

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP LỚN
PHÁT TRIỂN ỨNG DỤNG WEB

Nhóm học phần: CT449-04



ĐỀ TÀI: XÂY DỰNG WEBSITE
BẢN MÔ HÌNH FIGURE

Giảng viên hướng dẫn: Ths. Nguyễn Minh Trung

Sinh viên thực hiện: Lê Hoàng Khang

MSSV: B2017046

Lớp: DI20Z6A2

Khóa: 46

Cần Thơ, tháng 24/11/2023

LỜI CẢM ƠN

Để hoàn thành được bài báo cáo này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường Công nghệ thông tin và Truyền thông, những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Ths. Nguyễn Minh Trung - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài báo cáo này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài báo cáo này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của thầy để bài báo cáo ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	i
DANH MỤC SƠ ĐỒ	iv
DANH MỤC BIỂU BẢNG	v
DANH MỤC HÌNH ẢNH	vi
TÓM TẮT	vii
PHẦN GIỚI THIỆU	1
1. Đặt vấn đề	1
2. Mục tiêu đề tài	1
PHẦN NỘI DUNG	3
Chương 1. Đặc tả yêu cầu	3
Chương 2. Cơ sở lý thuyết	4
1. Tổng quan về MEVN STACK:	4
2. MongoDB:	5
3. Restful API:	5
4. Express.js:	6
4. Vue.js:	7
5. Node.js:	7
6. Tổng quan về Bootstrap 4:	8
7. Tổng quan về Mongoose:	8
Chương 3. Cài đặt giải pháp	9
1. Sơ đồ Use case	9
2. Sơ đồ use case đăng kí và đăng nhập	10
1.4 Sơ đồ use case quản lý tài khoản khách hàng	11
3. Cơ sở dữ liệu	12
Mô hình cơ sở dữ liệu CDM	12
.....	12
Hình : Mô hình CDM	13

Chương 4.	Kết quả đạt được	15
-----------	------------------------	----

DANH MỤC SƠ ĐỒ

Sơ đồ 1. 1: Sơ đồ use case tổng quát.....	9
Sơ đồ 1. 2: Sơ đồ use case Đăng Ký và Đăng Nhập.....	10
Sơ đồ 1. 3: Sơ đồ use case Giỏ Hàng.....	11
Sơ đồ 1. 4: Sơ đồ use Quản lí tài khoản khách hàng	11
Sơ đồ 1. 5: Sơ đồ use case: Quản lí sản phẩm	12

DANH MỤC BIỂU BẢNG

Bảng 1. 1: Đặc tả sơ đồ chức năng use case tổng quát9

Bảng 2. 1: Mô tả lớp 13

Bảng 2. 2: Mô tả user 13

Bảng 2. 3: Mô tả sản phẩm 13

Bảng 2. 4: Mô tả Giao hàng 14

DANH MỤC HÌNH ẢNH

Hình 1: Mô hình CDM	12
Hình 2: Giao diện Đăng Nhập	15
Hình 3: Giao diện Đăng Ký	16
Hình 4: Giao diện trang chủ	16
Hình 5: Giao diện trang liên hệ	17
Hình 6: Giao diện trang tìm kiếm	18
Hình 7: Giao diện Giỏ hàng	18
Hình 8: Giao diện quản lý sản phẩm	19
Hình 9: Giao diện quản lý người dùng	19

TÓM TẮT

Công nghệ thông tin đang ngày càng phát triển một cách nhảy vọt và mạnh mẽ. Sự ra đời của công nghệ thông tin làm cho bộ mặt xã hội, đời sống con người được nâng cao một cách rõ rệt, đóng góp to lớn cho sự phát triển của nhân loại. Với xu hướng toàn cầu hóa nền kinh tế thế giới, trong đó nhu cầu sử dụng Internet ngày càng nhiều và hình thức kinh doanh trên mạng ngày càng đa dạng. Vì vậy, nhiều cửa hàng đồng hồ đã áp dụng phương pháp bán hàng qua mạng để khách hàng dễ dàng tìm kiếm thông tin mà không cần tốn nhiều thời gian và công sức. Để đáp ứng nhu cầu đó, “**Website bán đồng hồ**” đã được xây dựng và áp dụng.

Mục tiêu hướng đến của đề tài “**Xây dựng Website bán đồng hồ**” làm theo công nghệ **MEVN STACK** (MongoDB, Express, Vue.js, Node.js). Để có thể hoàn thành mục tiêu đó cần phải có sự nghiên cứu và hiểu rõ về công nghệ **MEVN STACK** đang được sử dụng phổ biến ở các trang Web hiện nay.

Đề tài đạt được kết quả hướng đến là một “**Website bán đồng hồ**” với giao diện đơn giản, dễ sử dụng, bên cạnh đó giúp việc mua bán giày đá bóng diễn ra một cách nhanh chóng, an toàn, hiệu quả cao.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Sở hữu mô hình Anime Figure không chỉ đơn giản là một sở thích cá nhân mà còn là một đề tài đầy đa chiều và đầy tranh cãi trong cộng đồng yêu thích Anime. Việc này mở ra nhiều vấn đề và thách thức đáng quan ngại:

Giá trị Nghệ thuật và Suu tầm: Mô hình Anime Figure không chỉ là sản phẩm đồ chơi, mà còn được coi là tác phẩm nghệ thuật độc đáo, là nguồn cảm hứng cho người hâm mộ. Tuy nhiên, giá trị nghệ thuật cao cũng mang theo mức giá cao, đặt ra thách thức về quyết định suu tầm và đầu tư trong nghệ thuật.

Việc được xây dựng dự án đã áp dụng thành tựu của công nghệ thông tin vào lĩnh vực thương mại, tăng cường tính chất lượng và sự đa dạng của sản phẩm. Sự ra đời của website này giúp giải quyết vấn đề về mua sắm, cải thiện quy trình kinh doanh cho các cửa hàng bán Figure anime.

2. Mục tiêu đề tài

- Mục tiêu tổng quát:

Xây dựng website bán Figure trực tuyến với các chức năng hoàn chỉnh, dễ sử dụng và thuận tiện cho cả người quản lý và người mua hàng , giúp khách hàng có thể mua được những chiếc đồng chính hãng thể hiện phong cách và cá nhân của họ.

- Mục tiêu cụ thể:

Hệ thống bán mô hình Figure gồm có 2 website: **Website khách hàng** và **Website quản trị**.

 Website khách hàng:

- Website có 5 trang chính:

1. Trang chủ.
2. Liên hệ.
3. Đăng kí.
4. Đăng nhập.
5. Giỏ hàng.

- Chức năng:

1. Trang chủ: để hiển thị các loại mô hình bán chạy và được tìm mua nhiều
2. Trang liên hệ: phản hồi của khách hàng đối với cửa hàng.
3. Trang giỏ hàng: xem giỏ hàng và xóa sản phẩm.
4. Đăng ký: đăng ký tài khoản cho người dùng mới.
5. Đăng nhập: đăng nhập vào tài khoản để thực hiện mua sắm.
6. Tìm kiếm mô hình: tìm mô hình theo tên hoặc theo mô tả mô hình.



Website quản trị:

- Website quản trị có những trang chính:

1. Quản lý sản phẩm.
2. Quản lý tài khoản người dùng.

- Chức năng:

1. Quản lý sản phẩm: tìm kiếm sản phẩm theo tên hoặc theo mô tả, có thể thêm sửa hoặc xóa sản phẩm.
2. Quản lý tài khoản: tìm kiếm tài khoản khách hàng bằng email và xóa tài khoản khách hàng.

PHẦN NỘI DUNG

Chương 1. Đặc tả yêu cầu

Việc xây dựng “**Website bán mô hình Figure**” giúp tạo ra môi trường thuận tiện cho việc mua bán và quảng bá sản phẩm của người bán trên các nền tảng quảng cáo trực tuyến.

Về phía giao diện, “**Website bán mô hình Figure**” được xây dựng tốt nhằm hiển thị rõ ràng tất cả sản phẩm, cung cấp đầy đủ các thông tin cần thiết bao gồm tên đồng hồ, mô tả chi tiết, hình ảnh đồng hồ rõ nét. Mỗi đồng hồ đều được gán một ID đồng hồ riêng nhằm phân biệt với các loại giày đá bóng khác.

Về phía khách hàng, “**Website bán mô hình Figure**” cho phép khách hàng có thể xem thông tin cụ thể mô hình, giá tiền. Khi đặt hàng, khách hàng cần đăng nhập vào hệ thống trước để có thể lưu trữ được các sản phẩm mà khách hàng muốn đặt. Mỗi giỏ hàng bao gồm ID sản phẩm, số lượng sản phẩm, tổng hóa đơn.

Quy trình chung cho khách hàng bao gồm: Khách hàng chọn sản phẩm ưng ý, sau đó bấm nút thêm vào giỏ hàng. Khách hàng vào giỏ hàng có thể xem chi tiết thông tin về sản phẩm và có thể điều chỉnh xóa sản phẩm nếu cần.

Chung quy về việc quản lý website, người quản lý (Admin) có quyền thêm mới (khi có sản phẩm mới về), sửa đổi thông số của sản phẩm, và xóa đi những sản phẩm không còn hoặc đã lỗi. Bên cạnh đó người quản lý (Admin) còn quản lý thông tin tài khoản, quản lý kho hàng sản phẩm.

Chương 2. Cơ sở lý thuyết

1. Tổng quan về MEVN STACK:

Khái niệm:

MEVN STACK là sự kết hợp giữa của **MongoDB**, **Express.js**, **Vue.js**, **Node.js** và khiến cho việc xây dựng những ứng dụng web trở nên mạnh mẽ và đơn giản hơn bao giờ hết.

Ưu và nhược điểm của công nghệ MEVN STACK:

- Ưu điểm:

- + **Dễ dàng xây dựng ứng dụng web động:** MEVN stack cung cấp một giải pháp đầy đủ cho việc xây dựng các ứng dụng web động. Bằng cách kết hợp MongoDB để lưu trữ dữ liệu, Express.js để quản lý tương tác với dữ liệu, Vue.js để tạo giao diện người dùng và Node.js để xây dựng máy chủ, MEVN stack cung cấp một cách tiếp cận đơn giản và hiệu quả để phát triển ứng dụng web động.
- + **Tính linh hoạt:** Mỗi thành phần trong MEVN stack đều rất linh hoạt và cho phép các nhà phát triển tùy chỉnh ứng dụng của mình theo nhu cầu của họ. MongoDB cung cấp một cách tiếp cận linh hoạt cho việc lưu trữ dữ liệu, Express.js cho phép các nhà phát triển xây dựng các API RESTful động, Vue.js cho phép các nhà phát triển tạo giao diện người dùng phức tạp và Node.js cho phép các nhà phát triển xây dựng ứng dụng web động đa nền tảng.
- + **Hiệu năng cao:** MEVN stack sử dụng Node.js để xây dựng máy chủ, cung cấp khả năng xử lý đồng thời cao. Điều này giúp các ứng dụng xây dựng bằng MEVN stack có thể xử lý nhiều yêu cầu cùng một lúc, tăng hiệu suất và tốc độ phản hồi.
- + **Cộng đồng lớn:** MEVN stack được sử dụng rộng rãi trong cộng đồng phát triển web, vì vậy có rất nhiều tài liệu và hỗ trợ từ cộng đồng. Điều này giúp các nhà phát triển dễ dàng tìm kiếm thông tin và giải quyết các vấn đề khi phát triển ứng dụng.

- Nhược điểm:

- + **Không phù hợp cho các ứng dụng lớn:** MEVN stack không phù hợp cho các ứng dụng lớn và phức tạp, do các công nghệ trong stack này chủ yếu được sử dụng cho các ứng dụng web đơn giản và vừa phải.

Nếu bạn đang phát triển một ứng dụng web lớn và phức tạp hơn, bạn có thể cần sử dụng các công nghệ khác hoặc tùy chỉnh MEVN stack để đáp ứng nhu cầu của mình.

- + **Cần có kiến thức về nền tảng JavaScript:** Để phát triển các ứng dụng với MEVN stack, bạn cần có kiến thức về nền tảng JavaScript, bao gồm cả các framework Vue.js và Node.js. Nếu bạn không có kinh nghiệm với JavaScript, việc học và sử dụng MEVN stack có thể khó khăn.
- + **Độ phức tạp của MongoDB:** Mặc dù MongoDB là một cơ sở dữ liệu NoSQL rất phổ biến, nhưng nó có một số đặc điểm phức tạp, đặc biệt là khi sử dụng các tính năng phức tạp như đa lưu trữ hoặc quản lý giao dịch. Điều này có thể khiến cho việc phát triển và bảo trì ứng dụng với MEVN stack trở nên khó khăn hơn.
- + **Khả năng mở rộng giới hạn:** MEVN stack cung cấp một giải pháp đơn giản và nhanh chóng cho việc xây dựng ứng dụng web động, tuy nhiên, khả năng mở rộng của nó có thể bị giới hạn. Nếu ứng dụng của bạn phát triển mạnh mẽ và có nhu cầu mở rộng, bạn có thể cần phải tùy chỉnh MEVN stack hoặc chuyển sang các công nghệ khác để đáp ứng nhu cầu của mình.

2. MongoDB:

- Là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, phát triển bởi MongoDB. Các ứng dụng được phát triển trên MongoDB có thể được triển khai trên các nền tảng khác nhau, bao gồm các môi trường đám mây và máy chủ riêng.

- MongoDB được sử dụng rộng rãi trong các ứng dụng web, đặc biệt là các ứng dụng có nhu cầu xử lý dữ liệu lớn và phức tạp, như các ứng dụng thương mại điện tử, các hệ thống giám sát và các ứng dụng chạy trên đám mây.

3. Restful API:

Khái niệm:

RESTful API (Representational State Transfer) là một kiểu kiến trúc ứng dụng mạng dựa trên giao thức HTTP, giúp các ứng dụng giao tiếp với nhau thông qua các phương thức HTTP như GET, POST, PUT, DELETE. RESTful API giúp đơn giản hóa việc truyền tải dữ liệu giữa client và server dưới dạng JSON hoặc XML.

Ưu điểm và hạn chế:

- Ưu điểm:

- + Dễ hiểu và sử dụng: RESTful API sử dụng các phương thức HTTP thông dụng, giúp dễ dàng triển khai và tích hợp.
- + Tính nhất quán: RESTful API đảm bảo tính nhất quán trong việc truy cập và quản lý dữ liệu.
- + Khả năng mở rộng: RESTful API hỗ trợ khả năng mở rộng tốt, giúp dễ dàng thêm các tính năng mới.
- + Bảo mật: RESTful API hỗ trợ nhiều cơ chế xác thực và ủy quyền, giúp bảo vệ dữ liệu và tài nguyên.
- + Ngôn ngữ và nền tảng độc lập: RESTful API có thể hoạt động trên nhiều ngôn ngữ và nền tảng khác nhau.

- Hạn chế:

6. Trạng thái phiên (stateless): RESTful API không lưu trữ trạng thái phiên giữa các yêu cầu, điều này có thể gây khó khăn trong việc xử lý các tác vụ phức tạp.
7. Tốc độ truy vấn: RESTful API có thể chậm hơn so với các giao thức khác như gRPC hay GraphQL, đặc biệt khi cần truy vấn nhiều dữ liệu liên quan.
8. Over-fetching và under-fetching: RESTful API có thể trả về nhiều dữ liệu hơn hoặc ít hơn so với nhu cầu của client, dẫn đến lãng phí tài nguyên và băng thông.
9. Khó khăn trong phiên bản hóa: Việc cập nhật và quản lý phiên bản của RESTful API có thể gây khó khăn, khiến cho việc duy trì và nâng cấp hệ thống trở nên phức tạp.

4. Express.js:

- Express.js là một framework web được xây dựng trên nền tảng Node.js. Nó được phát triển nhằm giúp cho việc xây dựng ứng dụng web với Node.js trở nên đơn giản và nhanh chóng hơn. Express.js cung cấp cho các nhà phát triển các tính năng và công cụ để xử lý các yêu cầu HTTP, xử lý các phản hồi, quản lý session và cookie, và nhiều tính năng khác.

- Express.js được thiết kế để đơn giản hóa việc xây dựng các ứng dụng web và API với Node.js. Nó cho phép các nhà phát triển tạo ra các định tuyến

(routing) dựa trên các yêu cầu HTTP, tạo middleware để xử lý các yêu cầu, và cung cấp các công cụ để quản lý các kết nối và phiên làm việc.

4. Vue.js:

- Vue.js là một framework JavaScript mã nguồn mở được sử dụng để xây dựng các ứng dụng web động và giao diện người dùng (UI). Nó được thiết kế để cung cấp một cách tiếp cận dễ dàng cho việc xây dựng các ứng dụng phức tạp và tương tác với người dùng.

- Vue.js cho phép các nhà phát triển tạo ra các thành phần (components) tái sử dụng để xây dựng giao diện người dùng và quản lý trạng thái ứng dụng trong một cách hiệu quả. Nó cũng cung cấp các công cụ để liên kết dữ liệu (data binding) và xử lý sự kiện (event handling), giúp cho việc phát triển các ứng dụng web trở nên đơn giản và nhanh chóng hơn.

- Vue.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó hỗ trợ các tính năng như hot-reload (tự động tải lại khi có thay đổi), Vuex (trạng thái quản lý), Vue Router (định tuyến) và nhiều tính năng khác để giúp cho việc phát triển các ứng dụng web trở nên dễ dàng và linh hoạt hơn.

5. Node.js:

- Node.js là một nền tảng phát triển được xây dựng trên Chrome's V8 JavaScript engine để cho phép các nhà phát triển sử dụng JavaScript để phát triển các ứng dụng trên môi trường máy chủ. Nó cho phép các nhà phát triển xây dựng các ứng dụng web và các dịch vụ mạng khác một cách dễ dàng và hiệu quả.

- Node.js cho phép các nhà phát triển xây dựng các ứng dụng web với hiệu suất cao bằng cách sử dụng JavaScript để xử lý các yêu cầu từ phía máy khách và chuyển tiếp chúng đến máy chủ. Nó cũng cung cấp các tính năng để xử lý các yêu cầu không đồng bộ (asynchronous), giúp cho các ứng dụng web trở nên nhanh chóng và hiệu quả hơn.

- Node.js có thiết kế linh hoạt và có thể tích hợp vào các ứng dụng web hiện có một cách dễ dàng. Nó có thể được sử dụng để xây dựng các ứng dụng web đơn giản hoặc các ứng dụng web phức tạp, và hỗ trợ các tính năng như mã nguồn mở, quản lý gói, và phát triển cộng đồng.

6. Tổng quan về Bootstrap 4:

Bootstrap là một framework CSS (Cascading Style Sheets) được phát triển bởi Twitter để giúp các nhà phát triển web xây dựng các giao diện người dùng (UI) cho các ứng dụng web một cách dễ dàng và nhanh chóng. Nó cung cấp các công cụ để tạo ra các trang web thân thiện với người dùng, có thể trông đẹp và đáp ứng (responsive) trên các thiết bị khác nhau.

- Bootstrap có nhiều tính năng hữu ích, bao gồm:
 - + **Grid system:** Hỗ trợ tạo bố cục trang web với ô lưới tự động điều chỉnh kích thước, giúp các trang web đáp ứng tốt trên các thiết bị khác nhau.
 - + **Pre-built components:** Bao gồm nhiều thành phần (components) được thiết kế sẵn như navigation bars, forms, buttons, dropdowns, và modal windows, giúp giảm thời gian phát triển và tăng tính khả dụng của trang web.
 - + **JavaScript plugins:** Cung cấp các plugin JavaScript sẵn có như carousel, modals, tooltips, và popovers, giúp cho việc tạo các hiệu ứng trang web trở nên dễ dàng hơn.

7. Tổng quan về Mongoose:

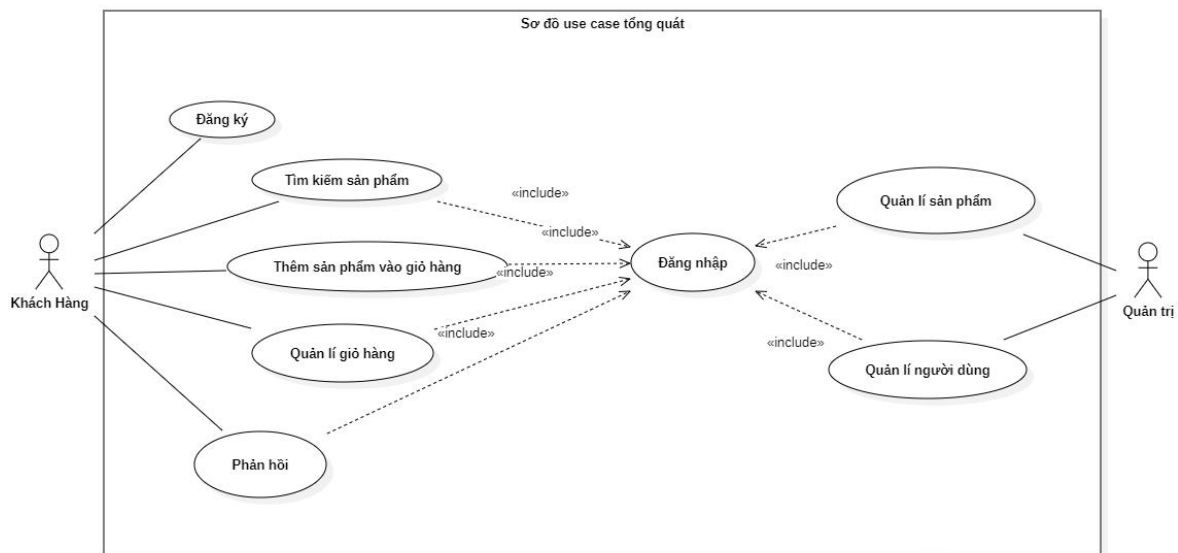
- Mongoose là một thư viện ODM (Object-Document Mapping) cho MongoDB và Node.js, cho phép lập trình viên dễ dàng tương tác với cơ sở dữ liệu MongoDB thông qua các đối tượng JavaScript.
- Mongoose cung cấp các tính năng như:
 - + **Validation:** Kiểm tra tính hợp lệ của dữ liệu trước khi lưu vào cơ sở dữ liệu.
 - + **Query building:** Xây dựng các truy vấn đến cơ sở dữ liệu MongoDB một cách dễ dàng.
 - + **Middleware:** Cho phép định nghĩa các hàm middleware được thực hiện trước hoặc sau các hoạt động như lưu hoặc xóa đối tượng.
 - + **Schema definition:** Cho phép định nghĩa các lược đồ (schema) cho các đối tượng được lưu trữ trong cơ sở dữ liệu.
 - + **Population:** Cho phép lấy dữ liệu từ các bảng liên quan đến nhau (references) trong cơ sở dữ liệu MongoDB.

Chương 3. Cài đặt giải pháp

1. Sơ đồ Use case

Sơ đồ Use case tổng quát của hệ thống

Sơ đồ 1. 1: Sơ đồ use case tổng quát



1.1. Đặc tả sơ đồ chức năng use case tổng quát

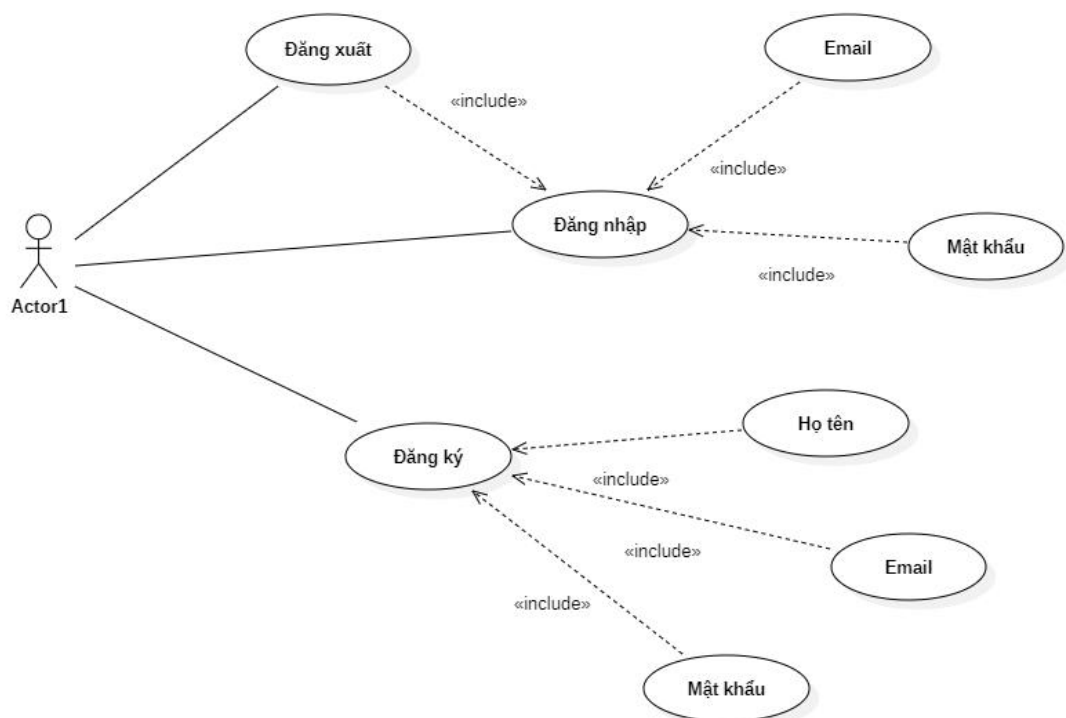
Bảng 1. 1: Đặc tả sơ đồ chức năng use case tổng quát

STT	Tên use case	Diễn giải
1	Đăng ký	Cho phép người dùng chưa có tài khoản đăng kí để xem và đặt hàng.
2	Đăng nhập	Đăng nhập vào website để có thể thao tác đầy đủ chức năng của website ..
3	Đăng xuất	Thoát khỏi tài khoản ra ra khỏi hệ thống.
4	Tìm kiếm sản phẩm	Giúp người dùng có thể tìm kiếm sản phẩm theo từ khóa bất kì.
5	Thêm vào giỏ hàng	Khách hàng có thể thêm các sản phẩm mà mình đặt vào giỏ hàng để dễ dàng thanh toán và đặt hàng.

6	Quản lý giỏ hàng	Khách hàng có thể xem và xóa sản phẩm
7	Phản hồi	Khách hàng có thể gửi góp ý đến cho cửa hàng để cửa hàng có chính sách thay đổi cho phù hợp.
8	Quản lý sản phẩm	Admin có thể thêm, sửa và xóa sản phẩm trên website. Có thể thêm các sản phẩm mới, điều chỉnh giá hoặc các chi tiết.
9	Quản lý tài khoản khách hàng	Admin có thể xóa tài khoản người dùng và tìm kiếm bằng địa chỉ email.

2. Sơ đồ use case đăng kí và đăng nhập

Sơ đồ 1. 2: Sơ đồ use case Đăng Ký và Đăng Nhập



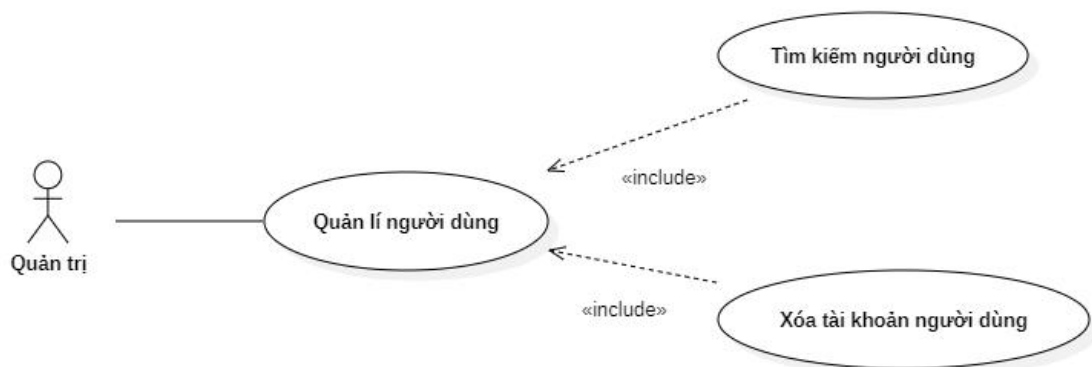
1.3 Sơ đồ Use case giỏ hàng

Sơ đồ 1. 3: Sơ đồ use case Giỏ Hàng



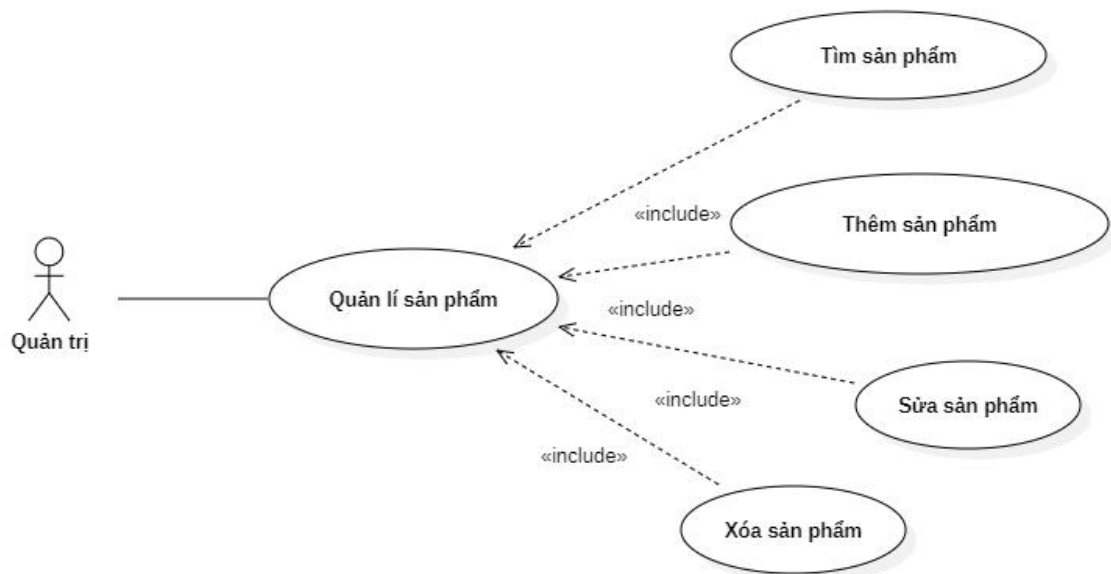
1.4 Sơ đồ use case quản lý tài khoản khách hàng

Sơ đồ 1. 4: Sơ đồ use Quản lí tài khoản khách hàng



1.5 Sơ đồ use case quản lý sản phẩm

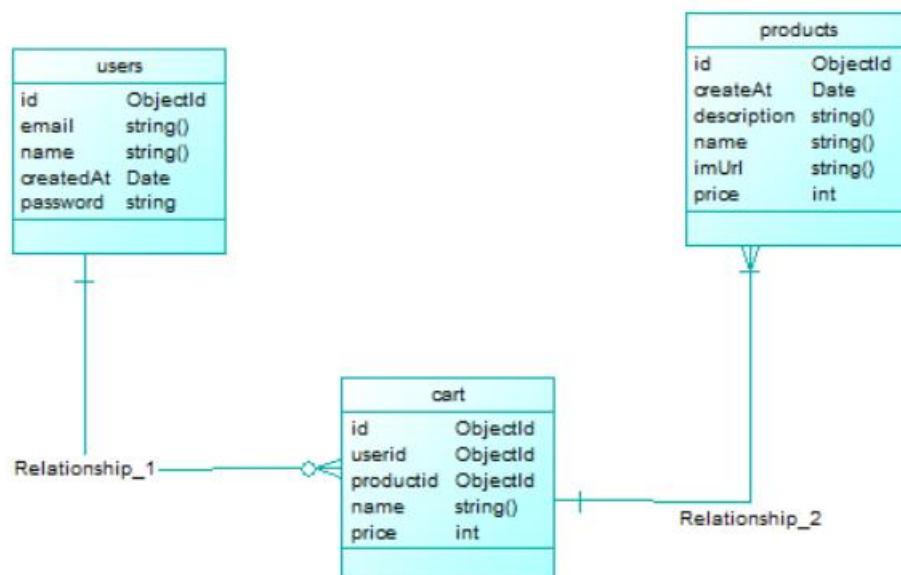
Sơ đồ 1. 5: Sơ đồ use case Quản lí sản phẩm



3. Cơ sở dữ liệu

Mô hình cơ sở dữ liệu CDM

Hình 1: Mô hình CDM



Hình : Mô hình CDM

Mô tả các lớp:

Bảng 2. 1: Mô tả lớp

STT	Tên bảng	Diễn giải
1	users	Danh sách tài khoản người dùng
2	product	Danh sách sản phẩm
3	Cart	Sản phẩm trong giỏ hàng

- Mô tả user:

Bảng 2. 2: Mô tả user

Tên thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Diễn giải
_id	ObjectId		ID của tài khoản
name	String		Họ và tên người dùng
email	String		Email
password	String		Mật khẩu
createdAt	Date		Ngày tạo tài khoản

- Mô tả product:

Bảng 2. 3: Mô tả sản phẩm

Tên thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Diễn giải
_id	ObjectId		Id kho hàng
createdAt	Date		Ngày tạo sản phẩm

			kho hàng
description	string		Mô tả
name	string		Tên sản phẩm
imageUrl	string		Hình ảnh
price	int		Giá

- Mô tả cart:

Bảng 2. 4: Mô tả Giỏ hàng

Tên thuộc tính	Kiểu dữ liệu	Giá trị mặc định	Diễn giải
_id	ObjectId		Id giỏ hàng
userid	ObjectId		Id của người dùng
productid	ObjectId		Id sản phẩm
name	string		Tên sản phẩm
price	Int		Giá sản phẩm

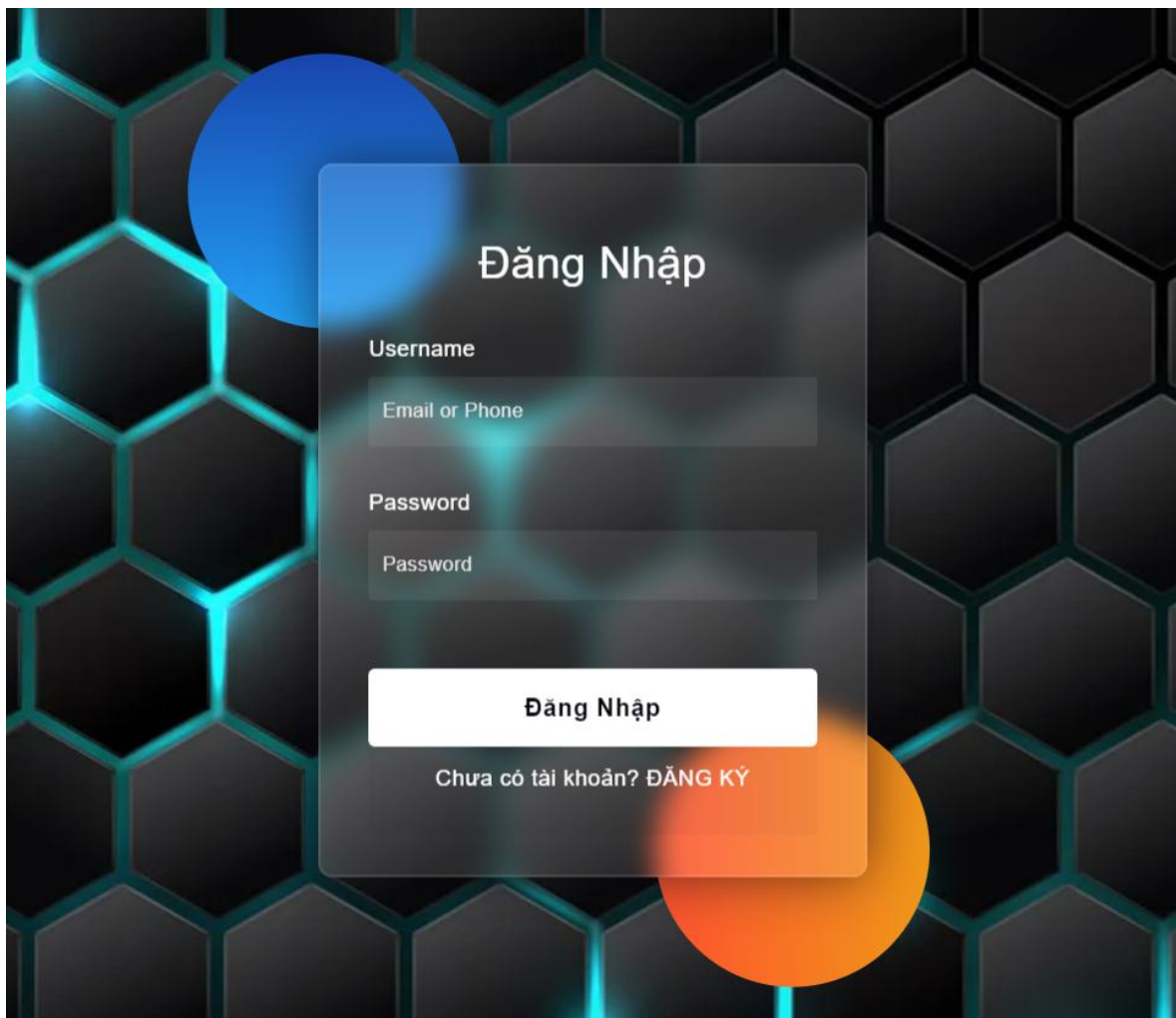
Chương 4. Kết quả đạt được

Website được xây dựng với giao diện thân thiện và đơn giản, mang tới trải nghiệm tốt cho khách hàng.

-Website ở phía khách hàng:

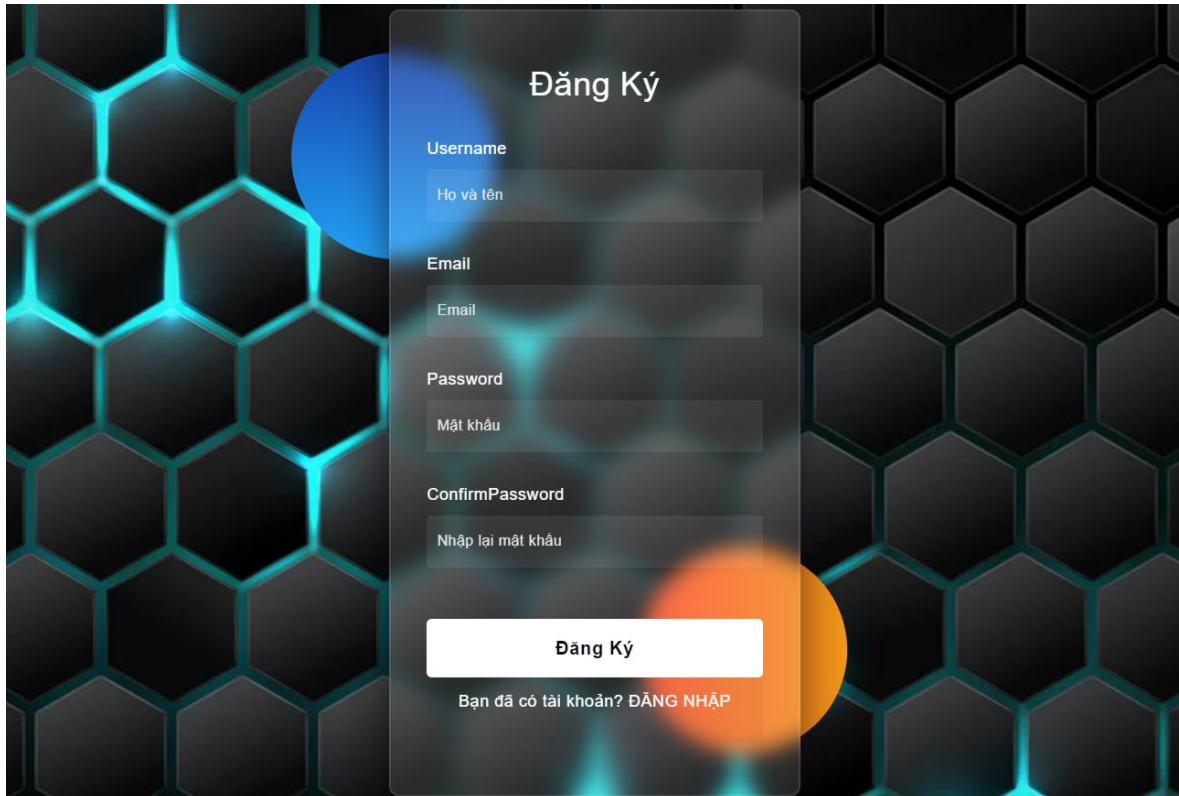
+ Trang đăng nhập:

Hình 2: Giao diện Đăng Nhập



+ Trang Đăng ký:

Hình 3: Giao diện Đăng Ký



Đăng Ký

Username

Ho và tên

Email

Email

Password

Mật khẩu

ConfirmPassword

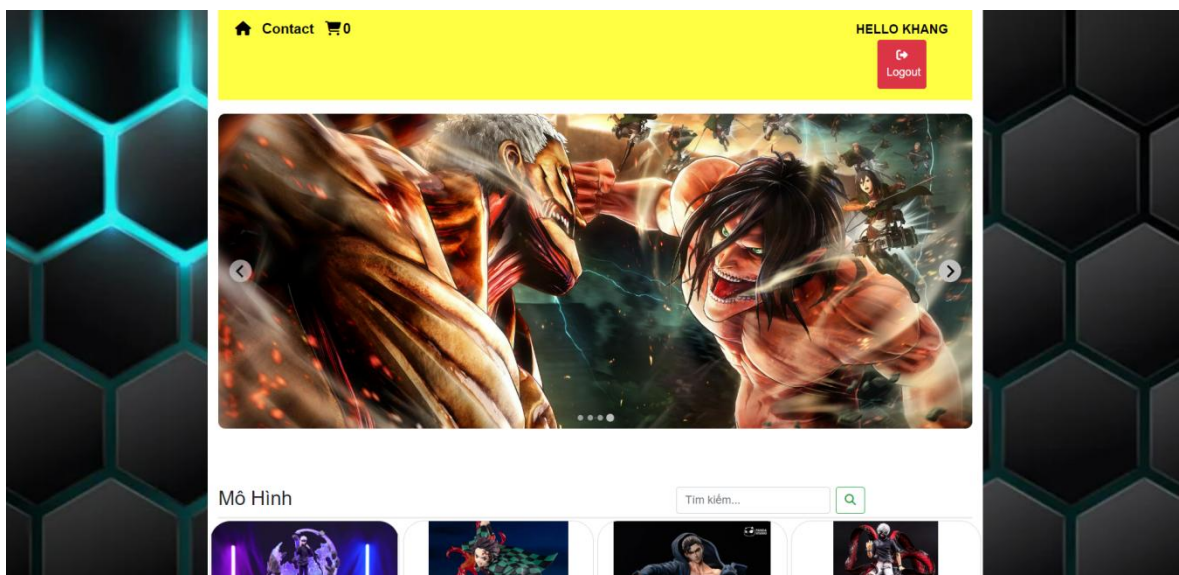
Nhập lại mật khẩu

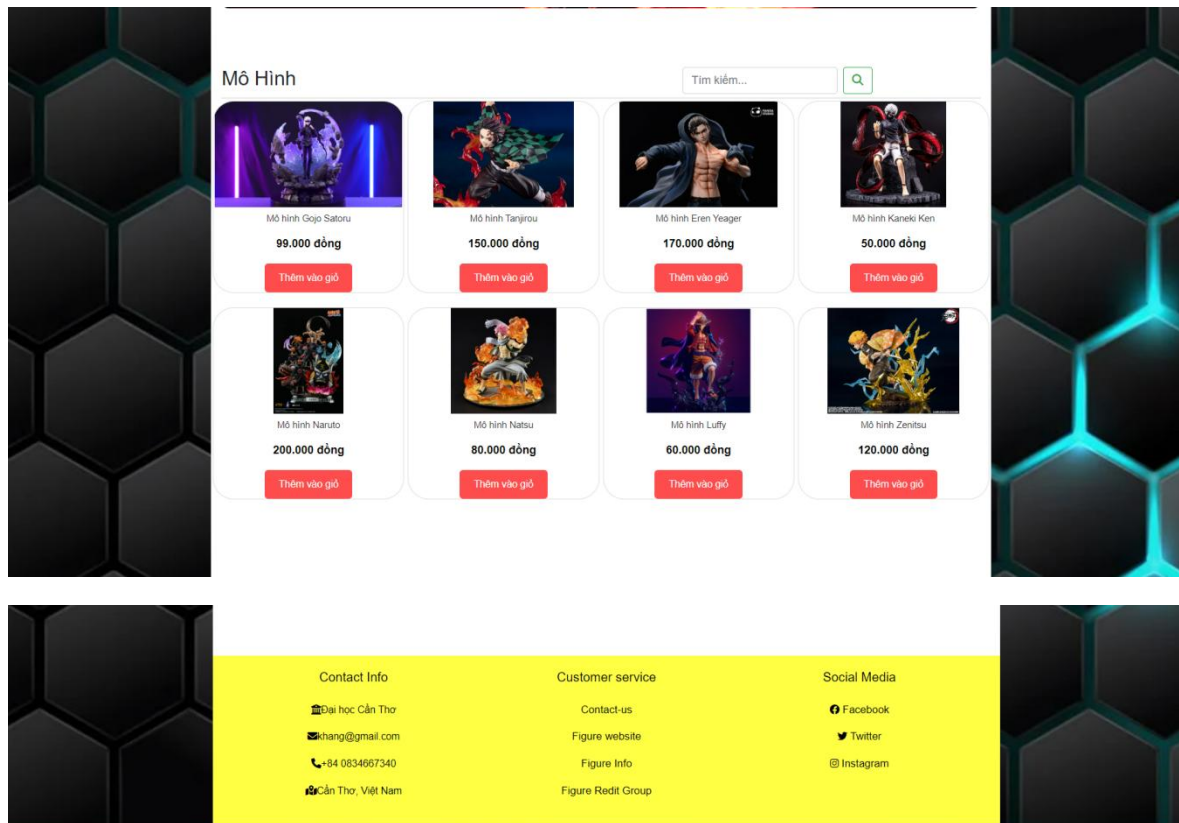
Đăng Ký

Bạn đã có tài khoản? **ĐĂNG NHẬP**

+ Trang chủ:

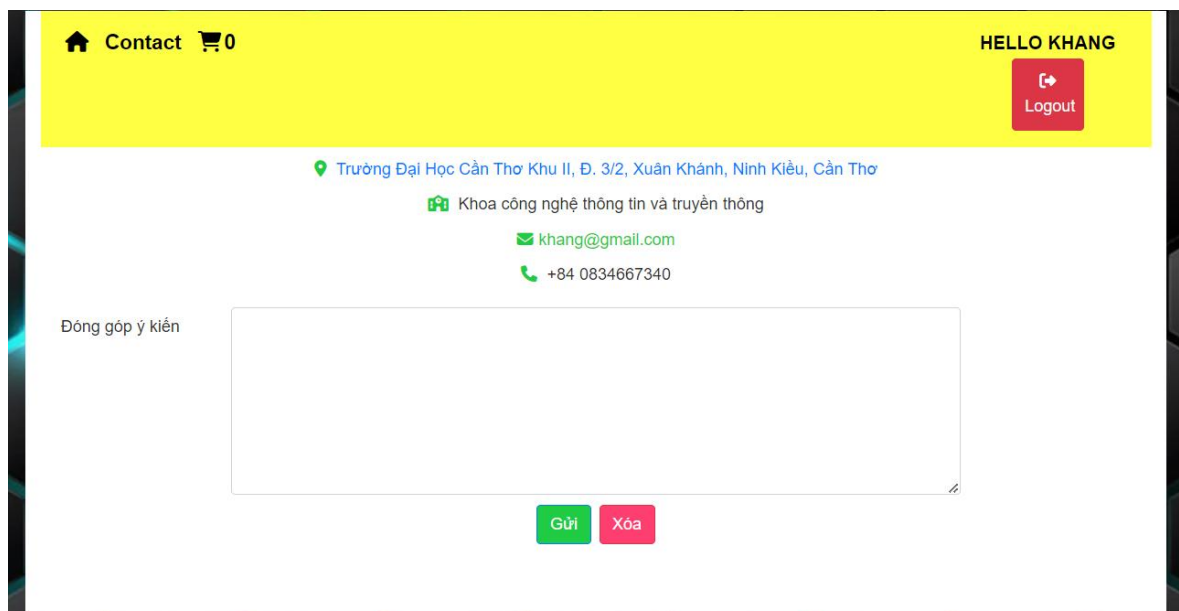
Hình 4: Giao diện trang chủ





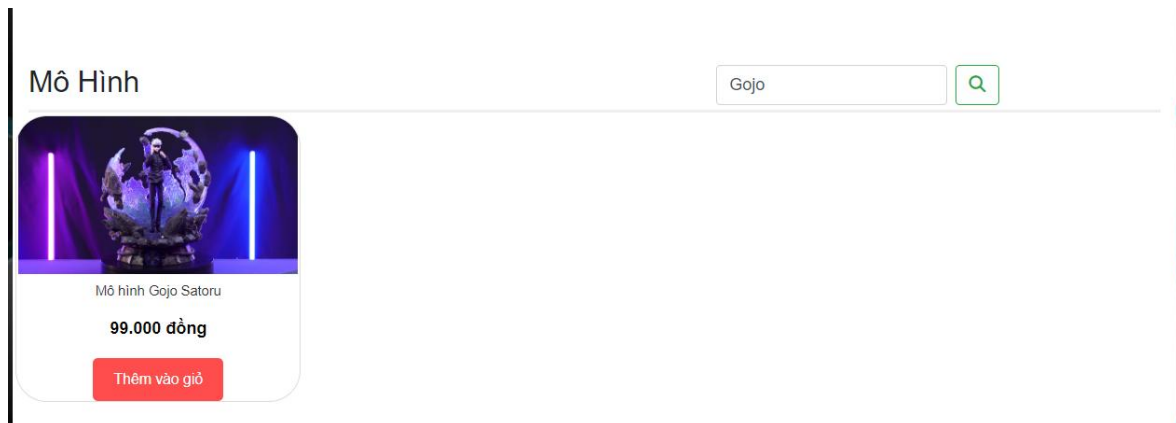
+ Trang Liên hệ:

Hình 5: Giao diện trang liên hệ



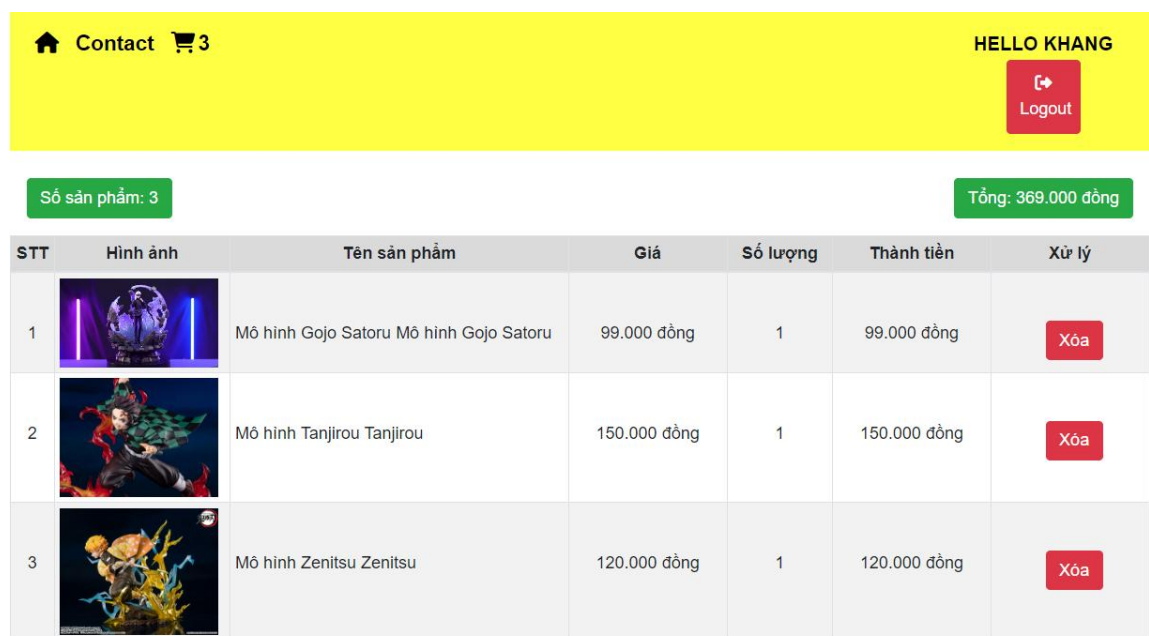
+ Trang Tìm kiếm sản phẩm:

Hình 6: Giao diện trang tìm kiếm



+ Trang Giỏ hàng:

Hình 7: Giao diện Giỏ hàng



-Website ở phía người quản lý (Admin):

+Giao diện quản lý sản phẩm :

Hình 8: Giao diện quản lý sản phẩm

<div> Product User </div>					
<div> <input type="text" value="Tìm kiếm..."/> <input type="button" value="Q"/> </div>			<div> <input type="button" value="+ Thêm mới"/> <input type="button" value="Xóa tất cả"/> </div>		
Hình ảnh	Tên sản phẩm	Giá	Mô tả	Ngày tạo/Cập nhật	Xử lý
	Mô hình Gojo Satoru	99.000 đồng	Mô hình Gojo Satoru	23:13:32 22/11/2023	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
	Mô hình Tanjiro	150.000 đồng	Tanjiro	03:11:06 24/11/2023	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
	Mô hình Eren Yeager	170.000 đồng	Eren	03:12:19 24/11/2023	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
	Mô hình Kaneki Ken	50.000 đồng	Kaneki	03:14:29 24/11/2023	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>
	Mô hình Naruto	200.000 đồng	Akatsuki	03:15:59 24/11/2023	<input type="button" value="Sửa"/> <input type="button" value="Xóa"/>

+ giao diện quản lý tài khoản khách hàng:

Hình 9: Giao diện quản lý người dùng

<div> <input type="button" value="Quay lại"/> <input type="text" value="Tìm kiếm..."/> <input type="button" value="Q"/> </div>					<div> <input type="button" value="Xóa tất cả"/> </div>
Vai trò	Tên	Email	Ngày tạo/Cập nhật	Xử lý	
user	khang	khang@gmail.com	20:27:46 22/11/2023	<input type="button" value="Xóa"/>	

Đường link đến github mã nguồn dự án:

https://github.com/HoangKhangSystem/B2017046_LEHOANGKHANG.git