

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Bài tập lớn

LẬP TRÌNH WEB

Đề tài:

Hệ thống bán điện thoại

GVHD: Nguyễn Hữu Hiếu

SV thực hiện:	Lê Nguyễn Hải Đăng	2113176
	Phan Nguyễn Xuân Lộc	2113971
	Lê Hoàng Anh Vũ	2115319
	Lê Phan Quốc Vũ	2115321
Email: loc.phan2113971@hcmut.edu.vn		

TP. HỒ CHÍ MINH, THÁNG 12/2023

Mục lục

1	Giới thiệu về tài	3
2	Cơ sở lý thuyết	3
2.1	ReactJS	3
2.2	Redux	5
2.3	HTML/CSS (Bootstrap)	7
2.4	Javascript (jQuery)	7
2.5	PHP	8
2.6	MySQL	9
2.7	API	10
2.8	Bảo mật	11
2.9	SEO	14
3	Thiết kế ứng dụng	15
3.1	Cơ sở dữ liệu	15
3.1.1	Accounts	15
3.1.2	Aboutimg	15
3.1.3	Carousel	16
3.1.4	CommentDB	16
3.1.5	CommentNews	16
3.1.6	Contact	16
3.1.7	ContactPage	17
3.1.8	News	17
3.1.9	Service	17
3.1.10	Statistic	17
3.1.11	Product	18
3.1.12	Tbl-Product	18
3.1.13	Store	18
3.1.14	Team	18
3.1.15	Cart	19
3.1.16	Testimonials	19
3.2	UseCase cho toàn bộ hệ thống	19
3.3	Các thao tác để thực hiện một số chức năng của Website	21
3.3.1	Đăng nhập	21
3.3.2	Mua hàng	22
3.3.3	Dánh giá về sản phẩm	23
3.3.4	Đăng bài báo của Admin	24
3.4	Cấu trúc mã nguồn	25
3.4.1	Font-end	25
3.4.2	Back-end	26
3.4.3	Mô hình Model View Controller(MCV)	27
4	Các bước nhập cơ sở dữ liệu trên phpMyAdmin.	28
5	Hiện thực	30
5.1	Người dùng khi chưa đăng nhập	30
5.2	Đăng kí	30
5.3	Đăng nhập	31
5.4	User sau khi đăng nhập	31
5.5	Tìm kiếm mặt hàng của trang Web	32
5.6	Chi tiết sản phẩm	32



5.7	Giỏ hàng	33
5.8	Tiến hành thanh toán	33
5.9	Dánh giá của người dùng	34
5.10	Danh sách đơn hàng được đặt	34
5.11	Tin tức	35
5.12	Bình luận về tin tức	35
5.13	Liên hệ	36
5.14	Địa chỉ của Shop	36
5.15	Người dùng chỉnh sửa thông tin các nhân	37
5.16	Người dùng chỉnh sửa mật khẩu	37
5.17	Trang quản lý của Admin	38
6	Cách thức cài đặt	38
6.1	Về Front-end	38
6.2	Về Back-end	38
	Tài liệu	39



1 Giới thiệu đề tài

Các phương thức mua bán trực tuyến đang dần thay thế các phương thức mua bán truyền thống trong thời đại ngày nay, khi công nghệ và khoa học ngày càng trở nên phổ biến. Chúng ta hoàn toàn có thể sở hữu cho mình bất kỳ món đồ nào ở bất kỳ địa điểm nào mà không cần đến trực tiếp đại lý bán hàng đó với một thiết bị có kết nối Internet. Việc mua hàng trực tuyến rất tiện lợi vì nó di động và linh hoạt về thời gian và không gian. Từ sau đại dịch COVID-19, chúng ta dần nhận thấy được việc mua sắm trực tuyến trở nên rất quan trọng trong cuộc sống của mỗi người. Tăng trưởng nhanh chóng của các ứng dụng thương mại điện tử và số lượng truy cập trong thời gian qua là bằng chứng rõ ràng nhất cho sức hút của lĩnh vực này.

Nhìn chung, các mặt hàng được bán trực tuyến rất đa dạng và phong phú, từ giày dép, quần áo cho đến các sản phẩm điện tử như laptop và điện thoại. Ngoài ra, việc tham khảo dịch vụ trực tuyến cũng rất phổ biến. Trong bối cảnh dịch bệnh hiện nay, việc mua hàng hóa trực tuyến giúp giảm thời gian di chuyển và hạn chế khả năng tiếp xúc với nhiều người.

Nhóm chúng em đã quyết định thiết kế một trang web bán điện thoại di động có tên VietPhone trong khuôn khổ bài tập lớn về lập trình web, trong đó đề tài là thiết kế giao diện và xây dựng các tính năng cơ bản cho trang web của một doanh nghiệp. Mục tiêu của trang web này là tiếp cận khách hàng của các cửa hàng. Mục đích của trang web là cung cấp cho khách hàng của mình một phương tiện mua tham khảo điện thoại vừa tiện lợi vừa dễ dàng hơn. Ngoài ra, trang web giúp quản lý kinh doanh dễ dàng hơn và tiếp cận nhiều đối tượng khách hàng hơn mà không bị cản trở về thời gian hoặc địa điểm. Trang web tích hợp các tính năng cần thiết của một trang web thương mại điện tử để tối ưu hóa trải nghiệm mua sắm của người dùng.

2 Cơ sở lý thuyết

Mô tả các thư viện, công nghệ

2.1 ReactJS

React là một thư viện javascript cho phép ta xây dựng UI của một ứng dụng web một cách linh hoạt, hiệu quả theo kiểu lập trình khai báo (declarative programming) nhằm tổ chức các thành phần trong trang web thành các component. Việc tổ chức các thành phần trang web thành các component cho phép chúng ta gom các logic xử lý một phần nhỏ của giao diện lại và tái sử dụng lại, hoặc mở rộng thêm chức năng của nó theo kiểu bao gồm (composition), một component lớn sẽ bao gồm nhiều component nhỏ, tự mỗi component sẽ tập trung giải quyết vấn đề riêng của nó (nguyên lý do one thing good). Chính khả năng tái sử dụng này là điều làm React trở nên nổi tiếng. Mỗi component sẽ tự quản lý state của riêng nó và nhận data từ các component khác truyền vào thông qua các props, và mỗi component sẽ chỉ render lại khi state của riêng nó thay đổi, điều này giúp trang web của ta giảm bớt các lần re-render không cần thiết.



Lý do nhóm sử dụng ReactJS để thực hiện Frontend



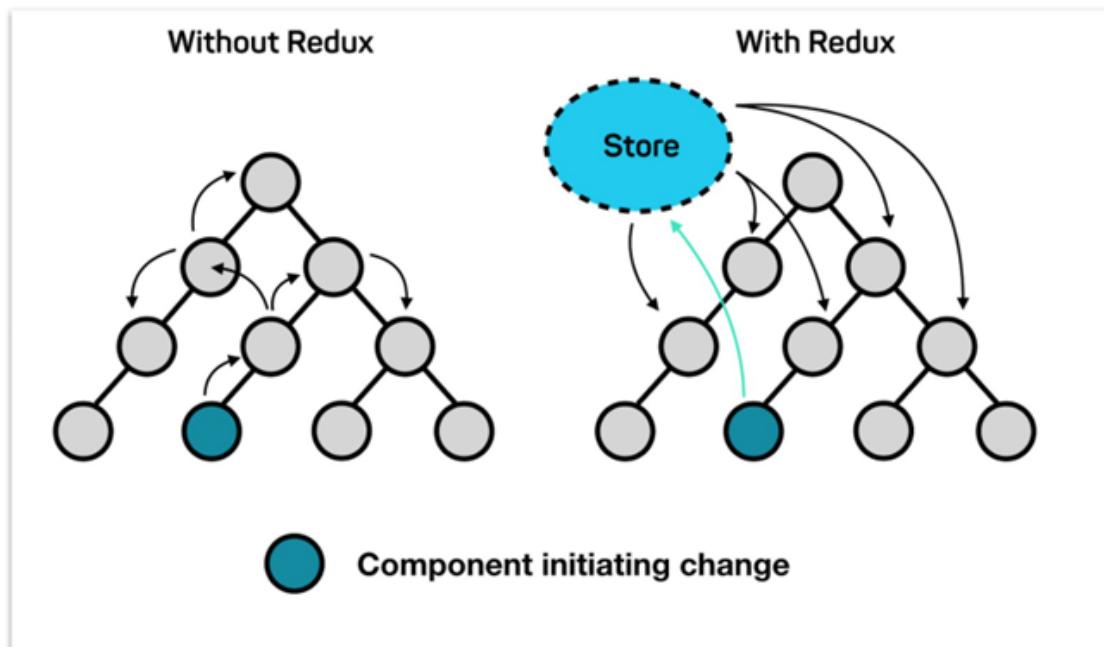
Hình 1: Thư viện React.

- Đơn giản: ReactJs tỏ ra dễ hiểu và dễ dùng hơn so với việc viết web truyền thống với HTML,CSS và manipulate các thành phần định nghĩa trên giao diện qua các hàm access vào DOM. React sử dụng ngôn ngữ JSX (một phiên bản mở rộng của JS cho phép nhúng syntax HTML vào) nên ta có thể xây dựng trang web hoàn toàn chỉ với JavaScript.
- Khả kiểm tra: Việc build ứng dụng web theo component cho phép ta viết test theo từng component. Khi từng component nhỏ được test, ta hoàn toàn có thể yên tâm sử dụng lại các component này để build một component khác lớn hơn.
- Flow dữ liệu dễ đoán: React sử dụng kiến trúc flux chỉ cho phép truyền dữ liệu từ component cha xuống các component con của nó (vẫn có cách để truyền ngược lại thông qua các hàm nhưng đó là một bad practice và React không khuyến khích điều này). Điều này khiến luồng dữ liệu trong ứng dụng của ta trở nên dễ đoán hơn và dễ debug khi ứng dụng của ta gặp lỗi.
- Cộng đồng lớn: ReactJs là thư viện Front-End JavaScript nổi tiếng nhất hiện nay. Do đó, sự hỗ trợ của cộng đồng là rất tốt khi ta gặp vấn đề lập trình. Bên cạnh đó, các thư viện, các hooks được phát triển và duy trì bởi hàng triệu developer trên toàn thế giới giúp ta dễ dàng tìm được đúng thư viện hữu ích cho bất kỳ chức năng nào của ứng dụng chúng ta.

Một số hạn chế của React:

- Khó khăn trong quản lý component: điểm yếu này chỉ là vấn đề khi codebase của phần mềm trở nên quá lớn và ta có quá nhiều component và việc tìm đúng component để sử dụng trở nên tốn thời gian hơn. Nhưng nếu ta viết lại component mới thì có khả năng component của ta bị trùng lặp về chức năng với một component đã có, lại làm codebase trở nên to hơn.
- Việc pass prop liên tục: khi cây component của ta trở nên phức tạp và có quá nhiều level, các component con ở nút lá có thể cần phải biết đến state của một component ở rất cao

so với nó. Điều này bắt buộc người lập trình phải truyền prop liên tục từ component cao xuống tận nút lá (React gọi hiện tượng này là drop-drilling), mặc dù là các component trung gian không hề đến luồng dữ liệu này. Điều này gây nên sự phức tạp không cần thiết cho ứng dụng. May mắn thay, các thư viện quản lý state như Recoil hay Redux giúp ta hoàn toàn khắc phục được vấn đề này.



Hình 2: Flow hoạt động của ứng dụng khi kết hợp với Redux

2.2 Redux

Tuy React là một thư viện mature và được cộng đồng lập trình web hưởng ứng rất nhiệt tình (React là thư viện lập trình Front-End nổi tiếng nhất hiện nay), React vẫn còn tồn tại một số nhược điểm, những điểm cần cải thiện. Điều này đặc biệt đúng khi codebase dần trở nên rộng lớn hơn và chúng ta có quá nhiều component để quản lý và các component lồng nhau tạo thành một cây component có chiều cao quá lớn. Như đã trình bày ở trên, khi một component ở nút lá quan tâm đến một component ở các nút nội cao hơn (hoặc tệ hơn cả là ở nút lá của cây component) thì chúng ta sẽ rơi vào trường hợp "props-drilling". Đây là một vấn đề lớn cả về mặt quản lý code lẫn về performance của ứng dụng (Khi được truyền nhiều tầng này thay đổi thì React sẽ render lại tất cả các component mà props này được truyền qua). Để giải quyết vấn đề này, thư viện Redux đã ra đời.

Redux là thư viện quản lý state cho các ứng dụng web (chứ không chỉ dùng chung cho các ứng dụng React như nhiều người thường làm tưởng). Redux kết hợp cực tốt với React, Redux giải quyết phần nào vấn đề "props-drilling" của React và hỗ trợ người lập trình debug ứng dụng của họ một cách dễ dàng hơn với việc gom tắt cả các trạng thái, các action - hành động và các reducer xử lý các action làm thay đổi trạng thái này vào một bộ chứa duy nhất mà Redux gọi là store. Các component React sẽ chỉ cần subscribe trực tiếp vào các state ở store và chỉ re-render

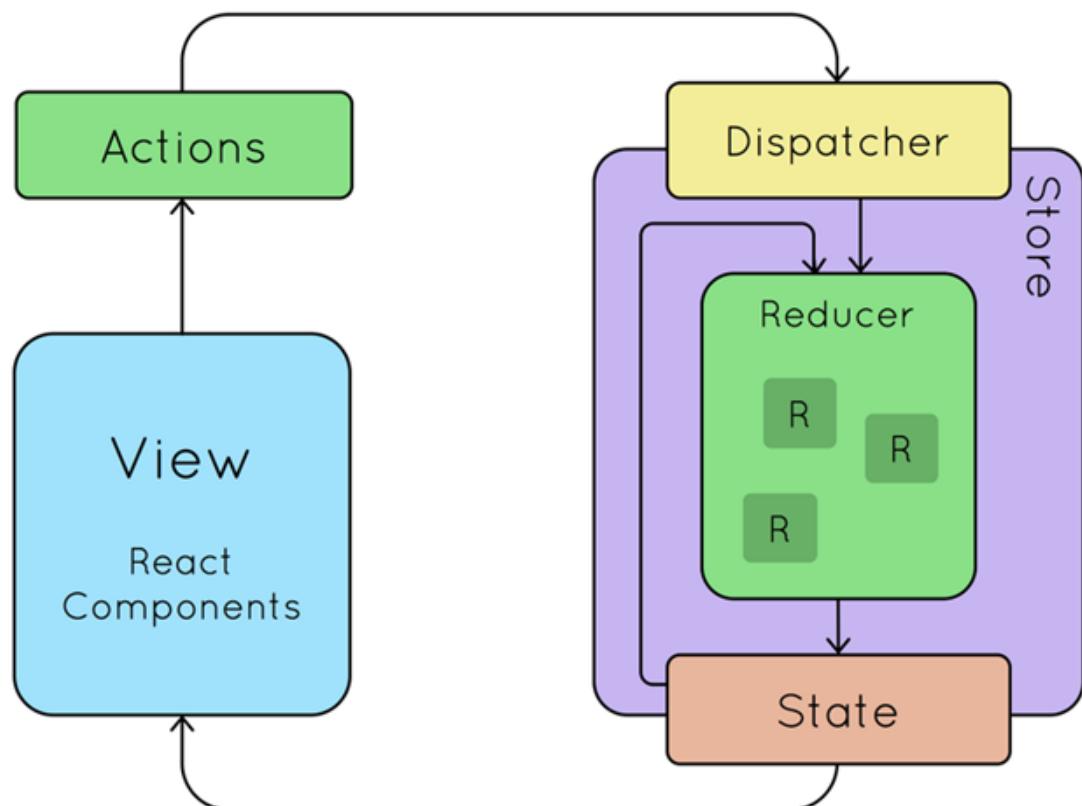
khi các state này thay đổi.

Như ta thấy ở **hình 2**, các component trong React sẽ subscribe thẳng vào state mà chúng quan tâm ở store, store trở thành Single Source Of Truth của ứng dụng nên việc debug sẽ trở nên dễ dàng hơn vì mọi sự thay đổi trạng thái ở ứng dụng đều có thể quan sát được thông qua dev tool của Redux.

MUI định nghĩa 3 theme lần lượt là sáng, tối, và normal với các màu sắc mặc định đã được chọn lựa kĩ càng cùng với các style mặc định về margin, padding và box-shadow. Tuy nhiên, ta hoàn toàn có thể tự định nghĩa theme mới cho ứng dụng với hàm createTheme tùy biến hoàn toàn hoặc chỉ một phần của theme mặc định.

Cơ chế hoạt động của một ứng dụng React - Redux Một ứng dụng React-Redux sẽ có luồng hoạt động cơ bản như sau:

- Dựa các state dùng chung và định nghĩa các action có thể có lên các state này lên store redux
- Định nghĩa các Reducer để xử lý các action được dispatch đến store của Redux
- Các component React sẽ subscribe vào các state ở store thông qua các hàm selector



Hình 3: Redux



Redux có 3 thành phần chính là Store, Action và Reducer.

- Store: Đây là nơi tập trung các state dùng chung của ứng dụng, nó là duy nhất. Các component có thể lấy ra dữ liệu từ đây, cập nhật state và lắng nghe sự thay đổi của các state.
- Action: Đây là các object sẽ được gửi vào store để cập nhật lại store. Cấu trúc của một action thường sẽ gồm 2 phần là type và payload. Phần type sẽ định nghĩa được đây là action loại nào để Redux chọn đúng Reducer mà xử lý. Phần payload là dữ liệu được cấp thêm mô tả giúp Reducer cập nhật lại store.
- Reducer: Đây là các hàm sẽ thực thi việc cập nhật lại store. Mỗi reducer sẽ có input là state hiện tại và một action. Khi reducer nhận được một action, reducer sẽ tính toán và cho ra State mới, cập nhật vào store.

Vì Redux chỉ cho phép các Reducer sử dụng các phương thức trong lập trình hàm (cụ thể là các hàm map, filter, reduce,...) để trả về một state mới mà không cho phép người dùng modify một state trực tiếp (mà thuật ngữ được dùng mô tả hiện tượng này là side-effect) lên store của Redux nên ứng dụng của ta sẽ tránh được các lỗi khó hiểu gây ra bởi việc một state nào đó bị thay đổi nằm ngoài sự theo dõi của người lập trình. Thật vậy, một điều tuyệt vời khi sử dụng Redux đó là DevTool của Redux hỗ trợ time-travel debugging cho phép người lập trình theo dõi sát sao mọi thay đổi của mọi state ở store. Từ đó, quá trình gỡ lỗi tốn ít thời gian hơn và thời gian hoàn thiện sản phẩm cũng được rút ngắn.

2.3 HTML/CSS (Bootstrap)

HTML (HyperText Markup Language): HTML là ngôn ngữ đánh dấu được sử dụng để xây dựng cấu trúc và nội dung của trang web. Chúng ta sử dụng HTML để định nghĩa các phần tử trên trang như đoạn văn bản, hình ảnh, liên kết, biểu mẫu, và nhiều thành phần khác. HTML cung cấp khung chứa cho các thông tin trên trang web.

CSS (Cascading Style Sheets): CSS là ngôn ngữ được sử dụng để định dạng và trình bày các phần tử HTML. Nó cho phép bạn thiết lập các kiểu, màu sắc, kích thước, vị trí và hiệu ứng cho các phần tử trên trang. CSS giúp tạo ra giao diện thẩm mỹ, tùy chỉnh và dễ đọc cho người dùng.

Nhóm sử dụng CSS Framework Bootstrap v5.0.2 được phát hành vào năm 2021. Bootstrap cung cấp một bộ các công cụ, giao diện và phương thức tạo layout và thiết kế web nhanh chóng. Nó bao gồm các tính năng như grid system, responsive design, typography, form controls, buttons, navigation và nhiều hơn nữa. Bootstrap đã trở thành một trong những Framework CSS nổi tiếng và được ưa chuộng nhất thế giới nhờ vào sự gọn nhẹ và dễ dàng sử dụng.

Một vài ưu điểm của Bootstrap có thể kể tới như tính tương thích cao đối với hầu hết các trình duyệt hiện tại; giao diện đầy đủ, hỗ trợ đầy đủ responsive design; dễ dàng tùy biến. Tuy vậy, Bootstrap vẫn còn tồn tại một số hạn chế như chưa có đủ các thư viện cần thiết nếu ta muốn phát triển web về lâu dài, và trên hết việc sử dụng Bootstrap quá nhiều dẫn đến hậu quả chúng ta có thể bị lệ thuộc vào nó mà hạn chế tính sáng tạo khi thiết kế giao diện.

2.4 Javascript (jQuery)

JavaScript: JavaScript là ngôn ngữ lập trình được sử dụng để thêm tính năng tương tác động cho trang web. Nó có khả năng tương tác với các phần tử HTML và CSS, điều khiển các hành



vi như thao tác người dùng, thay đổi nội dung động mà không cần tải lại trang web, và giao tiếp với máy chủ bằng AJAX. JavaScript cho phép xây dựng các ứng dụng web phức tạp và có tính tương tác cao mà HTML trước đây chưa thể mang lại được.

jQuery là một thư viện JavaScript phổ biến được sử dụng trong phát triển phía frontend của các ứng dụng web. Với mục tiêu đơn giản hóa việc tương tác với DOM (Document Object Model) và xử lý sự kiện trên trang web, jQuery đã giúp rút ngắn mã nguồn và tối ưu hóa quá trình lập trình trong nhiều năm. jQuery cung cấp cú pháp ngắn gọn để truy cập và thao tác các phần tử trong DOM, cho phép thay đổi nội dung, thuộc tính và kiểu dáng của chúng một cách dễ dàng. Nó cũng tương tác với sự kiện như nhấp chuột, nhập liệu và thao tác trên trang.

Một trong những điểm mạnh của jQuery là tích hợp AJAX, cho phép gửi và nhận dữ liệu từ máy chủ mà không cần phải tải lại trang. Điều này tạo ra các ứng dụng web động và tương tác mượt mà. Tuy nhiên, do sự phát triển của các tiêu chuẩn web và việc sử dụng JavaScript nguyên thuỷ được cải thiện, vai trò của jQuery đã giảm đi một phần. Nhiều tính năng mà trước đây chỉ có thể thực hiện bằng jQuery bây giờ có thể được thực hiện bằng cách sử dụng JavaScript nguyên thuỷ hoặc các thư viện và framework khác.

2.5 PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình phía server phổ biến và mạnh mẽ, thường được sử dụng để xây dựng các ứng dụng web động và tương tác với cơ sở dữ liệu.

Ưu điểm:

- Dễ học và sử dụng: PHP có cú pháp đơn giản và dễ học, khiến nó trở thành lựa chọn tốt cho người mới bắt đầu phát triển web. Nó tương tự như cú pháp kiểu C và có một cộng đồng lớn các nhà phát triển đóng góp vào tài liệu mở rộng của nó, giúp dễ dàng tìm kiếm tài nguyên và hỗ trợ.
- Cộng đồng và hệ sinh thái lớn: PHP có một cộng đồng các nhà phát triển lớn và tích cực, điều đó có nghĩa là có rất nhiều thư viện, khung và công cụ có sẵn để phát triển PHP. Các framework PHP phổ biến như Laravel, Symfony và CodeIgniter cung cấp các giải pháp mạnh mẽ và giàu tính năng để xây dựng các ứng dụng web một cách hiệu quả.
- Khả năng tương thích đa nền tảng: PHP là ngôn ngữ kịch bản đa nền tảng, có nghĩa là nó có thể chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, macOS, Linux và nhiều hệ điều hành khác. Điều này làm cho nó rất linh hoạt và cho phép các nhà phát triển khai thác các ứng dụng PHP trên các môi trường lưu trữ khác nhau.
- Tích hợp cơ sở dữ liệu: PHP có khả năng tích hợp mạnh mẽ với cơ sở dữ liệu, bao gồm MySQL, PostgreSQL, SQLite, v.v. Điều này giúp dễ dàng xây dựng các ứng dụng web động với các chức năng dựa trên cơ sở dữ liệu, chẳng hạn như xác thực người dùng, hệ thống quản lý nội dung và giải pháp thương mại điện tử.
- Chu kỳ phát triển nhanh: PHP được biết đến với chu kỳ phát triển nhanh, cho phép các nhà phát triển nhanh chóng xây dựng các ứng dụng web. PHP hỗ trợ nhiều công cụ phát triển web và IDE (Môi trường phát triển tích hợp), có thể nâng cao hơn nữa năng suất và tăng tốc quá trình phát triển.



Nhược điểm:

- Hỗ trợ hạn chế cho dữ liệu phi quan hệ: MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ, có nghĩa là nó không được tối ưu hóa để xử lý dữ liệu phi quan hệ, chẳng hạn như dữ liệu phân cấp hoặc dựa trên tài liệu. Mặc dù MySQL cung cấp hỗ trợ lưu trữ và truy xuất dữ liệu phi quan hệ, nhưng nó có thể không hiệu quả hoặc giàu tính năng như các cơ sở dữ liệu chuyên dụng khác được thiết kế cho dữ liệu phi quan hệ, chẳng hạn như MongoDB hoặc Cassandra.
- Thiếu các tính năng bảo mật nâng cao: Mặc dù MySQL cung cấp các tính năng bảo mật cơ bản như xác thực người dùng và kiểm soát truy cập, nhưng nó có thể thiếu một số tính năng bảo mật nâng cao có sẵn trong các cơ sở dữ liệu hiện đại khác. Ví dụ: các tính năng như mã hóa dữ liệu tự động, kiểm tra tích hợp hoặc kiểm soát truy cập chi tiết có thể yêu cầu cấu hình bổ sung hoặc plugin của bên thứ ba.
- Khó tiếp cận cho các tính năng nâng cao: Mặc dù MySQL rất dễ bắt đầu với các hoạt động cơ bản, nhưng nó có thể có đường cong học tập dốc hơn khi nói đến các tính năng nâng cao như tối ưu hóa hiệu suất, điều chỉnh cơ sở dữ liệu và tối ưu hóa truy vấn nâng cao. Kiến thức chuyên sâu về nội bộ cơ sở dữ liệu và kỹ thuật tối ưu hóa truy vấn có thể được yêu cầu để đạt được hiệu suất và khả năng mở rộng tối ưu.

2.6 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở rất phổ biến và được sử dụng rộng rãi trong các ứng dụng web và phần mềm. Dưới đây là một số ưu nhược điểm của MySQL:

Ưu điểm:

- Miễn phí và mã nguồn mở: MySQL là một sản phẩm mã nguồn mở, điều này có nghĩa là bạn có thể sử dụng, tùy chỉnh và phân phối nó mà không cần phải trả bất kỳ phí nào.
- Tốc độ và hiệu suất tốt: MySQL đã được tối ưu hóa để xử lý các truy vấn cơ sở dữ liệu nhanh chóng và hiệu quả. Điều này làm cho nó trở thành một lựa chọn phù hợp cho các ứng dụng có yêu cầu về tốc độ cao.
- Hỗ trợ nhiều loại dữ liệu: MySQL hỗ trợ nhiều loại dữ liệu như số, chuỗi, ngày tháng, hình ảnh và nhiều loại dữ liệu khác. Điều này giúp phù hợp với nhiều loại ứng dụng khác nhau.
- Quản lý đa người dùng và phân quyền: MySQL cung cấp khả năng quản lý người dùng và phân quyền, cho phép bạn kiểm soát quyền truy cập vào dữ liệu và các thao tác trên cơ sở dữ liệu.

Nhược điểm:

- Khả năng mở rộng hạn chế: Mặc dù MySQL có thể xử lý tải lớn, nhưng khi ứng dụng của bạn mở rộng và phải đổi mới với lượng dữ liệu lớn và truy vấn phức tạp, ta có thể gặp khó khăn về mặt hiệu suất so với các hệ cơ sở dữ liệu chuyên sâu hơn.
- Không hỗ trợ một số tính năng cao cấp: So với một số hệ quản trị cơ sở dữ liệu khác, MySQL không cung cấp một số tính năng cao cấp như khả năng lưu trữ dữ liệu không cấu trúc.
- Bảo mật và quản lý phiên khó khăn: Mặc dù MySQL có các tính năng bảo mật và quản lý phiên, nhưng cần phải cẩn thận khi cấu hình để đảm bảo an toàn tối đa.



- Khi sử dụng phiên bản cũ: Một số phiên bản cũ của MySQL có thể gặp vấn đề về hiệu suất, bảo mật hoặc sự ổn định, vì vậy cần cân nhắc khi chọn phiên bản sử dụng.

2.7 API

Cho việc giao tiếp giữa backend và frontend, nhóm ứng dụng phương thức giao tiếp REST. REST API (Representational State Transfer Application Programming Interface) là một phương thức để giao tiếp giữa các ứng dụng thông qua mạng. Nó cho phép các ứng dụng khác nhau trao đổi dữ liệu và thực hiện các thao tác trên các tài nguyên từ xa. REST API dựa trên các nguyên tắc của kiến trúc REST, đặc biệt là việc sử dụng các phương thức HTTP để thực hiện các thao tác trên các tài nguyên. Một REST API thường bao gồm các yếu tố sau:

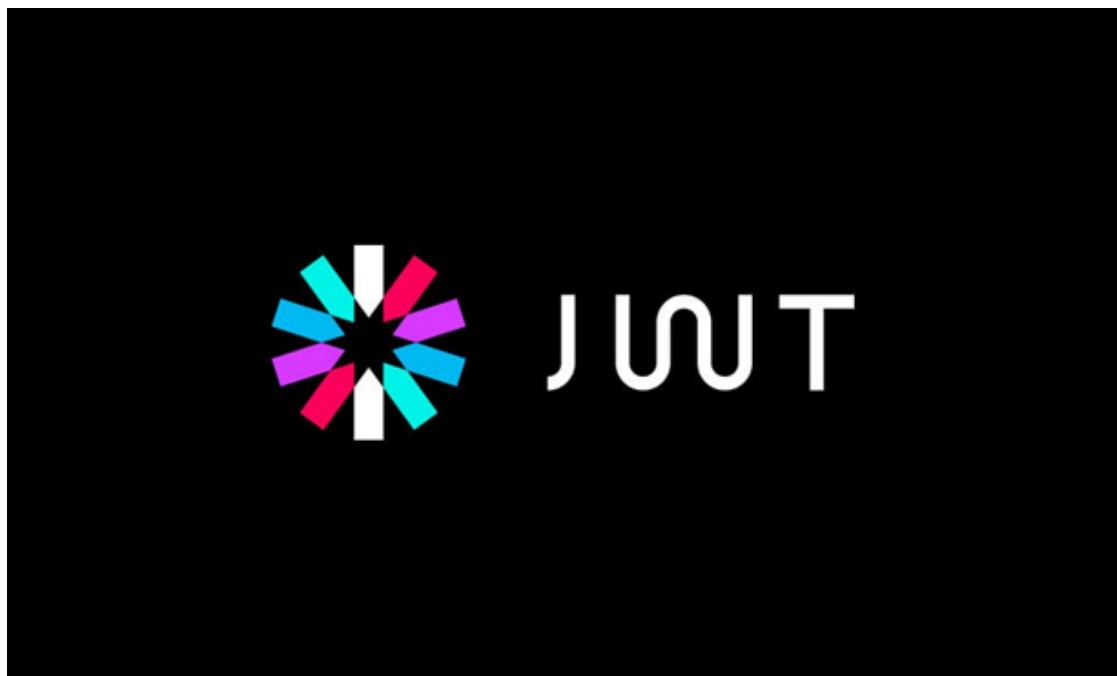
- Tài nguyên (Resources): Đại diện cho các đối tượng hoặc dữ liệu mà bạn muốn thao tác qua API. Mỗi tài nguyên được định danh bằng một URL duy nhất.
- Phương thức HTTP (HTTP Methods): Các phương thức HTTP như GET, POST, PUT và DELETE được sử dụng để thực hiện các thao tác trên tài nguyên. Ví dụ, GET để truy vấn thông tin, POST để tạo mới, PUT để cập nhật, và DELETE để xóa.
- Định dạng dữ liệu (Data Formats): Dữ liệu được trao đổi thông qua API thường sử dụng các định dạng như JSON (JavaScript Object Notation) hoặc XML (eXtensible Markup Language).
- Quản lý trạng thái (Stateless): REST API là không trạng thái, điều này có nghĩa rằng mọi yêu cầu từ phía client chứa đủ thông tin để server có thể hiểu và xử lý yêu cầu, không cần lưu trạng thái trước đó.

Ví dụ, để truy vấn thông tin về một người dùng từ một REST API, bạn có thể gửi một yêu cầu GET đến URL như sau: <https://api.example.com/users/123>, trong đó 123 là ID của người dùng. Server sẽ trả về dữ liệu người dùng dưới định dạng JSON hoặc XML. Tóm lại, REST API là một phương thức tiêu chuẩn để giao tiếp giữa các ứng dụng thông qua mạng bằng cách sử dụng các phương thức HTTP và định dạng dữ liệu như JSON hoặc XML. Điều này cho phép các ứng dụng tương tác với nhau một cách linh hoạt và hiệu quả.



2.8 Bảo mật

Xác thực và phân quyền người dùng với JWT

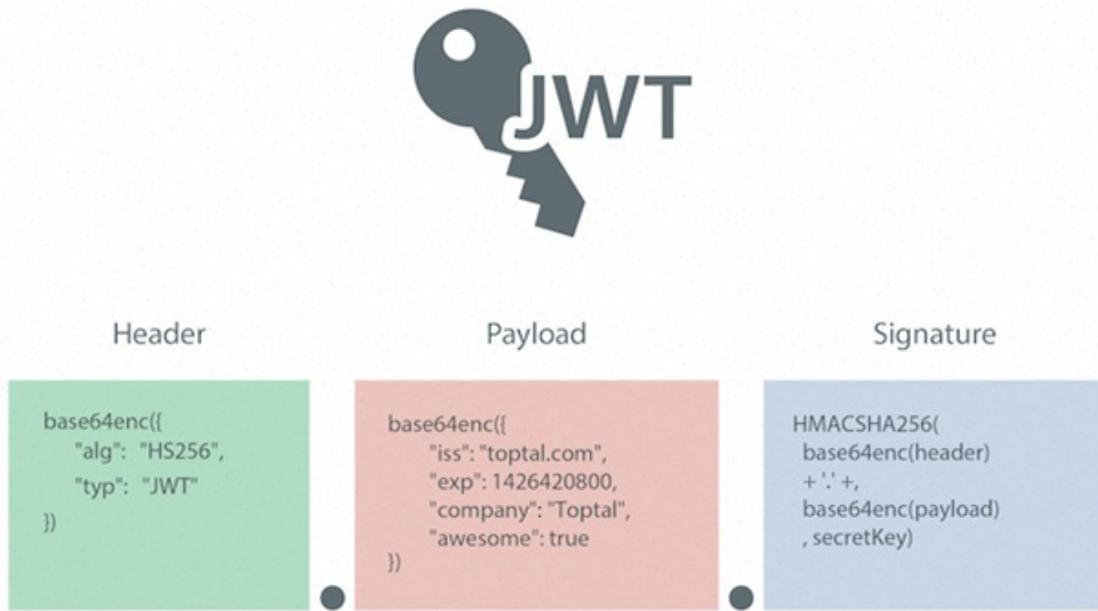


Hình 4: JSON Web Token

JSON Web Token - JWT JWT là một định nghĩa chuẩn về phương pháp xác thực đơn giản và ít tốn kém chi phí (về tài nguyên lưu trữ và tính toán) giữa các bên khác nhau trên internet thông qua các JSON object đã được mã hóa, thường được gọi là token. JWT hoạt động dựa trên cơ chế chữ ký số, JWT sẽ được ký bởi bên cung cấp token với một thông điệp bí mật (secret) và thuật toán HMAC hoặc sử dụng một cặp khóa công khai nếu thuật toán được sử dụng là RSA hay ECDSA.

Cấu trúc của một JWT Token: Một JWT token thường có 3 phần chính là header payload, signature được ngăn cách nhau bởi dấu "..". Cụ thể từng phần như sau:

- **Header** mô tả thuật toán được sử dụng để tạo chữ ký và loại token được sử dụng (cần có trường này vì ngoài JWT còn có các loại hình xác thực token khác như SWT và SAML).
- **Payload** là phần chứa các thông tin - được gọi là claims dùng để định danh một cá thể trong quá trình truyền nhận thông tin cùng một số thông tin thêm để kiểm tra tính hợp lệ của token.
- **Signature** là phần chữ ký được tạo ra bằng cách ký lên các thông tin gồm header, payload, và secret được dùng giữa các bên. Chữ ký này dùng để đảm bảo rằng thông tin không bị sửa đổi trong quá trình truyền nhận và xác thực được người gửi token này là ai (nếu token được ký với một khóa riêng trong mã hóa khóa công khai).

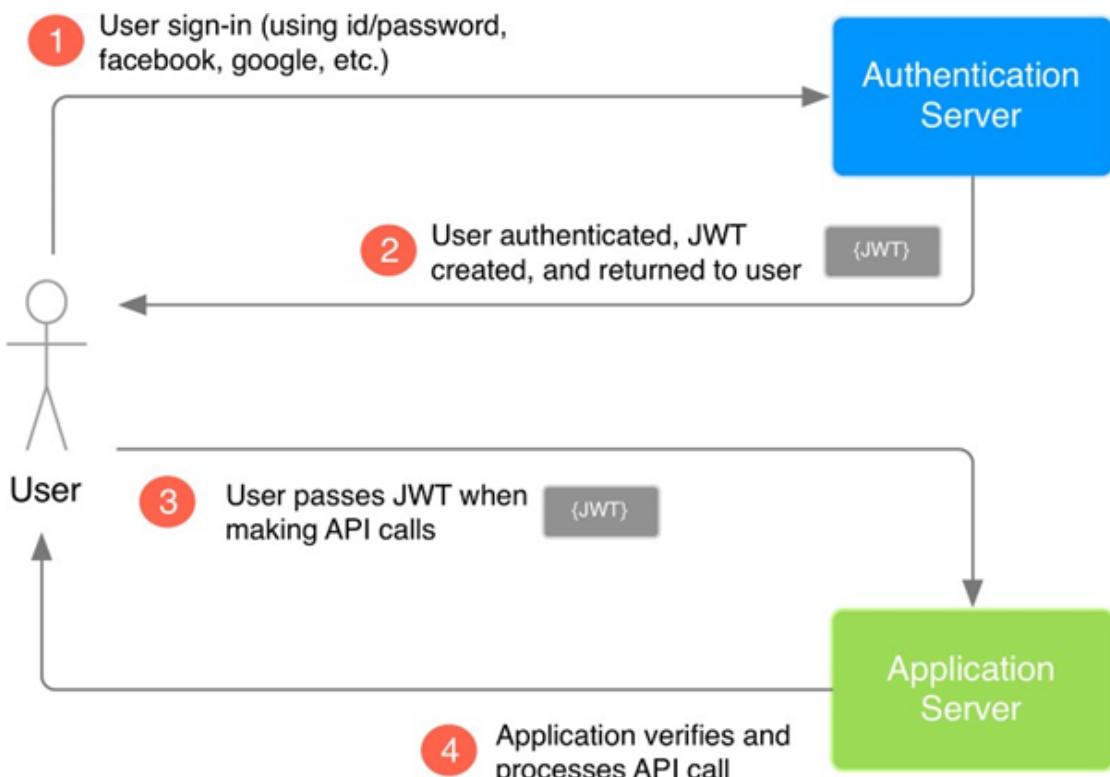


Hình 5: Các thành phần bên trong của JWT.

Quy trình của xác thực client với JWT:

1. Người dùng sẽ gửi thông tin xác thực của mình đến server xác thực.
2. Server xác thực sẽ thực hiện ký lên thông tin của người dùng kèm với một số claims khác như thời gian cấp, thời điểm hết hạn và thời điểm mà token bắt đầu trở nên hợp lệ.
3. Sau khi người dùng nhận được token, họ sẽ đính kèm token này vào các lời gọi API (thường là bỏ ở header Authorization của api request).
4. Server xử lý các api request sẽ tiến hành parse token và xử lý xác thực để cho phép người dùng truy cập các resource.

Tại sao chúng ta lại sử dụng JWT ?



Hình 6: Quy trình xác thực client với JWT

- Đơn giản, gọn nhẹ: Như ta có thể thấy ở trên, việc xác thực với JWT vô cùng đơn giản và gọn nhẹ. Việc lưu trữ token hoàn toàn nằm ở phía client giúp phần nào giảm tải được nhu cầu về tài nguyên lưu trữ và tính toán ở server. Đây là một ưu điểm nổi bật khi so sánh với các phương pháp truyền thống khác sử dụng Session hay Cookies.
- Dễ lập trình: Vì JWT được hỗ trợ và sử dụng rộng rãi nên hầu như tất cả ngôn ngữ lập trình backend đều hỗ trợ việc tạo và xác thực các JWT một cách tiện lợi.
- Dễ scale: Vì việc xác thực với JWT không yêu cầu lưu trữ thêm thông tin ở phía server, ứng dụng của ta có thể scale tốt hơn mà không phải tốn thêm quá nhiều chi phí cho các server và database xác thực người dùng.

Một vài điểm hạn chế: Vì JWT token nằm hoàn toàn ở phía client nên ta không thể hủy các token một cách chủ động từ backend. Để xử lý các phiên của người dùng và chống một số loại tấn công DOS ta phải ràng buộc thời hạn hợp lệ của token và hiện thực thêm các cơ chế cấp lại token mới cho người dùng. Trong quá trình hiện thực bài tập lớn, server cung cấp JWT sẽ bỏ vào payload lúc encode là lúc cấp và có giới hạn thời gian mà token này được xem và valid để ngăn chặn việc tấn công DOS và tránh được phần nào các lần đặt hàng không mong muốn. Các endpoint quan trọng đều được bảo vệ với JWT.



2.9 SEO

Nhóm đã tìm hiểu và thực hiện các nguyên tắc về SEO để đảm bảo khi deploy trang web luôn được index cao bởi search engine của google. Một số các nguyên tắc mà nhóm đã thực hiện trong quá trình hiện thực bài tập lớn như sau:

- Dùng đúng loại ảnh và không chứa ảnh trong database để đảm tăng tốc độ load trang; Các ảnh của nhóm được lưu trên dịch vụ storage của firebase nên database mySQL của nhóm sẽ chỉ phải lưu dạng chuỗi cho các hình ảnh.
- Các content luôn được đánh heading đúng với các tag h1,h2,h3 và component typography để đảm bảo là người dùng có thể phân cấp được nội dung và chú ý vào mục họ tìm kiếm nhanh hơn.
- Các URL dùng để navigate trong page luôn được chú ý đặt để trở nên ngắn và dễ hiểu nhất có thể đối với người dùng.

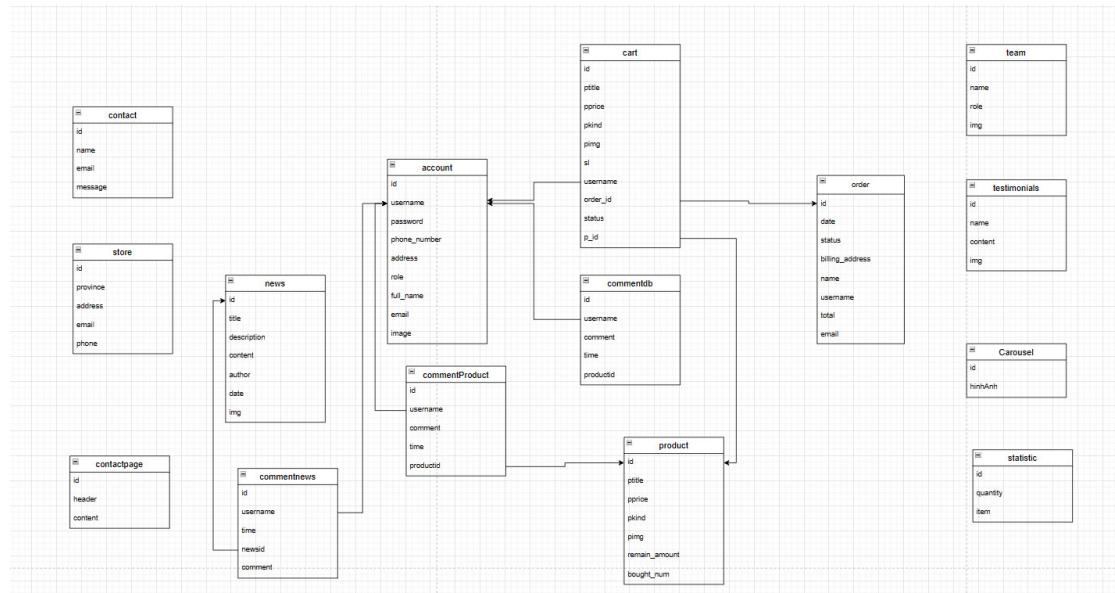
Việc tối ưu hóa trang web có thể bao gồm những công đoạn sắp xếp lại những nội dung của trang web sao cho hợp lý, giúp các công cụ tìm kiếm dễ dàng index; hoặc bao gồm việc tối ưu tốc độ tải trang web xuống mức tối thiểu, làm cho trang web tốn ít tài nguyên để index, nhờ đó tăng khả năng trang web được các công cụ tìm kiếm đưa lên kết quả.

Tuy nhiên, việc tối ưu hóa không có nghĩa là trang web của chúng ta chắc chắn sẽ được đưa lên kết quả trên các trang công cụ tìm kiếm. Ngoài ra, việc tối ưu là một công đoạn rất tốn thời gian và nguồn lực, đòi hỏi người làm web phải hiểu biết sâu về hệ thống cũng như có nội dung chất lượng ngay từ ban đầu.

3 Thiết kế ứng dụng

3.1 Cơ sở dữ liệu

Cơ sở dữ liệu của trang Web bao gồm 16 bảng dữ liệu: Accounts, Abouting, Carousel, CommentDB, CommentNews, Contact, ContactPage, News, Service, Statistic, Product, Tbl-Product, Store, Team, Cart, Testimonials.



Hình 7: Cơ sở dữ liệu

3.1.1 Accounts

Bảng dữ liệu Accounts lưu trữ thông tin của admin và user bao gồm các trường:

- **id:** Địa chỉ Id của Admin, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **user-name:** Tên đăng nhập của Admin, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **password:** Mật khẩu đăng nhập Admin, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **role:** Mô tả chức danh của tài khoản có độ dài nhỏ hơn hoặc bằng 50 kí tự.

3.1.2 Abouting

Bảng dữ liệu aboutimg lưu trữ hình ảnh phần giới thiệu trang web.

- **id:** Địa chỉ id của aboutimg, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **img:** Lưu trữ hình ảnh của phần about, kiểu medium text.



3.1.3 Carousel

Bảng dữ liệu Carousel lưu giữ thông tin hình ảnh của Carousel bao gồm các trường:

- **id:** Địa chỉ Id của Carousel, kiểu interger và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **hinhAnh:** Lưu giữ link của các hình ảnh, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 160 kí tự.

3.1.4 CommentDB

Bảng dữ liệu Commentdb lưu trữ thông tin chi đánh giá của khách hàng về sản phẩm.

- **id:** Địa chỉ id của Comment, kiểu interger và có độ dài nhỏ hơn hoặc bằng 11 kí tự.
- **username:** Tên người dùng để lại ý kiến, kiểu varchar và có độ dài không vượt quá 50 kí tự.
- **comment:** Ý kiến của người tiêu dùng, kiểu varchar và có độ dài không vượt quá 250 kí tự.
- **time:** Thời gian để lại ý kiến của khách hàng, kiểu Date.
- **productid:** Biểu thị id của sản phẩm khách hàng cho ý kiến, kiểu interger và có độ dài nhỏ hơn hoặc bằng 50 kí tự.

3.1.5 CommentNews

Bảng dữ liệu CommentNews lưu trữ thông tin đánh giá bài báo.

- **id:** Địa chỉ id của Comment, kiểu interger và có độ dài nhỏ hơn hoặc bằng 11 kí tự.
- **username:** Tên người dùng để lại ý kiến, kiểu varchar và có độ dài không vượt quá 100 kí tự.
- **time:** Thời gian đánh giá bài báo, kiểu Date.
- **newsid:** Địa chỉ id của bài báo, kiểu interger và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **newsid:** Nhận xét về bài báo, kiểu Mediumtext.

3.1.6 Contact

Bảng dữ liệu Contact lưu trữ thông tin liên hệ của khách hàng.

- **id:** Địa chỉ id của Contact, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **name:** Tên liên lạc của người dùng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **email:** Địa chỉ email của khách hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **message:** Những lời nhắn gửi của khách hàng, kiểu longtext.



3.1.7 ContactPage

Bảng dữ liệu Contactpage lưu trữ thông tin liên hệ của trang web.

- **id:** Địa chỉ id của contact, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **header:** Tiêu đề của trang liên lạc, kiểu mediumtext.
- **content:** Nội dung của trang liên lạc, kiểu mediumtext.

3.1.8 News

Bảng dữ liệu News lưu trữ thông tin về trang tin tức của Website:

- **id:** Địa chỉ id của News, kiểu integer và có độ dài nhỏ hơn hoặc bằng 11 kí tự.
- **title:** Miêu tả từng chủ đề của News, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 500 kí tự.
- **description:** Mô tả chủ đề, kiểu longtext.
- **content:** Miêu tả nội dung của chủ đề, kiểu longtext.
- **author:** Tác giả của bài viết, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **date:** Ngày đăng bài, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **img:** Lưu trữ hình ảnh của bài viết, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 500 kí tự.

3.1.9 Service

Bảng dữ liệu Service lưu trữ thông tin các dịch vụ của WebSite.

- **id:** Địa chỉ id của service, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **name:** Tên của các dịch vụ dành cho Website, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 150 kí tự.
- **description:** Mô tả dịch vụ Website, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 170 kí tự.

3.1.10 Statistic

Bảng dữ liệu statistic lưu trữ số liệu các đối tượng muốn hiển thị trên trang Giới thiệu.

- **id:** Địa chỉ id của statistic, kiểu interger và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **quantity:** Số liệu của đối tượng thống kê, kiểu interger và có độ dài nhỏ hơn hoặc bằng 150 kí tự.
- **item:** Tên của đối tượng thống kê, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 150 kí tự.



3.1.11 Product

Bảng dữ liệu Product lưu trữ thông tin của sản phẩm.

- **id:** Địa chỉ id của product, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **ptitle:** Tên của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 250 kí tự.
- **pprice:** Giá của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 250 kí tự.
- **pkind:** Loại sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **pimg:** Hình ảnh của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 250 kí tự.

3.1.12 Tbl-Product

Bảng dữ liệu tbl-product lưu trữ thông tin của sản phẩm.

- **id:** Địa chỉ id của product, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **ptitle:** Tên của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 250 kí tự.
- **pprice:** Giá của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **pfile:** Hình ảnh của sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 250 kí tự.
- **pstatus:**

3.1.13 Store

Bảng dữ liệu Store lưu trữ thông địa chỉ các chi nhánh của Shop.

- **id:** Địa chỉ id store, kiểu int và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **province:** Thành phố của cửa hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **address:** Địa chỉ cửa hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **email:** Email liên lạc của cửa hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.
- **phone:** Số điện thoại liên hệ cửa hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 100 kí tự.

3.1.14 Team

Bảng dữ liệu team lưu trữ thông tin các nhà phát triển Website.

- **id:** Địa chỉ id của các nhà phát triển Website, kiểu int và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **name:** Tên của nhà phát triển, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 150 kí tự.
- **role:** Vai trò của các nhà phát triển, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 150 kí tự.
- **img:** Địa chỉ hình ảnh, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 150 kí tự.



3.1.15 Cart

Bảng dữ liệu Cart lưu trữ thông tin giỏ hàng của sản phẩm:

- **id:** Địa chỉ id của cart, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **ptitle:** Tên của sản phẩm, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **pprice:** Giá của sản phẩm, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **pkind:** Loại sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **pimg:** Link hình ảnh của các sản phẩm, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **sl:** Số lượng sản phẩm trong giỏ hàng, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **username:** Biểu thị người sở hữu giỏ hàng, kiểu varchar và có độ dài nhỏ hơn hoặc bằng 50 kí tự.
- **p-id:** Địa chỉ id của người dùng sở hữu giỏ hàng, kiểu integer và có độ dài nhỏ hơn hoặc bằng 50 kí tự.

3.1.16 Testimonials

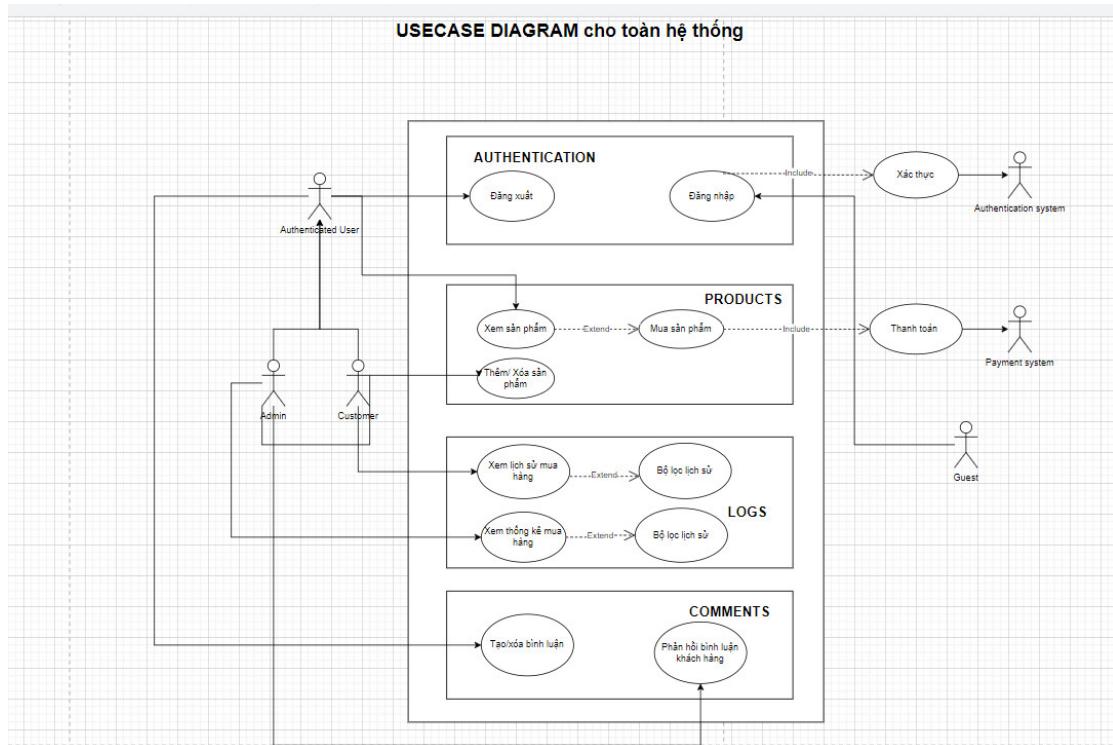
Bảng dữ liệu testimonials mô tả

- **id:** Địa chỉ id của testimonials, kiểu integer, có độ dài nhỏ hơn hoặc bằng 50 kí tự và là giá trị tự động tăng.
- **name:**
- **content:**
- **img:**

3.2 UseCase cho toàn bộ hệ thống

UseCase mô tả sự tương tác giữa người dùng và hệ thống. Thông thường có 2 cách thử phổ biến:

- Cách thức mà người dùng tương tác với hệ thống.
- Cách thức mà hệ thống tương tác với các hệ thống khác.



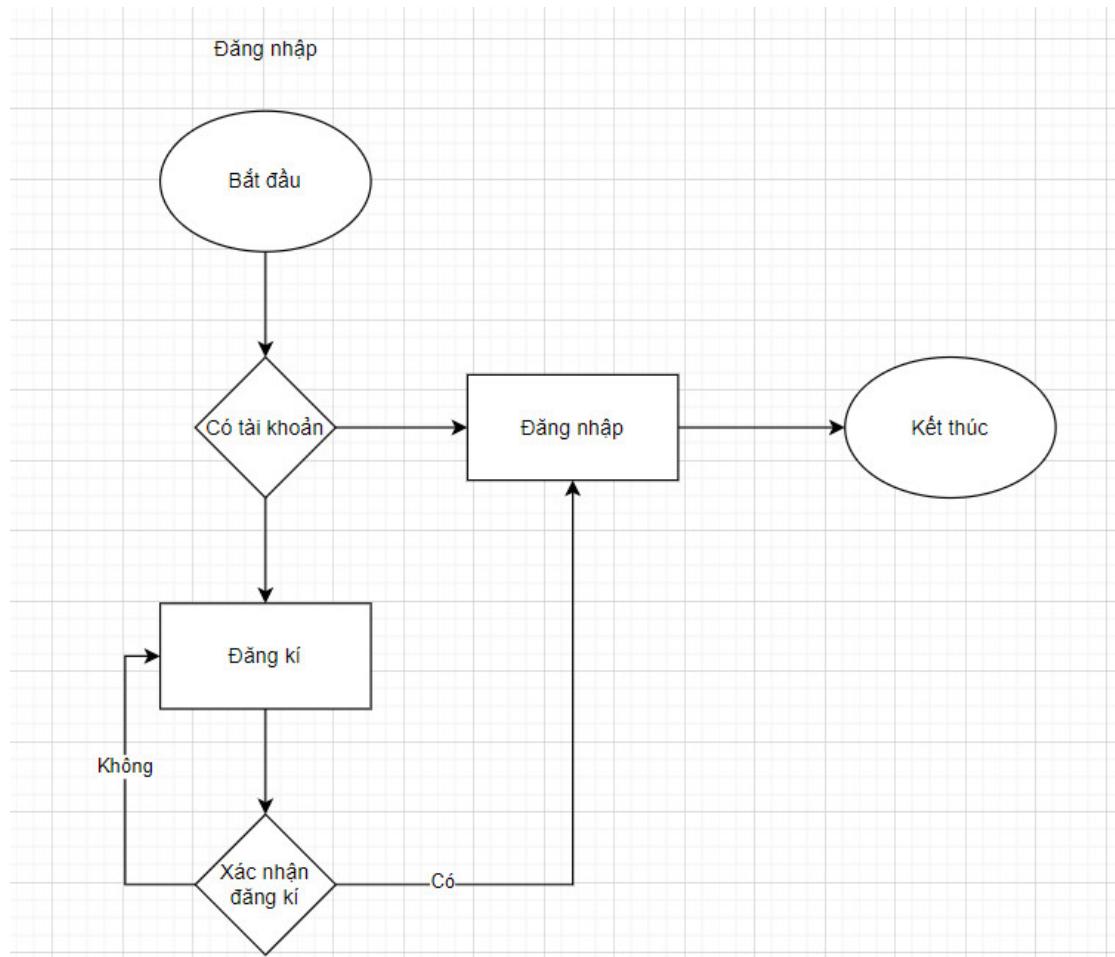
Hình 8: UseCase cho toàn bộ hệ thống

Mô tả ngắn gọn:

- **Guest:** Chỉ được quyền đăng nhập vào Website.
- **Admin** bao gồm những quyền sau:
 - Xem và chỉnh sửa thông tin của trang Web.
 - Xem chỉnh sửa giỏ hàng của khách hàng.
 - Xem thống kê mua hàng thông qua lịch sử đơn hàng.
 - Thay đổi thông tin và quản lý danh sách các khách hàng.
 - Phản hồi các bình luận của khách hàng.
- **Khách hàng** bao gồm những quyền sau:
 - Xem và lựa chọn sản phẩm của trang Web.
 - Thêm xóa sản phẩm và xem lịch sử mua hàng.
 - Tạo bình luận cho các sản phẩm của Shop.
 - Đăng xuất khỏi hệ thống.

3.3 Các thao tác để thực hiện một số chức năng của Website

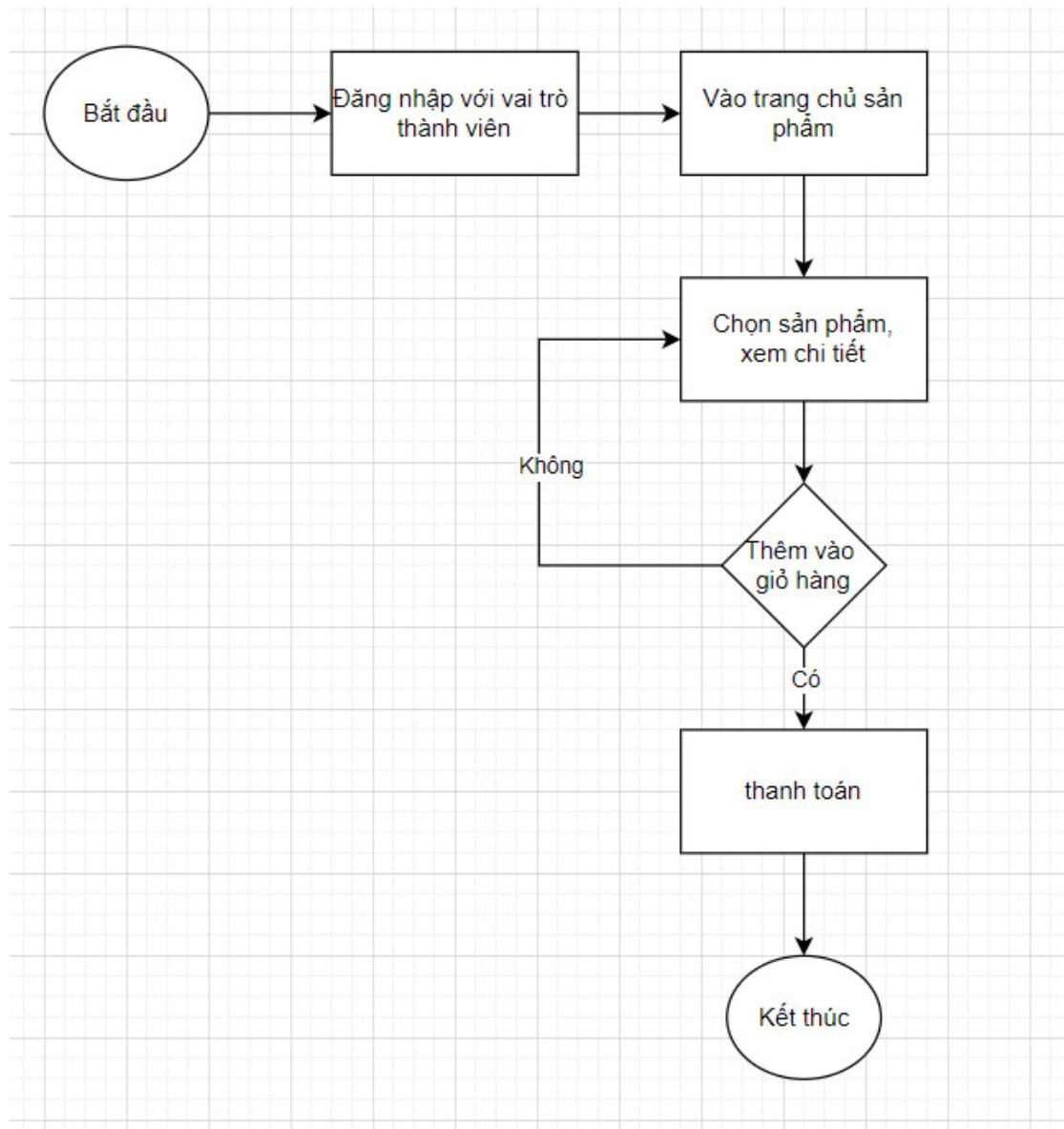
3.3.1 Đăng nhập



Hình 9: Quy trình đăng nhập tài khoản

Mô tả ngắn gọn: Khi người dùng truy cập vào Website, Để sử dụng được những dịch vụ của trang Web người dùng sẽ phải cần dùng một tài khoản đăng nhập. Nếu chưa có tài khoản người dùng sẽ phải đăng kí và được sự xác nhận từ phía admin. Trường hợp người dùng đã đăng kí thì chỉ cần tiến hành đăng nhập là sẽ hoàn tất.

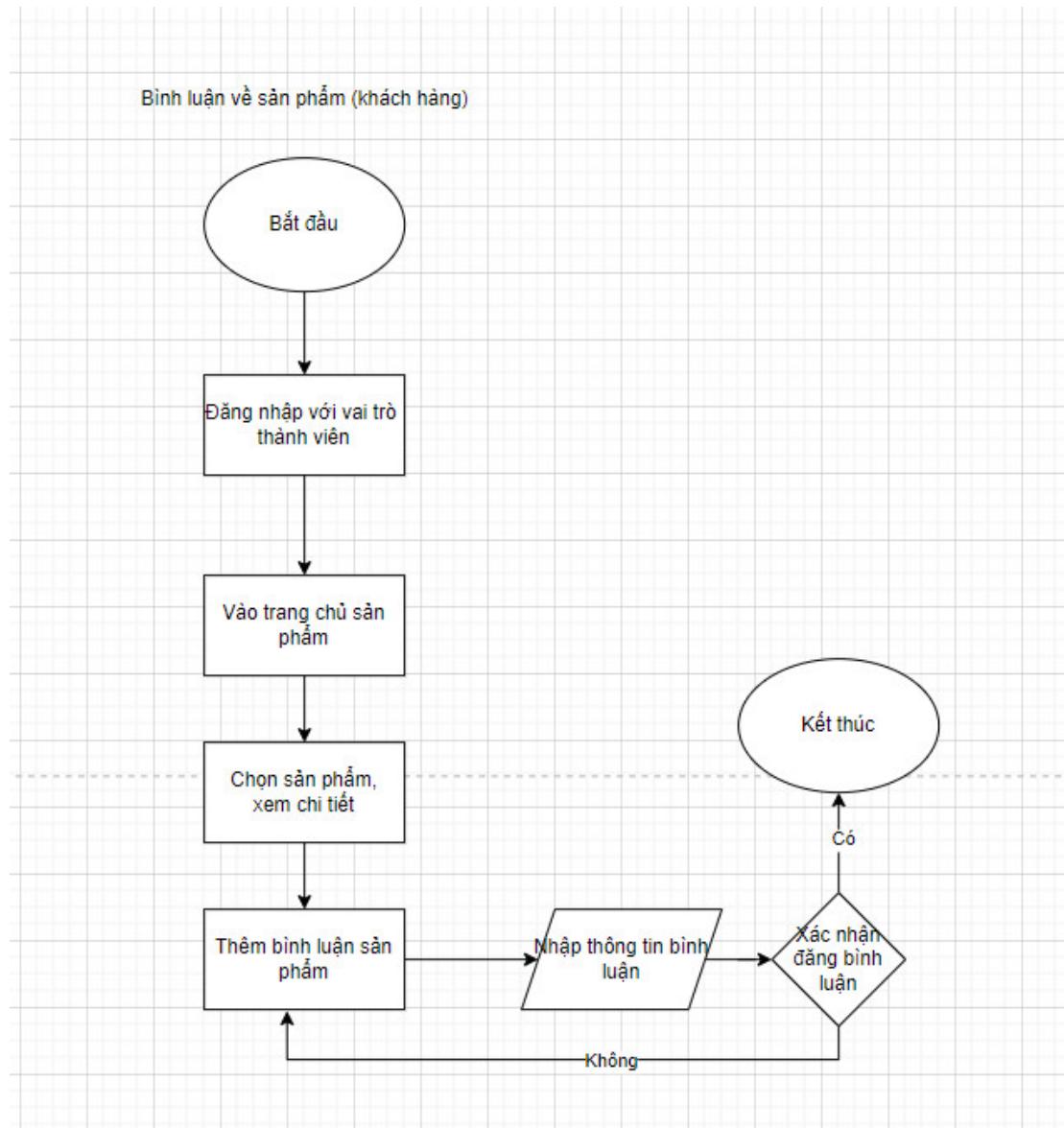
3.3.2 Mua hàng



Hình 10: Quy trình mua hàng

Mô tả ngắn gọn: Khi người dùng muốn mua mặt hàng của Shop, người dùng cần đăng nhập với vai trò là thành viên mua hàng, sau đó vào Trang chủ của sản phẩm, chọn sản phẩm cần mua rồi tiến hành thêm vào giỏ hàng. Ở trang giỏ hàng khi tiến hành thanh toán, Website sẽ dẫn người dùng đến trang thanh toán sau đó hoàn tất việc mua hàng. Nếu khách hàng chưa muốn thanh toán thì có thể quay lại Website để tìm thêm các sản phẩm dành cho người dùng.

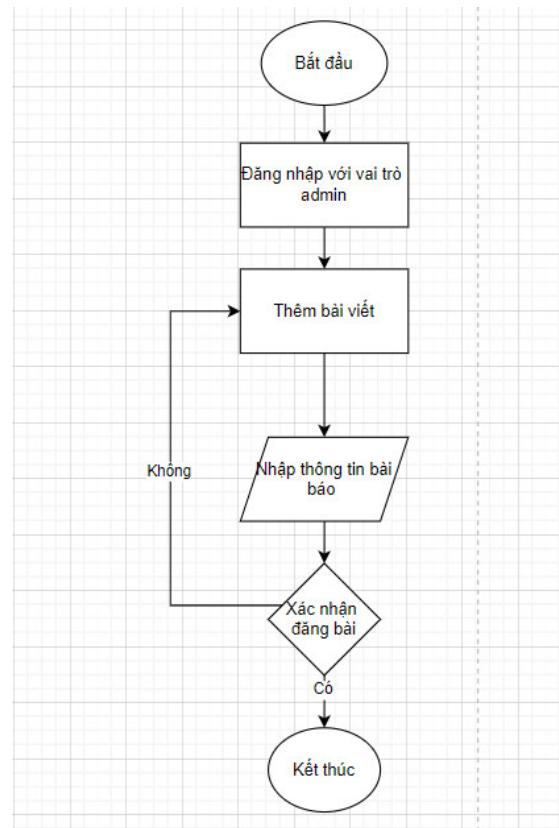
3.3.3 Đánh giá về sản phẩm



Hình 11: Quy trình thực hiện bình đánh giá sản phẩm

Mô tả ngắn gọn: Khi muốn để lại đánh giá về một loạt mặt hàng sản phẩm, người dùng cần đăng nhập với vai trò là một người tiêu dùng, vào Trang chủ của sản phẩm sau đó chọn chi tiết sản phẩm mà người dùng muốn để lại đánh giá. Chọn thêm bình luận sau đó nhập những thông tin mà khách hàng muốn nhận xét về sản phẩm. Chọn xác nhận để hoàn tất việc đánh giá sản phẩm.

3.3.4 Đăng bài báo của Admin



Hình 12: Quy trình thực hiện đăng 1 bài báo

Mô tả ngắn gọn: Khi Website cần cập nhật thêm một số bài báo cho trang Web, cần đăng nhập vào trang web với tư cách là một Admin, chọn vào thêm bài viết cho trang Web sau đó nhập thông tin cần thêm. Cuối cùng chỉ cần xác nhận đăng bài là hoàn tất việc đăng thêm một bài báo.



3.4 Cấu trúc mã nguồn

3.4.1 Font-end

Source Code Front-End gồm các folder chính sau:

```
WEBASSIGNMENT
  FE
    node_modules
    public
  src
    assets
    components
    data
    pages
    redux
    services
    templates
    App.css
    App.js
    App.test.js
    i18n.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
    theme.js
    .gitignore
    package-lock.json
    package.json
  OUTLINE
```

Hình 13: File code Front-End

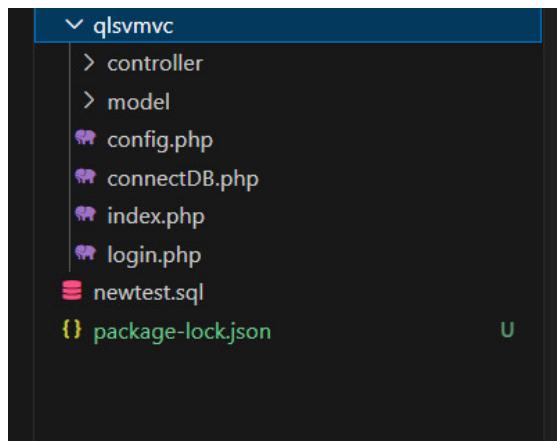
- **assets:** chứa ảnh logo của trang website/
- **components:** Chứa các component nhỏ hơn để dễ quản lý khi render theo vòng lặp như product .
- **data:** là thư mục chứa dữ liệu cứng để test trước khi lấy dữ liệu từ database về



- **pages:** chứa các trang của web được chia nhỏ hơn để dễ quản lý và bảo trì
- **redux:** là cách tổ chức cài đặt redux thunk. Redux Thunk là một middleware cho Redux, một thư viện quản lý trạng thái phổ biến trong các ứng dụng JavaScript, đặc biệt là những ứng dụng sử dụng React. Redux Thunk cho phép viết các action creators trả về một function thay vì một action. Mục đích chính là cho phép việc thực thi các action không đồng bộ. Trong Redux, mọi action đều được giả định là đồng bộ và ngay lập tức tạo ra các thay đổi trạng thái. Trong redux sẽ bao gồm action và reducer. Không update store trực tiếp được, muốn thay đổi cái store thì phải truyền action vào. Action (State is Read-Only) chỉ là một object trong Javascript. Quy định duy nhất của của object này là phải có type. Các file trong action sẽ gọi các hàm từ các file service để thao tác với với api lấy dữ liệu. Trong folder action sẽ có một folder types chứa các type khi người dùng muốn set một state bất kỳ. Reducer (Changes are made with Pure Functions). Sẽ có một cái hàm nhận cái action ở trên, rồi trả về state mới. Mà hàm này phải là pure function. Reducer là một cái hàm, nó trả về state mới. Sau đó ở component nào muốn xài thì liên hệ store để truy cập dữ liệu trong store.
- **service:** Sẽ chứa các hàm trực tiếp thao tác với api để lấy, thay đổi dữ liệu trả về action
- **template:** chứa các thư mục HomeTemplate, Layout, Usertemplate, và các file HomeTemplate, ProfileTempate, UserTemplate, việc tách ra các file riêng như vậy sẽ dễ bảo trì và làm việc nhóm.

3.4.2 Back-end

Source Code Back-End gồm các folder chính sau:

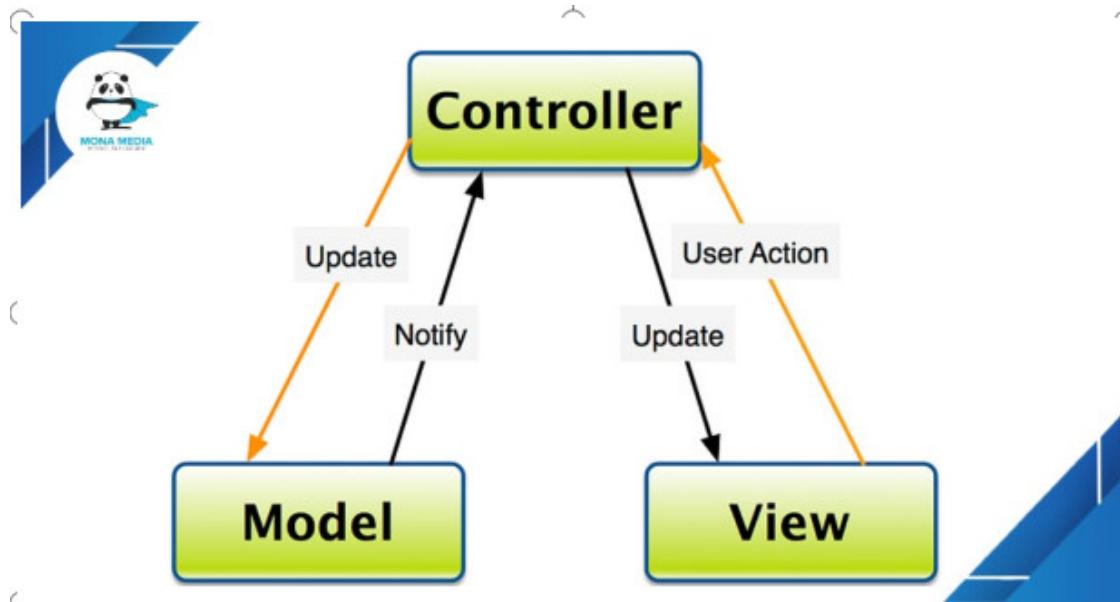


Hình 14: File code Back-End

- **controller:** là thành phần xử lý tương tác của người dùng, nhận dữ liệu từ Model, xử lý và chuyển tiếp thông tin đến View. Nó hoạt động như một trung gian giữa Model và View. Controller quản lý luồng dữ liệu trong ứng dụng, xử lý logic nghiệp vụ, và điều hướng. Nó nhận yêu cầu từ người dùng, xử lý yêu cầu đó (thường thông qua Model), và trả về phản hồi thích hợp.

- **model:** đại diện cho cấu trúc dữ liệu của ứng dụng, thường tương tác trực tiếp với cơ sở dữ liệu. Nó chứa các logic liên quan đến dữ liệu: cách truy xuất, lưu trữ và xác thực dữ liệu. Model chịu trách nhiệm cho việc lấy dữ liệu, xử lý và gửi trả dữ liệu đến Controller. Nó cũng có thể chứa logic nghiệp vụ như tính toán hoặc kiểm tra
- **index.js:** giống như một router đóng vai trò quan trọng trong việc điều hướng và kết nối giữa Controller và Model. Nó quản lý các yêu cầu đến từ client (thường là thông qua URL) và quyết định Controller nào sẽ xử lý yêu cầu đó, phân tích url và các thông tin yêu cầu khác (như phương thức HTTP) để xác định logic nào sẽ được thực thi. Nó có thể gọi một hàm trong Controller cụ thể, kích hoạt một chuỗi các sự kiện để xử lý yêu cầu.
- **file config:** được cài đặt các biến mặc định để mỗi khi thay đổi database và các thông tin khác ta chỉ cần vào đây thay đổi.
- **file connectDB:** dùng để kết nối đến database.

3.4.3 Mô hình Model View Controller(MCV)



Hình 15: Mô hình MVC

Về phía backend ta thực hiện theo mô hình MVC (Model – View – Controller). Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Model(M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,...



View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi. Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấp vào nút đó sẽ có một hành động trong Controller.

Controller (C): Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

Một ví dụ về việc sử dụng mô hình MVC cho Website.

Khi admin cần thêm một loại sản phẩm vào trong Shop của mình, lúc này mô hình MVC sẽ hoạt động như sau:

- **View:** Lúc này sẽ hiện thị ra một form để thêm sản phẩm cho cửa hàng.
- **Controller:** Lúc này sẽ nhận dữ liệu từ Form sau đó gửi tới cho Model.
- **Model:** Lúc này sẽ nhận dữ liệu từ Controller và lưu trữ những thông tin từ form vào hệ cơ sở dữ liệu.

Kết luận: Mỗi bộ phận thực hiện chức năng nhất định, nhưng chúng có sự thống nhất, liên kết với nhau tạo nên mô hình MVC. Mô hình này tương đối nhẹ. Nó có thể tích hợp được nhiều tính năng có trong ASP.NET hiện giờ. Ví dụ như authentication (quá trình xác thực).

4 Các bước nhập cơ sở dữ liệu trên phpMyAdmin.

- Tiến hành mở Hệ cơ sở dữ liệu XAMPP vào Admin để mở hệ cơ sở dữ liệu phpMyAdmin rồi tạo một hệ cơ sở dữ liệu mới.

Cơ sở dữ liệu



Hình 16: Tạo hệ cơ sở dữ liệu

- Chọn file sql để tiến hành import vào hệ cơ sở dữ liệu.



Nhập vào cơ sở dữ liệu "tbl_ltw"

Tập tin để nhập:

Tập tin có thể nén (gzip, bzip2) hoặc không.
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

Duyệt máy tính của bạn: (Tối đa 40MB)

Choose File

Bạn cũng có thể kéo thả một tập tin trên bất kỳ trang nào.

Bảng mã của tập tin:

utf-8

Nhập tung phản:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.
This might be a good way to import large files, however it can break transactions.

Bỏ qua số lượng truy vấn này (cho SQL) bắt đầu tính từ cái thứ nhất:

0

Hình 17: Import file sql vào hệ cơ sở dữ liệu

- Sau khi import vào hệ cơ sở dữ liệu, Trang web của chúng tôi bao gồm 16 bảng.

Table	Action	Rows	Type	Collation	Size	Overhead
aboutimg		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
account		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
carousel		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
cart		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
commentdb		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
commentnews		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contact		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contactpage		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
news		10	InnoDB	utf8mb4_general_ci	64.0 KiB	-
product		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
service		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
statistic		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
store		3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tbl_product		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
team		6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
testimonials		4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
16 tables	Sum				304.0 KiB	0 B

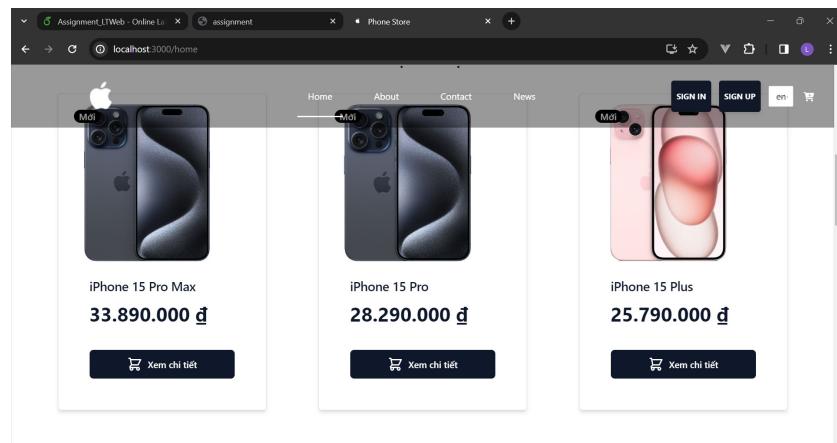
Hình 18: Các bảng trong hệ cơ sở dữ liệu



5 Hiện thực

5.1 Người dùng khi chưa đăng nhập

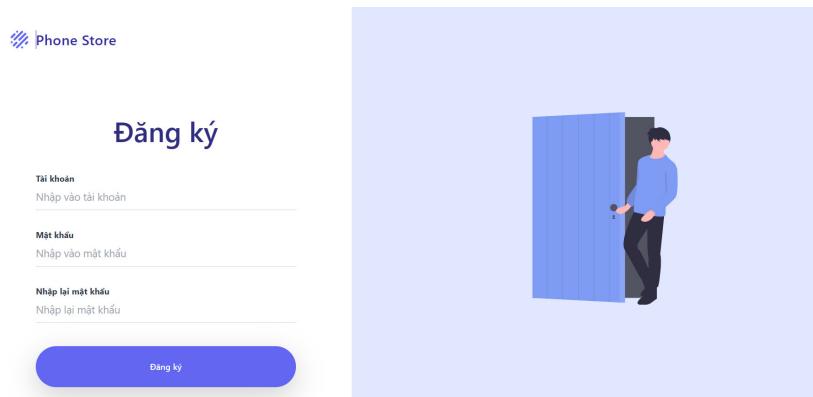
Là người có ít quyền nhất đối với Website. Tại đây người dùng có thể quan sát trang Web ở trang chủ và thực hiện quyền đăng ký hoặc đăng nhập tài khoản.



Hình 19: Trang chủ khi chưa đăng nhập

5.2 Đăng kí

Người dùng đi đến trang Đăng kí của Website

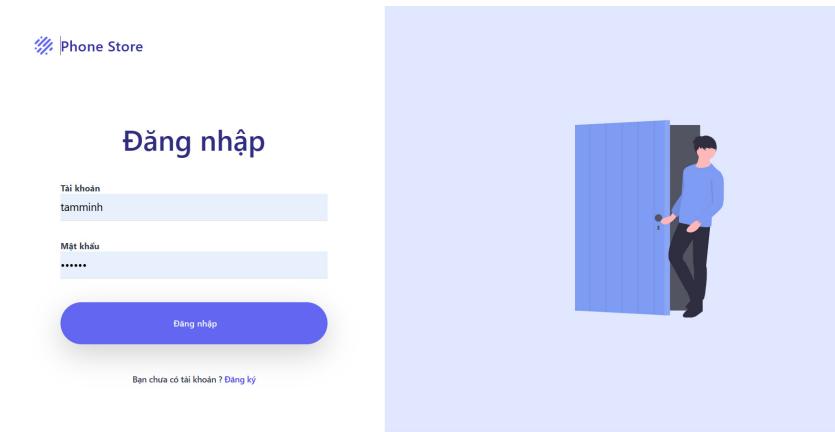


Hình 20: Đăng kí tài khoản



5.3 Đăng nhập

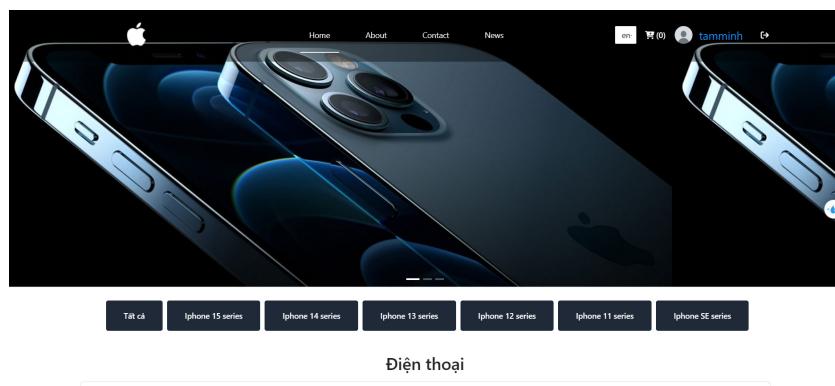
Người dùng đi đến trang Đăng nhập vào Website. Ở đây có thẻ đăng nhập với tư cách là khách hàng hoặc admin.



Hình 21: Đăng nhập tài khoản

5.4 User sau khi đăng nhập

Hình ảnh trang chủ của Website sau khi đã thực hiện đăng nhập.

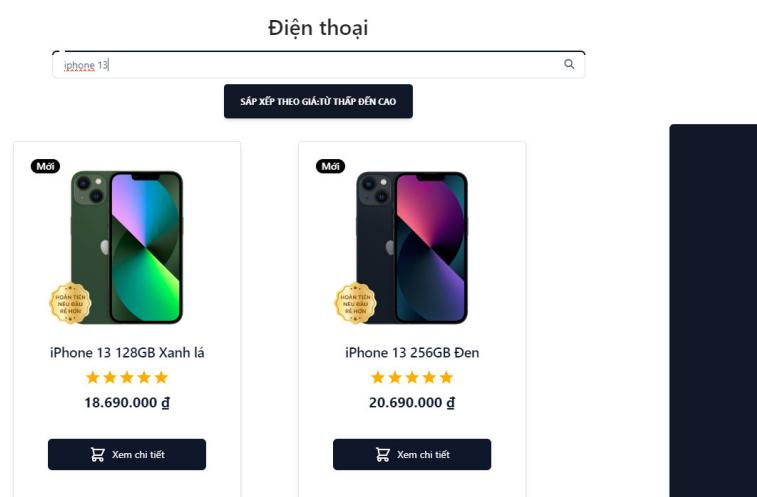


Hình 22: Xem trang chủ của website



5.5 Tìm kiếm mặt hàng của trang Web

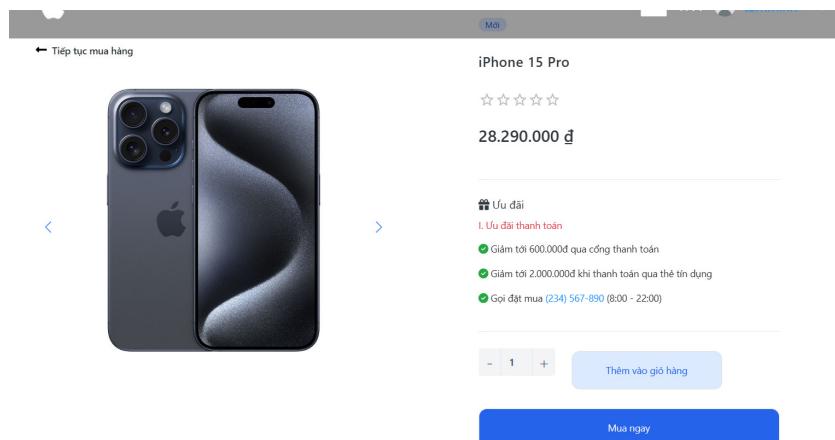
Người dùng sau khi đăng nhập có thể truy cập vào tìm kiếm và mua sản phẩm.



Hình 23: Tìm kiếm mặt hàng theo tên

5.6 Chi tiết sản phẩm

Người dùng có thể bấm vào từng mặt hàng để xem chi tiết sau đó thêm sản phẩm vào giỏ hàng.



Hình 24: Chi tiết của sản phẩm



5.7 Giỏ hàng

Sản phẩm sau khi được thêm vào giỏ hàng.

The screenshot shows a web browser window for a "Phone Store". The URL is "localhost:3000/cart". The page displays a shopping cart with the following items:

SẢN PHẨM	SỐ LƯỢNG	GIÁ	TỔNG
iPhone 15 Pro	8	28.290.000 đ	226.320.000 đ
iPhone 15 Plus	2	25.790.000 đ	51.580.000 đ
		TỔNG TIỀN	277.900.000 đ

At the bottom right of the cart area is a large black button labeled "THANH TOÁN".

Hình 25: Giỏ Hàng

5.8 Tiến hành thanh toán

Sau khi đã quyết định mua hàng người dùng sẽ bấm **Thanh toán** và đi đến trang thanh toán.

The screenshot shows a web browser window for a "Phone Store". The URL is "localhost:3000/checkout". The page displays a payment form with the following information:

Thanh tiền: 226.320.000 đ

Sản phẩm: iPhone 15 Plus
Đơn giá: 25790000
Số lượng: 2
Thành tiền: 51.580.000 đ

Đơn vị vận chuyển:

- viettel post**
Thời gian vận chuyển: 4-5 ngày
- J&T Express**
Thời gian vận chuyển: 2-4 ngày

Địa chỉ nhận hàng:

Ghi chú:

Tổng tiền hàng: 277.900.000 đ
Phi vận chuyển: 20.000 đ
Tổng thanh toán: 277.920.000 đ

A large black button labeled "Đặt hàng" is at the bottom right.

Hình 26: Thanh toán



5.9 Đánh giá của người dùng

Người dùng có thể để lại những đánh giá của mình về các sản phẩm của Shop.

The screenshot shows a product review interface. At the top, there's a navigation bar with links for Home, About, Contact, News, and a language switcher (en). A user profile 'tammminh' is visible. Below the navigation, there's a product image of a dark-colored electronic device. To the right of the image are buttons for 'Thêm vào giỏ hàng' (Add to cart) and 'Mua ngay' (Buy now). Underneath the product image is a review section titled 'Bình luận'. It shows a user profile 'tammminh' with a 5-star rating. There's a text input field with placeholder text 'Để lại bình luận của bạn' (Leave your comment) and a 'Đăng' (Post) button.

Hình 27: Đánh giá

5.10 Danh sách đơn hàng được đặt

The screenshot shows an order history page titled 'Danh sách đơn hàng'. The page has a header with links for Home, About, Contact, News, and a language switcher (en). A user profile 'tammminh' is visible. Below the header is a table with columns: Mã đơn hàng, Tổng thanh toán, Ngày thanh toán, Tình trạng, and Action. The table contains five rows of data:

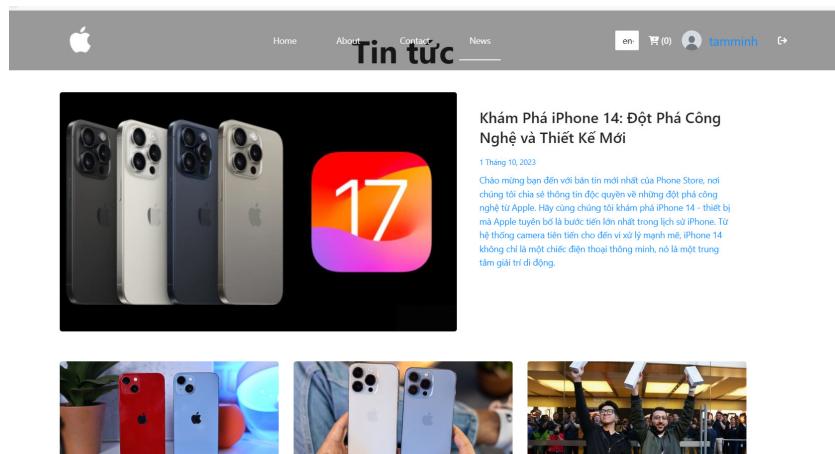
Mã đơn hàng	Tổng thanh toán	Ngày thanh toán	Tình trạng	Action
2	8388607	14/12/2023	Hủy	
3	8388607	14/12/2023	Đang tiếp nhận	
4	8388607	14/12/2023	Đang tiếp nhận	
5	8388607	14/12/2023	Đang tiếp nhận	

Hình 28: Danh sách đơn hàng



5.11 Tin tức

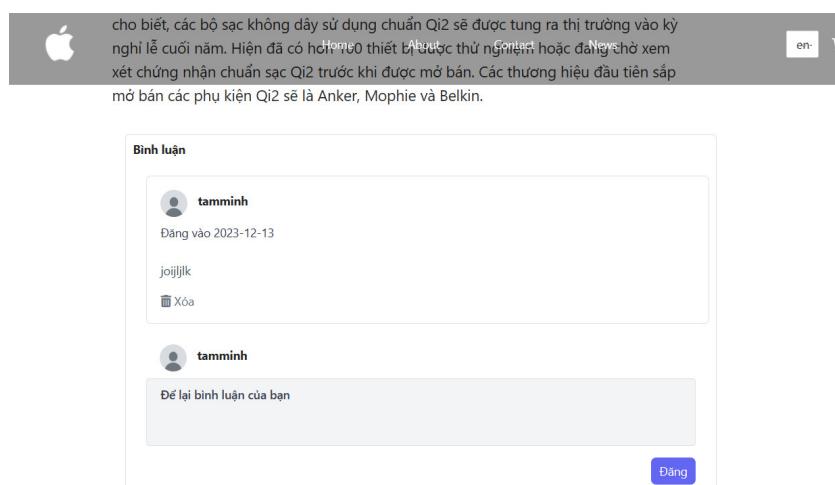
Người dùng có thể vào trang Tin tức để xem những thông tin mới nhất về Website.



Hình 29: Tin tức website

5.12 Bình luận về tin tức

Người dùng có thể vào trang Tin tức để xem những thông tin mới nhất về Website.



Lần đầu lô điện sau 15 năm: Đây là

Hình 30: Bình luận về tin tức



5.13 Liên hệ

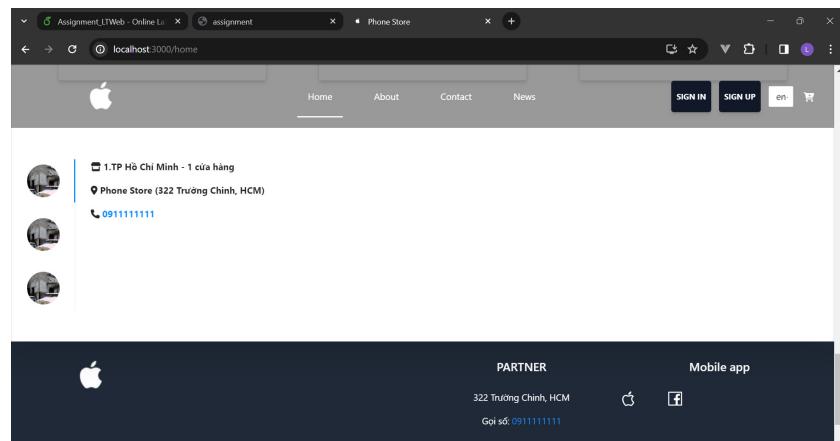
Người dùng có thể liên hệ với Shop qua trang liên lạc:

The screenshot shows a contact form on a website. At the top, there's a header with the university logo and navigation links: Home, About, Contact (which is underlined), and News. There are also language and user session controls. The main section is titled "Liên hệ chúng" (Contact us). It has fields for "Họ và tên" (Name) containing "Trần Văn A", "Địa chỉ email" (Email address) containing "trana@gmail.com", and "Lời nhắn" (Message). Below the form is a large text area for the message. At the bottom is a "Gửi" (Send) button. The footer contains links to "Thông tin liên hệ" (Store information), "Hệ thống cửa hàng" (System), "Nội quy cửa hàng" (Store rules), "Điều khoản phục vụ" (Service terms), "Hỗ trợ khách hàng" (Customer support), "Điều kiện giao dịch chung" (General trading conditions), and "Chính sách giao hàng" (Delivery policy).

Hình 31: Liên hệ Shop

5.14 Địa chỉ của Shop

Địa điểm của cửa hàng trên toàn quốc:



Hình 32: Địa chỉ cửa hàng



5.15 Người dùng chỉnh sửa thông tin các nhân

Thông tin cá nhân

Danh sách đơn hàng

THAY ĐỔI MẬT KHẨU

Email: Tâm Minh

Email: tammminh@gmail.com

Số điện thoại: 01234567899

Địa chỉ: 1

LƯU THÔNG TIN

Hình 33: Chỉnh sửa thông tin các nhân của người dùng

5.16 Người dùng chỉnh sửa mật khẩu

Thay đổi mật khẩu

Mật khẩu:

Mật khẩu mới:

Xác nhận mật khẩu mới:

XÁC NHẬN

Hình 34: Sửa đổi mật khẩu



5.17 Trang quản lý của Admin

The screenshot shows a web browser window with the URL `localhost:3000/manageOrder`. The title bar says "assignment". The main content is titled "Danh sách đơn hàng" (List of Orders). On the left, there is a sidebar with a user profile picture and a list of navigation items:

- Trang chủ
- Thông tin cá nhân
- Quản lý sản phẩm
- Quản lý người dùng
- Quản lý đơn đặt hàng
- Danh sách liên hệ
- Quản lý tin tức

The main area displays a table with the following data:

Mã đơn hàng	Người dùng	Tổng thanh toán	Ngày thanh toán	Địa chỉ nhận h...	Ghi chú	Tình trạng	
2	tamminh	8388607	14/12/2023	Bình thường	Hủy		
3	tamminh	8388607	14/12/2023	linh trung, thu...	Đang tiếp nhận		
4	tamminh	8388607	14/12/2023	le duan	Đang tiếp nhận		
5	tamminh	8388607	14/12/2023	dinh bo linh	nhe tay thoi n...	Đang tiếp nhận	

Hình 35: Trang quản lí của Admin

6 Cách thức cài đặt

6.1 Về Front-end

Để chạy front-end, ta cần thực hiện các bước sau:

1. Cài đặt Node và NPM
2. Clone source code từ repos của nhóm.
3. Chạy npm install (hoặc npm install yarn và sau đó chạy yarn).
4. Chạy lệnh npm start (chạy yarn start nếu bước 3 đã install yarn).
5. Truy cập vào localhost trên trình duyệt.

6.2 Về Back-end

Để chạy back-end, ta cần thực hiện các bước sau:

1. Cài đặt MySQL.
2. Cài đặt PHP và Composer.
3. Clone source code từ repos của nhóm
4. Bật server XAMP.
5. Server đang được chạy trên localhost:8080



Tài liệu

1. Website chính thức của W3School. Truy cập tại: <https://www.w3schools.com/>.
2. Document ngôn ngữ PHP <https://www.php.net/manual/en/>
3. Document thư viện React [1]<https://reactjs.org/>
4. Document thư viện Tailwind <https://tailwindcss.com/>
5. Goossens, M., Mittelbach, F., Samarin, A LaTeX Companion, Addison-Wesley, Reading, MA, 1994.
6. Kopka, H., Daly P.W., A Guide to LaTeX, Addison-Wesley, Reading, MA, 1999.
7. Pan, D., "A Tutorial on MPEG/Audio Compression," IEEE Multimedia, Vol.2, pp.60-74, Summer 1998.
8. Bài bài giảng môn học Lập trình Web.