

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



**LẬP TRÌNH WEB (CO3050)**

---

Báo cáo Bài tập lớn

**ONLINE SHOP  
DALLAS ORGANIC**

---

Giảng viên hướng dẫn: Nguyễn Đức Phúc  
Nhóm: L03\_1  
Trần Chí Công - 2010170  
Huỳnh Nguyên Phúc - 2110451  
Nguyễn Đặng Anh Khoa - 2010339  
Dương Phúc Thắng - 2112327

Email: [khoa.nguyenakaivn@hcmut.edu.vn](mailto:khoa.nguyenakaivn@hcmut.edu.vn)

Thành phố Hồ Chí Minh, tháng 4 năm 2023

# Danh sách thành viên

| STT | Họ và tên            | MSSV    | Nhiệm vụ   | Hoàn thành |
|-----|----------------------|---------|--|------------|
| 1   | Nguyễn Đặng Anh Khoa | 2010339 | <ul style="list-style-type: none"> <li>- Triển khai PHP server theo mô hình MVC.</li> <li>- Hiện thực giao diện phần trang chủ, cửa hàng, profile, blog.</li> <li>- Hiện thực bộ lọc sản phẩm, tính năng thêm sản phẩm vào giỏ hàng và đặt hàng, đăng tải bình luận, chỉnh sửa profile.</li> <li>- Hiện thực cơ sở dữ liệu: procedure, function, trigger.</li> </ul> | 100%       |
| 2   | Huỳnh Nguyên Phúc    | 2110451 | <ul style="list-style-type: none"> <li>- Tìm hiểu và trình bày cơ sở lý thuyết</li> <li>- SEO cho website</li> <li>- Hiện thực giao diện liên hệ, cửa hàng, sản phẩm.</li> <li>- Kiểm thử website.</li> </ul>  | 100%       |
| 3   | Trần Chí Công        | 2010170 | <ul style="list-style-type: none"> <li>- Hiện thực giao diện và tính năng cho quản trị viên.</li> <li>- Hiện thực giao diện giỏ hàng, đơn hàng.</li> <li>- Thiết kế ERD và hiện thực cơ sở dữ liệu: DDL, DML.</li> </ul>   | 100%       |

|   |                  |         |   |      |
|---|------------------|---------|---|------|
| 4 | Dương Phúc Thắng | 2112327 | <ul style="list-style-type: none"> <li>- Hiện thực giao diện và tính năng đăng nhập, đăng ký.</li> <li>- Tìm hiểu và trình bày cơ sở lý thuyết.</li> <li>- Kiểm thử website.</li> </ul> | 100% |
|---|------------------|---------|---|------|

# Mục lục

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Giới thiệu</b>  | <b>7</b>  |
| 1.1      | Khái niệm website doanh nghiệp . . . . .                   | 7         |
| 1.2      | Lợi ích khi thiết kế website doanh nghiệp . . . . .        | 7         |
| 1.2.1    | Đối với doanh nghiệp . . . . .                             | 7         |
| 1.2.2    | Đối với khách hàng . . . . .                               | 8         |
| 1.3      | Yêu cầu cần có của 1 website doanh nghiệp . . . . .        | 8         |
| 1.4      | Giới thiệu về Dallas Organic Online Shop Website . . . . . | 8         |
| <b>2</b> | <b>Cơ sở lý thuyết</b>                                     | <b>10</b> |
| 2.1      | Bootstrap 5 . . . . .                                      | 10        |
| 2.2      | AJAX . . . . .   | 11        |
| 2.3      | jQuery 3.4.1 . . . . .                                     | 11        |
| 2.3.1    | Giới thiệu về jQuery . . . . .                             | 11        |
| 2.3.2    | Giới thiệu về jQuery 3.4.1 . . . . .                       | 12        |
| 2.4      | REST API . . . . .   | 12        |
| 2.4.1    | Giới thiệu . . . . .                                       | 12        |
| 2.4.2    | Nguyên lý hoạt động . . . . .                              | 13        |
| 2.4.3    | Cách thức hoạt động . . . . .                              | 13        |
| 2.5      | Security . . . . .   | 14        |
| 2.6      | SEO . . . . .  | 14        |
| 2.6.1    | Giới thiệu . . . . .                                       | 14        |
| 2.6.2    | Ưu điểm và nhược điểm . . . . .                            | 15        |
| 2.6.2.1  | Ưu điểm . . . . .  | 15        |
| 2.6.2.2  | Hạn chế . . . . .  | 15        |
| 2.7      | Mô hình MVC . . . . .                                      | 15        |
| 2.7.1    | Giới thiệu . . . . .                                       | 15        |
| 2.7.2    | Ưu điểm và nhược điểm . . . . .                            | 17        |
| 2.7.2.1  | Ưu điểm . . . . .  | 17        |
| 2.7.2.2  | Hạn chế . . . . .  | 17        |
| 2.8      | Font Awesome 5.1 . . . . .                                 | 17        |
| 2.9      | Animate.css . . . . .                                      | 18        |
| 2.10     | WOW.js . . . . .   | 18        |
| 2.11     | Bootstrap Input Spinner . . . . .                          | 18        |
| 2.12     | Owl Carousel 2 . . . . .                                   | 19        |
| 2.13     | jquery.twbsPagination . . . . .                            | 19        |
| 2.14     | Chart.js . . . . .   | 20        |

|   |           |
|---|-----------|
| <b>3 Thiết kế</b>   | <b>21</b> |
| 3.1 Thiết kế cơ sở dữ liệu . . . . .  | 21        |
| 3.1.1 Lược đồ thực thể - mối liên kết (ERD) . . . . .                           | 21        |
| 3.1.2 Ánh xạ lược đồ thực thể-mối liên kết sang lược đồ cơ sở dữ liệu quan hệ . | 22        |
| 3.1.3 Hiện thực cơ sở dữ liệu. . . . .  | 23        |
| 3.2 Kiến trúc hệ thống . . . . .  | 26        |
| 3.3 Cấu trúc mã nguồn . . . . .   | 27        |
| <b>4 Hiện thực</b>  | <b>32</b> |
| 4.1 Đăng nhập, đăng ký và đăng xuất . . . . .                                   | 32        |
| 4.1.1 Đăng nhập . . . . .   | 32        |
| 4.1.2 Đăng xuất . . . . .   | 34        |
| 4.1.3 Đăng ký . . . . .   | 34        |
| 4.2 Danh sách sản phẩm, bộ lọc sản phẩm . . . . .                               | 37        |
| 4.3 Chi tiết sản phẩm, cho điểm đánh giá và đăng tải nhận xét . . . . .         | 41        |
| 4.4 Điều chỉnh giỏ hàng, Tạo đơn hàng . . . . .                                 | 44        |
| 4.5 Điều chỉnh profile . . . . .  | 47        |
| 4.6 Các tính năng quản lý dành cho quản trị viên . . . . .                      | 50        |
| <b>5 Demo</b>   | <b>56</b> |
| 5.1 Cài đặt môi trường . . . . .  | 56        |
| 5.2 Thiết lập các công cụ cần thiết . . . . .                                   | 56        |
| 5.3 Kết quả demo trang web . . . . .  | 57        |
| <b>Tài liệu tham khảo</b>   | <b>57</b> |

# Danh sách hình vẽ

|      |  |    |
|------|--|----|
| 1.1  | Nông trại hữu cơ Dallas Organic (Minh họa) . . . . .   | 8  |
| 2.1  | Sequence Diagram của AJAX . . . . .  | 11 |
| 2.2  | Cách thức hoạt động của REST API . . . . .   | 14 |
| 2.3  | Mô hình MVC . . . . .  | 16 |
| 3.1  | Lược đồ thực thể - mối liên kết cho cơ sở dữ liệu của hệ thống Dallas Organic . .  | 21 |
| 3.2  | Lược đồ cơ sở dữ liệu quan hệ cho cơ sở dữ liệu của hệ thống Dallas Organic . .  | 22 |
| 3.3  | Các bảng dữ liệu được tổ chức trong cơ sở dữ liệu . . . . .  | 23 |
| 3.4  | Mô hình kiến trúc MVC áp dụng cho hệ thống . . . . .   | 26 |
| 3.5  | Cấu trúc mã nguồn . . . . .  | 27 |
| 3.6  | Cấu trúc mã nguồn trong thư mục server . . . . .   | 28 |
| 3.7  | Cấu trúc mã nguồn trong thư mục client . . . . .   | 30 |
| 3.8  | Cấu trúc mã nguồn trong thư mục admin . . . . .  | 31 |
| 4.1  | Khi chưa đăng nhập, người dùng không thể truy cập vào giỏ hàng (khi nhấn biểu tượng giỏ hàng góc trên bên phải). Hiển thị popup yêu cầu đăng nhập hoặc đăng ký . | 33 |
| 4.2  | Mật khẩu không đúng định dạng (khi validate tại client-side hoặc server-side) .  | 33 |
| 4.3  | Xác thực không thành công (sau khi server trả về kết quả) . . . . .  | 34 |
| 4.4  | Sau khi đăng nhập thành công, người dùng có thể truy cập đến một số trang và sử dụng các tính năng cá nhân hóa . . . . .   | 34 |
| 4.5  | Biểu mẫu đăng ký tài khoản . . . . .   | 35 |
| 4.6  | Đăng ký tài khoản thành công . . . . .   | 36 |
| 4.7  | Tài khoản mới được thêm vào cơ sở dữ liệu . . . . .  | 36 |
| 4.8  | Validate dữ liệu thất bại (số điện thoại không hợp lệ) . . . . .   | 36 |
| 4.9  | Server trả về lỗi tài khoản đã tồn tại . . . . .   | 37 |
| 4.10 | Danh sách sản phẩm hiển thị với pagination . . . . .   | 39 |
| 4.11 | Áp dụng bộ lọc theo giá: nhỏ hơn 50.000đ . . . . .   | 40 |
| 4.12 | Áp dụng bộ lọc theo thể loại: thịt - hải sản . . . . .   | 40 |
| 4.13 | Áp dụng bộ lọc theo thể loại: thịt - hải sản và theo giá: nhỏ hơn 50.000đ. Không có sản phẩm nào thỏa mãn. . . . .   | 41 |
| 4.14 | Sắp xếp các sản phẩm rau củ hữu cơ theo giá tăng dần . . . . .   | 41 |
| 4.15 | Giao diện chi tiết sản phẩm . . . . .  | 42 |
| 4.16 | Người dùng điều chỉnh số lượng và nhấn nút Mua ngay để chuyển tiếp đến giỏ hàng đã thêm sản phẩm . . . . .   | 42 |
| 4.17 | Phản đánh giá sản phẩm . . . . .   | 43 |
| 4.18 | Kết quả sau khi người dùng đăng tải đánh giá của mình. Điểm đánh giá cũng được cập nhật (Hình 4.16) . . . . .  | 43 |

|   |    |
|---|----|
| 4.19 Kết quả sau khi người dùng đăng tải đánh giá của mình. Điểm đánh giá cũng<br>được cập nhật (Hình 4.16) . . . . . | 44 |
| 4.20 Điều chỉnh lượng số sản phẩm trong giỏ hàng . . . . .  | 44 |
| 4.21 Kết quả sau khi điều chỉnh lượng số sản phẩm trong giỏ hàng . . . . .  | 45 |
| 4.22 Thông báo đặt đơn hàng thành công . . . . .  | 45 |
| 4.23 Theo dõi danh sách đơn hàng . . . . .  | 46 |
| 4.24 Theo dõi cụ thể một đơn hàng đã đặt . . . . .  | 47 |
| 4.25 Profile cũ . . . . .   | 49 |
| 4.26 Update mật khẩu . . . . .  | 49 |
| 4.27 Upload avatar . . . . .  | 50 |
| 4.28 Chính sửa profile . . . . .  | 50 |
| 4.29 Chính sửa blog đã đăng . . . . .   | 51 |
| 4.30 Quản lý danh mục sản phẩm . . . . .  | 51 |
| 4.31 Quản lý danh sách đơn hàng . . . . .   | 52 |
| 4.32 Cập nhật trạng thái đơn hàng sang đã hoàn thành . . . . .  | 52 |
| 4.33 Chính sửa thông tin tài khoản . . . . .  | 53 |
| 4.34 Quản lý danh sách sản phẩm, có thể lọc theo thể loại . . . . .   | 53 |
| 4.35 Báo cáo doanh thu theo sản phẩm . . . . .  | 54 |
| 4.36 Báo cáo doanh thu theo danh mục sản phẩm . . . . .   | 54 |
| 4.37 Dashboard trực quan hóa số liệu thống kê kinh doanh (top sản phẩm bán chạy) .                                    | 55 |
| 4.38 Dashboard trực quan hóa số liệu thống kê kinh doanh (Tổng kết doanh thu) . . . . .                               | 55 |
| 5.1 Giao diện chính của trang web . . . . .   | 57 |

# Chương 1

## Giới thiệu

### 1.1 Khái niệm website doanh nghiệp

Về cơ bản, website doanh nghiệp có thể được hiểu là một tập hợp các trang web (web page) bao gồm văn bản, hình ảnh, video,... về doanh nghiệp nằm trên một tên miền trên World Wide Web của Internet. Website doanh nghiệp cung cấp đến người truy cập đầy đủ các thông tin về doanh nghiệp từ quá trình hoạt động, sản phẩm, dịch vụ cũng như các chương trình khuyến mại doanh nghiệp đang cung cấp. Với mỗi góc nhìn khác nhau, website doanh nghiệp lại có một định nghĩa khác:

- *Từ góc độ truyền thông:* Website doanh nghiệp là một công cụ quan trọng của doanh nghiệp, là nơi cung cấp đầy đủ các thông tin về doanh nghiệp giúp khách hàng, đối tác có thể hiểu rõ hơn về tính chất cũng như chất lượng sản phẩm, dịch vụ của doanh nghiệp.
- *Từ góc độ quản lý:* Website doanh nghiệp được tích hợp các công cụ giúp việc quản lý công việc trở nên đơn giản và dễ dàng hơn, nâng cao hiệu suất công việc, mở rộng không gian phát triển.

### 1.2 Lợi ích khi thiết kế website doanh nghiệp

#### 1.2.1 Đối với doanh nghiệp

- Tạo ra 1 kênh truyền thông trên mạng internet cho doanh nghiệp, là nơi phát ngôn các thông tin chính thức của doanh nghiệp, hạn chế các tin tức không chính xác của đối thủ cạnh tranh.
- Tạo cho doanh nghiệp 1 kênh quảng bá tự nhiên hiệu quả, hỗ trợ tích cực cho các hoạt động truyền thông, tăng hiệu suất và tiết kiệm chi phí cho doanh nghiệp.
- Tiếp cận nguồn khách hàng tiềm năng lớn.
- Nâng cao khả năng cạnh tranh với các đối thủ, rút ngắn khoảng cách giữa doanh nghiệp nhỏ và lớn.
- Hầu hết các thiết kế website doanh nghiệp hiện nay đều cung cấp đến doanh nghiệp khả năng tích hợp các phần mềm quản lý, giúp việc quản lý doanh nghiệp đơn giản, nâng cao hiệu suất cũng như tiết kiệm chi phí nhân công.

- Tăng doanh thu cho doanh nghiệp.
- Nâng cao chất lượng, độ uy tín của doanh nghiệp trong mắt khách hàng.

### 1.2.2 Đối với khách hàng

- Giúp việc tìm kiếm và tra cứu thông tin của khách hàng dễ dàng hơn.
- Cung cấp cho khách hàng 1 nguồn mua hàng uy tín, chất lượng, hạn chế tình trạng hàng giả, hàng nhái đang được bày bán trên thị trường.
- Giúp khách hàng hiểu rõ hơn về doanh nghiệp.

## 1.3 Yêu cầu cần có của 1 website doanh nghiệp

Dưới đây là các yêu cầu cần phải có cho việc thiết kế 1 website doanh nghiệp.

- Thiết kế đẹp, ấn tượng, đúng chuẩn bộ nhận diện thương hiệu
- Website doanh nghiệp phải có UX thuận tiện
- Website doanh nghiệp cần có nội dung chất lượng
- Website doanh nghiệp cần tối ưu SEO
- Website doanh nghiệp cần tương thích với nhiều thiết bị (Responsive Web Design - RWD).
- Website doanh nghiệp được quản trị dễ dàng
- Website doanh nghiệp phải được bảo mật

## 1.4 Giới thiệu về Dallas Organic Online Shop Website



Hình 1.1: Nông trại hữu cơ Dallas Organic (Minh họa)

Dallas Organic Online Shop Website là trang web thương mại điện tử được thiết kế để bày bán các mặt hàng cho nông trại Dallas Organic. Đây là một nông trại có điều kiện tự nhiên lí tưởng, trong lành, cùng với cơ sở vật chất và khoa học kỹ thuật tân tiến, đội ngũ chuyên gia hàng đầu cả nước, là nguồn cung cấp nông sản và thực phẩm sạch hàng đầu cả nước, với chất lượng không phải bàn, và vô cùng hợp vệ sinh, an toàn cho sức khỏe. Với giá trị cốt lõi là sạch sẽ, an toàn, chất lượng và thân thiện với môi trường, các sản phẩm của nông trại luôn được đánh giá rất cao từ người dùng, và hiện đã có mặt tại tất cả các siêu thị, cửa hàng tiện lợi trên toàn quốc.

Trang web cung cấp cho người dùng danh sách sản phẩm vô cùng chất lượng với giá cả vô cùng hợp lý, giúp người dùng có thể dễ dàng đặt mua hàng online mà không cần phải ra siêu thị hay cửa hàng tiện lợi. Ngoài ra, với việc trở thành một khách hàng của Dallas Organic, người dùng sẽ được tư vấn và nhận những lời khuyên trực tiếp của những chuyên gia hàng đầu về chế biến nông sản và sức khỏe của nông trại cũng như đọc các bài viết liên quan đến chủ đề được đăng tải trên diễn đàn của trang web, từ đó nâng cao chất lượng cuộc sống.

## Chương 2

# Cơ sở lý thuyết

Dưới đây là các công nghệ, framework và thư viện được nhóm sử dụng trong quá trình hiện thực website cho nông trại Dallas Organic.

### 2.1 Bootstrap 5

Bootstrap là một framework CSS miễn phí được phát triển bởi Twitter giúp người dùng phát triển web nhanh hơn và dễ dàng hơn. Bootstrap được sử dụng phổ biến vì nó giúp chúng ta dễ xây dựng các responsive website tương thích trên nhiều thiết bị khác nhau.

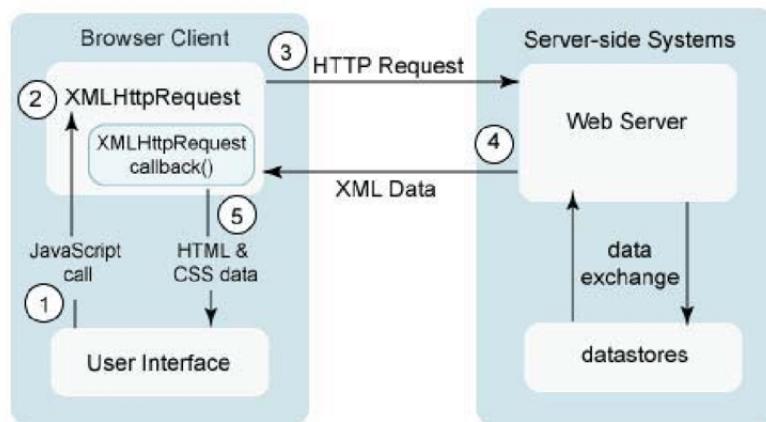
Phiên bản mới nhất của Bootstrap là Bootstrap 5, được ra mắt vào tháng 5 năm 2021. Bootstrap 5 được bổ sung thêm nhiều tính năng mới và cải tiến hơn so với các phiên bản trước đó, bao gồm:

- *Hỗ trợ tốt hơn cho CSS Grid:* Bootstrap 5 cải tiến việc sử dụng CSS Grid, giúp cho việc xây dựng bố cục trang web, xây dựng responsive web trở nên dễ dàng hơn.
- *Tính tương thích cao:* Bootstrap 5 được thiết kế để tương thích tốt với nhiều trình duyệt và thiết bị khác nhau, bao gồm nhiều loại điện thoại di động và máy tính bảng. Ngoài ra, Bootstrap 5 tương thích với tất cả các trình duyệt phổ biến hiện nay như Chrome, Firefox, Edge, Safari, Opera... Lưu ý rằng nếu bạn cần hỗ trợ có thể xem trên IE11 thì hãy sử dụng Bootstrap 4 hoặc Bootstrap 3.
- *Không còn sử dụng jQuery:* Bootstrap 5 đã bỏ đi jQuery và thay thế nó bằng Vanilla JavaScript. Điều này giúp tăng tốc độ tải trang web và giảm kích thước của các tệp JavaScript.
- *Cung cấp thêm nhiều components mới:* Bootstrap 5 cung cấp nhiều components mới bao gồm: toast, offcanvas, accordion, spinners, progress bar...
- *Thêm nhiều tiện ích cho biểu mẫu:* Bootstrap 5 cải tiến và cung cấp nhiều tiện ích hơn cho biểu mẫu, bao gồm trường nhập liệu kiểu mới và hỗ trợ cho các loại định dạng dữ liệu khác nhau.
- *Thiết kế đơn giản và tối giản:* Bootstrap 5 sử dụng thiết kế đơn giản và tối giản để giúp người dùng dễ dàng tùy chỉnh và điều chỉnh giao diện của trang web của mình.

## 2.2 AJAX

AJAX, viết tắt của Asynchronous JavaScript and XML, là một công nghệ phát triển web cho phép tải dữ liệu trang web mà không cần phải tải lại toàn bộ trang. Thay vì tải lại trang hoàn toàn, AJAX cho phép trang web tải và hiển thị dữ liệu mới trong phần nội dung chỉ cần một phần được tải lại. Điều này giúp giảm thời gian tải trang, không cần tải lại toàn bộ trang web. Điều đó sẽ làm cho trang web trở nên sinh động hơn, cải thiện chất lượng trải nghiệm của người dùng và tiết kiệm băng thông mạng, nhất là đối với những server vừa và nhỏ.

Công nghệ AJAX được sử dụng bằng cách sử dụng JavaScript để tương tác với máy chủ và đọc hoặc ghi dữ liệu từ hoặc đến máy chủ. Kết quả được trả về là dữ liệu XML hoặc JSON. AJAX cho phép các trang web được tạo ra mà không cần phải tải lại trang hoàn toàn, giúp cho người dùng cảm thấy thoải mái và thuận tiện hơn khi sử dụng trang web.



Hình 2.1: Sequence Diagram của AJAX

Với AJAX, các trang web có thể thực hiện các chức năng phức tạp một cách nhanh chóng và hiệu quả. Các chức năng như phân trang, lọc dữ liệu và tìm kiếm trên trang web có thể được thực hiện một cách nhanh chóng và hiệu quả, mà không cần phải tải lại trang.

## 2.3 jQuery 3.4.1

### 2.3.1 Giới thiệu về jQuery

jQuery là một thư viện JavaScript mã nguồn mở, được phát triển bởi John Resig vào năm 2006 và hiện nay đã trở thành một trong những thư viện JavaScript phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các dự án web lớn và nhỏ. Nó được thiết kế để tối ưu hóa việc tương tác giữa trình duyệt và các phần tử HTML trên trang web. Nó cung cấp các tính năng hữu ích giúp cho việc tối ưu hóa thao tác với HTML, CSS và DOM của trang web, đồng thời cải thiện tốc độ và hiệu suất của các ứng dụng web. Một số tính năng nổi bật của nó bao gồm:

- *Tương tác với các phần tử HTML và DOM*: jQuery giúp người phát triển dễ dàng truy vấn và thực hiện các thao tác hoặc sửa đổi các phần tử HTML trên trang web. Nó cung cấp các tính năng tiện lợi để truy cập các phần tử DOM và thao tác với chúng.
- *Hiệu ứng và tương tác trên trang web*: jQuery cung cấp một số hiệu ứng và các tính năng tương tác trên trang web như di chuyển, hiển thị ẩn, kéo thả và thay đổi kích thước của các phần tử. Nó cũng hỗ trợ tạo các sự kiện tương tác như click, hover hay scroll.
- *AJAX và đọc/gửi dữ liệu*: jQuery cung cấp một số tính năng AJAX để đọc và gửi dữ liệu đến máy chủ một cách dễ dàng. Điều này giúp cải thiện trải nghiệm của người dùng và tăng tốc độ tải trang.
- *Tích hợp với các thư viện khác*: jQuery có thể tích hợp với nhiều thư viện JavaScript khác như Bootstrap, AngularJS và React để tạo ra các ứng dụng web phức tạp. Điều này cũng giúp cải thiện trải nghiệm của người dùng.

Ngoài những tính năng trên, jQuery còn rất dễ học và dễ sử dụng. Nó cung cấp nhiều documentation và ví dụ minh họa trên trang web chính thức của nó ([Ấn vào đây để truy cập](#)), giúp cho người dùng mới bắt đầu có thể nhanh chóng tiếp cận và hiểu được cách sử dụng jQuery.

### 2.3.2 Giới thiệu về jQuery 3.4.1

Phiên bản jQuery mà nhóm sử dụng là 3.4.1, được phát hành vào ngày 1 tháng 5 năm 2019. Phiên bản này cung cấp nhiều tính năng mới và cải tiến so với phiên bản trước đó. Các tính năng chính của jQuery 3.4.1 bao gồm:

- *Sự cải tiến về hiệu suất*: jQuery 3.4.1 được cải tiến về hiệu suất, giúp cho các trang web và các ứng dụng web chạy nhanh hơn và mượt hơn.
- *Hỗ trợ cho ES6*: jQuery 3.4.1 bao gồm hỗ trợ cho các tính năng của ES6 như const và let, giúp cho việc phát triển các ứng dụng web trở nên dễ dàng hơn.
- *Cải tiến thêm cho AJAX*: jQuery 3.4.1 có nhiều cải tiến về AJAX, giúp cho việc tải dữ liệu trở nên nhanh hơn và dễ dàng hơn.
- *Hỗ trợ cho các trình duyệt mới*: jQuery 3.4.1 được phát triển để hỗ trợ cho các trình duyệt mới nhất, bao gồm Microsoft Edge và Firefox.

## 2.4 REST API

### 2.4.1 Giới thiệu

REST API, với REST là viết tắt của Representational State Transfer, API là viết tắt của Application Programming Interface, là một kiểu thiết kế phần mềm để phục vụ cho việc truyền tải dữ liệu giữa các ứng dụng web. REST API được sử dụng để cho phép các ứng dụng khác nhau giao tiếp với nhau thông qua giao thức HTTP (Hypertext Transfer Protocol) một cách hiệu quả và tiêu tốn ít tài nguyên.

REST API sử dụng định dạng dữ liệu phổ biến nhất hiện nay là JSON (JavaScript Object Notation) để truyền tải dữ liệu giữa các ứng dụng. Điều này giúp cho các ứng dụng có thể giao tiếp với nhau một cách dễ dàng và đáng tin cậy.

REST API được sử dụng rộng rãi trong các ứng dụng web hiện đại, từ các ứng dụng di động đến các ứng dụng web phức tạp. Nó cung cấp một phương tiện hiệu quả để các ứng dụng có thể chia sẻ và truy cập dữ liệu một cách an toàn và bảo mật.

#### 2.4.2 Nguyên lý hoạt động

REST API hoạt động dựa trên các nguyên lý cơ bản của kiến trúc REST. Các nguyên lý này bao gồm:

- *Statelessness*: REST API không lưu trữ bất kỳ thông tin trạng thái nào về phiên làm việc của người dùng. Thay vào đó, mọi thông tin về trạng thái phải được gửi đầy đủ thông qua request của người dùng.
- *Client-Server architecture*: REST API tách rời cơ chế lưu trữ dữ liệu và giao diện người dùng. Điều này cho phép các phần của hệ thống hoạt động độc lập với nhau.
- *Uniform interface*: REST API sử dụng một giao diện (interface) chung giữa các thành phần của hệ thống, đồng thời bao gồm các phương thức HTTP (GET, POST, PUT, DELETE) và định dạng truyền tải dữ liệu để cho phép các thành phần giao tiếp với nhau.
- *Cacheability*: REST API hỗ trợ caching dữ liệu để tăng tốc độ truy cập dữ liệu.
- *Layered system*: REST API cho phép có nhiều tầng trung gian giữa client và server để cải thiện tính bảo mật và tăng tính linh hoạt của hệ thống.

#### 2.4.3 Cách thức hoạt động

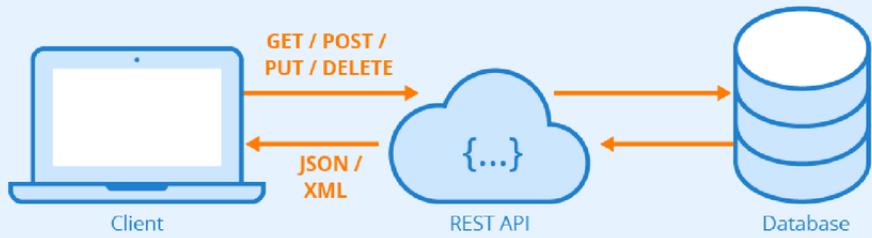
REST API hoạt động theo mô hình client-server, trong đó client và server sẽ tương tác với nhau bằng cách sử dụng giao thức HTTP (Hypertext Transfer Protocol) để truyền tải các request và phản hồi.

Các request của client được gửi đến server thông qua các phương thức HTTP như GET, POST, PUT hoặc DELETE. Mỗi yêu cầu này đều chứa đầy đủ các thông tin cần thiết để server có thể hiểu và thực hiện một hành động cụ thể trên tài nguyên được yêu cầu. Các thông tin này thường được định dạng theo các ngôn ngữ chuẩn như JSON (JavaScript Object Notation) hoặc XML (Extensible Markup Language).

Server sẽ xử lý yêu cầu và trả về phản hồi tương ứng thông qua các mã trạng thái HTTP như 200 OK (yêu cầu được xử lý thành công), 404 Not Found (không tìm thấy tài nguyên yêu cầu), hoặc 500 Internal Server Error (lỗi máy chủ)... Phản hồi này cũng thường được định dạng dưới dạng JSON hoặc XML để client có thể hiểu và xử lý kết quả trả về.

REST API được thiết kế để tối giản hóa việc lưu trữ dữ liệu trên server và cho phép client truy cập và tương tác với các tài nguyên này một cách dễ dàng và linh hoạt. Mỗi tài nguyên được đại diện bởi một URI (Uniform Resource Identifier), và các phương thức HTTP được sử dụng để thực hiện các hoạt động cơ bản trên tài nguyên này, bao gồm GET (lấy dữ liệu), POST (tạo mới tài nguyên), PUT (cập nhật tài nguyên), và DELETE (xóa tài nguyên).

# A Deep Look into RESTful APIs



Hình 2.2: Cách thức hoạt động của REST API

## 2.5 Security

Hiện Bảo mật trang web là một trong những yếu tố quan trọng nhất trong việc phát triển và duy trì một trang web. Với sự phát triển của công nghệ và mức độ phổ biến của Internet, việc bảo mật thông tin cá nhân của người dùng trên trang web đã trở nên cực kỳ cần thiết. Các vấn đề bảo mật trang web bao gồm việc bảo vệ dữ liệu của người dùng khỏi các cuộc tấn công mạng, đảm bảo tính toàn vẹn của trang web và đảm bảo tính bảo mật của các thông tin nhạy cảm được truyền tải trên không gian mạng.

Để bảo vệ trang web của mình, developer triển cần phải thực hiện nhiều biện pháp bảo mật khác nhau. Một trong những biện pháp quan trọng nhất là sử dụng các giao thức bảo mật như HTTPS và SSL để mã hóa thông tin truyền tải giữa máy tính của người dùng và máy chủ web. Ngoài ra, người phát triển cần thường xuyên cập nhật các phần mềm và bản vá bảo mật mới nhất để đảm bảo tính an toàn của trang web.

Một số biện pháp bảo mật khác bao gồm việc sử dụng các cơ chế xác thực và xác minh như hình ảnh CAPTCHA; sử dụng các mật khẩu mạnh để bảo vệ tài khoản người dùng; giới hạn quyền truy cập của người dùng trên trang web, sử dụng tường lửa để ngăn chặn những truy cập từ cái địa chỉ IP không cho phép. Đồng thời cũng cần phải thực hiện sao lưu thường xuyên để tránh việc bị mất dữ liệu, thường xuyên kiểm tra mã độc, kiểm tra tính bảo mật của trang web.

## 2.6 SEO

### 2.6.1 Giới thiệu

SEO, viết tắt của "Search Engine Optimization", là một quá trình tối ưu hóa trang web để cải thiện vị trí và thứ hạng của nó trên các trang kết quả tìm kiếm của các công cụ tìm kiếm như Google, Bing, Yahoo và nhiều công cụ tìm kiếm khác. SEO là một phần quan trọng của

kinh doanh trực tuyến hiện nay, vì nó giúp tăng khả năng tìm thấy của trang web của bạn trên Internet, từ đó giúp thu hút những khách hàng tiềm năng đến với trang web của bạn.

Các chiến lược SEO bao gồm cải thiện nội dung trang web để đảm bảo chúng có giá trị cho người dùng và đáp ứng nhu cầu của họ, tối ưu hóa các từ khóa để tăng cơ hội xuất hiện trên các kết quả tìm kiếm và xây dựng các liên kết trang web để cải thiện độ tin cậy và uy tín của trang web của bạn trong mắt của các công cụ tìm kiếm.

Một số công cụ SEO phổ biến bao gồm Google Analytics và Google Search Console. Nó cung cấp thông tin về lưu lượng truy cập, tìm kiếm từ khóa... để giúp bạn cải thiện chiến lược SEO của mình.

## 2.6.2 Ưu điểm và nhược điểm

### 2.6.2.1 Ưu điểm

- *Tăng lượng truy cập trang web:* SEO giúp tăng lượng truy cập trang web thông qua việc tối ưu hóa các từ khóa, thẻ tiêu đề và mô tả trang. Điều này giúp cải thiện vị trí của trang web trong kết quả tìm kiếm, dẫn đến sự tăng trưởng của lưu lượng truy cập.
- *Giảm chi phí quảng cáo:* Nếu website của bạn đạt được vị trí cao trong kết quả tìm kiếm tự nhiên thì nó sẽ giúp giảm chi phí quảng cáo vì bạn không cần tốn tiền cho Google AdWords hoặc các dịch vụ quảng cáo trả tiền khác.
- *Độ tin cậy cao:* Với vị trí tốt trong kết quả tìm kiếm, người dùng sẽ có sự tin tưởng và độ tin cậy cao hơn vào nội dung trang web của bạn. Điều đó sẽ giúp tăng doanh thu cho dự án của bạn.

### 2.6.2.2 Hạn chế

- *Kết quả chậm:* SEO là một quá trình dài và có thể mất nhiều tháng hoặc năm để đạt được kết quả tốt. Vì vậy, bạn cần phải kiên trì và có sự hiểu biết về cách thực hiện SEO.
- *Dộ phức tạp:* SEO là một quá trình phức tạp đòi hỏi người thực hiện cần có kiến thức kỹ thuật để thực hiện tối ưu hóa trang web. Điều này có thể khiến cho việc thực hiện SEO trở nên khó khăn đối với những người không có kiến thức, kỹ thuật.
- *Không có kết quả đảm bảo:* Mặc dù bạn có thực hiện tốt SEO, không có nghĩa là bạn có thể khiến cho vị trí của trang web của bạn trở nên cao hơn trong kết quả tìm kiếm. Các thuật toán của các công cụ tìm kiếm có thể thay đổi bất kỳ lúc nào và luôn luôn được cập nhật thường xuyên và dẫn đến mất vị trí của trang web của bạn. Vì thế cần phải luôn thường xuyên đảm bảo chất lượng cho trang web của bạn.

## 2.7 Mô hình MVC

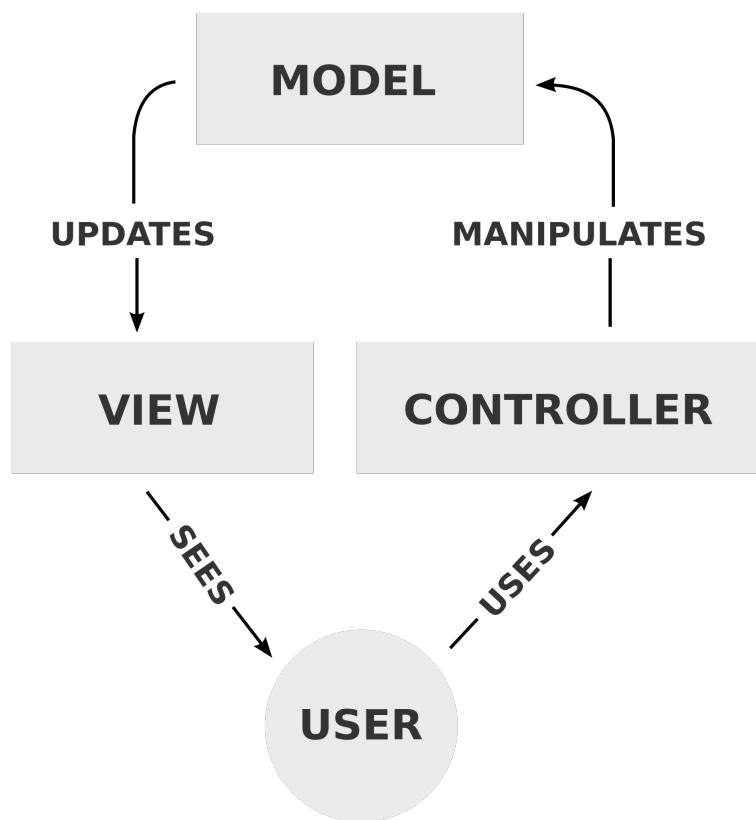
### 2.7.1 Giới thiệu

Mô hình MVC, viết tắt của Model-View-Controller là một kiến trúc phần mềm được sử dụng rộng rãi trong các ứng dụng web cũng như các ứng dụng di động. Nó được thảo luận lần đầu vào năm 1979 bởi Trygve Reenskaug. MVC được thiết kế để tách biệt các thành phần của ứng

dụng, giúp cho việc phát triển và bảo trì trở nên dễ dàng hơn.

MVC bao gồm ba thành phần chính: Model, View và Controller.

- **Model** đại diện cho dữ liệu và các thao tác với dữ liệu. Nó cung cấp các phương thức để truy xuất và cập nhật dữ liệu, và đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
- **View** đại diện cho giao diện người dùng. Nó hiển thị dữ liệu cho người dùng và cho phép người dùng tương tác với ứng dụng.
- **Controller** đóng vai trò điều khiển ứng dụng, xử lý tương tác của người dùng. Nó là cầu nối giữa phần Model và phần View. Controller làm nhiệm vụ xử lý các yêu cầu từ người dùng, tương tác với Model để truy xuất và cập nhật dữ liệu, và thay đổi giao diện người dùng (View) để phản ánh các thay đổi đó.



Hình 2.3: Mô hình MVC

Với MVC, các thành phần được tách biệt rõ ràng, cho phép các developers có thể làm việc độc lập với nhau. Nó cũng cho phép tái sử dụng code dễ dàng hơn, vì các thành phần có thể được sử dụng lại trong các ứng dụng khác nhau.

## 2.7.2 Ưu điểm và nhược điểm

### 2.7.2.1 Ưu điểm

- *Tách biệt các thành phần khác nhau của ứng dụng:* Với việc tách biệt Model, View và Controller, các developers có thể làm việc độc lập với nhau. Điều đó giúp cải thiện hiệu suất làm việc và tăng tính linh hoạt của ứng dụng.
- *Tái sử dụng code:* các thành phần của MVC có thể được sử dụng lại trong các ứng dụng khác, giảm thiểu thời gian và chi phí phát triển ứng dụng.
- *Dễ bảo trì:* Với các thành phần được tách biệt rõ ràng, việc bảo trì và sửa lỗi trở nên dễ dàng hơn.
- *Độ phân tầng cao:* MVC được sử dụng rộng rãi trong các ứng dụng lớn vì tính phân tầng cao của nó, giúp cho việc phát triển và mở rộng ứng dụng trở nên dễ dàng hơn.

### 2.7.2.2 Hạn chế

- *Độ phức tạp cao:* Với nhiều thành phần và liên kết giữa chúng, việc hiểu và triển khai MVC có thể khó khăn đối với những người mới bắt đầu hoặc những người không có năng lực tốt.
- *Thời gian phát triển tăng:* MVC có thể yêu cầu thời gian phát triển lâu hơn so với các mô hình khác, nhất là đối với các ứng dụng vừa và nhỏ.
- *Hiệu quả giảm:* Với các ứng dụng nhỏ và đơn giản, việc sử dụng MVC có thể không hiệu quả, vì độ phức tạp của nó có thể không cần thiết và tốn kém hơn so với mô hình khác.

## 2.8 Font Awesome 5.1

Font Awesome là một thư viện, toolkit dành cho các biểu tượng được phát triển bởi Dave Gandy. Phiên bản Font Awesome 5.1 là một bản cập nhật mới nhất của thư viện này, cung cấp cho các nhà phát triển web và thiết kế đồ họa hàng loạt biểu tượng bắt mắt và dễ sử dụng. Nó được sử dụng bởi hàng triệu designers, developers và người sáng tạo nội dung trên toàn thế giới.

Với Font Awesome 5.1, chúng ta có thể được sử dụng hơn 1.500 biểu tượng, bao gồm cả các biểu tượng động SVG, biểu tượng logo và biểu tượng trực quan. Chúng ta có thể dễ dàng thêm các biểu tượng các biểu tượng này vào trang web hoặc ứng dụng của mình, chỉ cần sao chép mã HTML hoặc CSS tương ứng của biểu tượng vào trang web.

Font Awesome cũng cung cấp một loạt các tính năng mới trong phiên bản 5.1, bao gồm tùy chỉnh tùy ý hoàn toàn cho các biểu tượng, khả năng tải xuống các biểu tượng theo yêu cầu và tính năng phân loại các biểu tượng để bạn có thể dễ dàng tìm kiếm chúng.

Một điểm nổi bật khác của Font Awesome là thư viện này rất dễ sử dụng và tích hợp vào các trang web và các ứng dụng. Với các công cụ như NPM, Bower và Rubygems, việc cài đặt và quản lý Font Awesome trở nên đơn giản hơn bao giờ hết.

## 2.9 Animate.css

Animate.css là một thư viện phổ biến được sử dụng để tạo các hiệu ứng animation trên trang web. Thư viện này được tạo ra và phát triển bởi Daniel Eden và những người bạn của anh ta.

Với hơn 70 hiệu ứng chuyển động khác nhau, Animate.css giúp cho các nhà phát triển web có thể tạo ra các hiệu ứng chuyển động đẹp mắt mà không cần phải viết mã JavaScript phức tạp. Các hiệu ứng được cung cấp bao gồm các hiệu ứng phát sáng, xoay, nhấp nháy, lật, phóng to/thu nhỏ...

Animate.css cũng hỗ trợ các trình duyệt phổ biến như Google Chrome, Firefox, Safari và Microsoft Edge. Ngoài ra, thư viện này cũng có thể tương thích với các framework CSS như Bootstrap, Foundation và Materialize.

## 2.10 WOW.js

WOW.js được phát triển bởi Matthieu Aussaguel, một kỹ sư lập trình web hiện đang sống và làm việc tại Thái Lan. WOW.js là sự kết hợp giữa jQuery và thư viện animation.css được viết sẵn các kịch bản. Khi scroll website thì jQuery sẽ bắt sự kiện và kích hoạt chuyển động, các đối tượng sẽ chuyển động và di chuyển với sự hỗ trợ của CSS 3 Animation và CSS 3 Transition.

Ngoài việc hỗ trợ các animation scroll-triggered, Wow.js cũng có thể được tùy chỉnh để phù hợp với các nhu cầu thiết kế của người dùng. Wow.js cung cấp các tùy chọn để điều chỉnh tốc độ, thời gian trễ và các cài đặt khác cho các animation. Ngoài ra, Wow.js cũng có thể được sử dụng để kích hoạt các animation bằng cách sử dụng các sự kiện khác như nhấp chuột hoặc di chuyển chuột. Wow.js là một thư viện JavaScript mã nguồn mở và miễn phí sử dụng cho các dự án web.

Một số tiện ích của WOW.js có thể kể đến:

- Tự động phát hiện các phần tử khi cuộn trang: Wow.js sử dụng kỹ thuật phát hiện khi một phần tử trên trang được cuộn đến vị trí được xác định để kích hoạt animation.
- Hỗ trợ đa trình duyệt: Wow.js tương thích với các trình duyệt phổ biến như Chrome, Firefox, Safari và Internet Explorer.
- Đơn giản để sử dụng: Người dùng có thể sử dụng Wow.js một cách dễ dàng bằng cách chỉ định lớp "wow" cho các phần tử mà họ muốn kích hoạt animation.
- Dễ tùy chỉnh: Wow.js cho phép người dùng tùy chỉnh các tham số như tốc độ và thời gian trễ của animation để phù hợp với thiết kế của trang web.
- Nhẹ và nhanh: Wow.js được viết bằng JavaScript thuần túy và có kích thước nhỏ, giúp giảm thời gian tải trang và tăng trải nghiệm của người dùng.

## 2.11 Bootstrap Input Spinner

Bootstrap Input Spinner là một plugin JavaScript miễn phí cho Bootstrap, giúp tạo ra các phần tử input số với các nút tăng/giảm giá trị, cho phép người dùng dễ dàng thay đổi giá trị của input một cách chính xác và thuận tiện hơn. Plugin này cung cấp một giao diện đơn giản và dễ sử

dụng cho người dùng, và có thể được tùy chỉnh để phù hợp với các nhu cầu thiết kế của trang web.

Bootstrap Input Spinner cung cấp một số tính năng hữu ích như:

- Tích hợp với Bootstrap: Plugin này được thiết kế để tích hợp với Bootstrap, do đó người dùng có thể sử dụng nó một cách dễ dàng trong các dự án Bootstrap của mình.
- Tùy chỉnh linh hoạt: Bootstrap Input Spinner cho phép người dùng tùy chỉnh nhiều thông số khác nhau, bao gồm giá trị tối thiểu/tối đa, bước tăng/giảm, đơn vị tiền tệ và nhiều hơn nữa.
- Hỗ trợ các sự kiện: Plugin này cung cấp các sự kiện JavaScript để người dùng có thể thực hiện các hành động như khi giá trị được thay đổi, khi nút tăng/giảm được nhấn và khi giá trị đạt giá trị tối thiểu/tối đa.
- Dễ dàng sử dụng: Bootstrap Input Spinner có thể được triển khai một cách đơn giản bằng cách chỉ cần thêm một vài dòng mã HTML và JavaScript.

## 2.12 Owl Carousel 2

Owl Carousel 2 là một thư viện được sử dụng để tạo ra các slideshow hoặc carousel trên trang web của bạn một cách dễ dàng và hiệu quả. Owl Carousel 2 có biểu tượng là hình một con cú ẩn mình dưới lớp rèm. Thư viện này được xây dựng trên nền tảng của jQuery và Bootstrap, với nhiều tính năng và tùy chọn cấu hình để bạn có thể tùy chỉnh carousel của mình theo ý muốn.

Owl Carousel 2 cung cấp cho người dùng một giao diện trực quan và dễ sử dụng. Với các lựa chọn tùy chỉnh dễ dàng để thay đổi kích thước, tốc độ, hình dạng và kiểu dáng của các mục trong carousel, bạn có thể tạo ra các hiệu ứng hoạt hình độc đáo và phù hợp với phong cách thiết kế của trang web của mình.

Các tính năng chính của Owl Carousel 2 bao gồm:

- *Hỗ trợ responsive design:* Cho phép người dùng tùy chỉnh carousel để phù hợp với kích thước của các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động hay máy tính bảng.
- *Hỗ trợ các hiệu ứng chuyển động đa dạng:* Owl Carousel 2 sử dụng các giải pháp tăng tốc phần cứng và CSS3 Translate3d giúp cho các hiệu ứng animation trở nên rất mượt và thu hút người dùng. Không những thế Owl còn cung cấp rất nhiều sự lựa chọn khác nhau về animation cho người dùng.
- *Hỗ trợ các nút điều khiển và thanh trượt để điều khiển carousel:* Giúp cho người dùng có thể dễ dàng tương tác với carousel của họ, tăng trải nghiệm cho người dùng.
- *Dễ dàng tùy chỉnh và cấu hình:* Với các tùy chọn cấu hình đơn giản và dễ hiểu, người dùng có thể thay đổi các thuộc tính của carousel một cách nhanh chóng và dễ dàng.

## 2.13 jquery.twbsPagination

jquery.twbsPagination là một plugin JavaScript miễn phí dành cho jQuery, giúp tạo ra các phân trang (pagination) trên trang web. Plugin này cung cấp một giao diện đơn giản và dễ sử

dụng, cho phép người dùng tùy chỉnh các thông số như số trang hiển thị, số trang tối đa, định dạng các nút phân trang, v.v.

Một số tính năng của jquery.twbsPagination bao gồm:

- Tích hợp với Bootstrap: Plugin này được thiết kế để tích hợp với Bootstrap, do đó người dùng có thể sử dụng nó một cách dễ dàng trong các dự án Bootstrap của mình.
- Tùy chỉnh linh hoạt: jquery.twbsPagination cho phép người dùng tùy chỉnh nhiều thông số khác nhau, bao gồm số trang hiển thị, số trang tối đa, định dạng các nút phân trang, v.v.
- Hỗ trợ các sự kiện: Plugin này cung cấp các sự kiện JavaScript để người dùng có thể thực hiện các hành động như khi trang được thay đổi, khi nút phân trang được nhấn, v.v.
- Dễ dàng sử dụng: jquery.twbsPagination có thể được triển khai một cách đơn giản bằng cách chỉ cần thêm một vài dòng mã HTML và JavaScript.

## 2.14 Chart.js

Chart.js là một thư viện JavaScript miễn phí cho việc tạo các biểu đồ trên trang web. Thư viện này sử dụng HTML5 Canvas để vẽ biểu đồ và hỗ trợ nhiều loại biểu đồ, bao gồm đường, tròn, thanh, bánh, v.v. Chart.js cung cấp một giao diện đơn giản và dễ sử dụng, cho phép người dùng tùy chỉnh các thông số như màu sắc, chủ đề, tiêu đề, v.v.

Một số tính năng của Chart.js bao gồm:

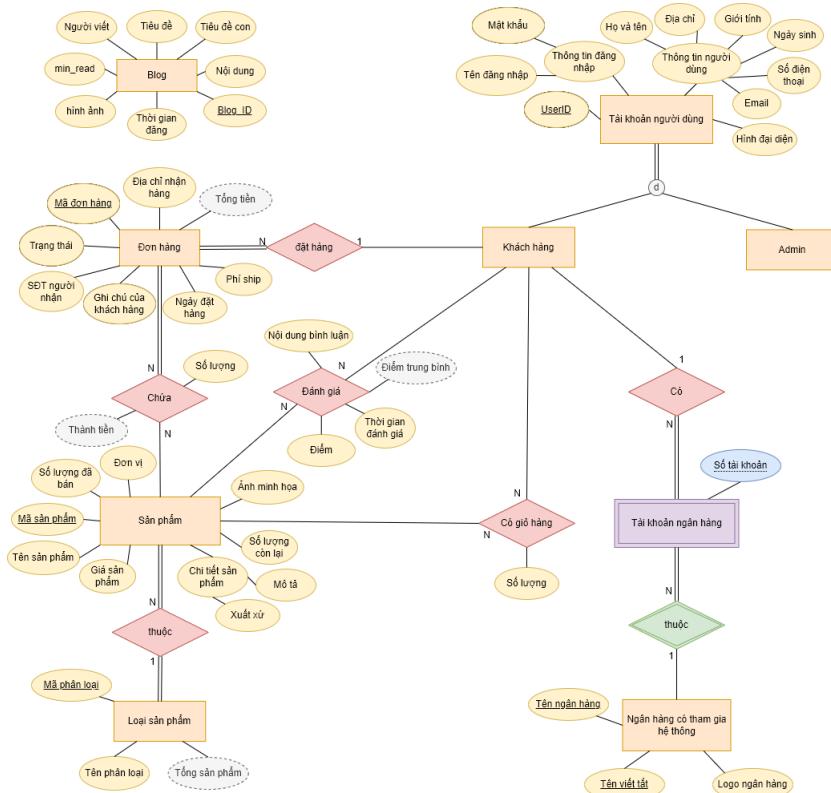
- Hỗ trợ nhiều loại biểu đồ: Chart.js hỗ trợ nhiều loại biểu đồ khác nhau, bao gồm đường, tròn, thanh, bánh, v.v.
- Tùy chỉnh linh hoạt: Thư viện này cung cấp rất nhiều tính năng tùy chỉnh để người dùng có thể tùy biến và điều chỉnh biểu đồ theo ý muốn. Người dùng có thể điều chỉnh các thông số như màu sắc, kích thước, chủ đề, tiêu đề, v.v.
- Dữ liệu đa dạng: Chart.js có thể nhận dữ liệu từ nhiều nguồn khác nhau, bao gồm các tệp JSON hoặc API trên trang web. Điều này giúp người dùng có thể tạo các biểu đồ động và dễ dàng cập nhật dữ liệu.
- Dễ dàng sử dụng: Chart.js có thể được triển khai một cách đơn giản bằng cách chỉ cần thêm một vài dòng mã JavaScript và HTML.

# Chương 3

## Thiết kế

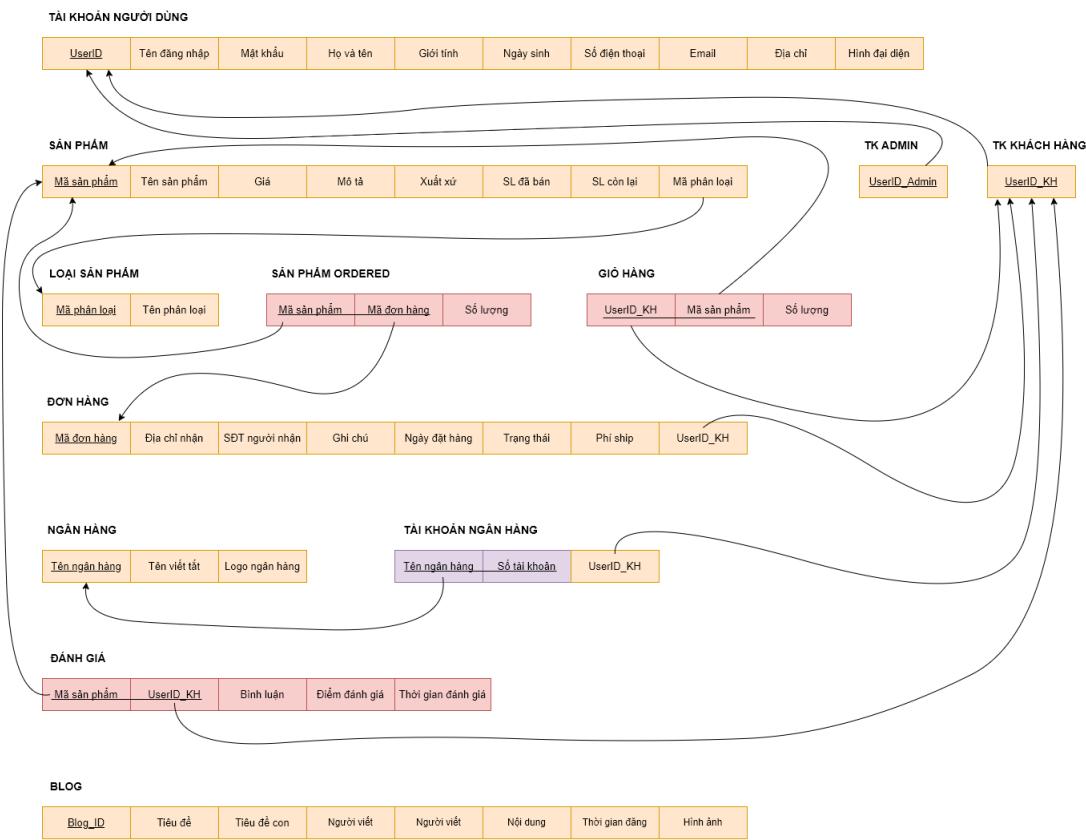
### 3.1 Thiết kế cơ sở dữ liệu

#### 3.1.1 Lược đồ thực thể - mối liên kết (ERD)



Hình 3.1: Lược đồ thực thể - mối liên kết cho cơ sở dữ liệu của hệ thống Dallas Organic

### 3.1.2 Ánh xạ lược đồ thực thể-mối liên kết sang lược đồ cơ sở dữ liệu quan hệ



Hình 3.2: Lược đồ cơ sở dữ liệu quan hệ cho cơ sở dữ liệu của hệ thống Dallas Organic

### 3.1.3 Hiện thực cơ sở dữ liệu.

Xuất phát từ lược đồ cơ sở dữ liệu quan hệ đã được thiết kế dành cho hệ thống như trên, ta tiến hành hiện thực cơ sở dữ liệu sử dụng cho hệ thống web Dallas Organic bằng hệ quản trị cơ sở dữ liệu MySQL. Cơ sở dữ liệu gồm những bảng được thể hiện trong hình sau:

| Bảng             | Hành động |
|------------------|-----------|
| admin_account    |           |
| banks            |           |
| bank_account     |           |
| blogs            |           |
| cart             |           |
| category         |           |
| customer_account |           |
| feedback         |           |
| ordered_product  |           |
| orders           |           |
| products         |           |
| user_account     |           |

Hình 3.3: Các bảng dữ liệu được tổ chức trong cơ sở dữ liệu

#### 1. Bảng user\_account:

Bảng **user\_account** chứa thông tin tài khoản của toàn bộ người dùng sử dụng hệ thống, bao gồm các trường:

- **id** (unique): mã định danh của tài khoản.
- **username**: tên tài khoản của người dùng, cũng được sử dụng làm tên đăng nhập.
- **password**: mật khẩu tài khoản người dùng.
- **fullname**: họ và tên thật của chủ tài khoản.
- **sex**: giới tính của chủ tài khoản.
- **DoB**: ngày sinh của chủ tài khoản.
- **phonenumer**: số điện thoại người dùng.
- **email**: địa chỉ email người dùng.
- **address**: địa chỉ nhà của người dùng.
- **avatar**: link hình ảnh mà người dùng mong muốn đặt làm ảnh đại diện.

Các tài khoản trong hệ thống sẽ thuộc một và chỉ một trong hai vai trò là: khách hàng (Customer) và quản trị viên (Admin). Để phân biệt hai loại tài khoản này trong cơ sở dữ liệu, ta có hai bảng dữ liệu sau:

- Bảng **admin\_account** có trường **userID\_admin** lưu mã định danh của tất cả các tài khoản có vai trò Admin trong hệ thống, được tham chiếu đến trường **id** của bảng **user\_account**.
- Bảng **customer\_account** có trường **userID\_customer** lưu mã định danh của tất cả các tài khoản có vai trò Customer trong hệ thống, được tham chiếu đến trường **id** của bảng **user\_account**.

2. Bảng **category**:

Bảng **category** chứa thông tin các loại sản phẩm có trong cửa hàng, có các trường:

- **id** (unique): mã định danh của phân loại sản phẩm.
- **cate\_name**: tên phân loại sản phẩm.

3. Bảng **products**:

Bảng **products** chứa thông tin của các sản phẩm được bày bán trên hệ thống. Các trường thông tin trong bảng như sau:

- **id** (unique): mã định danh của sản phẩm.
- **product\_name**: tên sản phẩm.
- **price**: giá sản phẩm.
- **unit**: đơn vị đo của sản phẩm (ví dụ: kg, túi 300G, hộp, ...).
- **description**: mô tả sản phẩm.
- **origin**: nguồn gốc sản phẩm.
- **sold\_number**: số lượng sản phẩm đã bán.
- **remain\_number**: số lượng sản phẩm còn lại trong kho.
- **image**: link ảnh minh họa của sản phẩm.
- **category\_id**: mã định danh của phân loại sản phẩm, được tham chiếu đến trường **id** của bảng **category**.

4. Bảng **banks**:

Bảng **banks** chứa thông tin của các ngân hàng có tham gia vào hệ thống.

- **bank\_name** (unique): tên ngân hàng.
- **bank\_shortname**: tên viết tắt của ngân hàng.
- **bank\_logo**: link logo của ngân hàng.

5. Bảng **bank\_account**:

Bảng **bank\_account** chứa thông tin tài khoản được liên kết của các tài khoản khách hàng, bao gồm các trường:

- **bank\_name**: tên ngân hàng, được tham chiếu đến trường **bank\_name** của bảng **banks**.
- **acc\_number**: số tài khoản ngân hàng của khách hàng.

- **customerID**: chứa mã định danh của tài khoản khách hàng tương ứng với tài khoản ngân hàng được liên kết, tham chiếu tới trường **userID\_customer** của bảng **customer\_account**.

#### 6. Bảng **blogs**:

Bảng **blogs** chứa thông tin các bài đăng của admin lên hệ thống, gồm các trường như sau:

- **id** (unique): mã định danh của bài blog.
- **title**: tiêu đề của bài blog.
- **subtitle**: tiêu đề phụ của bài blog.
- **created\_by**: người đăng bài, mặc định là "admin".
- **min\_read**: thời gian tối thiểu để đọc bài blog.
- **content**: nội dung bài blog.
- **created\_at**: thời gian đăng bài.
- **image**: link hình ảnh chủ đạo của bài đăng.

#### 7. **cart**:

Bảng **cart** chứa thông tin giỏ hàng của tài khoản người dùng, thông tin các trường:

- **userID**: mã định danh tài khoản người dùng, tham chiếu đến trường **id** của **user\_account**.
- **productID**: mã định danh của sản phẩm, tham chiếu đến trường **id** của bảng **products**.
- **quantity**: số lượng sản phẩm.

#### 8. Bảng **orders**:

Bảng **orders** chứa thông tin đơn hàng của người dùng:

- **id** (unique): mã định danh của đơn hàng.
- **recieve\_address**: địa chỉ nhận hàng.
- **recieve\_phonenum**: số điện thoại người nhận.
- **note**: ghi chú của khách hàng.
- **order\_date**: ngày đặt hàng.
- **order\_status**: tình trạng đơn hàng.
- **ship\_fee**: phí vận chuyển của đơn hàng.
- **userID\_ordcus**: mã định danh của tài khoản người dùng mua hàng, tham chiếu đến trường **userID\_customer** của bảng **customer\_account**.

#### 9. Bảng **ordered\_product**:

Bảng **ordered\_product** chứa thông tin các sản phẩm được đặt trong đơn hàng:

- **productID**: mã định danh của sản phẩm, tham chiếu đến trường **id** của bảng **products**.
- **orderID**: mã định danh của đơn hàng, tham chiếu đến trường **id** của bảng **orders**.
- **quantity**: số lượng sản phẩm.

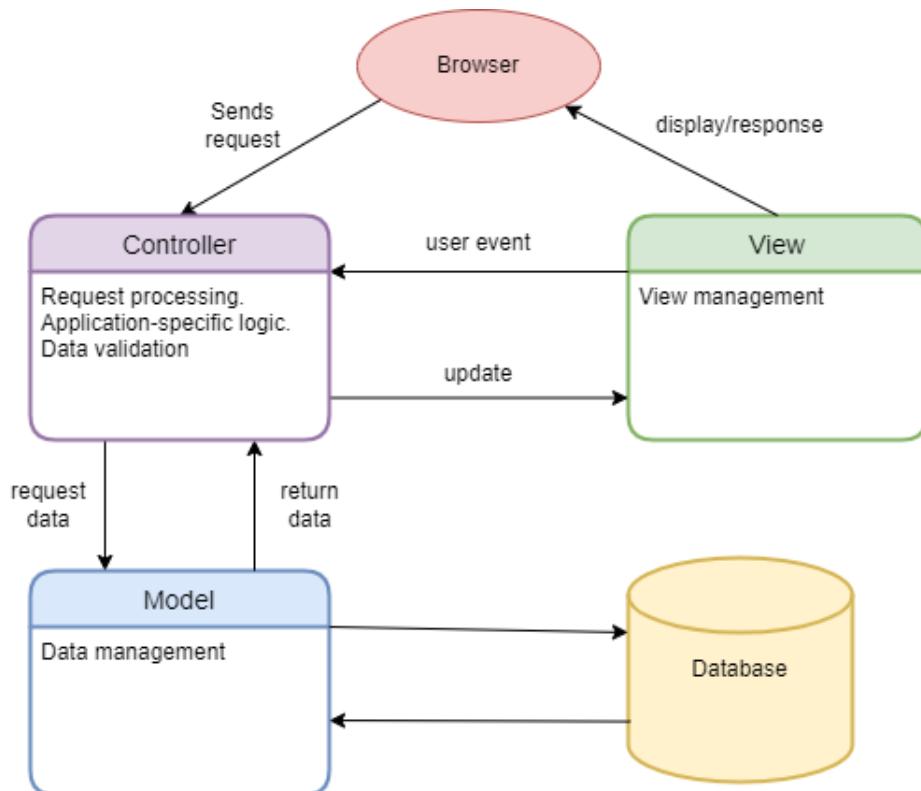
#### 10. Bảng feedback:

Bảng **feedback** bao gồm thông tin đánh giá của khách hàng đối với sản phẩm, bao gồm các trường:

- **productID**: mã định danh của sản phẩm, tham chiếu đến trường **id** của bảng **products**.
- **customer**: mã định danh của tài khoản người dùng, tham chiếu đến trường **userID\_customer** của bảng **customer\_account**.
- **comment**: nội dung bình luận của khách hàng.
- **rating**: điểm đánh giá của khách hàng.
- **feedback\_datetime**: thời gian khách hàng đánh giá.

### 3.2 Kiến trúc hệ thống

Để hiện thực hệ thống như mong muốn, nhóm chọn hướng tiếp cận kiến trúc theo mô hình kiến trúc MVC (Model - Controller - View) như sau:



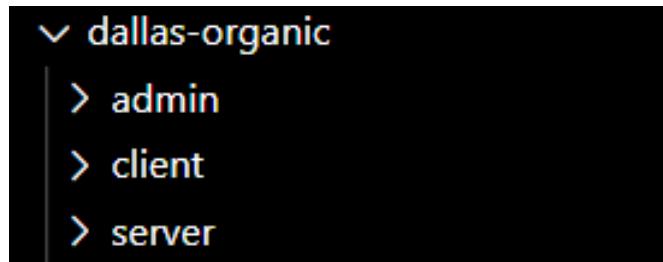
Hình 3.4: Mô hình kiến trúc MVC áp dụng cho hệ thống

Trong đó:

- Người dùng truy cập và sử dụng hệ thống trên trình duyệt (**browser**).
- **View**: có thể xem đây là phần giao diện dành cho người sử dụng, là nơi quản lý về “phần nhìn” (View management), tức thực hiện việc hiển thị các đối tượng trong hệ thống, các biểu mẫu, thông báo,... để người dùng có thể quan sát và tương tác được. Ngoài ra, đây còn là nơi lắng nghe các sự kiện và thu nhận dữ liệu đầu vào từ người dùng (input) khi họ tương tác với hệ thống, sau đó chuyển giao cho **Controller** xử lý, rồi thực hiện hiển thị ra màn hình những kết quả/ dữ liệu được **Controller** xử lý và cập nhật.
- **Controller**: là nơi điều khiển sự tương tác giữa hai thành phần **Model** và **View**.
  - Request processing: **Controller** thu nhận và xử lý các yêu cầu từ người dùng được đưa đến thông qua View, nhận vào input rồi thực hiện các cập nhật tương ứng.
  - Application-specific logic: **Controller** là nơi chứa các phương thức xử lý chính của hệ thống, thực hiện các chức năng cụ thể để phục vụ cho hệ thống.
  - Data validation: kiểm tra tính hợp lệ của dữ liệu nhập vào (tên tài khoản, mật khẩu, thông tin tài khoản, ..).
- **Model**: là nơi quản lý dữ liệu của hệ thống, làm việc với database (nơi lưu trữ dữ liệu hệ thống). Mỗi **model** là dữ liệu được sử dụng bởi chương trình. Model cũng cung cấp những phương thức cơ bản xử lý logic nghiệp vụ.

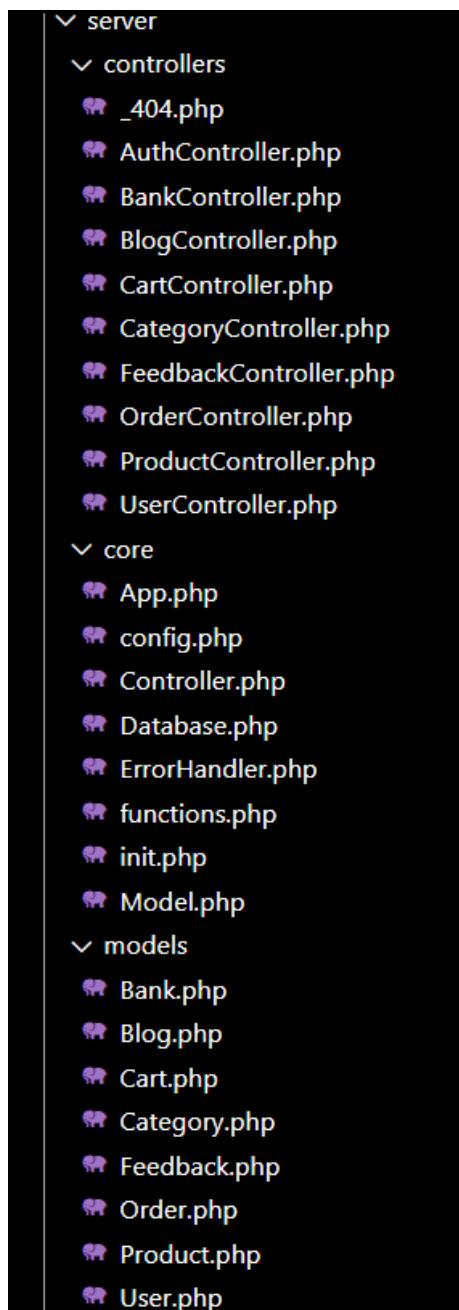
### 3.3 Cấu trúc mã nguồn

Như đã đề cập ở phần trên, nhóm chọn hướng tiếp cận kiến trúc hệ thống theo mô hình kiến trúc MVC. Vì thế cấu trúc mã nguồn cũng được tổ chức tương đồng và có những nét đặc trưng của mô hình này:



Hình 3.5: Cấu trúc mã nguồn

1. **Thư mục server:** Thư mục server bao gồm các file hiện thực server cho website, kết nối đến cơ sở dữ liệu, xử lý logic nghiệp vụ và cung cấp dịch vụ web ra bên ngoài thông qua các RESTful API.



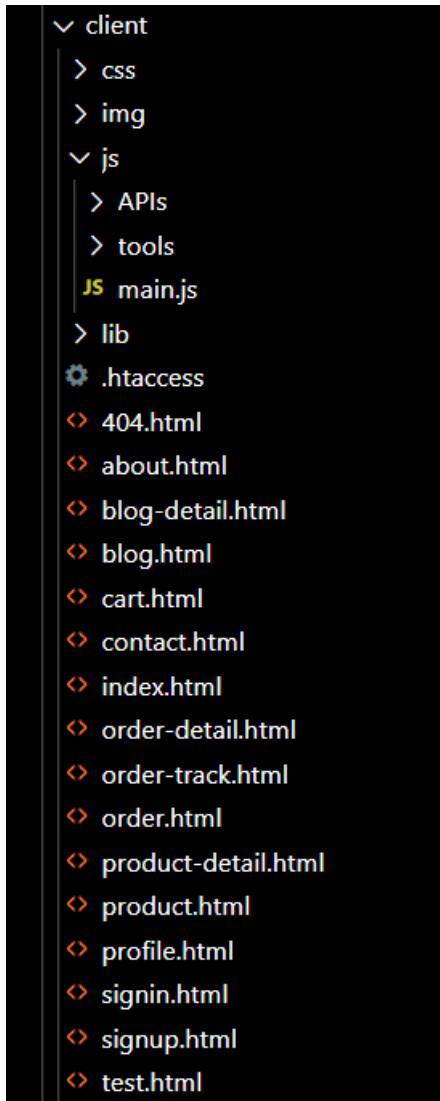
Hình 3.6: Cấu trúc mã nguồn trong thư mục server

- File **index.php** là nơi đầu tiên của server tiếp nhận request từ phía client thông qua

các API. Nó tiến hành khởi tạo phiên làm việc (session) trong ứng dụng PHP; thiết lập hàm xử lý lỗi tự định nghĩa được gọi khi có lỗi xảy ra trong ứng dụng, thiết lập hàm xử lý ngoại lệ tự định nghĩa được gọi khi có ngoại lệ xảy ra trong ứng dụng; thiết lập tiêu đề HTTP cho phép các yêu cầu từ bất kỳ nguồn nào được truy cập tài nguyên, thiết lập tiêu đề HTTP cho biết kiểu dữ liệu trả về là JSON với bộ ký tự UTF-8, thiết lập tiêu đề HTTP cho phép yêu cầu HTTP cho các phương thức GET, POST, PUT, DELETE. Sau đó, nó tạo một thể hiện của lớp App, được khởi tạo để xử lý các yêu cầu HTTP, và gọi phương thức processAPI của đối tượng App để xử lý yêu cầu HTTP.

- **Thư mục core** chứa các file lõi của server.
  - **File init.php** khởi tạo các module chính để server có thể vận hành.
  - **App.php** hiện thực class App có phương thức chính giúp xử lý các API gửi về server, phân giải nó để khởi tạo đối tượng của Controller tương ứng với request và gọi phương thức index của Controller để ánh xạ từ API sang hàm xử lý.
  - **config.php** chứa các thông tin cấu hình cho server để kết nối với cơ sở dữ liệu và **functions.php** chứa các hàm phụ trợ.
  - **Database.php** hiện thực trait Database chứa các phương thức giúp kết nối và truy vấn cơ sở dữ liệu.
  - **Controller.php** hiện thực trait Controller là Controller trung tâm (cha) của server, chứa phương thức index giúp ánh xạ REST API sang phương thức xử lý tương ứng. Ngoài ra, nó gồm các hàm xử lý request chính sử dụng các phương thức truy vấn từ đối tượng Model tương ứng của nó để lấy, thêm, sửa, xóa, lọc dữ liệu và trả về mã trạng thái HTTP cùng đối tượng JSON mà nó lấy được hoặc một phản hồi đơn.
  - **Model.php** hiện thực trait Model là Model trung tâm (cha) của server, chứa các phương thức truy vấn giao tiếp với cơ sở dữ liệu để thực hiện logic nghiệp vụ.
- **Thư mục model** chứa các file Model đặc thù có chức năng làm việc với database. Những file này cung cấp các phương thức cơ bản để quản lý truy xuất dữ liệu được chương trình sử dụng, như dữ liệu về tài khoản người dùng (**User**), sản phẩm **Product**, loại sản phẩm **Category**, tài khoản ngân hàng (**Bank**), đơn hàng (**Order**), giỏ hàng (**Cart**), đánh giá (**Feedback**), các bài đăng (**Blog**). Mỗi Model cụ thể sẽ được hiện thực là một class kế thừa từ Model trung tâm để có thể sử dụng những phương thức cơ bản nhất, cũng như bao gồm các phương thức của riêng nó để xử lý logic nghiệp vụ đặc thù.
- **Thư mục controller** chứa các file Controller đặc thù chứa mã nguồn làm nhiệm vụ xử lý các yêu cầu từ người dùng, thực hiện các chức năng chính của hệ thống và kiểm tra, xác thực dữ liệu đầu vào. Các file mã nguồn trên thực hiện các chức năng như xác thực dữ liệu người dùng khi đăng nhập, đăng ký và xử lý yêu cầu của họ; xử lý khi người dùng đọc blog; thêm sản phẩm vào giỏ hàng, mua hàng, thanh toán; người dùng đánh giá sản phẩm; người dùng xem sản phẩm, lọc sản phẩm hay tìm kiếm theo nhu cầu của mình. Tương tự như Model, mỗi Controller cụ thể sẽ được hiện thực là một class kế thừa từ Controller trung tâm để có thể sử dụng những phương thức cơ bản nhất, cũng như bao gồm các phương thức của riêng nó để xử lý các yêu cầu đặc thù.

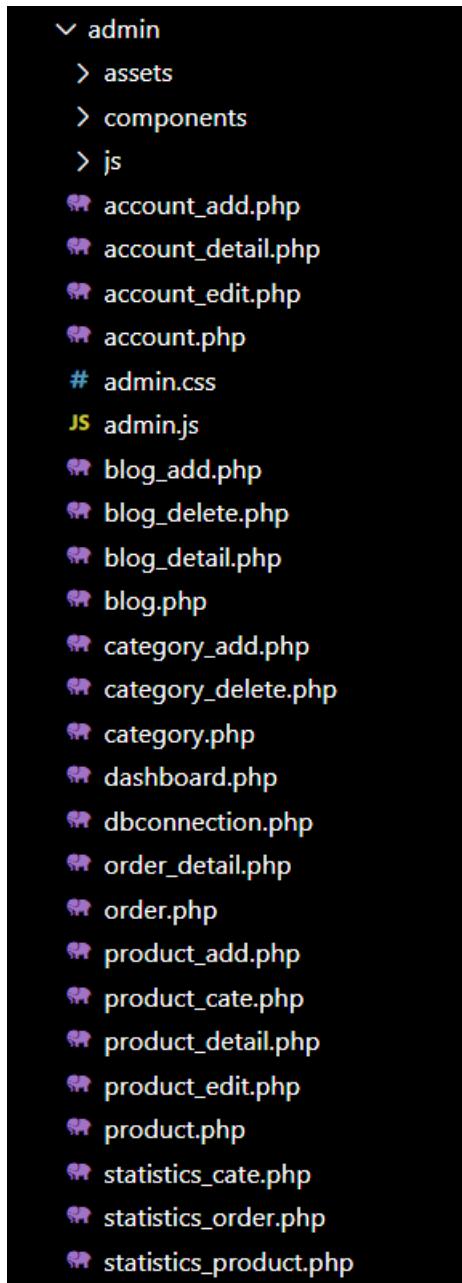
## 2. Thư mục client:



Hình 3.7: Cấu trúc mã nguồn trong thư mục client

- Các file html trong **thư mục client** giữ vai trò tạo ra giao diện người dùng, các file css trong **thư mục css** làm cho giao diện người dùng trở nên đẹp mắt hơn. Có thể xem đây là thư mục đóng vai trò **View** trong mô hình kiến trúc MVC.
- Trong quá trình tạo nên giao diện người dùng, cần phải có dữ liệu được lấy từ database để hiển thị. Lúc này dữ liệu sẽ được lấy thông qua APIs, các file trong **thư mục js/APIs** thực hiện quá trình này, cụ thể sử dụng AJAX jQuery với hàm \$.ajax() để gửi yêu cầu đến các API đã được định nghĩa và cung cấp bởi server. Ngoài ra các file javascript còn lại dùng để hỗ trợ các thao tác của người dùng, ví dụ: người dùng click chuột vào một đối tượng trên web,...

### 3. Thư mục admin:



```
└── admin
    ├── assets
    ├── components
    ├── js
    │   ├── account_add.php
    │   ├── account_detail.php
    │   ├── account_edit.php
    │   ├── account.php
    │   ├── admin.css
    │   ├── admin.js
    │   ├── blog_add.php
    │   ├── blog_delete.php
    │   ├── blog_detail.php
    │   ├── blog.php
    │   ├── category_add.php
    │   ├── category_delete.php
    │   ├── category.php
    │   ├── dashboard.php
    │   ├── dbconnection.php
    │   ├── order_detail.php
    │   ├── order.php
    │   ├── product_add.php
    │   ├── product_cate.php
    │   ├── product_detail.php
    │   ├── product_edit.php
    │   ├── product.php
    │   ├── statistics_cate.php
    │   ├── statistics_order.php
    │   └── statistics_product.php
```

Hình 3.8: Cấu trúc mã nguồn trong thư mục admin

- **Thư mục admin** chứa các file php hiện thực giao diện và các tính năng của admin như quản lý sản phẩm (thêm, xóa, sửa), quản lý tài khoản (thêm, xóa, sửa), quản lý đơn hàng, quản lý blog, ... Ngoài ra còn có trang dashboard để admin xem những thông tin chung và một số trang thống kê doanh thu.

# Chương 4

## Hiện thực

### 4.1 Đăng nhập, đăng ký và đăng xuất

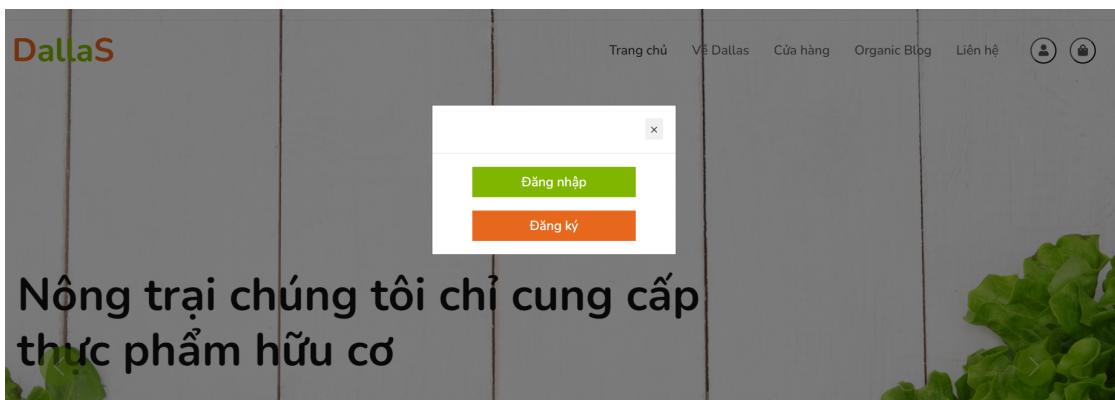
Dăng nhập và đăng ký là hai chức năng cơ bản của một website bán hàng. Hai chức năng này được hiện thực bằng cách sử dụng superglobal variables trong PHP là `$_SESSION` và sử dụng `cookies` của trình duyệt.

AuthController và User Model sẽ đảm nhiệm việc tiếp nhận và xử lý các request từ người dùng liên quan đến các tính năng đăng nhập, đăng ký và đăng xuất.

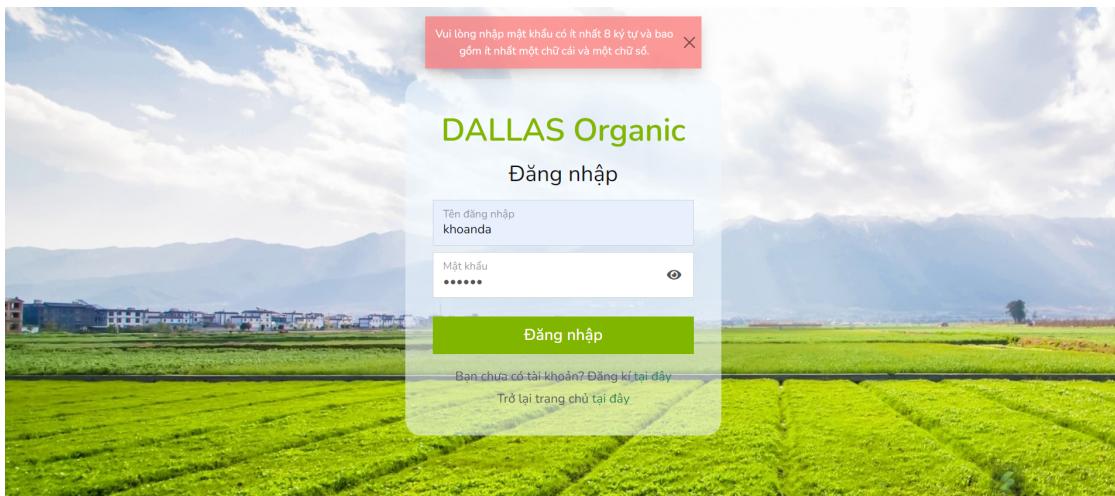
#### 4.1.1 Đăng nhập

Người dùng cần nhập tài khoản và mật khẩu của mình tại trang đăng nhập. Ở phía client-side, tài khoản và mật khẩu của người dùng được validate trước để đảm bảo đúng định dạng yêu cầu trước khi được gửi đến server thông qua phương thức HTTP POST hiện thực bằng hàm `$.ajax()` của JQuery.

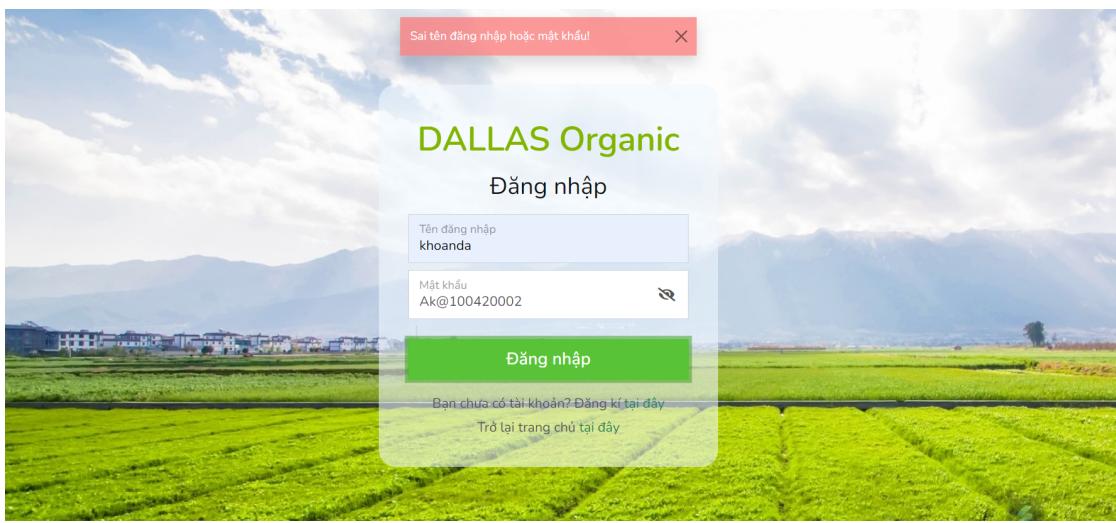
Tại server-side, thông tin đăng nhập (tên tài khoản và mật khẩu) sẽ lại được validate một lần nữa. Nếu hợp lệ, mật khẩu sẽ được verify để so trùng với mật khẩu (đã được hash) của tài khoản tương ứng được lưu trữ trong cơ sở dữ liệu. Server trả về lỗi trong trường hợp validate ở server side không thành công, mật khẩu không khớp hoặc không tồn tại tài khoản của người dùng. Nếu xác thực thành công, ở server-side sẽ thiết đặt `$_SESSION['id']` là id của người dùng và ở phía client-side sẽ thiết đặt cookie với tên là `user_id` và giá trị là id người dùng. Cookie `user_id` và biến `$_SESSION['id']` tồn tại từ khi người dùng đăng nhập vào tài khoản của mình và sẽ gỡ đi khi người dùng đăng xuất khỏi hệ thống. Chúng được dùng để giới hạn người truy cập website trong việc sử dụng một số tính năng (như xem profile, đăng tải bình luận, cho đánh giá sản phẩm, xem giỏ hàng, xem đơn hàng và tiến hành đặt hàng). Ngoài ra, các tính năng được cá nhân hóa như xem lịch sử đơn hàng, giỏ hàng, profile cũng sử dụng id được lưu trữ trong cookie đã thiết đặt này để gọi API lấy về các dữ liệu của riêng người dùng đó.



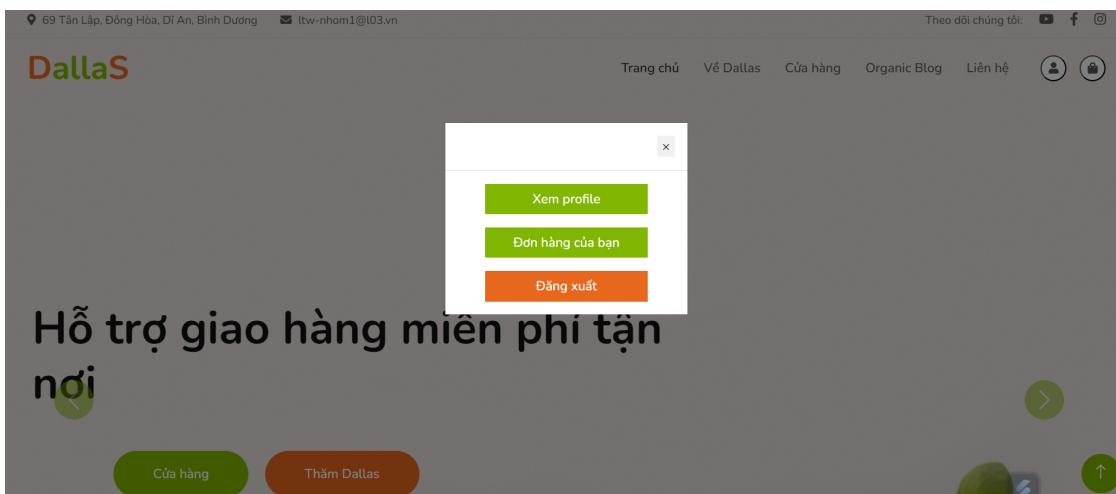
Hình 4.1: Khi chưa đăng nhập, người dùng không thể truy cập vào giỏ hàng (khi nhấn biểu tượng giỏ hàng góc trên bên phải). Hiển thị popup yêu cầu đăng nhập hoặc đăng ký



Hình 4.2: Mật khẩu không đúng định dạng (khi validate tại client-side hoặc server-side)



Hình 4.3: Xác thực không thành công (sau khi server trả về kết quả)



Hình 4.4: Sau khi đăng nhập thành công, người dùng có thể truy cập đến một số trang và sử dụng các tính năng cá nhân hóa

#### 4.1.2 Đăng xuất

Khi người dùng đăng xuất khỏi hệ thống, cookie `user_id` và biến `$_SESSION['id']` sẽ được gỡ đi tương ứng ở phía trình duyệt và phía server.

#### 4.1.3 Đăng ký

Khách có thể đăng ký tài khoản thông qua biểu mẫu đăng ký. Tương tự đăng nhập, thông tin đăng ký trước hết được validate ở phía client-side, nếu mọi thông tin là hợp lệ, hàm `$.ajax()` được dùng để gửi dữ liệu cho server. Server tiếp tục validate dữ liệu đăng ký. Nếu validate thành

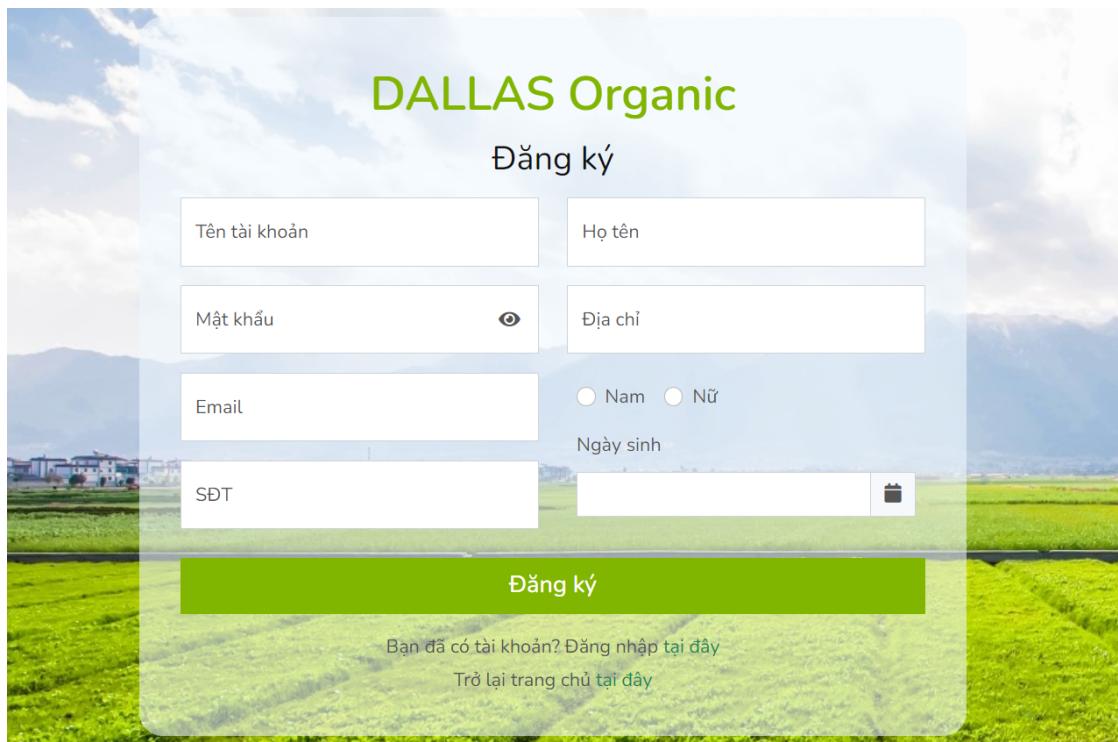
công, mật khẩu được hash trước khi thông tin người dùng lưu vào cơ sở dữ liệu. Các regex pattern được dùng để validate thông tin đăng ký:

```
1 var usernamePattern = /^[a-zA-Z0-9_-]{3,16}$/;
2 var emailPattern = /^[^s@]+@[^s@]+\.[^s@]+$/;
3 var passwordPattern =
4 /^(?=.*[A-Za-z])(?=.*\d)(?=.*[$!%*#?&])[A-Za-z\d$!%*#?&]{8,})$/;
5 var phonePattern = /^[0-9]{10,}$/;
```

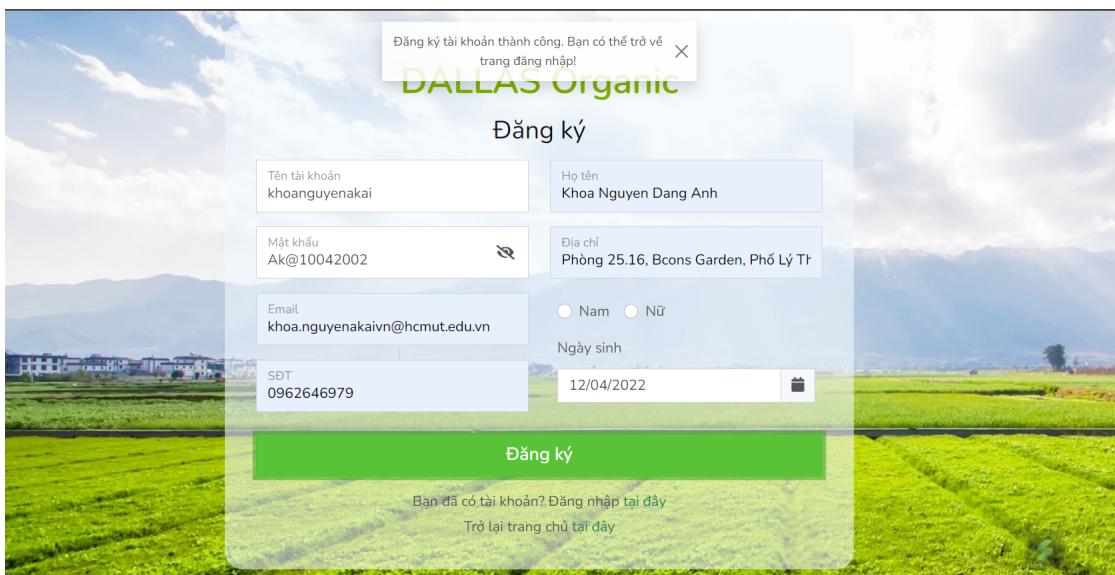
Hàm được sử dụng để hash mật khẩu phía server:

```
$data['password'] = password_hash($data['password'], PASSWORD_DEFAULT);
```

Nếu đăng ký tài khoản thành công, người dùng có thể đăng nhập và sử dụng các tính năng cá nhân. Server trả về thông báo đăng ký thất bại nếu tài khoản đã tồn tại hoặc việc validate phía server không thành công.



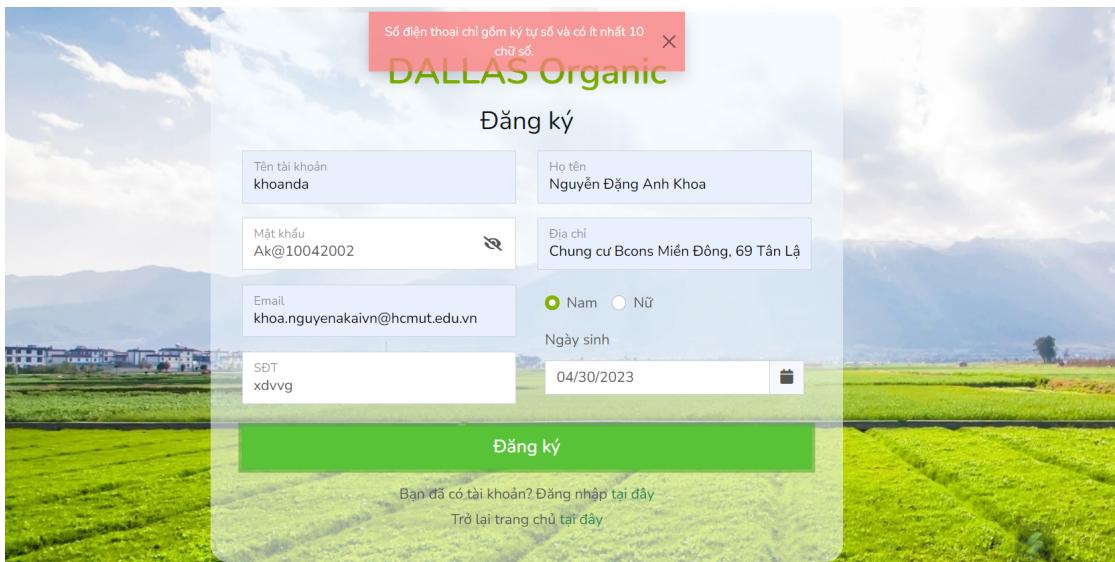
Hình 4.5: Biểu mẫu đăng ký tài khoản



Hình 4.6: Đăng ký tài khoản thành công



Hình 4.7: Tài khoản mới được thêm vào cơ sở dữ liệu



Hình 4.8: Validate dữ liệu thất bại (số điện thoại không hợp lệ)



Hình 4.9: Server trả về lỗi tài khoản đã tồn tại

## 4.2 Danh sách sản phẩm, bộ lọc sản phẩm

Để hiển thị danh sách sản phẩm, một HTTP request với phương thức GET được gửi đến API tương ứng của server sử dụng hàm `$.ajax()` của JQuery. Nếu server truy vấn dữ liệu sản phẩm thành công, một JSON object được trả về cho client. JSON Object có thể truy xuất đơn giản thông qua tham số của hàm trả về tại trường `success` của `$.ajax()` dưới dạng mảng các object trong javascript nếu request thành công. Sau đó, chúng được phân giải để hiển thị lên giao diện trang Cửa hàng.

```

1  $.ajax({
2      url:
3          "http://localhost/dallas-organic/server/product/getAllTitle?offset=
4              +
5                  offset +
6                  "&so=" +
7                  sort_order +
8                  "&sf=" +
9                  sort_field +
10                 "&price=" +
11                 price_filter +
12                 "&cate=" +
13                 cate_filter +
14                 "&name=" +
15                 product_name +
16                 "" ,
17                 type: "GET",
18                 dataType: "json",
19                 success: function (products) {
20                     productRender(products);
21                 },
22                 error: function (xhr, status, error) {
23                     console.log("Error:", error);
24                 },
25             }
26         )
27     )
28     
```

Nhóm hiện thực tính năng pagination cho phép mỗi lần client chỉ yêu cầu server lấy một số lượng sản phẩm nhất định để hiển thị lên. Nếu đổi qua trang khác, \$.ajax() sẽ lại được gọi, truyền vào offset tương ứng với chỉ mục trang, trả về danh sách sản phẩm mới vào cập nhật lại trên giao diện mà không cần tải lại trang. Việc hiện thực pagination giúp hạn chế tình trạng một lúc client yêu cầu một lượng quá lớn dữ liệu không cần thiết, ảnh hưởng đến tốc độ tải trang.

Ở bộ lọc sản phẩm, người dùng có thể lọc sản phẩm theo mức giá, lọc theo thể loại. Có thể sắp xếp theo giá cả, số lượng còn lại, số lượng đã bán và được phép chọn thứ tự sắp xếp. Khi nhấp vào nút áp dụng bộ lọc, một request sẽ được gửi đến server để server xử lý, lọc ra và trả về các sản phẩm cụ thể cập lại lại trên giao diện. Tính năng tìm kiếm sản phẩm hoạt động tương tự.

Procedure trong mySQL lọc ra các sản phẩm và sắp xếp chúng tùy theo các thuộc tính mà người dùng đã chọn ở bộ lọc. Procedure gồm các tham số đầu vào: IN p\_offset int, IN sort\_order varchar(5), IN sort\_field varchar(10), IN p\_price int, IN p\_cate int, IN p\_name varchar(50)

```

BEGIN
    DECLARE off_set int;
    declare price_pattern text;
    declare cate_pattern text;
    declare name_pattern text;
    set off_set = p_offset*6;

    IF p_name = '' THEN
        set name_pattern = CONCAT('product_name LIKE ', "%");
    ELSE
        set name_pattern = CONCAT('product_name LIKE ', "%", p_name, "%");
    END IF;

    IF p_cate = -1 THEN
        set cate_pattern = CONCAT('category_id LIKE ', "%");
    ELSE
        set cate_pattern = CONCAT('category_id=', p_cate);
    END IF;

    IF p_price = -1 THEN
        set price_pattern = CONCAT('price LIKE ', "%");
    END IF;

    IF (p_price >= 0 && p_price < 200) THEN
        set price_pattern = CONCAT('price>=',
        p_price*1000,
        ' AND price<=',
        p_price*1000+50000);
    END IF;

    IF (p_price >= 200) THEN
        set price_pattern = CONCAT('price>=',
        p_price*1000);
    END IF;

    SET @t1 =CONCAT('SELECT id,product_name,price,image,sold_number

```

```

    FROM products WHERE',
        price_pattern,
        'AND',
        cate_pattern,
        'AND',
        name_pattern,
        'ORDER BY',
        sort_field,
        '',
        sort_order,
        '',
        'LIMIT 6 OFFSET',
        off_set);
PREPARE stmt FROM @t1;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END //
DELIMITER ;

```

Áp dụng

Tìm theo tên...

Sắp xếp bởi | Tăng dần

Lọc với giá

- Tất cả sản phẩm
- 0đ - 50.000đ
- 50.000đ - 100.000đ
- 100.000đ - 150.000đ
- 150.000đ - 200.000đ
- Từ 200.000đ

Lọc với thể loại

- Tất cả sản phẩm
- Rau củ hữu cơ
- Trái cây hữu cơ
- Thực phẩm khô
- Thịt - Hải sản
- Thực uống hữu cơ
- Bánh mì hữu cơ

|   | Hot   | Hot   | Hot                                   |
|---|---|---|---------------------------------------|
|    | <b>Cà rốt</b><br>42,000đ 52,000đ                |  | <b>Táo đỏ Mỹ</b><br>100,000đ 110,000đ |
|  | <b>Bưởi da xanh túi lưới</b><br>59,000đ 69,000đ |   |                                       |
| <a href="#">Xem thêm</a>  | <a href="#">Giao 24H</a>                        | <a href="#">Xem thêm</a>  | <a href="#">Giao 24H</a>              |

First Previous 1 2 3 4 5 Next Last

↑

Hình 4.10: Danh sách sản phẩm hiển thị với pagination

Tất cả sản phẩm  
 0đ - 50.000đ  
 50.000đ - 100.000đ  
 100.000đ - 150.000đ  
 150.000đ - 200.000đ  
 Từ 200.000đ

Lọc với thể loại

Tất cả sản phẩm  
 Rau củ hữu cơ  
 Trái cây hữu cơ  
 Thực phẩm khô  
 Thịt - Hải sản  
 Thức uống hữu cơ  
 Bánh mì hữu cơ



**Cà rốt**

42,000đ 52,000đ

[Xem thêm](#) [Giao 24H](#)



**Cải bó xôi hữu cơ USDA Food King**

40,250đ 50,250đ

[Xem thêm](#) [Giao 24H](#)



**Dưa leo VietGAP**

19,500đ 29,500đ

[Xem thêm](#) [Giao 24H](#)

Hình 4.11: Áp dụng bộ lọc theo giá: nhỏ hơn 50.000đ

Tất cả sản phẩm  
 0đ - 50.000đ  
 50.000đ - 100.000đ  
 100.000đ - 150.000đ  
 150.000đ - 200.000đ  
 Từ 200.000đ

Lọc với thể loại

Tất cả sản phẩm  
 Rau củ hữu cơ  
 Trái cây hữu cơ  
 Thực phẩm khô  
 Thịt - Hải sản  
 Thức uống hữu cơ  
 Bánh mì hữu cơ



**Ba chỉ bò Obe hữu cơ**

225,000đ 235,000đ

[Xem thêm](#) [Giao 24H](#)



**Ba rọi rút sườn heo organic**

197,500đ 207,500đ

[Xem thêm](#) [Giao 24H](#)



**Bắp bò hữu cơ obe - obe shin**

255,000đ 265,000đ

[Xem thêm](#) [Giao 24H](#)

Hình 4.12: Áp dụng bộ lọc theo thể loại: thịt - hải sản

First | Previous | 1 | 2 | 3 | 4 | 5 | Next | Last

Tất cả sản phẩm  
 0đ - 50.000đ  
 50.000đ - 100.000đ  
 100.000đ - 150.000đ  
 150.000đ - 200.000đ  
 Từ 200.000đ

Không tìm thấy sản phẩm!

---

Lọc với thể loại

Tất cả sản phẩm  
 Rau củ hữu cơ  
 Trái cây hữu cơ  
 Thực phẩm khô  
 Thịt - Hải sản  
 Thức uống hữu cơ  
 Bánh mì hữu cơ

↑
↓

Hình 4.13: Áp dụng bộ lọc theo thể loại: thịt - hải sản và theo giá: nhỏ hơn 50.000đ. Không có sản phẩm nào thỏa mãn.

Tìm theo tên... 🔍

Giá ▼

Làng dân ▼

**Lọc với giá**

Tất cả sản phẩm  
 0đ - 50.000đ  
 50.000đ - 100.000đ  
 100.000đ - 150.000đ  
 150.000đ - 200.000đ  
 Từ 200.000đ

**Lọc với thể loại**

Tất cả sản phẩm  
 Rau củ hữu cơ  
 Trái cây hữu cơ  
 Thực phẩm khô  
 Thịt - Hải sản  
 Thức uống hữu cơ



**Dưa leo VietGAP**  
19,500đ 29,500đ



**Đậu cove hữu cơ USDA Food King**  
35,000đ 45,000đ

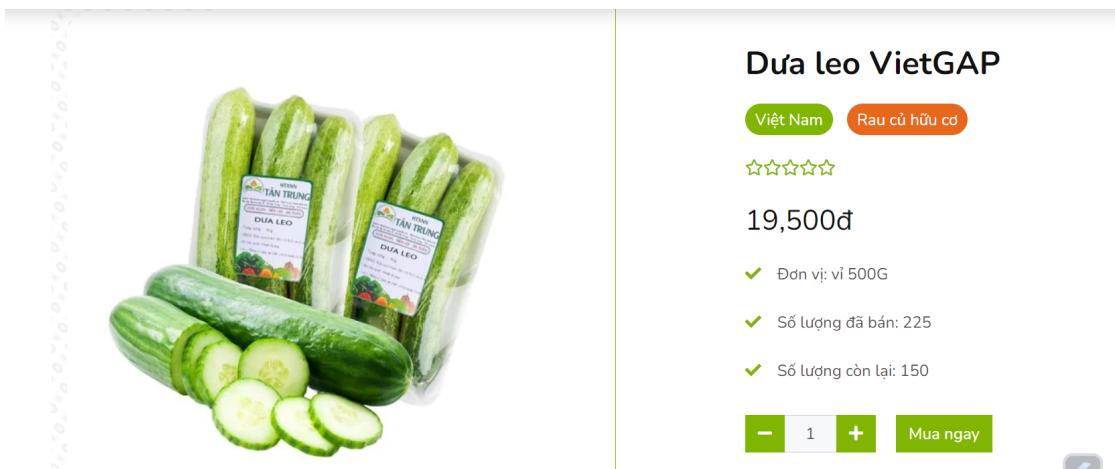


**Cải bó xôi hữu cơ USDA Food King**  
40,250đ 50,250đ

Hình 4.14: Sắp xếp các sản phẩm rau củ hữu cơ theo giá tăng dần

### 4.3 Chi tiết sản phẩm, cho điểm đánh giá và đăng tải nhận xét

Trang chi tiết về sản phẩm sẽ hiển thị thông tin chi tiết về sản phẩm tương ứng khi người dùng nhấp chuột vào một sản phẩm nào đó tại trang cửa hàng. Ở đây, người dùng có thể điều chỉnh số lượng sản phẩm mình muốn mua vào thêm vào giỏ hàng. Hệ thống sẽ chuyển tiếp người dùng qua giỏ hàng của mình.



Hình 4.15: Giao diện chi tiết sản phẩm



Hình 4.16: Người dùng điều chỉnh số lượng và nhấn nút Mua ngay để chuyển tiếp đến giỏ hàng đã thêm sản phẩm

Ngoài ra, người dùng có thể xem điểm đánh giá trung bình của sản phẩm, các đánh giá của từng người dùng khác đã trải nghiệm qua sản phẩm. Bản thân người dùng cũng có thể đăng tải đánh giá của mình, gồm việc cho điểm đánh giá và gửi nhận xét. Khi người dùng cho điểm đánh giá, website sẽ tự động cập nhật lại điểm đánh giá trung bình của sản phẩm và thêm đánh giá của người dùng vào danh sách đánh giá đang hiển thị. Hàm `$.ajax()` nhận dữ liệu là một JSON object chứa thông tin đánh giá của người đăng tải, sau đó gửi đến server thông qua phương thức POST.

```

1  function postReview() {
2    var FormData = {
3      productId: Number(id),
4      customerID: Number(user_id),
5      comment: $("#message").val(),
6      rating: $("#point").val(),

```

```

7      };
8      $.ajax({
9        url: "http://localhost/dallas-organic/server/feedback/createItem",
10       type: "POST",
11       data: JSON.stringify(formData),
12       contentType: "application/json",
13       success: function (result) {
14         alert("    ng    t i b nh l u n th nh c ng !");
15         window.location.href = "product-detail.html?id=" + formData.productID;
16       },
17       error: function (xhr, status, error) {
18         alert("    ng    t i b nh l u n th ht b i !");
19       },
20     });
21   }

```

The screenshot shows a product detail page with a review section. At the top, there are two tabs: 'Mô tả' (Description) and 'Đánh giá' (Reviews), with 'Đánh giá' being the active tab. Below the tabs, it says '1 đánh giá cho sản phẩm này' (1 review for this product). A single review is listed:

**Nguyễn Đăng Anh Hồng**  
@khoanda  
*Thời gian đăng tải: 2023-04-28 02:21:17*

**★★★★★**  
Bưởi dở quá

The review text is: **Bưởi ngon lắm**. The 'Gửi đánh giá' (Send review) button is visible at the bottom.

Hình 4.17: Phần đánh giá sản phẩm

The screenshot shows a product detail page with a review section. At the top, there are two tabs: 'Mô tả' (Description) and 'Đánh giá' (Reviews), with 'Đánh giá' being the active tab. Below the tabs, it says '2 đánh giá cho sản phẩm này' (2 reviews for this product). Two reviews are listed:

**Nguyễn Đăng Anh Hồng**  
@khoanda  
*Thời gian đăng tải: 2023-04-28 02:21:17*

**★★★★★**  
Bưởi dở quá

**Nguyễn Đăng Anh Khoa**  
@khoand  
*Thời gian đăng tải: 2023-04-28 04:43:42*

**★★★★★**  
Bưởi ngon lắm

The 'Gửi đánh giá' (Send review) button is visible at the bottom.

Hình 4.18: Kết quả sau khi người dùng đăng tải đánh giá của mình. Điểm đánh giá cũng được cập nhật (Hình 4.16)

## 4.4 Điều chỉnh giỏ hàng, Tạo đơn hàng

Tại giỏ hàng, người dùng có thể điều chỉnh số lượng sản phẩm mỗi loại, xóa sản phẩm ra khỏi giỏ hàng, xóa hết sản phẩm trong giỏ hàng hoặc chuyển qua bước tiếp theo là tạo đơn hàng. Và khi điều chỉnh số lượng hoặc xóa, thành tiền ở mỗi sản phẩm cùa sản phẩm cùng tổng tiền của giỏ hàng cũng sẽ được cập nhật. Ví dụ, đối với tính năng xóa sản phẩm khỏi giỏ hàng, ta sử dụng hàm `$.ajax()` để thực hiện một request xóa sản phẩm thông qua phương thức HTTP `DELETE`, truyền cho nó 2 tham số đó là id của sản phẩm và id của tài khoản sở hữu giỏ hàng.

Kiểm tra thật kỹ đơn hàng trước khi xác nhận nhé!

| Sản phẩm  | Giá                   | Đơn vị | Số lượng                               | Tổng tiền   |
|---|-----------------------|--------|--|---|
|  | Bưởi da xanh túi lưới | 59,000 | trái                                   | <input type="button" value="-"/> 5 <input type="button" value="+"/> <span style="color:red;">X</span> |
| Tổng tiền đơn hàng:   |                       |        |  | 295,000 vnđ   |
|   |                       |        | <input type="button" value="Xóa hết"/> | <input type="button" value="Xác nhận"/>   |

Hình 4.19: Kết quả sau khi người dùng đăng tải đánh giá của mình. Điểm đánh giá cũng được cập nhật (Hình 4.16)

Kiểm tra thật kỹ đơn hàng trước khi xác nhận nhé!

| Sản phẩm  | Giá                   | Đơn vị | Số lượng                               | Tổng tiền   |
|---|-----------------------|--------|--|---|
|  | Bưởi da xanh túi lưới | 59,000 | trái                                   | <input type="button" value="-"/> 5 <input type="button" value="+"/> <span style="color:red;">X</span> |
| Tổng tiền đơn hàng:   |                       |        |  | 295,000 vnđ   |
|   |                       |        | <input type="button" value="Xóa hết"/> | <input type="button" value="Xác nhận"/>   |

Hình 4.20: Điều chỉnh lượng số sản phẩm trong giỏ hàng

**Giỏ hàng của bạn**

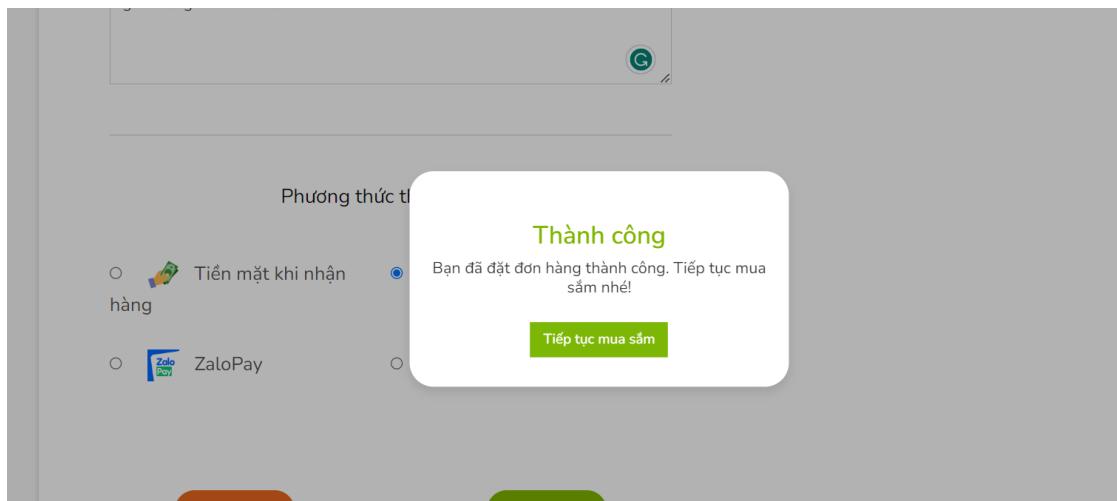
Kiểm tra thật kỹ đơn hàng trước khi xác nhận nhé!

| Sản phẩm              | Giá    | Đơn vị | Số lượng   | Tổng tiền |
|-----------------------|--------|--------|--|-----------|
| Bưởi da xanh túi lưới | 59,000 | trái   | <input type="button" value="-"/> <input type="button" value="3"/> <input type="button" value="+"/> | 177,000   |

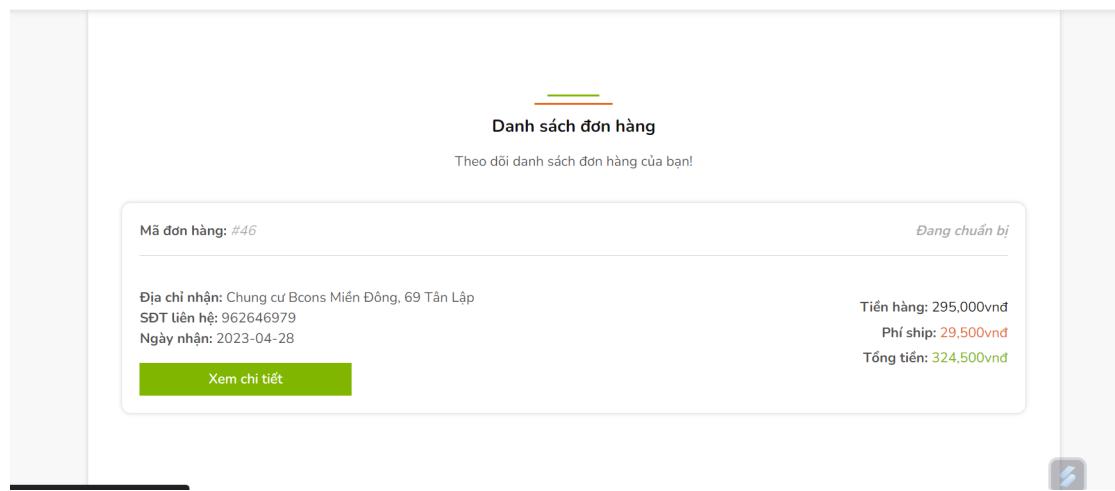
Tổng tiền đơn hàng: 177,000 vnd

Hình 4.21: Kết quả sau khi điều chỉnh lượng số sản phẩm trong giỏ hàng

Tại trang đơn hàng, người dùng sẽ có thông tin giao hàng mặc định là thông tin được lấy từ profile (cung cấp lúc đăng ký). Người dùng cũng có thể thay đổi số điện thoại liên hệ, địa chỉ nhận và thêm ghi chú cho đơn hàng, sau đó chọn phương thức thanh toán và nhấn xác nhận đặt hàng. Nếu đặt hàng thành công, thông tin về đơn hàng sẽ được lưu và hiển thị ở mục đơn hàng của tôi. Người dùng có thể theo dõi danh sách đơn hàng hoặc một đơn hàng cụ thể để nắm bắt trạng thái đơn hàng của mình.



Hình 4.22: Thông báo đặt đơn hàng thành công



Hình 4.23: Theo dõi danh sách đơn hàng

## Đơn hàng của bạn

Họ tên khách hàng: Nguyễn Đặng Anh Khoa  
SĐT liên hệ: 962646979  
Địa chỉ nhận: Chung cư Bcons Miền Đông, 69 Tân Lập

| Mã đơn hàng : | Ngày đặt : | Trạng thái :         |
|---------------|------------|----------------------|
| #46           | 2023-04-28 | <b>Đang chuẩn bị</b> |

|   |           |
|---|-----------|
|  Bưởi da xanh túi lưới<br>5 trái | 59,000vnđ |
|---|-----------|

Tổng cộng: 295,000  
Phí vận chuyển: 29,500  
**Thành tiền: 324,500**

Kiểm tra thông tin đơn hàng của bạn và báo với chúng tôi khi cần thiết!  
*Cảm ơn đã ủng hộ nông trại của chúng tôi.*

**Dallas Organic**

Hình 4.24: Theo dõi cụ thể một đơn hàng đã đặt

## 4.5 Điều chỉnh profile

Người dùng có thể xem, chỉnh sửa profile, upload ảnh đại diện và thay đổi mật khẩu (có validate mật khẩu mới) ở trang profile của tôi. Chẳng hạn, khi người dùng muốn cập nhật profile của mình, người dùng thực hiện chỉnh sửa và dữ liệu mới dạng JSON object sẽ được gửi đến server thông qua phương thức PUT được cung cấp bởi `$.ajax()`. Nếu chỉnh sửa thành công, server sẽ trả về mã 200 và client thông báo đến người dùng profile đã được cập nhật. Nếu không, người dùng sẽ nhận thông báo về lỗi.

Ví dụ về hiện thực cho tính năng cập nhật mật khẩu và cập nhật profile phía client (đóng gói thông tin mới và gửi đến API từ server):

```
1  var formData = {  
2      fullname: $("#name").val(),  
3      email: $("#email").val(),  
4      phonenumber: $("#phone").val(),  
5      address: $("#address").val(),  
6      };  
7      $.ajax({
```

```

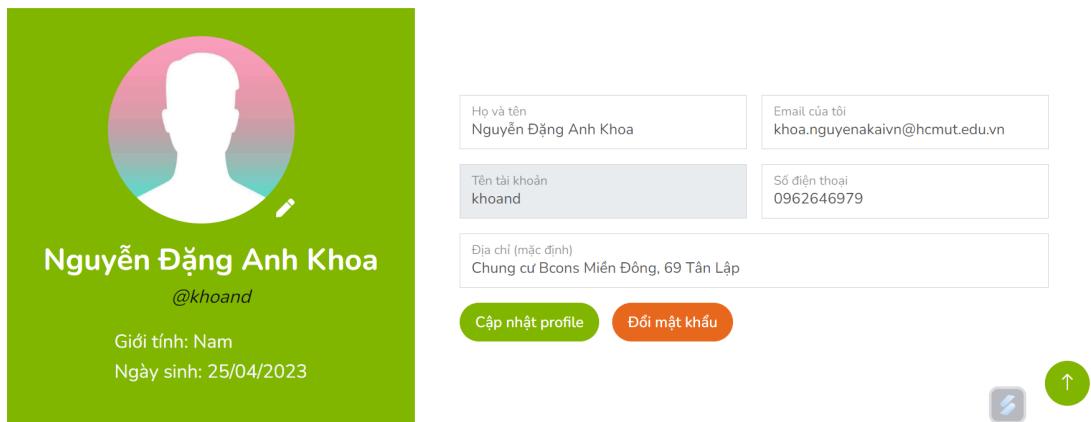
8         url:
9             "http://localhost/dallas-organic/server/user/editItem?id=" +
10            user_id,
11            type: "PUT",
12            data: JSON.stringify(formData),
13            contentType: "application/json",
14            success: function (result) {
15                document.getElementById("name-left").innerHTML = $("#name").val
16                    ();
17                createToast("Cap nhat thong tin thanh cong!");
18            },
19            error: function (xhr, status, error) {
20                createToast("Cap nhat thong tin that bai: " + error, true);
21            },
22        });


---

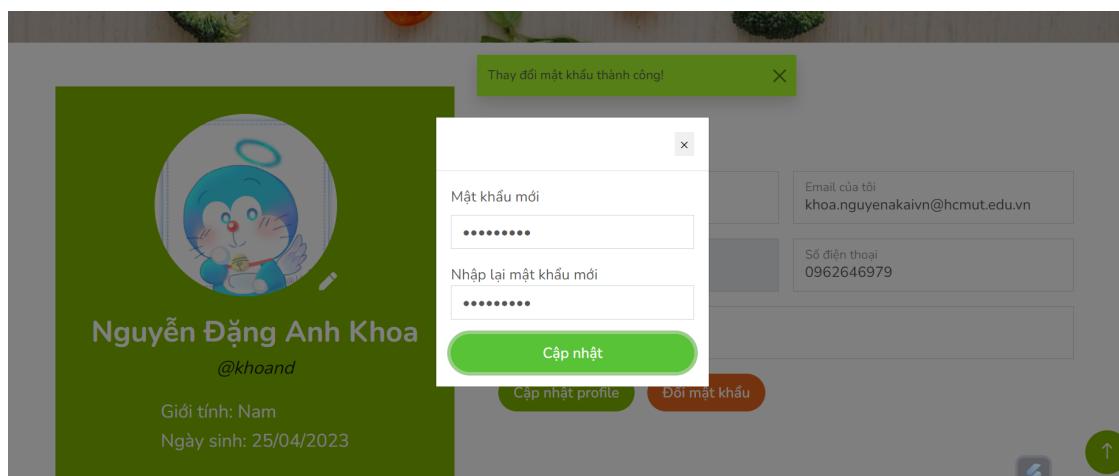

1 $("#upd-pass-btn").click(function (e) {
2     e.preventDefault();
3     var passwordPattern =
4         /^(?=.*[A-Za-z])(?=.*\d)(?=.*[$@!%*#?&])[A-Za-z\d@$!%*#?&]{8,}$/;
5     var pass = $("#password").val();
6     var pass2 = $("#password2").val();
7     if (!pass || !pass2 || pass != pass2) {
8         createToast("Mat khau moi khong khop!", true);
9     } else if (!passwordPattern.test(pass)) {
10        createToast(
11            "Vui long nhap mat khau dung dinh dang!",
12            true
13        );
14    } else {
15        $.ajax({
16            url: "http://localhost/dallas-organic/server/user/editItem?id=" +
17                user_id,
18                type: "PUT",
19                data: JSON.stringify({ password: pass }),
20                contentType: "application/json",
21                success: function (result) {
22                    createToast("Thay doi mat khau thanh cong!");
23                },
24                error: function (xhr, status, error) {
25                    createToast("Thay doi mat khau that bai: " + error, true);
26                },
27            });
28    }
29 });

```

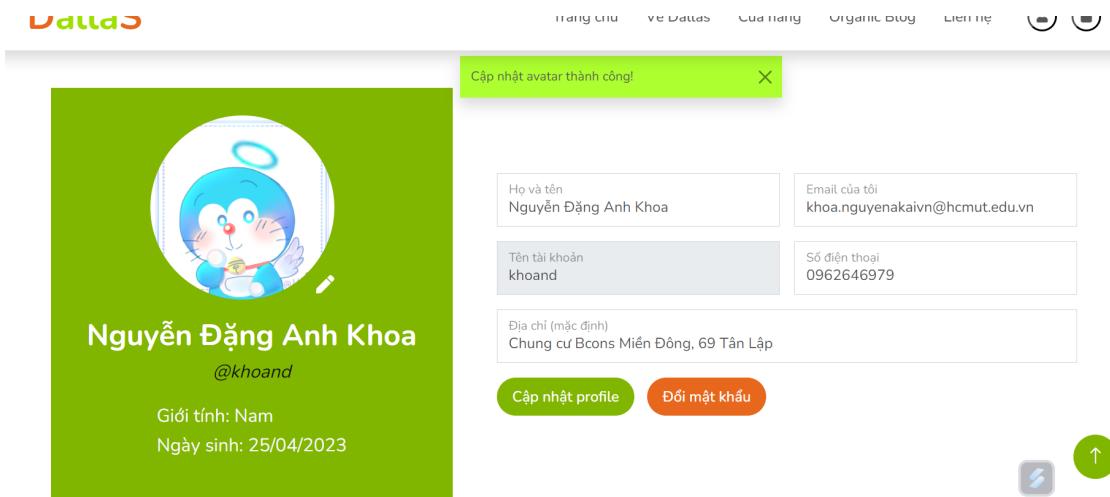
---



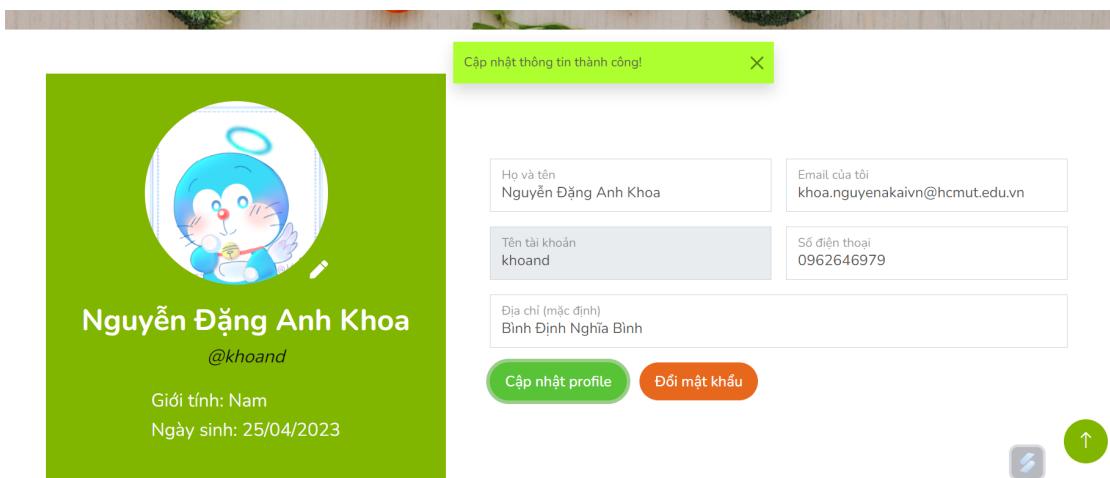
Hình 4.25: Profile cũ



Hình 4.26: Update mật khẩu



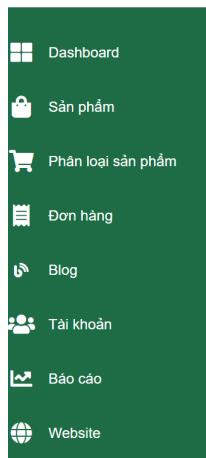
Hình 4.27: Upload avatar



Hình 4.28: Chỉnh sửa profile

## 4.6 Các tính năng quản lý dành cho quản trị viên

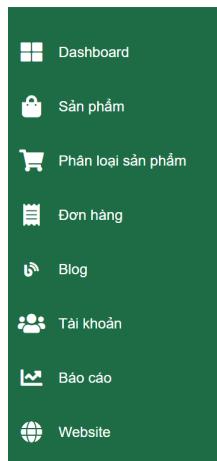
Quản trị viên được cung cấp các tính năng quản lý tài nguyên và cơ sở dữ liệu của website, thực hiện các thao tác quản lý danh sách, xem chi tiết, thêm, xóa, sửa dữ liệu.



### Chi tiết - Chính sửa Blog

|                  |   |  |
|------------------|---|--|
| Tiêu đề*:        | 5 Healthy Lifestyle Tips to Improve Your Overall Health   |  |
| Tiêu đề phụ*:    | Reduce your tiredness and boost your health with these simple changes   |  |
| Người đăng*:     | Admin   |  |
| Thời gian đăng*: | 23/04/2023  |  |
| Min-read*:       | 2   |  |
| Nội dung*:       | <div style="display: flex; justify-content: center;">  </div> |  |

Hình 4.29: Chính sửa blog đã đăng



### Quản lý phân loại sản phẩm

| ID | Phân Loại       | Số lượng mặt hàng | Đã bán | Kho   | Thao tác                 |                                      |   |
|----|-----------------|-------------------|--------|-------|--------------------------|--------------------------------------|---|
| 1  | Rau củ hữu cơ   | 7                 | 1,450  | 3,986 | <a href="#">Chi tiết</a> | <a href="#">Sửa tên phân loại...</a> | <a href="#">Chỉnh sửa</a> <a href="#">Xóa</a> |
| 2  | Trái cây hữu cơ | 7                 | 936    | 2,141 | <a href="#">Chi tiết</a> | <a href="#">Sửa tên phân loại...</a> | <a href="#">Chỉnh sửa</a> <a href="#">Xóa</a> |
| 3  | Thực phẩm khô   | 8                 | 539    | 649   | <a href="#">Chi tiết</a> | <a href="#">Sửa tên phân loại...</a> | <a href="#">Chỉnh sửa</a> <a href="#">Xóa</a> |
| 4  | Thịt - Hải sản  | 10                | 839    | 368   | <a href="#">Chi tiết</a> | <a href="#">Sửa tên phân loại...</a> | <a href="#">Chỉnh sửa</a> <a href="#">Xóa</a> |
| 5  | Thức uống       | 5                 | 307    | 67    | <a href="#">Chi tiết</a> | <a href="#">Sửa tên phân loại...</a> | <a href="#">Chỉnh sửa</a> <a href="#">Xóa</a> |

Hình 4.30: Quản lý danh mục sản phẩm

| ID | Họ tên người nhận | Tài khoản  | Số điện thoại | Ngày đặt   | Tổng đơn    | Trạng thái    | Thao tác   |
|----|-------------------|------------|---------------|------------|-------------|---------------|--|
| 1  | Trần Chí Công     | congtran14 | 908402431     | 20-12-2023 | 433,000 vnđ | Đang chuẩn bị | <button>Chi tiết</button><br><button>Đang chuẩn bị</button><br><button>Đang giao hàng</button><br><button>Hoàn thành</button><br><button>Đã hủy</button><br><button>Chỉnh sửa</button> |
| 2  | Trần Chí Công     | congtran14 | 908402431     | 09-11-2023 | 878,000 vnđ | Đang chuẩn bị | <button>Chi tiết</button><br><button>Đang chuẩn bị</button><br><button>Chỉnh sửa</button>  |
| 3  | Trần Chí Công     | congtran14 | 908402431     | 09-10-2023 | 878,000 vnđ | Đang chuẩn bị | <button>Chi tiết</button><br><button>Đang chuẩn bị</button><br><button>Chỉnh sửa</button>  |

Hình 4.31: Quản lý danh sách đơn hàng

### Quản lý Đơn hàng

| ID | Họ tên người nhận | Tài khoản  | Số điện thoại | Ngày đặt   | Tổng đơn    | Trạng thái    | Thao tác  |
|----|-------------------|------------|---------------|------------|-------------|---------------|---|
| 1  | Trần Chí Công     | congtran14 | 908402431     | 20-12-2023 | 433,000 vnđ | Hoàn thành    | <button>Chi tiết</button><br><button>Hoàn thành</button><br><button>Chỉnh sửa</button>    |
| 2  | Trần Chí Công     | congtran14 | 908402431     | 09-11-2023 | 878,000 vnđ | Đang chuẩn bị | <button>Chi tiết</button><br><button>Đang chuẩn bị</button><br><button>Chỉnh sửa</button> |
| 3  | Trần Chí Công     | congtran14 | 908402431     | 09-10-     | 878,000     | Đang          | <button>Chi tiết</button><br><button>Chỉnh sửa</button>                                   |

Hình 4.32: Cập nhật trạng thái đơn hàng sang đã hoàn thành

**Chỉnh sửa thông tin người dùng**

Tên đăng nhập\*: congtran14

Họ và tên\*: Trần Chí Công

Giới tính\*: Nam

Ngày sinh\*: 14/05/2002

Số điện thoại\*: 0908999997

Email\*: cong.tran1462002@hcmut.edu.vn

Địa chỉ\*: 175 Trương Định, Phường 9, Quận 3, TPHCM

Hình 4.33: Chính sửa thông tin tài khoản

**Quản lý sản phẩm**

| ID | Tên sản phẩm                     | Loại          | Giá   | Đơn vị   | Đã bán | Kho | Thao tác  |
|----|----------------------------------|---------------|-------|----------|--------|-----|---|
| 1  | Cà rốt                           | Rau củ hữu cơ | 42000 | kg       | 215    | 385 | <button>Chi tiết</button> <button>Chỉnh sửa</button> <button>Xóa</button> |
| 9  | Cải bó xôi hữu cơ USDA Food King | Rau củ hữu cơ | 40250 | túi 250G | 212    | 98  | <button>Chi tiết</button> <button>Chỉnh sửa</button> <button>Xóa</button> |
| 10 | Dưa leo VietGAP                  | Rau củ hữu cơ | 19500 | vỉ 500G  | 225    | 150 | <button>Chi tiết</button> <button>Chỉnh sửa</button> <button>Xóa</button> |
| 11 | Cải xoăn Kale hữu cơ Food King   | Rau củ hữu cơ | 44000 | túi 250G | 120    | 80  | <button>Chi tiết</button> <button>Chỉnh sửa</button> <button>Xóa</button> |

Hình 4.34: Quản lý danh sách sản phẩm, có thể lọc theo thẻ loại

Ngoài ra, hệ thống còn cung cấp trang dashboard trực quan hóa số liệu kinh doanh và tính năng tạo báo cáo doanh thu theo từng sản phẩm, theo danh mục sản phẩm hoặc doanh thu theo đơn đặt hàng trong ngày.

## Báo cáo Doanh thu

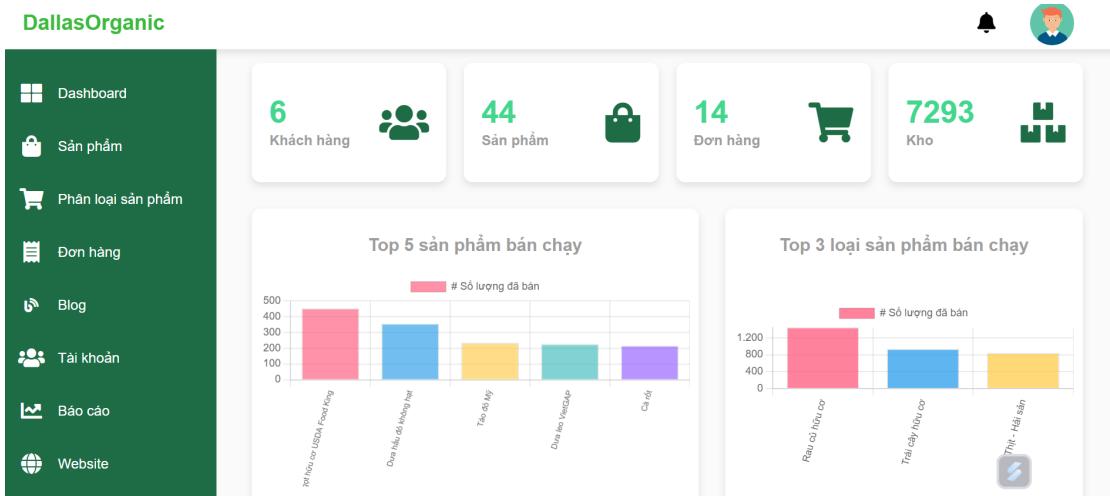
| Theo sản phẩm | Theo loại sản phẩm    | Theo đơn hàng   |
|---------------|-----------------------|-----------------|
| Tim kiếm      |                       |                 |
|               |                       |                 |
| ID            | Tên sản phẩm          | Loại            |
| 1             | Cà rốt                | Rau củ hữu cơ   |
| 2             | Táo đỏ Mỹ             | Trái cây hữu cơ |
| 3             | Bưởi da xanh túi lưới | Trái cây hữu cơ |
| 4             | Dưa lưới giòn         | Trái cây hữu cơ |
|               |                       |                 |
|               |                       | Chi tiết        |

Hình 4.35: Báo cáo doanh thu theo sản phẩm

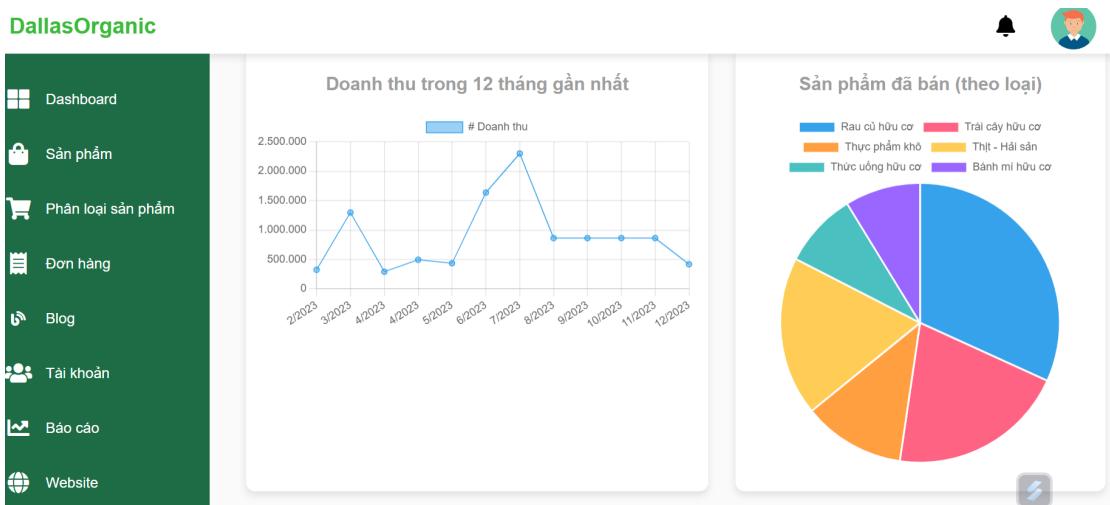
## Báo cáo Doanh thu

| Theo sản phẩm | Theo loại sản phẩm | Theo đơn hàng  |
|---------------|--------------------|----------------|
| Tim kiếm      |                    |                |
|               |                    |                |
| ID            | Phân loại          | Số mặt hàng    |
| 1             | Rau củ hữu cơ      | 7              |
| 2             | Trái cây hữu cơ    | 7              |
| 3             | Thực phẩm khô      | 8              |
| 4             | Thịt - Hải sản     | 10             |
| 5             | Thức uống hữu cơ   | 5              |
| 6             | Ramen hữu cơ       | 7              |
|               |                    |                |
|               |                    | Doanh thu      |
|               |                    | 69,577,350vnđ  |
|               |                    | 122,502,000vnđ |
|               |                    | 56,909,000vnđ  |
|               |                    | 181,963,000vnđ |
|               |                    | 75,583,000vnđ  |
|               |                    | 39,937,000vnđ  |

Hình 4.36: Báo cáo doanh thu theo danh mục sản phẩm



Hình 4.37: Dashboard trực quan hóa số liệu thống kê kinh doanh (top sản phẩm bán chạy)



Hình 4.38: Dashboard trực quan hóa số liệu thống kê kinh doanh (Tổng kết doanh thu)

# Chương 5

## Demo

### 5.1 Cài đặt môi trường

Ứng dụng được chạy trên web server localhost được tạo bởi chương trình XAMPP. XAMPP là một web server miễn phí, mã nguồn mở được phát triển bởi Apache Friends. XAMPP là tên viết tắt của "X (cross-platform hay đa nền tảng), Apache, MariaDB, PHP và Perl", đại diện cho những thành phần chính của nó bao gồm database, và interpreters dành cho những đối tượng sử dụng ngôn ngữ PHP và Perl. Ngoài ra XAMPP còn tích hợp sẵn các công cụ khác như phpMyAdmin dùng để quản lý hệ cơ sở dữ liệu MySQL, FTP Server, Mail Server, Open SLL...

XAMPP giúp cho người dùng dễ dàng cài đặt một Web Server trên thiết bị cá nhân của mình. Với XAMPP, người dùng có thể thực hiện phát triển và kiểm tra các ứng dụng web mà không cần phải đăng ký tên miền và thuê máy chủ. Nó cho phép bạn thực hiện kiểm tra và thử nghiệm trang web của mình trên máy tính cá nhân của mình trước khi triển khai nó trên một máy chủ web thực tế trên Internet. Nó còn là một công cụ hữu hiệu để học tập hay nghiên cứu các kỹ thuật lập trình web mới.

Website mà nhóm thực hiện có thể chạy trên các phiên bản khác nhau của XAMPP. Tuy vậy nhóm thống nhất sử dụng phiên bản XAMPP 8.2.0. Phiên bản của các ứng dụng bên trong nó tương ứng như sau:

- PHP 8.2.0
- Apache 2.4.54
- MariaDB 5.4.27
- phpMyAdmin 5.2.0
- OpenSSL 1.1.1p (Windows) and OpenSSL 1.1.1s (Linux and OS X)

### 5.2 Thiết lập các công cụ cần thiết

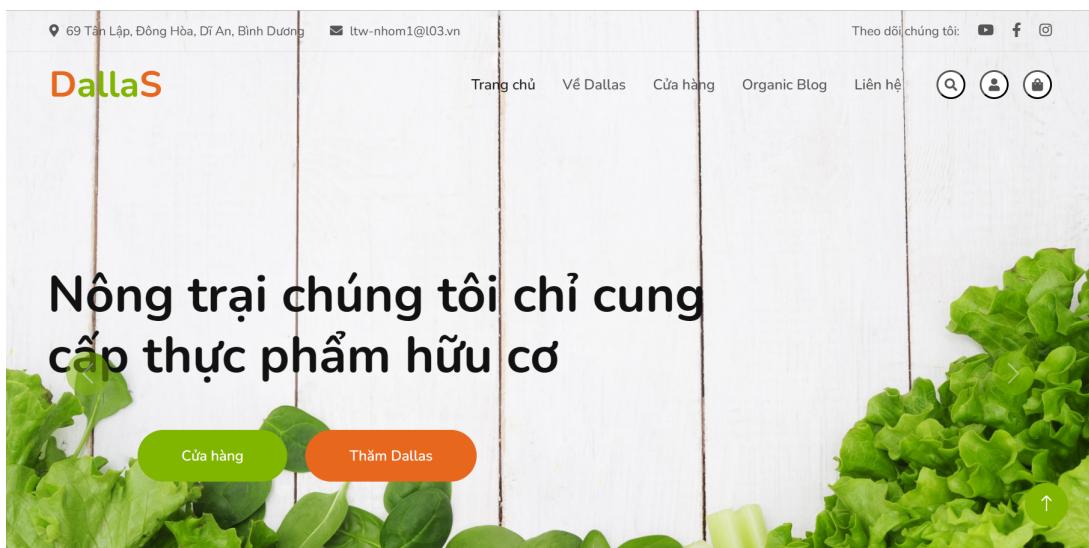
Để có thể chạy được website, chúng ta cần phải thực hiện các bước sau:

- **Bước 1:** Chỉnh đường dẫn của Document Root nằm trong file httpd.conf của XAMPP tới thư mục nơi chứa source code của project. Sau khi đã thiết lập xong đường dẫn thì

- **Bước 2:** Khởi động XAMPP, sau đó khởi động Apache server và MySQL server.
- **Bước 3:** Vào trang phpMyAdmin, sau đó thực hiện import các database có trong thư mục script nằm trong phần server.
- **Bước 4:** Truy cập vào trình duyệt web nào đó như Microsoft Edge hoặc Google Chrome, nhập đường dẫn localhost/dallas-organic/client. Trang web sẽ được hiển thị thành công.

### 5.3 Kết quả demo trang web

Sau khi thực hiện các bước ở trên, ta sẽ vào được giao diện chính của trang web như hình dưới đây.



Hình 5.1: Giao diện chính của trang web

# Tài liệu tham khảo

- [1] Website Doanh Nghiệp Là Gì? Lợi Ích Khi Thiết Kế Website Doanh Nghiệp  
<https://tatthanh.com.vn/website-doanh-nghiep-la-gi>
- [2] Website doanh nghiệp cần có gì? Câu hỏi đặt ra cho các trưởng dự án  
<https://beau.vn/vi/goc-nhin/website-doanh-nghiep-can-co-gi>
- [3] Bootstrap 5.3 Documentation  
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [4] Animate.css Documentation  
<https://animate.style>
- [5] Owl Carousel 2 Documentation  
<https://owlcarousel2.github.io/OwlCarousel2/docs/start#welcome.html>
- [6] Font Awesome 5.1 Documentation  
<https://fontawesome.com/v5/docs>
- [7] AJAX Tutorial  
<https://www.tutorialspoint.com/ajax/index.htm>
- [8] jQuery  
<http://api.jquery.com>
- [9] Introduction to REST API Principles  
<https://rapidapi.com/guides/rest-api-principles/>
- [10] wow.js — Reveal Animations When Scrolling  
<https://wowjs.uk/>
- [11] Chart.js | Open source HTML5 Charts for your website  
<https://www.chartjs.org/>