tuy nhiên, thực tế cho thấy vẫn còn thiếu những hệ thống tích hợp thông minh, dễ sử dụng và phù hợp với người dùng Việt Nam để đáp ứng nhu cầu này một cách toàn diện.

Bên cạnh đó, sự phát triển vượt bậc của công nghệ web đã mở ra nhiều cơ hội để xây dựng các ứng dụng thực tiễn, trong đó PHP và Framework Laravel nổi bật như một bộ đôi mạnh mẽ. PHP là ngôn ngữ lập trình phổ biến với cộng đồng phát triển lớn, dễ tiếp cận và linh hoạt, trong khi Laravel – một framework hiện đại – mang lại hiệu suất cao, khả năng bảo mật tốt và hỗ trợ phát triển nhanh chóng nhờ cấu trúc rõ ràng cùng hệ sinh thái phong phú. Sự kết hợp này không chỉ phù hợp để giải quyết bài toán thực tế mà còn là cơ hội để áp dụng những tiến bộ công nghệ mới nhất vào một sản phẩm hữu ích.

Vì vậy em chọn đề tài “Xây dựng website chia sẻ đồ học tập thông minh sử dụng PHP và Framework Laravel” để đáp ứng nhu cầu thực tế của cộng đồng mà còn là dịp để em củng cố kiến thức đã học, nâng cao kỹ năng lập trình, thiết kế hệ thống và quản lý dự án.

3. Mục tiêu của đề tài

Xây dựng một website cho phép người dùng đăng tải, tìm kiếm và chia sẻ đồ dùng một cách dễ dàng, nhanh chóng.

Tích hợp các tính năng thông minh như gợi ý sản phẩm, tìm kiếm nâng cao và quản lý thông tin người dùng.

Ứng dụng PHP và Framework Laravel để đảm bảo website hoạt động ổn định, bảo mật và dễ dàng mở rộng trong tương lai

Đóng góp một giải pháp thiết thực hỗ trợ cộng đồng học tập, đồng thời hoàn thiện kỹ năng phát triển ứng dụng web của bản thân.

4. Nội dung nghiên cứu

Nghiên cứu tổng quan về đề tài:

- Nghiên cứu về ngôn ngữ PHP và Framework Laravel
- Phân tích thiết kế
- Phân tích các use case
- Xây dựng biểu đồ use case
- Thiết kế cơ sở dữ liệu

5. Phạm vi của đề tài

Đề tài tập trung vào việc xây dựng một website với các chức năng cơ bản như đăng ký/đăng nhập, đăng tải đồ dùng học tập, tìm kiếm và quản lý thông tin cá nhân.

Sử dụng ngôn ngữ PHP và Framework Laravel làm công nghệ chính, kết hợp với cơ sở dữ liệu MySQL.

Đối tượng phục vụ chủ yếu là học sinh, sinh viên và những người có nhu cầu chia sẻ đồ dùng học tập.

Phạm vi triển khai giới hạn trong môi trường thử nghiệm, chưa mở rộng ra thực tế quy mô lớn.

6. Phương pháp thực hiện

Phương pháp nghiên cứu lý thuyết: Thu thập tài liệu, phân tích các hệ thống tương tự và học hỏi từ các nguồn tài liệu liên quan đến PHP, Laravel và phát triển web.

Phương pháp thực nghiệm: Tiến hành thiết kế, lập trình và kiểm thử từng module của website.

Phương pháp phân tích và tổng hợp: Đánh giá kết quả sau khi triển khai, từ đó rút ra ưu điểm, hạn chế và hướng cải tiến.

Sử dụng các công cụ hỗ trợ như Visual Studio Code, XAMPP, Git để phát triển và quản lý mã nguồn.

7. Bố cục của đề tài

Báo cáo được chia thành 4 chương chính:

Chương 1: Tổng quan về nội dung nghiên cứu

- Chương này cung cấp nền tảng kiến thức lý thuyết cần thiết cho việc nghiên cứu và phát triển website
- Khảo sát hệ thống: Thu thập yêu cầu từ người dùng, phân tích nhu cầu thị trường và xác định các chức năng cần thiết cho ứng dụng.

Chương 2: Phân tích thiết kế hệ thống

- Phân tích và thiết kế hệ thống:
- Phân tích yêu cầu chức năng và phi chức năng.
- Thiết kế cơ sở dữ liệu.

Chương 3: Cài đặt và kết quả thực nghiệm: Trình bày về kết quả của quá trình phát triển

Chương 4: Kiểm thử hệ thống: Trình bày phương pháp và công cụ kiểm thử thủ công, cùng với lịch trình kiểm thử cụ thể. Cung cấp các testcase chi tiết, kiểm tra các tính năng của hệ thống.

CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

1.1. Giới thiệu chung

1.2. Cơ sở lý thuyết

1.2.1. PHP

❖ Khái niệm

PHP (viết tắt của Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản chủ yếu được sử dụng để phát triển các ứng dụng web. Nó được thiết kế để nhúng trực tiếp vào mã HTML, giúp tạo ra các trang web động và tương tác.

❖ Đặc điểm

- Dễ học và sử dụng: PHP có cú pháp đơn giản và dễ hiểu, giúp những lập trình viên mới bắt đầu nhanh chóng trở nên thành thạo.
- Tích hợp sẵn với HTML: PHP được thiết kế để hoạt động tốt với HTML, cho phép lập trình viên dễ dàng nhúng mã PHP vào các trang web.
- Hiệu suất cao: PHP có hiệu suất tốt, đặc biệt là khi sử dụng các framework và công cụ hiện đại.
- Đa nền tảng: PHP có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, Linux và macOS.
- Cộng đồng lớn và dễ tiếp cận: PHP có một cộng đồng lập trình viên rộng

lớn, cung cấp nhiều tài nguyên, thư viện và framework hỗ trợ.

- Linh hoạt và mở rộng: PHP được cấu trúc một cách linh hoạt, cho phép lập trình viên dễ dàng mở rộng và tích hợp với các công nghệ khác.
- PHP thường được sử dụng để xây dựng các ứng dụng web động, quản lý nội dung, diễn đàn, blog, hệ thống quản lý học tập, TMĐT và nhiều loại ứng dụng web khác. Với sự phát triển liên tục của PHP và các framework/thư viện hỗ trợ, ngôn ngữ này vẫn giữ vị trí quan trọng trong lĩnh vực phát triển web.

❖ Ứng dụng trong dự án

PHP đóng vai trò là nền tảng chính để phát triển backend, chịu trách nhiệm:

- Sử dụng để phát triển các API RESTful, hỗ trợ giao tiếp giữa frontend và backend.
- Xử lý các nghiệp vụ phức tạp, như quản lý giỏ hàng, đặt hàng, xử lý thanh toán, và quản lý trạng thái đơn hàng.
- Xây dựng hệ thống quản trị cho phép quản lý tài khoản người dùng, quản lý đơn đặt hàng, quản lý sản phẩm và quản lý phản hồi.

1.2.2. Hệ quản trị cơ sở dữ liệu MySQL

❖ Khái niệm

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở được phát triển bởi MySQL AB, hiện thuộc sở hữu của Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc SQL (*Structured Query Language*) để quản lý và thao tác dữ liệu. Nó được thiết kế để xử lý khối lượng dữ liệu lớn và hỗ trợ nhiều người dùng đồng thời.

❖ Đặc điểm

- Hỗ trợ đa nền tảng: MySQL có sẵn cho nhiều hệ điều hành khác nhau như Linux, Windows và macOS, cho phép bạn chạy cơ sở dữ liệu trên nhiều môi trường.
- Tính sẵn sàng cao: MySQL được thiết kế để có tính sẵn sàng cao, với các tính năng như nhân bản, cân bằng tải và khả năng phục hồi dữ liệu.

- Hiệu suất cao: MySQL được tối ưu hóa để xử lý các tải công việc lớn và cung cấp các công cụ để tinh chỉnh hiệu suất, bao gồm các chỉ mục và truy vấn song song.

- MySQL là một lựa chọn phổ biến cho các ứng dụng web và doanh nghiệp, đặc biệt là khi cần một hệ quản trị cơ sở dữ liệu nhanh, đáng tin cậy và dễ sử dụng.

❖ Ứng dụng trong dự án

- MySQL là nơi lưu trữ dữ liệu chính trong các ứng dụng web, bao gồm thông tin người dùng, sản phẩm, đơn hàng và các dữ liệu liên quan khác.

- MySQL được sử dụng để thực hiện các truy vấn phức tạp và phân tích dữ liệu.

- Khi xây dựng các API RESTful, MySQL đóng vai trò làm cơ sở lưu trữ dữ liệu được truy xuất hoặc cập nhật qua các endpoint API.

- Lưu trữ dữ liệu của các bảng dữ liệu như sản phẩm, danh mục sản phẩm, tài khoản người dùng, đơn hàng, chi tiết đơn hàng, giỏ hàng, contact, blog.

1.2.3. Mô hình MVC

❖ Khái niệm

Mô hình MVC (Model-View-Controller) là một mô hình kiến trúc phần mềm được sử dụng rộng rãi trong phát triển ứng dụng. Nó giúp tách biệt logic xử lý dữ liệu (Model), giao diện người dùng (View) và quản lý sự tương tác giữa chúng (Controller). Mô hình MVC giúp cải thiện tính mô-đun, tái sử dụng và bảo trì của ứng dụng.

- Model

Chịu trách nhiệm quản lý dữ liệu và các quy trình nghiệp vụ. Model giao tiếp trực tiếp với cơ sở dữ liệu và xử lý tất cả các tương tác dữ liệu, bao gồm cả việc truy vấn và lưu trữ thông tin.

- View

Là giao diện người dùng, nơi dữ liệu được trình bày. View hiển thị dữ liệu được cung cấp bởi Model và được điều khiển bởi Controller. View không

thực hiện bất kỳ xử lý nào liên quan đến dữ liệu và chỉ tập trung vào việc hiển thị.

- **Controller**

Đóng vai trò như một trung gian giữa Model và View. Controller xử lý các yêu cầu đầu vào từ người dùng, thông qua các hành động, truy vấn dữ liệu từ Model và sau đó đẩy dữ liệu đó tới View để hiển thị.

Mô hình MVC hỗ trợ tính năng phân tách mối quan tâm, cho phép một sự phân chia rõ ràng giữa logic nghiệp vụ và giao diện người dùng, điều này làm cho ứng dụng dễ dàng bảo trì và mở rộng hơn.

❖ **Ưu điểm**

- **Phát triển độc lập:** MVC cho phép phát triển các thành phần Model, View và Controller một cách độc lập, giúp tăng hiệu quả công việc và dễ dàng trong quản lý dự án.
- **Bảo trì dễ dàng:** Do sự phân tách rõ ràng giữa các thành phần, MVC giúp việc bảo trì và nâng cấp hệ thống trở nên thuận tiện và nhanh chóng hơn.
- **Tối ưu cho SEO:** Các ứng dụng sử dụng MVC có thể dễ dàng tạo ra các URL thân thiện với SEO, từ đó cải thiện khả năng hiển thị trên các công cụ tìm kiếm.
- **Hỗ trợ phát triển song song:** MVC hỗ trợ phát triển song song, cho phép các nhóm lập trình viên làm việc trên các thành phần khác nhau của ứng dụng mà không ảnh hưởng đến nhau.
- **Kiểm thử dễ dàng:** Do mỗi thành phần của MVC có thể được kiểm thử độc lập, việc kiểm thử và debug trở nên đơn giản và hiệu quả hơn.

❖ **Nhược điểm**

- **Độ phức tạp cao:** MVC yêu cầu phải thiết kế rõ ràng và phức tạp hơn so với các mô hình đơn giản hơn, điều này có thể làm tăng đáng kể chi phí phát triển, đặc biệt là với các dự án nhỏ hoặc ngắn hạn.
- **Khó khăn trong việc học tập:** Đối với những lập trình viên mới, việc hiểu và nắm vững mô hình MVC có thể mất nhiều thời gian hơn do sự

phân tách rõ ràng giữa logic nghiệp vụ và giao diện người dùng.

- **Quản lý tài nguyên:** MVC có thể dẫn đến việc sử dụng tài nguyên không hiệu quả khi không được triển khai một cách cẩn thận, bởi vì mỗi thành phần có thể tạo ra nhiều phiên bản đối tượng, đòi hỏi nhiều tài nguyên hơn cho việc quản lý trạng thái và đối tượng.
- **Tăng thời gian phát triển:** Do tính chất độc lập của các thành phần, các dự án sử dụng MVC có thể mất thêm thời gian để tích hợp và kiểm thử toàn diện hơn các mô hình khác.
- **Khó khăn trong bảo trì:** Mặc dù mỗi thành phần có thể được bảo trì độc lập, nhưng sự phụ thuộc lẫn nhau giữa các thành phần có thể làm cho quá trình bảo trì trở nên khó khăn, đặc biệt là trong các dự án lớn hoặc phức tạp.

❖ Cách thức hoạt động của mô hình MVC

B1: Khởi tạo và Yêu cầu: Người dùng gửi yêu cầu qua trình duyệt, thường là dưới dạng một HTTP Request. Yêu cầu này được nhận bởi Controller.

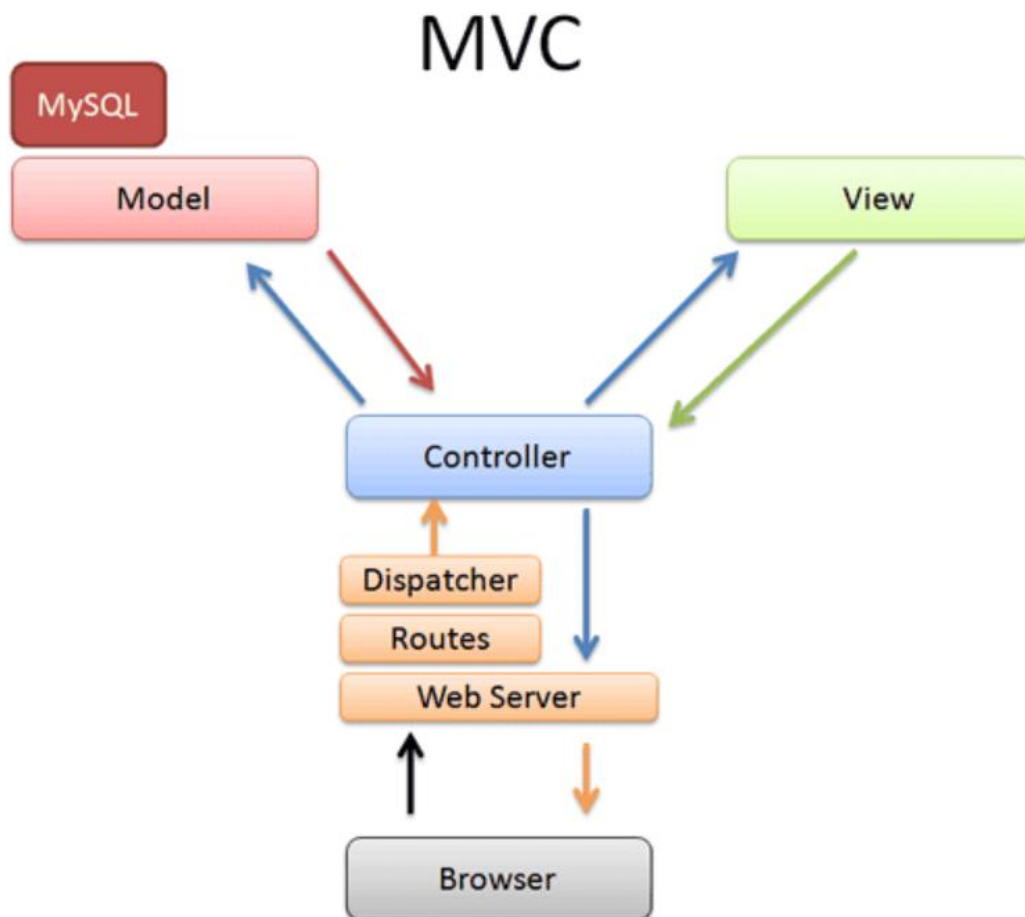
B2: Xử lý bởi Controller: Controller kiểm tra yêu cầu và quyết định liệu có cần truy xuất dữ liệu từ Model không. Nếu cần, Controller sẽ giao tiếp với Model để lấy dữ liệu cần thiết.

B3: Tương tác giữa Model và Database: Model có nhiệm vụ tương tác với cơ sở dữ liệu (database) để truy xuất hoặc cập nhật thông tin, sau đó trả dữ liệu trở lại cho Controller.

B4: Dữ liệu được gửi tới View: Sau khi nhận dữ liệu từ Model, Controller sẽ chuyển dữ liệu đến View.

B5: Hiển thị thông tin: View sử dụng dữ liệu nhận được để tạo giao diện người dùng phù hợp, sau đó gửi giao diện này trở lại trình duyệt để hiển thị cho người dùng.

B6: Kết thúc xử lý: Quá trình từ khi nhận yêu cầu đến khi hiển thị kết quả cho người dùng được hoàn tất, đồng thời đóng gói và gửi dữ liệu cuối cùng về phía client để hiển thị.



Hình: Cách thức hoạt động của mô hình MVC

1.2.4. HTML, CSS, Bootstrap và Javascript

❖ Tổng quan về HTML, CSS, Bootstrap, Javascript

HTML (HyperText Markup Language): ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng và cấu trúc trang web. HTML cung cấp các phần tử (elements) để mô tả cấu trúc của trang web, từ văn bản và hình ảnh đến các đối tượng như biểu mẫu và các liên kết.

CSS (Cascading Style Sheets): là một ngôn ngữ đánh dấu được sử dụng để định dạng và trang trí nội dung của trang web được viết bằng HTML. CSS giúp tách biệt nội dung và kiểu dáng của trang, giúp việc quản lý và cập nhật giao diện trở nên dễ dàng hơn.

Bootstrap là một framework front-end mã nguồn mở được phát triển bởi Twitter. Nó cung cấp một bộ công cụ thiết kế và mã nguồn để phát triển

nhANH chóng các trang web và ứng dụng web có giao diện người dùng đẹp mắt, linh hoạt, và tương thích trên nhiều thiết bị.

JavaScript: là một ngôn ngữ lập trình dựa trên văn bản, đa nền tảng, được sử dụng chủ yếu để tạo ra các trang web tương tác và động. JavaScript là một phần quan trọng của phát triển web và cho phép các trang web thực hiện các chức năng phức tạp như tương tác người dùng, cập nhật nội dung trang mà không cần làm mới trang, và giao tiếp với máy chủ để lấy hoặc gửi dữ liệu.

❖ Ứng dụng trong dự án

HTML, CSS dùng để xây dựng front-end phía người dùng như trang giới thiệu, liên hệ, blog, tạo giao diện thân thiện đẹp mắt, dễ sử dụng.

Bootstrap dùng trong việc xây dựng giao diện phía người quản trị như quản lý sản phẩm, quản lý tài khoản, quản lý đơn đặt hàng, quản lý phản hồi. Ngoài ra nó còn giúp xây dựng giao diện responsive, đảm bảo tương thích trên mọi thiết bị.

Javascript giúp tạo giao diện người dùng động, thay đổi nội dung, cấu trúc HTML một cách linh hoạt, phản hồi các tương tác của người dùng. Tạo các hiệu ứng tương tác như tooltip, modal, dropdown. Đảm bảo dữ liệu nhập vào hợp lệ trước khi gửi đi. Lấy dữ liệu từ server bằng AJAX

1.2.5. Framework Laravel

❖ Khái niệm

Laravel được hiểu đơn giản là một framework PHP dùng để xây dựng các ứng dụng web trên nhiều nền tảng khác nhau.

Laravel cung cấp cho nhà phát triển một thư viện khổng lồ chứa các tính năng đã được lập trình sẵn. Nhờ vào đó mà người dùng xây dựng trang web trở nên nhanh chóng và đơn giản hơn do giảm số lượng và thời gian code. Các chức năng mà Laravel cung cấp có thể dễ dàng mở rộng và bảo trì. Đồng thời, các nhà phát triển còn có thể thêm thắt chức năng vào ứng dụng phần mềm của mình một cách liền mạch nhờ vào hệ thống đóng gói module và

quản lý thật chặt.

Laravel là một framework chủ yếu hỗ trợ phát triển trên Backend, tuy nhiên nó cũng hỗ trợ một số tính năng như frontend nhưng có nhiều tính năng không khả dụng.

❖ Ưu điểm

- Cú pháp đơn giản, dễ học.
- Hỗ trợ mô hình MVC (Model-View-Controller): Giúp tổ chức mã nguồn gọn gàng, dễ bảo trì và mở rộng. Tách biệt phần hiển thị (View) và xử lý logic (Controller, Model), giúp phát triển ứng dụng dễ dàng hơn.
- Hệ thống Routing mạnh mẽ: Cung cấp hệ thống định tuyến linh hoạt giúp dễ dàng quản lý các URL. Hỗ trợ Route Middleware để kiểm soát quyền truy cập hiệu quả.
- Eloquent ORM – Truy vấn CSDL dễ dàng: giúp làm việc với cơ sở dữ liệu dễ dàng hơn so với truy vấn SQL thuần. Hỗ trợ các thao tác như CRUD, quan hệ giữa các bảng một cách tự nhiên.
- Hỗ trợ Migration & Seeding cho cơ sở dữ liệu: giúp quản lý phiên bản cơ sở dữ liệu mà không cần sửa trực tiếp. Seeding giúp dễ dàng tạo dữ liệu giả để kiểm thử ứng dụng.
- Hỗ trợ Blade Template Engine: Blade giúp tối ưu hiệu suất hiển thị giao diện với cú pháp dễ hiểu và tái sử dụng code tốt.
- Tích hợp hệ thống xác thực và phân quyền mạnh mẽ: Laravel cung cấp sẵn hệ thống xác thực (Authentication) và phân quyền (Authorization), giúp quản lý người dùng dễ dàng.
- Bảo mật tốt.
- Tích hợp Artisan CLI – Công cụ dòng lệnh mạnh mẽ: Artisan giúp tự động hóa nhiều tác vụ như tạo model, controller, migration, seeding, giúp tiết kiệm thời gian phát triển.
- Khả năng mở rộng và tích hợp tốt.

❖ Nhược điểm

- Hiệu suất chậm hơn so với PHP thuần hoặc framework nhẹ: Do có nhiều

tính năng tích hợp sẵn nên Laravel có thể chậm hơn so với các framework nhẹ như Lumen hoặc CodeIgniter. Cần tối ưu bộ nhớ cache và sử dụng công cụ như Redis để cải thiện hiệu suất.

- Tiêu tốn tài nguyên nhiều hơn.
- Độ phức tạp cao với dự án lớn.
- Cập nhật thường xuyên, có thể gây khó khăn khi nâng cấp.
- Không phù hợp cho ứng dụng thời gian thực (Real-time) nếu không có hỗ trợ Redis/WebSockets.

❖ Vòng đời của 1 request trong Laravel

1. Client gửi Request

- Người dùng (hoặc một ứng dụng) gửi yêu cầu đến hệ thống Laravel thông qua trình duyệt, ứng dụng di động hoặc API Client.
- Request có thể là GET, POST, PUT, DELETE tùy theo hành động cần thực hiện.

2. Tiếp nhận Request tại Web Server

- Request đầu tiên sẽ đi qua Web Server (Apache/Nginx).
- Web Server kiểm tra URL và chuyển hướng request đến Laravel thông qua file `public/index.php`.

3. Xử lý trong Laravel

a. Middleware

- Laravel trước tiên sẽ đi qua lớp Middleware để kiểm tra request, ví dụ:
 - Xác thực (Authentication)
 - Phân quyền (Authorization)
 - Xử lý CORS
 - Logging request

b. Route & Controller

- Request được chuyển đến Route phù hợp (`routes/web.php` hoặc `routes/api.php`).
- Route điều hướng request đến Controller xử lý logic nghiệp vụ.

c. Xử lý Business Logic

- Controller có thể gọi các Service, Repository hoặc xử lý trực tiếp.
- Nếu cần dữ liệu từ Database, Controller sẽ gọi đến Model hoặc Repository.

d. Truy vấn Database qua Model (Eloquent ORM)

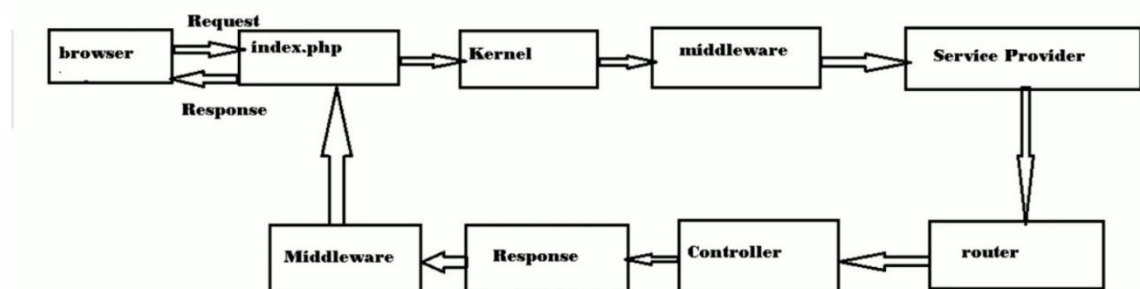
- Laravel sử dụng Eloquent ORM để tương tác với Database.
- Các truy vấn có thể bao gồm:
 - Lấy dữ liệu (SELECT)
 - Thêm mới (INSERT)
 - Cập nhật (UPDATE)
 - Xóa (DELETE)

4. Phản hồi Response về Client

- Sau khi xử lý xong, Laravel trả kết quả về dạng JSON hoặc HTML.
- Response có thể chứa dữ liệu, mã lỗi hoặc thông báo từ hệ thống.
- Response đi ngược lại qua Middleware để kiểm tra lần cuối trước khi gửi về Client.

5. Client nhận Response và hiển thị kết quả

- Trình duyệt hoặc ứng dụng xử lý Response để hiển thị thông tin cho người dùng.



Laravel Request Life Cycle

Hình: Vòng đời request trong Laravel

1.3. Khảo sát cục bộ

1.3.1. Mục tiêu

- **Thu nhập nhu cầu thực tế:** Xác định nhu cầu sử dụng và chia sẻ đồ dùng

học tập thông minh của người dùng để định hướng phát triển website

- **Đánh giá thói quen và khó khăn:** Tìm hiểu thói quen sử dụng đồ dùng học tập, các vấn đề gặp phải trong việc tìm kiếm hoặc chia sẻ tài nguyên học tập hiện nay.

- **Xác định tính khả thi:** Đánh giá mức độ đón nhận của người dùng đối với một nền tảng trực tuyến phục vụ chia sẻ đồ dùng học tập thông minh.

- **Cung cấp dữ liệu thiết kế:** Thu thập thông tin để tối ưu hóa giao diện, tính năng và trải nghiệm người dùng (UX/UI) cho website.

1.3.2. Đối tượng khảo sát

- **Học sinh và sinh viên:** Nhóm người dùng chính, thường xuyên cần các đồ dùng học tập như sách, tài liệu, thiết bị học tập thông minh (máy tính bảng, bút cảm ứng, v.v.).

- **Giáo viên:** Những người có thể đóng góp tài liệu, ý tưởng hoặc sử dụng website để hỗ trợ giảng dạy.

- **Phụ huynh:** Đối tượng gián tiếp, có thể tham gia chia sẻ hoặc tìm kiếm đồ dùng học tập cho con em mình.

- **Nhà phát triển và chuyên gia công nghệ:** Để lấy ý kiến về tính khả thi kỹ thuật và các yêu cầu hệ thống của website..

1.3.3. Phương pháp

Kế hoạch phỏng vấn người dùng

Kế hoạch phỏng vấn	
Người được hỏi: Nguyễn Thị A	Người hỏi: Nguyễn Thanh Mai
Địa chỉ: Trường đại học Công nghiệp Hà Nội	Thời gian hẹn: 14h - 22/03/2025
	Thời gian bắt đầu: 14h15
	Thời gian kết thúc: 15h15
Đối tượng: Sinh viên	Các yêu cầu đòi hỏi: Người có nhu cầu tìm kiếm, trao đổi đồ dùng học tập

Chương trình	Thời gian phỏng vấn
• Giới thiệu	3 phút
• Tổng quan về dự án	3 phút
• Tổng quan về phỏng vấn	3 phút
• Chủ đề sắp đề cập (xin phép được ghi âm)	3 phút
• Chủ đề 1: Hành vi và nhu cầu của người dùng	10 phút
• Chủ đề 2: Đánh giá mức độ tin tưởng của mô hình chia sẻ đồ dùng học tập	10 phút
• Chủ đề 3: Các tính năng mong muốn trên website	10 phút
• Chủ đề 4: Thanh toán và chính sách bảo vệ quyền lợi	10 phút
• Chủ đề 5: Đối tượng sử dụng và mở rộng cộng đồng	5 phút
	Tổng thời gian dự kiến: 57 phút

Phiếu phỏng vấn người dùng

PHIẾU PHỎNG VẤN	
Dự án: xây dựng website chia sẻ đồ dùng học tập thông minh	
Người được hỏi: Nguyễn Thị A	Người hỏi: Nguyễn Thanh Mai Ngày: 22/03/2025
Câu hỏi	Ghi chú:
Câu hỏi 1: Bạn thường mua hoặc mượn đồ dùng học tập từ đâu? (Hiệu sách, chợ online, bạn bè, trường học)	Trả lời
Câu hỏi 2: Bạn có thường xuyên chia sẻ hoặc cho mượn đồ dùng học tập	Trả lời

không? Vì sao?	
Câu hỏi 3: Bạn có từng gặp khó khăn khi tìm kiếm hoặc mua dụng cụ học tập không?	Trả lời
Câu hỏi 4: Bạn có sẵn sàng sử dụng nền tảng này để mượn hoặc cho mượn đồ không?	Trả lời
Câu hỏi 5: Điều gì có thể khiến bạn tin tưởng hoặc lo lắng khi sử dụng nền tảng	Trả lời
Câu hỏi 6: Bạn mong đợi những tính năng nào? (Tìm kiếm thông minh, chat, đặt cọc, đánh giá,..)	Trả lời
Câu hỏi 7: Bạn có nghĩ rằng cần có chính sách đảm bảo hoặc bảo hiểm khi mượn đồ không?	Trả lời
Câu hỏi 8: Bạn có muốn nền tảng hỗ trợ xử lý tranh chấp nếu có vấn đề phát sinh không?	Trả lời
Câu hỏi 9: Bạn nghĩ ai sẽ quan tâm đến nền tảng này nhất? (Học sinh, sinh viên, giáo viên, phụ huynh,...)	Trả lời

1.4. Hoạt động của hệ thống

- Về phía người dùng:

✓ Người dùng đăng ký và đăng nhập:

- Tạo tài khoản với email/số điện thoại hoặc tài khoản Google/Facebook.
- Xác thực tài khoản để đảm bảo tính bảo mật.

✓ Đăng tải và tìm kiếm đồ dùng học tập

- Người cho mượn đăng sản phẩm với thông tin chi tiết (mô tả, hình ảnh, điều kiện mượn).
- Người mượn tìm kiếm và lọc sản phẩm theo danh mục, địa điểm, giá cả, đánh giá...

✓ **Quy trình mượn/trả đồ**

- Người mượn gửi yêu cầu đến người cho mượn.
- Hai bên thỏa thuận về thời gian, địa điểm giao nhận.
- Hệ thống có thể yêu cầu đặt cọc hoặc xác nhận thông tin trước khi giao dịch.

✓ **Đánh giá và phản hồi**

- Sau khi hoàn tất giao dịch, cả hai bên có thể đánh giá nhau để tăng độ tin cậy.
- Hệ thống có thể cảnh báo hoặc hạn chế người dùng vi phạm quy định.

- Về phía người quản trị (admin):

✓ **Quản lý người dùng**

- Kiểm tra danh tính, thông tin người dùng khi đăng ký (nếu có xác thực).
- Cấp quyền hoặc hạn chế người dùng khi cần thiết

✓ **Quản lý sản phẩm, quản lý nội dung và hoạt động cộng đồng**

- Xác minh thông tin và hình ảnh đồ dùng học tập trước khi hiển thị trên nền tảng.
- Loại bỏ sản phẩm không phù hợp hoặc vi phạm chính sách.
- Thêm, chỉnh sửa, xóa danh mục sản phẩm để tối ưu tìm kiếm.

✓ **Giám sát và xử lý giao dịch**

- Kiểm tra giao dịch đang diễn ra, đảm bảo đúng quy trình.
- Hỗ trợ khi có tranh chấp hoặc khiếu nại.
- Giám sát các khoản thanh toán, đặt cọc.
- Hỗ trợ hoàn tiền hoặc xử lý vi phạm khi có yêu cầu.

1.5. Phân tích yêu cầu

1.5.1. Yêu cầu của người dùng

Đối với người mượn và người cho mượn website mang đến cho người dùng tính minh bạch, đơn giản và dễ sử dụng, đăng tin, trao đổi đồ dùng dễ dàng.

Đối với người quản trị mang đến cho họ đầy đủ nhu cầu và tính năng để quản lý sản phẩm, đồ dùng 1 cách khoa học và chính xác hơn.

1.5.2. Yêu cầu về chức năng

✓ Các chức năng chính của hệ thống:

- Mượn đồ
- Đăng tải thông tin đồ dùng
- Tìm kiếm
- Đánh giá và phản hồi
- Quản lý người dùng
- Quản lý sản phẩm
- Quản lý nội dung

✓ Yêu cầu phi chức năng

- Giao diện dễ dùng.
- Hệ thống được bảo mật khỏi sự truy cập trái phép.
- Hệ thống phải có khả năng xử lý số lượng người dùng cần thiết mà không có bất kỳ sự suy giảm nào về hiệu suất.
- Hệ thống phải đáng tin cậy và đáp ứng các yêu cầu của người sử dụng.
- Hệ thống phải dễ sử dụng và dễ hiểu.
- Hệ thống phải tương thích với các hệ thống khác.

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Biểu đồ use case

2.1.1. Biểu đồ use case tổng quát

2.1.2. Biểu đồ use case phân rã