

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NÔNG LÂM THÀNH PHỐ HỒ CHÍ MINH



TIỂU LUẬN MÔN ĐIÊN TOÁN ĐÁM MÂY

ĐỀ TÀI: XÂY DỰNG VÀ PHÁT TRIỂN
WEBSITE BÁN CÂY TRỒNG

GVHD: Th.S Nguyễn Đình Long

Nhóm sinh viên thực hiện: Nhóm 13

Lớp: DH22HM

Võ Thành Nhân 22166064

Nguyễn Nữ Kim Linh 22166043

Nguyễn Quỳnh Nhi 22166065

Du Thị Bích Ngọc 22166058

Thành phố Hồ Chí Minh, tháng 12 năm 2024.

LỜI MỞ ĐẦU

Trong bối cảnh hiện nay, nhu cầu trồng cây cảnh và cây xanh không chỉ để làm đẹp không gian sống mà còn góp phần cải thiện môi trường đang ngày càng gia tăng. Tuy nhiên, việc tìm kiếm, lựa chọn và mua cây trồng thường gặp khó khăn do hạn chế về thông tin, địa điểm, cũng như sự tương tác giữa người bán và người mua.

Trước thực tế đó, nhóm đã thực hiện đề tài **“Thiết kế website bán cây trồng”** với mục tiêu tạo ra một nền tảng trực tuyến hiện đại, thân thiện và dễ sử dụng. Website không chỉ cung cấp thông tin chi tiết về các loại cây trồng mà còn hỗ trợ người dùng trong việc tìm kiếm, so sánh giá cả, và đặt hàng trực tuyến một cách nhanh chóng, tiện lợi.

Đề tài này không chỉ mang tính thực tiễn cao, mà còn góp phần thúc đẩy xu hướng ứng dụng công nghệ thông tin trong lĩnh vực thương mại và nông nghiệp. Báo cáo này sẽ trình bày chi tiết về quy trình thiết kế, các chức năng chính của website, và những giá trị mà nó mang lại cho cộng đồng.

Nhóm đã nỗ lực trong quá trình nghiên cứu và hoàn thiện đề tài nhưng vẫn còn những thiếu sót. Vì vậy, kính mong nhận được sự góp ý của thầy và các bạn.

MỤC LỤC

LỜI MỞ ĐẦU	i
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài.....	1
1.2 Mục tiêu của website.....	1
1.3 Đối tượng website hướng đến.....	2
CHƯƠNG 2: MÔ HÌNH ĐỀ XUẤT.....	4
2.1 Mô hình kiến trúc website.....	4
2.1.1 Frontend (Giao diện người dùng):	4
2.1.2 Backend (Hệ thống quản trị):	4
2.2 Mô hình dữ liệu.....	4
2.3 Quy trình vận hành.....	5
2.4 Mô hình triển khai.....	6
CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU VÀ KẾT QUẢ	7
3.1 Introduction(Giới thiệu).	8
3.2 Sign up(Đăng ký).	10
3.3 Log in(Đăng nhập).	12
3.4 Forgot password(Quên mật khẩu).....	13
3.5 Cách chăm sóc.....	18
3.6 Connection(Liên hệ).....	21
3.7 Policy(Chính sách).	23
3.8 Trang tin tức.....	24
3.9 Chức năng giỏ hàng nhỏ.	26
3.10 Chức năng thêm sản phẩm vào giỏ hàng.	28
3.11 Chức năng giỏ hàng.....	31

3.12 Chức năng tìm kiếm.....	39
3.13 Trang chủ.....	41
3.14 Danh mục sản phẩm.....	42
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	44
4.1 Kết luận.....	44
4.2 Hướng phát triển.....	45
TÀI LIỆU THAM KHẢO	47

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài.

- Trong bối cảnh hiện nay, lối sống xanh và việc bảo vệ môi trường đang trở thành xu hướng được nhiều người quan tâm. Cây trồng không chỉ mang lại không gian sống gần gũi với thiên nhiên mà còn giúp cải thiện chất lượng không khí và tạo ra giá trị tinh thần cho con người. Tuy nhiên, thị trường cây trồng truyền thống gặp phải nhiều hạn chế như:
 - Thiếu thông tin chi tiết và rõ ràng về các loại cây trồng.
 - Khó khăn trong việc tiếp cận các nhà cung cấp uy tín.
 - Thiếu các công cụ hỗ trợ người dùng so sánh và lựa chọn sản phẩm phù hợp.
- Với sự phát triển mạnh mẽ của công nghệ thông tin, việc xây dựng một nền tảng trực tuyến hỗ trợ mua bán cây trồng trở thành giải pháp thiết thực để giải quyết các vấn đề trên. Một website chuyên nghiệp không chỉ giúp người dùng dễ dàng tìm kiếm thông tin và mua cây trồng mà còn giúp các nhà cung cấp mở rộng thị trường và tiếp cận khách hàng hiệu quả hơn.

1.2 Mục tiêu của website.

- Tạo kênh bán hàng trực tuyến hiệu quả
 - Cung cấp nền tảng để các nhà cung cấp cây trồng dễ dàng giới thiệu và bán sản phẩm đến người tiêu dùng mà không bị giới hạn bởi vị trí địa lý.
- Cung cấp thông tin chi tiết về cây trồng:
 - Giới thiệu đầy đủ về các loại cây, bao gồm đặc điểm, cách chăm sóc, giá cả, và công dụng.
 - Hướng dẫn người dùng chọn loại cây phù hợp với nhu cầu và không gian sống.

- Hỗ trợ khách hàng tìm kiếm và mua sắm dễ dàng:
 - Xây dựng hệ thống tìm kiếm thông minh giúp khách hàng dễ dàng tìm thấy sản phẩm mong muốn.
 - Đơn giản hóa quy trình đặt hàng và thanh toán trực tuyến.
- Xây dựng cộng đồng yêu cây trồng:
 - Tạo không gian cho người dùng chia sẻ kinh nghiệm, kiến thức chăm sóc cây trồng.
 - Kết nối người bán và người mua, tạo sự gắn kết trong cộng đồng yêu cây.
- Thúc đẩy xu hướng sống xanh:
 - Góp phần nâng cao nhận thức về bảo vệ môi trường thông qua việc trồng và chăm sóc cây xanh.
 - Khuyến khích người tiêu dùng xây dựng lối sống hài hòa với thiên nhiên.
- Tối ưu hóa trải nghiệm người dùng:
 - Giao diện thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động.
 - Đảm bảo tốc độ tải trang nhanh và cung cấp trải nghiệm mượt mà cho khách hàng.

1.3 Đối tượng website hướng đến.

- Người yêu cây cảnh và cây trồng:
 - Những cá nhân quan tâm đến việc trồng cây để làm đẹp không gian sống và làm việc.

Những người yêu thích việc chăm sóc cây như một sở thích hoặc hoạt động thư giãn.

- Người học và nghiên cứu về cây trồng:
 - Học sinh, sinh viên, và nhà nghiên cứu cần tìm kiếm thông tin và tài liệu về các loại cây trồng và kỹ thuật chăm sóc.
- Các nhà cung cấp và nhà vườn:
 - Các cá nhân hoặc tổ chức kinh doanh cây trồng cần một nền tảng trực tuyến để quảng bá sản phẩm và mở rộng thị trường.

CHƯƠNG 2: MÔ HÌNH ĐỀ XUẤT

2.1 Mô hình kiến trúc website.

2.1.1 Frontend (Giao diện người dùng):

- Giao diện trang chủ với các danh mục chính như: *Danh mục sản phẩm, Giới thiệu, Cách chăm sóc, Tin tức, Liên hệ và Chính sách.*
- Thanh tìm kiếm và biểu tượng giỏ hàng/tài khoản người dùng ở góc trên.
- Phần nội dung nổi bật như: câu khẩu hiệu "Hãy chung tay vì một cộng đồng xanh" và danh sách cây trồng bán chạy.
- Phần chân trang web với thông tin liên hệ.

2.1.2 Backend (Hệ thống quản trị):

- Quản lý sản phẩm: thêm, xóa, sửa thông tin cây cảnh.
- Quản lý đơn hàng và khách hàng.
- Quản lý bài viết về cách chăm sóc cây và tin tức.

2.2 Mô hình dữ liệu.

- Plant model:
 - Plant_name: Tên cây.
 - Plant_type: Loại cây.
 - Description: Mô tả chi tiết.
 - Image: Liên kết ảnh cây.
 - Price: Giá bán.
 - Quality: Số lượng hiện sản phẩm hiện có trong kho.

- User model
 - Username: Tên người dùng
 - Password: Mật khẩu
- User contact model
 - Email: email dùng để theo dõi đơn hàng và cập nhật mật khẩu
 - Phone number: liên hệ nhận hàng
 - Address: địa chỉ giao hàng
- Invoice model:
 - User: người đặt đơn
 - Total amount: tổng thanh toán
 - Status: trạng thái đơn hàng
 - Create at: thời gian tạo
 - Update at: thời gian cập nhật
- Invoice item model:
 - Invoice: đơn hàng
 - Product: sản phẩm
 - Quantity: số lượng
 - Price at time for purchase: giá mặt hàng tại thời điểm mua
 - Total price: tổng thanh toán

2.3 Quy trình vận hành.

- Người dùng truy cập website → Kiểm tra tình trạng đăng nhập → cấp quyền

- Người dùng chưa đăng nhập có thể → tìm kiếm sản phẩm, xem mặt hàng, thông tin liên hệ, cách chăm sóc và tin tức.
- Người dùng đã đăng nhập → Xem giỏ hàng, thêm vào giỏ hàng, mua hàng.
- Quản trị viên nhận thông báo khi khách hàng đặt đơn → xử lý đơn hàng → giao sản phẩm.

2.4 Mô hình triển khai.

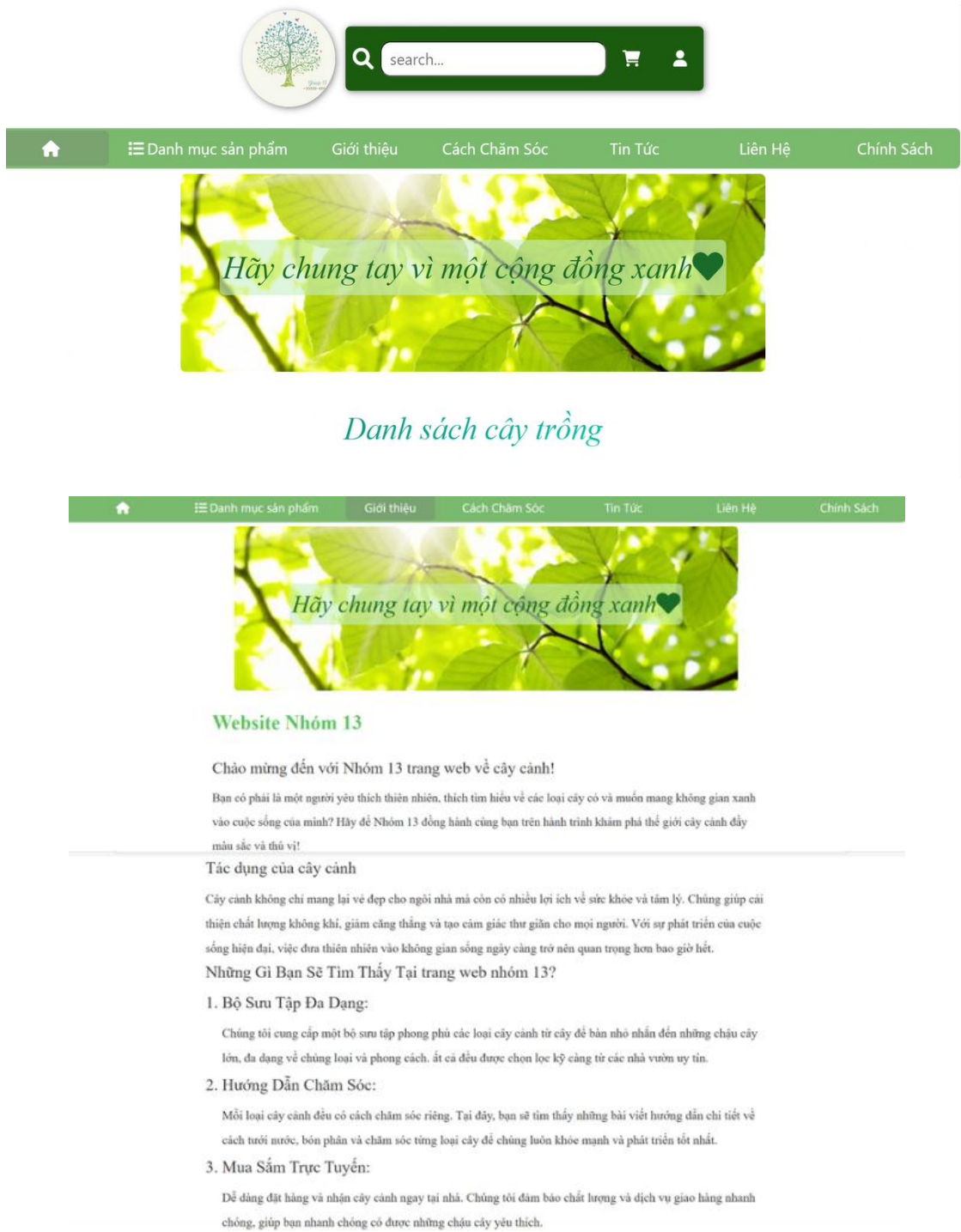
- Sử dụng framework, ngôn ngữ lập trình cơ bản phát triển:
 - Html, Css, Bootstrap cho frontend
 - Django, Ajax, Cloud Storage, Cloud SQL cho backend

CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU

- Nghiên cứu thị trường
 - Xác định đối tượng khách hàng: Nghiên cứu đối tượng khách hàng mục tiêu (ví dụ: người yêu thích làm vườn, các hộ gia đình, các doanh nghiệp, vườn ươm, trường học, công viên công cộng...). Hiểu rõ nhu cầu, thói quen mua sắm, và kỳ vọng của khách hàng đối với các sản phẩm cây trồng.
 - Phân tích đối thủ cạnh tranh: Nghiên cứu các website bán cây trồng khác trên thị trường, tìm hiểu về mô hình kinh doanh, sản phẩm, giao diện website, chiến lược marketing, và những điểm mạnh/yếu của họ.
 - Phân tích nhu cầu thị trường: Nghiên cứu xu hướng cây trồng đang được ưa chuộng, các cây trồng phổ biến, cây trồng dễ chăm sóc, các loại cây trồng đặc biệt và các sản phẩm bổ trợ như phân bón, dụng cụ làm vườn.
- Nghiên cứu hành vi người dùng.
 - Khảo sát khách hàng: Tiến hành khảo sát hoặc phỏng vấn khách hàng tiềm năng để hiểu rõ nhu cầu, mong muốn và vấn đề họ gặp phải khi mua cây trồng trực tuyến. Các câu hỏi có thể liên quan đến thông tin họ muốn tìm kiếm, cách họ so sánh giá, các yếu tố quyết định khi lựa chọn cây trồng, v.v.
 - Phân tích hành vi trực tuyến: Sử dụng các công cụ phân tích web như Google Analytics để hiểu về cách người dùng duyệt web, từ đó thiết kế giao diện và cấu trúc website sao cho thuận tiện và dễ sử dụng.
- Nghiên cứu về sản phẩm.
 - Lựa chọn các loại cây trồng: Nghiên cứu các loại cây trồng phổ biến, cây có nhu cầu cao, cây dễ chăm sóc hoặc cây đặc biệt mà thị trường chưa đáp ứng đầy đủ. Cân nhắc các yếu tố như khí hậu, yêu cầu chăm sóc cây, và sự phát triển của các loại cây.
- Nghiên cứu công nghệ và thiết kế website
 - Lựa chọn nền tảng xây dựng website: Nghiên cứu các nền tảng và công cụ xây dựng website thương mại điện tử.
 - Tính năng và giao diện: Xác định các tính năng cần có như giỏ hàng, thanh toán, bộ lọc tìm kiếm cây trồng.
- Nghiên cứu về chính sách
 - Chính sách đổi trả và bảo hành: Cung cấp các chính sách hợp lý về việc đổi trả sản phẩm, bảo hành cây trồng, và hỗ trợ khách hàng khi sản phẩm gặp vấn đề trong quá trình vận chuyển.

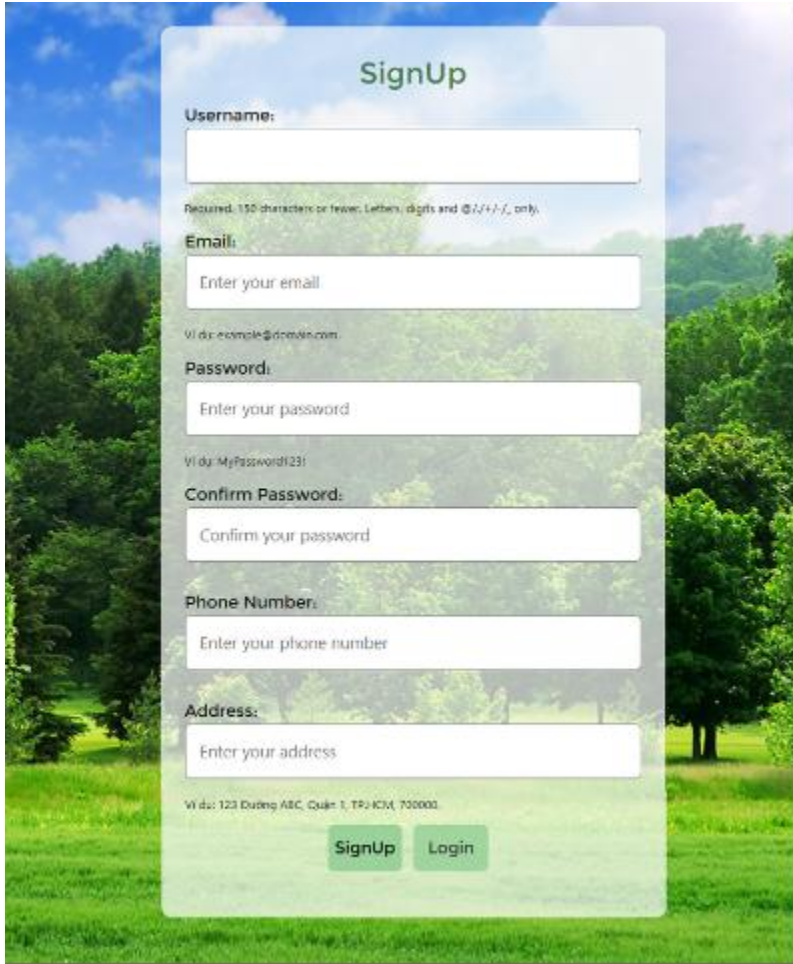
CHƯƠNG 4: KẾT QUẢ

4.1 Introduction(Giới thiệu).



- Mục đích của trang "Introduction":
 - Tạo ấn tượng đầu tiên với khách hàng: Phần "Giới thiệu" là nơi khách hàng hiểu về giá trị, sứ mệnh, và tầm nhìn của doanh nghiệp. Đây là cơ hội để xây dựng niềm tin và kết nối cảm xúc với khách hàng.
 - Khẳng định uy tín và chuyên môn: Giới thiệu về lịch sử hình thành, kinh nghiệm trong lĩnh vực, hoặc những điểm nổi bật giúp thể hiện sự chuyên nghiệp của doanh nghiệp.
- Lợi ích đối với người dùng:
 - Hiểu về giá trị mà doanh nghiệp mang lại: Khách hàng nhận biết được cam kết của doanh nghiệp đối với chất lượng sản phẩm và dịch vụ.
 - Tạo sự kết nối cảm xúc: Một phần giới thiệu được viết tốt có thể khiến khách hàng cảm thấy được đồng cảm và trân trọng.

4.2 Sign up(Đăng ký).



SignUp

Username:

Required, 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email:

Ví dụ: example@domain.com

Password:

Ví dụ: MyPassword123!

Confirm Password:

Phone Number:

Address:

Ví dụ: 123 Đường ABC, Quận 1, TP HCM, 750000.

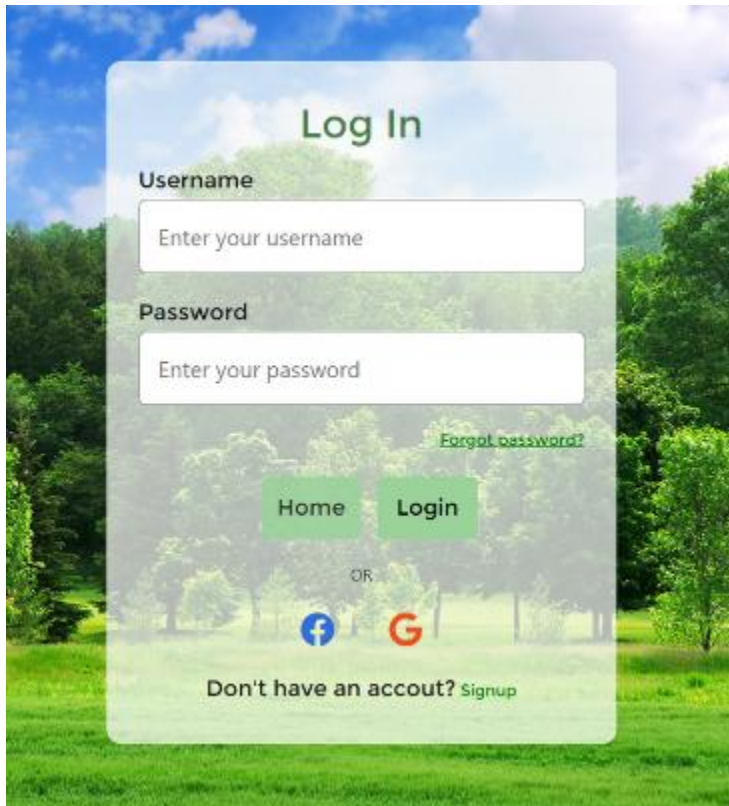
SignUp **Login**

– Mục đích của trang đăng ký:

- **Xác thực người dùng:** Trang đăng ký giúp hệ thống xác định danh tính của người dùng, qua đó quản lý và hỗ trợ người dùng tốt hơn.
- **Tạo tài khoản cá nhân hóa:** Người dùng có thể tạo tài khoản để lưu trữ thông tin cá nhân, lịch sử giao dịch, và sở thích, giúp tăng trải nghiệm sử dụng.
- **Cải thiện dịch vụ khách hàng:** Có thông tin đăng ký giúp đội ngũ quản lý dễ dàng hỗ trợ người dùng qua các kênh liên hệ hoặc cung cấp các chương trình ưu đãi phù hợp thông qua email, số điện thoại.

- Quản lý khách hàng: Một hệ thống bán hàng hiệu quả cần quản lý thông tin khách hàng để tối ưu hóa việc bán hàng và chăm sóc khách hàng.
 - Xây dựng lòng trung thành: Khi khách hàng đã có tài khoản, họ sẽ dễ dàng quay lại và sử dụng các tính năng như lưu giỏ hàng, nhận thông báo khuyến mãi.
 - Hỗ trợ pháp lý và bảo mật: Xác thực tài khoản giúp bảo vệ giao dịch và thông tin người dùng, đồng thời đảm bảo tuân thủ các quy định pháp luật liên quan đến thương mại điện tử.
 - Tăng cơ hội tiếp thị: Thông tin đăng ký (như email, số điện thoại) giúp doanh nghiệp thực hiện các chiến dịch marketing trực tiếp và cá nhân hóa hơn.
- Lợi ích mà trang đăng ký mang lại cho người dùng:
- Người dùng có thể lưu lại thông tin cá nhân, giúp tiết kiệm thời gian khi mua sắm lần sau.
 - Hệ thống cung cấp đề xuất sản phẩm phù hợp với nhu cầu dựa trên lịch sử mua hàng.
 - Tham gia các chương trình khuyến mãi, tích điểm, hoặc nhận thông báo sớm về các sản phẩm mới.
 - Theo dõi lịch sử mua hàng và quản lý đơn hàng dễ dàng.

4.3 Log in(Đăng nhập).



- Truy cập form đăng nhập:
 - Form hiển thị tiêu đề "Log In" và yêu cầu nhập thông tin:
 - Username (Tên người dùng).
 - Password (Mật khẩu).
- Nhập thông tin:
 - Username: Điền tên người dùng mà bạn đã đăng ký.
 - Password: Điền mật khẩu tương ứng với tài khoản.
- Tùy chọn khác:
 - Nếu chưa có tài khoản, bạn có thể nhấp vào "Signup" để tạo tài khoản.

- Nếu quên mật khẩu, bạn có thể nhấp vào liên kết "Forgot password?" để đặt lại mật khẩu.
- Mục đích của trang đăng nhập:
- Truy cập tài khoản cá nhân: Trang đăng nhập cho phép người dùng truy cập vào tài khoản đã tạo, nơi lưu trữ thông tin cá nhân, lịch sử giao dịch, và các tùy chọn cá nhân hóa.
 - Bảo mật thông tin và giao dịch: Đăng nhập giúp hệ thống xác định đúng danh tính người dùng, từ đó bảo vệ các dữ liệu quan trọng như thông tin giao dịch, địa chỉ nhận hàng, và phương thức thanh toán.
 - Tăng tính liên tục trong trải nghiệm: Người dùng có thể tiếp tục thực hiện các tác vụ còn dang dở, như kiểm tra giỏ hàng, theo dõi đơn hàng hoặc tiếp tục mua sắm từ lần truy cập trước.

4.4 Forgot password(Quên mật khẩu).

- Chức năng này được django hỗ trợ với các đường dẫn và giao diện có sẵn, Nhóm thực hiện thay đổi giao diện và nội dung email phù hợp

```
from django.contrib.auth import views as auth_views

path('password_reset/', auth_views.PasswordResetView.as_view(
    template_name='registration/password_reset.html',
    form_class=UserPasswordResetForm,
    email_template_name = 'registration/password_reset_email.txt'), name='password_reset'),

path('password_reset_sent/', auth_views.PasswordResetDoneView.as_view(
    template_name='registration/password_reset_sent.html'), name='password_reset_done'),

path('password_reset_confirm/<uidb64>/<token>', auth_views.PasswordResetConfirmView.as_view(
    template_name = "registration/set-password.html", form_class=CustomSetPasswordForm), name = 'password_reset_confirm'),

path('password_reset_complete/', auth_views.PasswordResetCompleteView.as_view(
    template_name = 'registration/password_reset_complete.html'), name = 'password_reset_complete'),
```

- Chức năng được thực hiện khi người dùng nhấn vào forgot password trong giao diện login

Log In

Username

Enter your username

Password

Enter your password

[Forgot password?](#)

Home Login

OR

[f](#) [G](#)

Don't have an account? [Signup](#)

- Người dùng sẽ được điều hướng tới trang xác nhận email để đổi mật khẩu

Khôi phục mật khẩu

Enter your email address

Submit

- Email khôi phục sẽ được gửi về mail người dùng

Khôi phục mật khẩu thành công!

Email khôi phục mật khẩu đã được gửi đến bạn. Hãy kiểm tra hộp thư của mình và làm theo hướng dẫn để đặt lại mật khẩu.



22166064@st.hcmuaf.edu.vn

đến tôi ▾

16:41 (0 pl

Xin chào nhóm13,

Hãy nhấn vào link phía bên dưới để đặt lại mật khẩu mới

http://127.0.0.1:8000/password_reset_confirm/MQ/cig3pb-549b23fab71a191a27cf96f84cf1d280/

Nếu như bạn đã tìm lại được hoặc không cần đổi vui lòng bỏ qua mail này

Thanks,
The Nhom13



- Nhấn vào link và tạo mật khẩu mới

Password Reset Form

Please enter your new password so we can verify.

- Người dùng có thể trở về trang login sau đó để đăng nhập

Your password has been reset successfully!

You can now log in with your new password.

[Log in](#)

- Thay đổi giao diện các trang và các form tương ứng
 - Trang password reset giao diện

```

<body>
  <div class = "container-reset">
    <h1>Khôi phục mật khẩu</h1>
    <form method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit" class = 'btn_login_signup'>Submit</button>
    </form>
  </div>
</body>

```

- Form

```

class UserPasswordResetForm(PasswordResetForm):
    Tabnine | Edit | Test | Explain | Document | Ask
    def __init__(self, *args, **kwargs):
        super(UserPasswordResetForm, self).__init__(*args, **kwargs)

        email = forms.EmailField(label='', widget=forms.EmailInput(attrs={
            'class': 'input_medium',
            'placeholder': 'Enter your email address',
            'type': 'email',
            'name': 'email'
        }))

```

- Trang password reset sent

```

<body>
  <h2>Khôi phục mật khẩu thành công!</h2>
  <p>Email khôi phục mật khẩu đã được gửi đến bạn.
  Hãy kiểm tra hộp thư của mình và làm theo hướng dẫn để đặt lại mật khẩu.</p>
</body>

```

- Nội dung Email khôi phục

```
{% autoescape off %}
Xin chào {{ user.get_full_name|default:user.username }},

Hãy nhấn vào link phía bên dưới để đặt lại mật khẩu mới
{% block password_reset_url %}
{{ protocol }}://{{ domain }}{% url 'password_reset_confirm' uidb64=uid token=token %}
{% endblock %}

Nếu như bạn đã tìm lại được hoặc không cần đổi vui lòng bỏ qua mail này

Thanks,
The Nhom13
{% endautoescape %}
```

- Trang set-password

```
<body>
  {% block content %}
  <div class="text-center" style="width: 80%; margin: 0 auto">
    <h1>Password Reset Form</h1>
    <h6 style="width: 60%; margin: 0 auto">
      Please enter your new password so we can verify.
    </h6>
  </div>
  <br />
  <form action="" method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class = 'btn_login_signup'>Submit</button>
  </form>
  {% endblock %}
</body>
```

- Form

```
class CustomSetPasswordForm(SetPasswordForm):
    new_password1 = forms.CharField(label="",
                                    widget=forms.PasswordInput(attrs={
                                        'placeholder': 'Enter your new password',
                                        'type': 'password',
                                    })),
    new_password2 = forms.CharField(label="",
                                    widget=forms.PasswordInput(attrs={
                                        'placeholder': 'Confirm your new password',
                                        'type': 'password',
                                    })),
```

- Trang password reset complete

```
<body>
  <h2>Your password has been reset successfully!</h2>
  <p>You can now log in with your new password.</p>
  <a href="{% url 'login' %}">Log in</a>
</body>
```

- Các trang được khi đề vào file gốc của django nhờ khai báo `template_name` và `form_class` trong đường dẫn của chúng

```
path('password_reset/', auth_views.PasswordResetView.as_view(
    template_name='registration/password_reset.html',
    form_class=UserPasswordResetForm,
    email_template_name = 'registration/password_reset_email.txt'), name='password_reset'),
```

- Mục đích của trang quên mật khẩu:
 - Khôi phục quyền truy cập: Trang quên mật khẩu giúp người dùng lấy lại quyền truy cập tài khoản khi họ không nhớ mật khẩu.
 - Tăng sự tiện lợi: Đảm bảo rằng người dùng không bị gián đoạn trải nghiệm do quên mật khẩu, tránh việc tạo tài khoản mới không cần thiết.
 - Bảo mật tài khoản: Quy trình quên mật khẩu được thiết kế để đảm bảo rằng chỉ chủ tài khoản mới có thể đặt lại mật khẩu, ngăn chặn hành vi truy cập trái phép.
- Lợi ích mà trang quên mật khẩu mang lại cho người dùng: khôi phục nhanh chóng, người dùng có thể lấy lại quyền truy cập vào tài khoản mà không cần tạo tài khoản mới hoặc liên hệ bộ phận hỗ trợ.

4.5 Cách chăm sóc.

- *Care CTS*

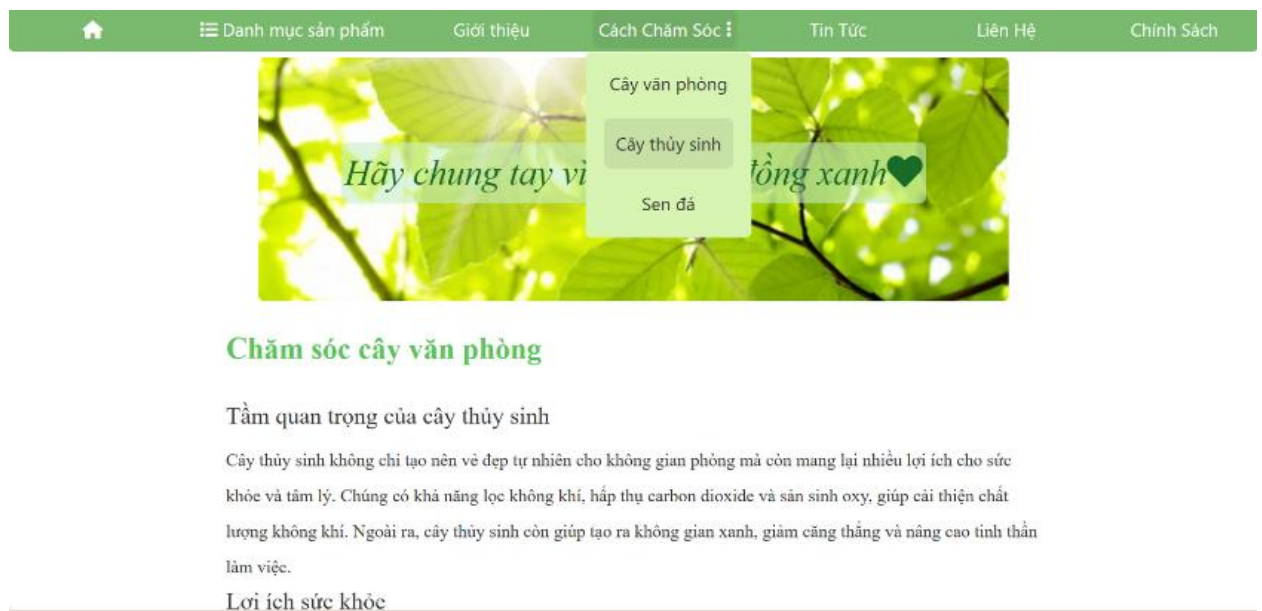


– Mục tiêu:

- "Hưỡng dẫn chăm sóc cây"
 - i. Ánh sáng: Cây thủy sinh cần ánh sáng vừa phải. Đặt cây ở nơi có ánh sáng tự nhiên nhưng tránh ánh sáng mặt trời trực tiếp để không làm hại cây.
 - ii. Phân bón: Sử dụng phân bón chuyên dụng cho cây thủy sinh để cung cấp đủ chất dinh dưỡng cho cây. Thường xuyên kiểm tra và bổ sung khi cần thiết.
 - iii. Nước: Đảm bảo nước luôn sạch và được thay thường xuyên. Nên thay nước mỗi tuần một lần để tránh sự phát triển của vi khuẩn.
 - iv. Vệ sinh: Kiểm tra lá cây thường xuyên để loại bỏ tảo và bụi bẩn. Nếu cần, có thể rửa sạch lá bằng nước sạch.
- "Giải pháp khắc phục sự cố":
 - i. Cây héo úa: Cây có thể héo do thiếu nước hoặc dinh dưỡng. Kiểm tra mức nước và bổ sung phân bón nếu cần.

- ii. Cây vàng lá: Nguyên nhân thường do nước ô nhiễm hoặc thiếu ánh sáng. Đảm bảo nước luôn sạch và cung cấp đủ ánh sáng cho cây.
- iii. Tảo phát triển: Nếu thấy tảo bám trên lá hoặc trong nước, có thể giảm ánh sáng hoặc thay nước thường xuyên hơn để kiểm soát sự phát triển của tảo.

– Care CVP:



– Mục đích của trang:

- Hỗ trợ người dùng: Trang chăm sóc cung cấp thông tin và hướng dẫn giúp người dùng giải quyết các vấn đề hoặc thắc mắc liên quan đến sản phẩm và dịch vụ.
- Cải thiện trải nghiệm khách hàng: Tạo một kênh giao tiếp hiệu quả giữa doanh nghiệp và khách hàng, tăng mức độ hài lòng và xây dựng lòng trung thành.
- Tăng giá trị sản phẩm: Đặc biệt với cây trồng, trang chăm sóc còn có thể cung cấp kiến thức về cách trồng và bảo dưỡng cây, giúp khách hàng sử dụng sản phẩm hiệu quả hơn.

– Lợi ích cho doanh nghiệp:




- Cải thiện hình ảnh thương hiệu: Một trang chăm sóc khách hàng chất lượng thể hiện sự chuyên nghiệp và tận tâm.
- Tăng doanh số gián tiếp: Khi khách hàng cảm thấy được hỗ trợ tốt, họ có xu hướng quay lại mua hàng và giới thiệu cho người khác.
- Tạo cơ hội tương tác: Doanh nghiệp có thể sử dụng phản hồi từ khách hàng để cải thiện sản phẩm hoặc phát triển các dịch vụ mới.

4.6 Connection(Liên hệ).


Liên hệ

Liên hệ trực tiếp tại
SĐT: 0909090909
Email: Group13@gmail.com
Địa chỉ: Đại Học Nông Lâm TP HCM

Liên hệ thông qua các nền tảng khác

 Facebook
 TikTok
 Youtube

Bản Đồ Địa Điểm



Giờ Làm Việc
Thứ Hai - Thứ Sáu: 9:00 - 17:00

Liên hệ	Thành viên	Theo dõi chúng tôi trên
SĐT: 0909090909	Võ Thành Nhân	 Facebook
Email: Group13@gmail.com	Nguyễn Nữ Kim Linh	 TikTok
Địa chỉ: Đại Học Nông Lâm TP HCM	Du Thị Bích Ngọc	 Youtube
	Nguyễn Quỳnh Nhi	

– Thông tin liên hệ :

- SĐT: 090909090909
- Email: Group13@gmail.com
- Địa chỉ: Đại Học Nông Lâm TP HCM

– Liên kết các nền tảng mạng xã hội:

- Facebook
- Tiktok
- Youtube

– Mục đích của trang "Liên hệ":

- Tạo kênh kết nối với khách hàng: Cung cấp thông tin liên hệ trực tiếp như số điện thoại, email, và địa chỉ. Điều này giúp khách hàng dễ dàng liên lạc khi có thắc mắc hoặc cần hỗ trợ về sản phẩm, dịch vụ.
- Xây dựng niềm tin và sự minh bạch: Việc hiển thị địa chỉ thực tế và bản đồ giúp khách hàng cảm thấy tin tưởng hơn khi giao dịch với website.
- Hỗ trợ đa kênh: Liên kết đến các nền tảng mạng xã hội (Facebook, TikTok, Youtube) giúp tăng cơ hội tương tác và quảng bá thương hiệu.
- Thông tin giờ làm việc: Hỗ trợ người dùng nắm bắt khung giờ hỗ trợ, tránh liên hệ ngoài giờ và giảm thiểu sự thất vọng.

– Lợi ích đem lại cho người dùng:

- Tăng cơ hội được hỗ trợ kịp thời: Nhờ có số điện thoại, email và các kênh mạng xã hội, người dùng luôn có một phương tiện phù hợp để giao tiếp.
- Định vị và chỉ dẫn dễ dàng: Bản đồ tích hợp giúp người dùng dễ dàng tìm đường tới địa chỉ được cung cấp.

4.7 Policy(Chính sách).

Chính Sách

Chính sách bảo hành này nhằm đảm bảo quyền lợi của khách hàng khi mua sản phẩm cây trồng hoặc các sản phẩm liên quan từ website của chúng tôi.!

Thời Gian Bảo Hành

Cây Trồng: Chúng tôi cam kết bảo hành cho cây trồng trong thời gian 7 ngày kể từ ngày nhận hàng. Trong thời gian này, nếu cây không phát triển hoặc chết do lỗi của chúng tôi, khách hàng có thể yêu cầu đổi hoặc hoàn tiền.

Điều Kiện Bảo Hành

- Cây trồng phải được chăm sóc đúng cách theo hướng dẫn của chúng tôi.
- Khách hàng cần cung cấp hóa đơn mua hàng và thông tin chi tiết về tình trạng của cây hoặc sản phẩm khi yêu cầu bảo hành.
- Không bảo hành cho những cây đã bị hư hại do thiên tai, sâu bệnh, hoặc sự bất cẩn trong quá trình chăm sóc.

Quy Trình Đổi/Hoàn Tiền

- Khách hàng cần liên hệ với chúng tôi qua email hoặc số điện thoại trong thời gian bảo hành.
- Chúng tôi sẽ yêu cầu gửi lại hình ảnh hoặc thông tin liên quan để xác minh yêu cầu.
- Sau khi xác minh, chúng tôi sẽ tiến hành đổi sản phẩm hoặc hoàn tiền theo yêu cầu của khách hàng.

Chúng tôi không chỉ đơn thuần là một cửa hàng cây cảnh; chúng tôi là nơi để bạn khám phá, học hỏi và kết nối với những người có cùng đam mê. Hãy theo dõi blog của chúng tôi để cập nhật những thông tin mới nhất về cây cảnh, xu hướng thiết kế không gian xanh và mẹo chăm sóc cây.

Miễn Trừ Trách Nhiệm

Chúng tôi không chịu trách nhiệm cho những thiệt hại phát sinh do việc không tuân thủ hướng dẫn chăm sóc hoặc các yếu tố bên ngoài không thể kiểm soát.

Cây Cảnh Tươi Xanh

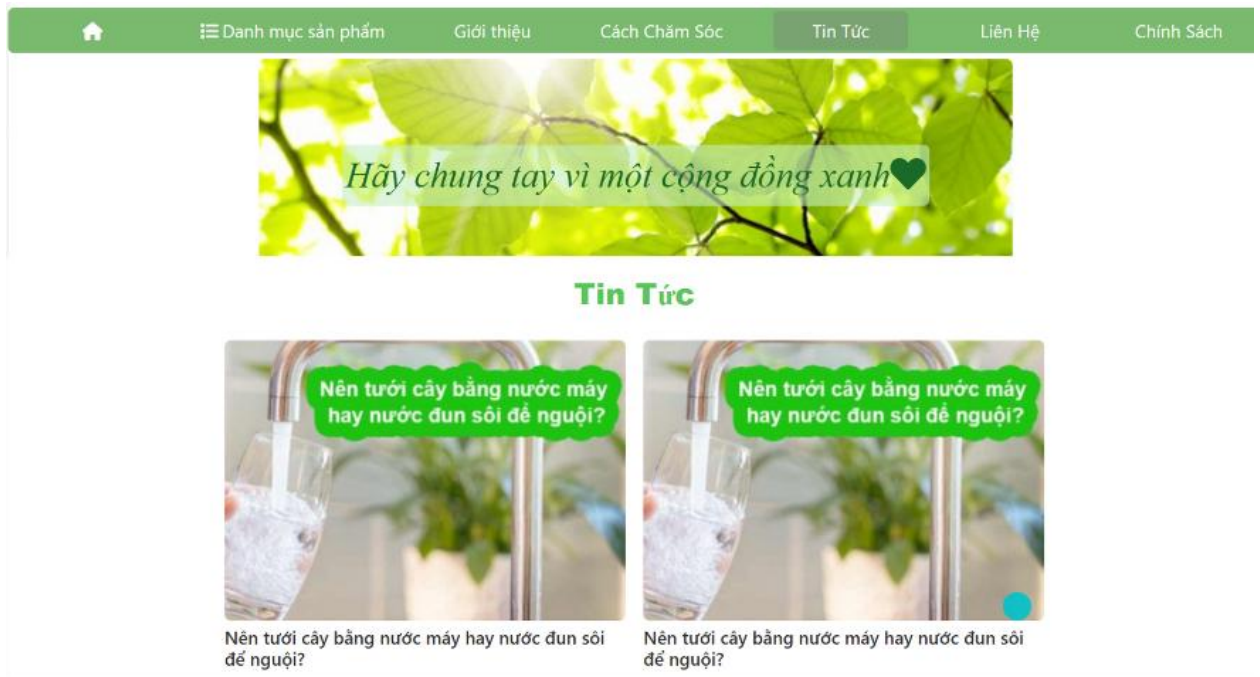
– Mang thiên nhiên đến gần hơn với cuộc sống của bạn!

– Mục đích của trang "Chính sách":

- **Đảm bảo quyền lợi khách hàng:** Trang này được thiết kế để khách hàng hiểu rõ các quy định liên quan đến bảo hành, đổi trả, và trách nhiệm khi mua hàng, giúp họ yên tâm khi mua sắm.
- **Xây dựng niềm tin với khách hàng:** Việc minh bạch các chính sách bảo hành và đổi trả giúp khách hàng tin tưởng hơn vào chất lượng sản phẩm và dịch vụ của website.
- **Giảm thiểu tranh chấp:** Khi có vấn đề phát sinh, các chính sách rõ ràng sẽ là căn cứ để giải quyết xung đột một cách minh bạch và công bằng.

- Tuân thủ pháp lý: Các quy định này thường là yêu cầu bắt buộc đối với các trang thương mại điện tử để đảm bảo quyền lợi của khách hàng.
- Cải thiện trải nghiệm người dùng: Khi có vấn đề xảy ra, người dùng dễ dàng tìm thấy hướng dẫn xử lý mà không cần liên hệ trực tiếp.
- Thể hiện tính chuyên nghiệp: Một website có chính sách rõ ràng cho thấy sự đầu tư và trách nhiệm của doanh nghiệp đối với khách hàng.

4.8 Trang tin tức.



– Mục đích của trang "Tin tức":

- Cung cấp thông tin hữu ích cho khách hàng: Trang "Tin tức" được sử dụng để chia sẻ kiến thức, mẹo chăm sóc cây trồng, xu hướng về cây cảnh, hoặc thông tin liên quan đến lĩnh vực nông nghiệp.
- Xây dựng sự tương tác: Việc chia sẻ nội dung giá trị có thể thu hút người dùng quay lại website thường xuyên để cập nhật thông tin mới.
- Tăng giá trị thương hiệu: Thể hiện website không chỉ là nơi bán hàng mà còn là một nguồn thông tin uy tín và chuyên sâu, qua đó khẳng định sự chuyên nghiệp.

- Nhu cầu tìm hiểu của người dùng: Khách hàng không chỉ tìm nơi mua cây mà còn muốn biết cách chăm sóc, chọn cây phù hợp, hoặc học hỏi những xu hướng mới.
- Giữ chân khách hàng: Một trang tin tức liên tục cập nhật sẽ tạo lý do để khách hàng quay lại, ngay cả khi họ chưa có nhu cầu mua hàng.
- Hỗ trợ tăng doanh số: Các bài viết tin tức có thể lồng ghép thông tin về sản phẩm hoặc dịch vụ, từ đó gián tiếp thúc đẩy quyết định mua hàng

4.9 Chức năng giỏ hàng nhỏ.

- Chức năng giúp hiển thị giỏ hàng nhỏ trên mỗi trang giúp người dùng tiện theo dõi sản phẩm mình đã lựa chọn. Được hiển thị khi người dùng di chuột vào biểu tượng giỏ hàng trên thanh điều hướng.



- Giỏ hàng nhỏ sẽ được cập nhật khi người dùng đăng nhập vào trang chủ với hình ảnh, tên, số lượng và giá của chúng.

```
def index(request):
    products = Plant.objects.all()# lấy tất cả sản phẩm
    paginator = Paginator(products, 6) # Mỗi trang sẽ có 6 sản phẩm
    page_number = request.GET.get('page') # Lấy số trang từ URL
    page_obj = paginator.get_page(page_number) # Lấy trang hiện tại
    cartItems = {}#tạo nơi chứa sản phẩm cho giỏ hàng
    if request.user.is_authenticated:#kiểm tra xem người dùng có đăng nhập chưa
        cartItems = CartItem.objects.filter(user = request.user)
        #lấy các sản phẩm tương ứng với user
    return render(request,'index.html',{'page_obj':page_obj,'cart_items': cartItems})
```

- Phương thức hiển thị lên trang web tương ứng với các trường hợp khác nhau

```

<div class = "cart">
  <div class = "cart-title"> Sản phẩm đã thêm</div>
  {% if user.is_authenticated %} <!-- kiểm tra user đã đăng nhập hay chưa -->
  {% if cart_items %} <!-- kiểm tra giỏ hàng người dùng có sản phẩm hay không -->
  <div class = "cart-content scrollable">
    {% for item in cart_items %}<!-- hiển thị sản phẩm -->
    <div class = "product-cart">
      
      <h4>{{item.product.plant_name}}</h4>
      <span class="price">{{item.product.price}}Đ</span>
      <span class="quantity">{{item.quantity}}Cây</span>
    </div>
    {% endfor %}
  </div>
  {% else %} <!-- hiển thị khi người dùng chưa thêm sản phẩm -->
  <div class="product-none" style = "color:black;"><strong>Giỏ hàng của bạn trống</strong></div>
  {% endif %}
  {% else %} <!-- hiển thị khi người dùng chưa đăng nhập -->
  <div class="product-none" style = "color:black;"><strong>Hãy đăng nhập để xem giỏ hàng</strong></div>
  {% endif %}

```

- Giỏ hàng nhỏ sẽ được cập nhật tương ứng với thao tác thêm sản phẩm mới hay xóa sản phẩm khỏi giỏ hàng của người dùng

```

def cart_mini_update(request):
    if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'GET':
        try:
            cart_items = CartItem.objects.filter(user=request.user) # lấy danh sách sản phẩm mới tương ứng người dùng
            product_list = [] # tạo nơi chứa sản phẩm
            for item in cart_items: # dùng for lấy những thông tin cần thiết của từng sản phẩm
                item_data = {
                    'id': item.id,
                    'name': item.product.plant_name,
                    'price': item.product.price,
                    'image': item.product.image,
                    'quantity': item.quantity,
                }
                product_list.append(item_data)
            # trả về dữ liệu
            return JsonResponse({
                'status': 'success',
                'cartProducts': product_list,
            })
        except CartItem.DoesNotExist: # trả về lỗi khi model không hoạt động đúng
            return JsonResponse({'status': 'error', 'message': 'Không tìm thấy sản phẩm trong giỏ hàng!'})

```

4.10 Chức năng thêm sản phẩm vào giỏ hàng.

- Người dùng có thể sử dụng chức năng này khi nhấn vào hiệu tượng xe đẩy trên mỗi sản phẩm



Cây Kim Tiền

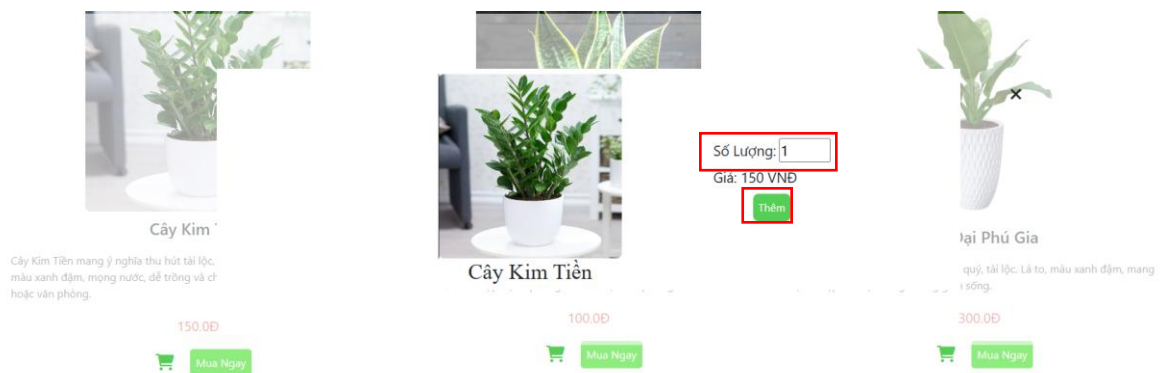
Cây Kim Tiền mang ý nghĩa thu hút tài lộc, may mắn và thịnh vượng. Lá có màu xanh đậm, mọng nước, dễ trồng và chăm sóc, thích hợp đặt trong nhà hoặc văn phòng.

150.0Đ



Mua Ngay

- Trang web sẽ hiển thị giao diện để người dùng lựa chọn số lượng sản phẩm và hoàn tất việc thêm



- Trang web sẽ hiển thị thông báo sau khi người dùng thêm hoàn tất



- Cách hoạt động

- Tạo giao diện hiển thị sản phẩm người dùng muốn thêm dựa trên dữ liệu từ server

```

$('.add').on('click',function() { // lắng nghe sự kiện người dùng click icon giỏ hàng trên mỗi sản phẩm

    var ItemId = $(this).data('cart-item-id'); // lấy id của sản phẩm tương ứng
    // Gửi yêu cầu AJAX để cập nhật số lượng
    $.ajax({
        url: addURL,
        type: 'GET',
        data: {
            'item_id': ItemId,
            'csrfmiddlewaretoken': csrfToken
        },
        success: function(response) { // xử lý dữ liệu trả về thành công

            if (response.status === 'success') { // hiển thị display với sản phẩm tương ứng mà người dùng muốn thêm
                var imageUrl = "/static/images/" + response.product.image; // tạo đường link tới ảnh của sản phẩm
                var price = ' '+response.product.price + ' VND' // tạo hiển thị giá sản phẩm
                $('#product-cart-img').attr('src', imageUrl); // hiển thị ảnh
                $('#cart-price-content').html(price); // hiển thị giá sản phẩm
                $('#product-cart-name').html(response.product.name); // hiển thị tên sản phẩm
                $('#add_complete').attr('value', response.product.id); // truyền id sản phẩm vào nút thêm vào giỏ hàng
                $('#add-to-cart-display').removeClass('hidden');

            } else {
                alert(response.message); // hiển thị thông báo lỗi
            }
        },
        error: function() { // hiển thị thông báo lỗi
            alert('Có lỗi xảy ra. Vui lòng thử lại!');
        }
    });
}

```

- Phía server

```

@login_required
Tabnine | Edit | Test | Explain | Document | Ask
def add(request):
    if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'GET':
        item_id = request.GET.get('item_id')
        try:
            item = Plant.objects.get(id=item_id) # lấy sản phẩm tương ứng

            return JsonResponse({
                'status': 'success',
                'product': {
                    'id': item_id, # id sản phẩm
                    'name': item.plant_name, # tên sản phẩm
                    'price': item.price, # giá sản phẩm
                    'image': item.image, # tên ảnh
                }
            })
        except:
            return JsonResponse({
                'status': 'error',
                'message': 'Sản phẩm không tồn tại'
            })

```

- Thực hiện thêm vào cơ sở dữ liệu giỏ hàng của người dùng

```

    $.add_complete').on('click',function() {
        var ItemId = $(this).attr('value'); // lấy id sản phẩm
        var quantity = $('cart-quantity').value; // số lượng của sản phẩm
        // gửi yêu cầu
        $.ajax({
            url: addURL,
            type: 'POST',
            data: {
                'item_id': ItemId,
                'quantity': quantity,
                'csrfmiddlewaretoken': csrfToken
            },
            success: function(response) {
                if (response.status === 'success') {
                    $('.add-to-cart-display').addClass('hidden'); // ẩn lớp thêm sản phẩm
                    update_cart_item(); // gọi hàm cập nhập sản phẩm trong giỏ hàng
                    alert('Thêm vào giỏ hàng thành công!!!') // thông báo đến người dùng
                } else {
                    alert(response.message); // thông báo lỗi
                }
            },
            error: function() {
                alert('Có lỗi xảy ra. Vui lòng thử lại!');
            }
        });
    });
});

```

- Phía server

```

if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'POST':
    item_id = request.POST.get('item_id') # lấy id sản phẩm
    quantity = int(request.POST.get('quantity',1)) # số lượng sản phẩm
    try:
        plant = Plant.objects.get(id=item_id) # lấy dữ liệu sản phẩm tương ứng

        # Kiểm tra xem sản phẩm đã có trong giỏ hàng của người dùng chưa
        cart_item = CartItem.objects.filter(product=plant, user=request.user).first()
        if cart_item:
            # Nếu sản phẩm đã có trong giỏ hàng, tăng số lượng
            cart_item.quantity += quantity # tăng số lượng sản phẩm
            cart_item.save()
        else:
            # Nếu không có, tạo mới sản phẩm trong giỏ hàng
            cartItem = CartItem.objects.create(product=plant, quantity=quantity, user = request.user) # tạo một hàng mới tương ứng
            cartItem.save()
        return JsonResponse({
            'status': 'success'
        })
    except Plant.DoesNotExist:
        return JsonResponse({
            'status': 'error'
        })

```

- Mục đích của "Giỏ hàng nhỏ":

- Tăng tính tiện lợi: Giúp người dùng nhanh chóng kiểm tra các sản phẩm đã thêm vào giỏ mà không cần chuyển sang trang giỏ hàng chính.
- Tối ưu hóa thời gian thao tác: Người dùng có thể dễ dàng xem, chỉnh sửa (tăng/giảm số lượng, xóa sản phẩm) mà không cần rời khỏi trang hiện tại.

- Tạo cảm giác tiện nghi: Một giỏ hàng nhỏ để truy cập giúp người dùng cảm thấy website được thiết kế chu đáo và hiện đại.

4.11 Chức năng giỏ hàng.

- Trang hiển thị thông tin về số lượng, hình ảnh và tên sản phẩm, tạo môi trường thân thiện cho người dùng thao tác xóa sản phẩm, tăng giảm số lượng sản phẩm, mua một hoặc nhiều sản phẩm cùng lúc.

Giỏ hàng của bạn

Sản phẩm

Đơn giá

Số lượng

Số tiền

Thao tác

Cây Kim Tiền

150.0

+ 2 -

300.0

Cây Lưỡi Hổ

100.0

+ 2 -

200.0

Mã giảm giá

Chọn hoặc nhập mã

-0 ngàn

Chọn tất cả

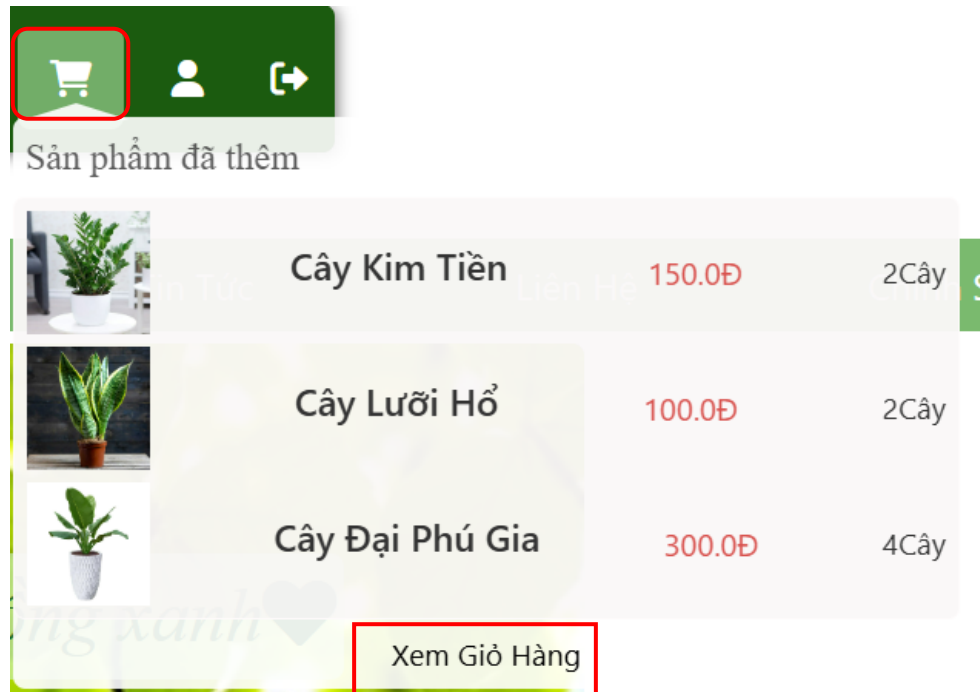
Xóa tất cả

Tổng thanh toán

0.00VNĐ

Mua ngay

- Hiển thị giỏ hàng khi người dùng nhấn vào biểu tượng giỏ hàng trên thanh điều hướng hoặc nút xem giỏ hàng trong giỏ hàng nhỏ.

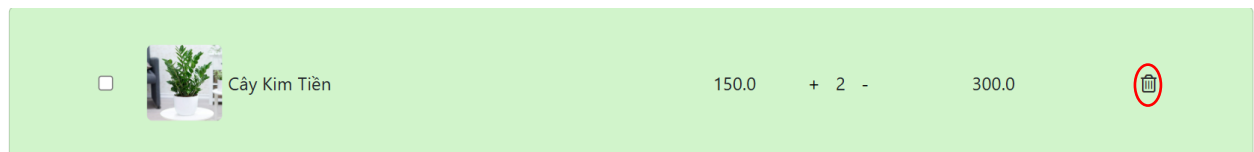


- Lấy dữ liệu cho hiển thị trang web

```
@login_required # yêu cầu người dùng cần đăng nhập để xem được trang
Tabnine | Edit | Test | Explain | Document | Ask
def cart_view(request):
    cart_items = CartItem.objects.filter(user=request.user) # lấy sản phẩm ứng với user tương ứng
    return render(request, 'cart.html', {'cart_items': cart_items})
```

- Thao tác xóa sản phẩm gồm 2 thao tác

- Xóa 1 sản phẩm đơn lẻ



- Cách hoạt động : Xóa hiển thị sản phẩm và gửi yêu cầu xóa sản phẩm trong cơ sở dữ liệu giỏ hàng của người dùng

```

$('.delete-cart-item').click(function() {
    var cartItemId = $(this).data('cart-item-id'); // Lấy ID của sản phẩm
    var divCha = $('#'+cartItemId); // lấy thẻ chứa sản phẩm

    // Gửi yêu cầu AJAX để xóa sản phẩm khỏi giỏ hàng
    $.ajax({
        url: "{% url 'delete_cart_item' %}",
        type: 'POST',
        data: {
            'cart_item_id': cartItemId
        },
        success: function(response) {
            if (response.status === 'success') {
                // Nếu xóa thành công, loại bỏ sản phẩm khỏi giỏ hàng
                divCha.remove(); // xóa bỏ thẻ chứa
                $('#cart-item-' + cartItemId).remove(); // Loại bỏ phần tử có id cart-item-{cartItemId} khỏi bảng
            }
        }
    });
});

```

- Hoặc trả về lỗi khi hàm hoạt động không đúng

```

    } else {
        alert(response.message);
    }
},
error: function() {
    alert('Có lỗi xảy ra. Vui lòng thử lại!');
}

```

- Thông tin sẽ được gửi về server, xóa sản phẩm trong cơ sở dữ liệu

```

def delete_cart_item(request):
    if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'POST':
        cart_item_id = request.POST.get('cart_item_id') # lấy id sản phẩm trong giỏ hàng

        try:
            # Lấy CartItem từ cơ sở dữ liệu tương ứng với id
            cart_item = CartItem.objects.get(id=cart_item_id, user=request.user)

            # Xóa sản phẩm trong cơ sở dữ liệu của giỏ hàng
            cart_item.delete()

            # Trả lại phản hồi JSON
            return JsonResponse({
                'status': 'success',
                'message': 'Sản phẩm đã được xóa khỏi giỏ hàng!'
            })

        except CartItem.DoesNotExist:
            return JsonResponse({'status': 'error', 'message': 'Đã có lỗi xảy ra!'})

    return JsonResponse({'status': 'error', 'message': 'Yêu cầu không hợp lệ!'})

```

- Xóa toàn bộ sản phẩm

Mã giảm giá

Chọn hoặc nhập mã

-0 ngàn

☐ Chọn tất cả

Xóa tất cả

Tổng thanh toán 0.00VNĐ

Mua ngay

- Cách hoạt động

```
def delete_all(request):
    cart_items = CartItem.objects.filter(user=request.user) # lấy sản phẩm tương ứng với user
    cart_items.delete() # xóa toàn bộ sản phẩm
    return redirect('cart') # điều hướng về lại trang |
```

- Kết quả

Giỏ hàng của bạn

<input type="checkbox"/> Sản phẩm	Đơn giá	Số lượng	Số tiền	Thao tác
Giỏ hàng của bạn hiện đang trống.				
Mã giảm giá <input type="text"/> Chọn hoặc nhập mã				
<input type="checkbox"/> Chọn tất cả	Xóa tất cả	Tổng thanh toán	0.00VNĐ	<input type="button" value="Mua ngay"/>

-0 ngàn

- Thao tác tăng giảm số lượng sản phẩm

<input type="checkbox"/>		Cây Kim Tiền	150.0	<input type="button" value="+"/> 1 <input type="button" value="-"/>	150.0	
--------------------------	--	--------------	-------	---	-------	--

- Cách thức hoạt động của thao tác tăng số lượng

```
// Tăng số lượng sản phẩm
$('.quantity-increase').click(function() {
    var cartItemId = $(this).data('cart-item-id'); // lấy id sản phẩm
    var quantityElement = $('#quantity-' + cartItemId); // lấy số lượng sản phẩm hiện tại
    var quantity = parseInt(quantityElement.text()) + 1; // tăng số lượng lên 1
    var total = $('#total-price-' + cartItemId); // lấy phần tử hiển thị tổng giá của sản phẩm tương ứng
    var priceElement = $('#product-price-' + cartItemId); // lấy phần tử hiển thị giá sản phẩm tương ứng
    var price = parseInt(priceElement.text()); // chuyển đổi giá sang int
    var totalPrice = quantity * price; // tính lại tổng giá của sản phẩm

    // Gửi yêu cầu AJAX để cập nhật số lượng
    $.ajax({
        url: "{% url 'update_cart_item_quantity' %}",
        type: 'POST',
        data: {
            'cart_item_id': cartItemId,
            'quantity': quantity,
            'csrfmiddlewaretoken': '{{ csrf_token }}'
        },
    });
```

```

    csrfMiddlewareToken : { csrf_token }
  },
  success: function(response) {
    if (response.status === 'success') {
      // Cập nhật lại số lượng
      quantityElement.text(quantity); // cập nhật lại phần tử hiển thị số lượng
      total.text(totalPrice + 'VNĐ') // cập nhật lại phần tử hiển thị giá
      calculateTotalPayment(); // gọi hàm cập nhật lại tổng thanh toán nếu sản phẩm đang được chọn
    } else {
      alert(response.message); // gửi lỗi nếu hàm không hoạt động đúng
    }
  },
  error: function() {
    alert('Có lỗi xảy ra. Vui lòng thử lại!'); // gửi lỗi nếu dữ liệu trả về không đúng
  }
});

```

– Xử lý bên phía cơ sở dữ liệu

```

def update_cart_item_quantity(request):
    if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'POST':
        cart_item_id = request.POST.get('cart_item_id')
        new_quantity = int(request.POST.get('quantity'))

        try:
            cart_item = CartItem.objects.get(id=cart_item_id, user=request.user) # lấy sản phẩm tương ứng trong giỏ hàng của người dùng

            # Kiểm tra xem số lượng có hợp lệ không
            if new_quantity < 1:
                return JsonResponse({'status': 'error', 'message': 'Số lượng phải lớn hơn hoặc bằng 1'})

            # Cập nhật số lượng trong giỏ hàng
            cart_item.quantity = new_quantity
            cart_item.save()

            # Tính lại tổng giá trị giỏ hàng

            return JsonResponse({
                'status': 'success',
                'message': 'Số lượng đã được cập nhật!'
            })
        except:
            return JsonResponse({'status': 'error', 'message': 'Có lỗi xảy ra'})

```

– Thao tác giảm số lượng sản phẩm

```

// Giảm số lượng sản phẩm
$('.quantity-decrease').click(function() {
    var cartItemId = $(this).data('cart-item-id'); // lấy id sản phẩm
    var quantityElement = $('#quantity-' + cartItemId); // lấy số lượng sản phẩm
    var quantity = parseInt(quantityElement.text()) - 1; // chuyển kiểu dữ liệu và trừ đi 1
    var total = $('#total-price-' + cartItemId); // lấy phần tử hiển thị tổng giá của sản phẩm
    var priceElement = $('#product-price-' + cartItemId); // lấy phần tử hiển thị giá sản phẩm
    var price = parseInt(priceElement.text()); // chuyển giá sản phẩm sang integer
    var totalPrice = quantity * price; // tính lại tổng giá của sản phẩm
    // Kiểm tra số lượng phải lớn hơn 1
    if (quantity < 1) {
        alert('Số lượng phải lớn hơn hoặc bằng 1');
        return;
    }

    // Gửi yêu cầu AJAX để cập nhật số lượng
    $.ajax({
        url: "{% url 'update_cart_item_quantity' %}",
        type: 'POST',
        data: {

```

```

url: "{% url 'update_cart_item_quantity' %}",
type: 'POST',
data: {
    'cart_item_id': cartItemId,
    'quantity': quantity,
    'csrfmiddlewaretoken': '{{ csrf_token }}'
},
success: function(response) {
    if (response.status === 'success') {
        // Cập nhật lại số lượng và tổng giá của sản phẩm
        quantityElement.text(quantity); // cập nhật lại số lượng
        total.text(totalPrice + 'VNĐ') // cập nhật lại tổng giá
        calculateTotalPayment(); // cập nhật lại tổng thanh toán nếu sản phẩm đang được chọn
    } else {
        alert(response.message);
    }
}

```

– Xử lý bên phía cơ sở dữ liệu

```

def update_cart_item_quantity(request):
    if request.headers.get('X-Requested-With') == 'XMLHttpRequest' and request.method == 'POST':
        cart_item_id = request.POST.get('cart_item_id')
        new_quantity = int(request.POST.get('quantity'))

        try:
            cart_item = CartItem.objects.get(id=cart_item_id, user=request.user) # lấy sản phẩm tương ứng trong giỏ hàng của người dùng

            # Kiểm tra xem số lượng có hợp lệ không
            if new_quantity < 1:
                return JsonResponse({'status': 'error', 'message': 'Số lượng phải lớn hơn hoặc bằng 1'})







            # Cập nhật số lượng trong giỏ hàng
            cart_item.quantity = new_quantity
            cart_item.save()

            # Tính lại tổng giá trị giỏ hàng

            return JsonResponse({
                'status': 'success',
                'message': 'Số lượng đã được cập nhật!'
            })

```

- Thao tác chọn sản phẩm sẽ cho người dùng cái nhìn tổng quan về tổng thanh toán
 - Chọn 1 hoặc các sản phẩm riêng lẻ

		Cây Kim Tiền	150.0	+	2	-	300VNĐ	
		Cây Lưỡi Hổ	100.0	+	1	-	100.0	

Mã giảm giá Chọn hoặc nhập mã

-0 ngàn

☐ Chọn tất cả ☐ Xóa tất cả

Tổng thanh toán 0.00 VND

- Cách hoạt động

```
// Hàm tính toán tổng thanh toán

// Lắng nghe sự kiện thay đổi trạng thái của checkbox (tick/uncheck)
$('.product-checkbox').change(function() {
    calculateTotalPayment(); // Gọi hàm tính tổng thanh toán mỗi khi checkbox thay đổi
});

function calculateTotalPayment() {
    var totalPayment = 0; // tạo biến tổng thanh toán

    // Duyệt qua tất cả các checkbox đã được chọn
    $('.product-checkbox:checked').each(function() {
        var cartItemId = $(this).attr('value'); // lấy id sản phẩm
        var quantityElement = $('#quantity-' + cartItemId); // số lượng sản phẩm
        var quantity = parseInt(quantityElement.text()); // chuyển đổi số lượng sang integer
        var priceElement = $('#product-price-' + cartItemId); // lấy giá sản phẩm
        var price = parseInt(priceElement.text()); // chuyển đổi sang dạng integer
        var totalPrice = price * quantity; // tính tổng giá của sản phẩm
        totalPayment += totalPrice; // thực hiện cộng dồn ph

    });

    // Cập nhật tổng thanh toán vào giao diện
    $('#total-payment').text(totalPayment.toFixed(2) + ' VND');
}
```

- Kết quả

Giỏ hàng của bạn

<input type="checkbox"/> Sản phẩm	Đơn giá	Số lượng	Số tiền	Thao tác
<input checked="" type="checkbox"/>  Cây Kim Tiền	150.0	+ 3 -	450.0	
<input type="checkbox"/>  Cây Lưỡi Hổ	100.0	+ 1 -	100.0	

Mã giảm giá Chọn hoặc nhập mã






-0 ngàn

☐ Chọn tất cả ☐ Xóa tất cả

Tổng thanh toán 450.00 VND

- Chọn tất cả








Giỏ hàng của bạn

<input type="checkbox"/> Sản phẩm	Đơn giá	Số lượng	Số tiền	Thao tác
<input type="checkbox"/>  Cây Kim Tiền	150.0	+ 3 -	450.0	
<input type="checkbox"/>  Cây Lưỡi Hổ	100.0	+ 1 -	100.0	
Mã giảm giá Chọn hoặc nhập mã				
<input type="checkbox"/> Chọn tất cả		Xóa tất cả	Tổng thanh toán 0.00VND	Mua ngay

- Cách hoạt động

```
// Chọn/deselect tất cả sản phẩm
$('#select-all').change(function() {
    var isChecked = $(this).prop('checked'); // Lấy trạng thái của checkbox "Chọn tất cả"
    $('.product-checkbox').prop('checked', isChecked); // Thay đổi trạng thái của tất cả các checkbox
    calculateTotalPayment(); // Cập nhật lại tổng thanh toán
});
$('#select-all-2').change(function() {
    var isChecked = $(this).prop('checked'); // Lấy trạng thái của checkbox "Chọn tất cả"
    $('.product-checkbox').prop('checked', isChecked); // Thay đổi trạng thái của tất cả các checkbox
    calculateTotalPayment(); // Cập nhật lại tổng thanh toán
});
```

- Kết quả

<input type="checkbox"/> Sản phẩm	Đơn giá	Số lượng	Số tiền	Thao tác
<input checked="" type="checkbox"/>  Cây Kim Tiền	150.0	+ 3 -	450.0	
<input checked="" type="checkbox"/>  Cây Lưỡi Hổ	100.0	+ 1 -	100.0	
<input checked="" type="checkbox"/>  Cây Đại Phú Gia	300.0	+ 1 -	300.0	
Mã giảm giá Chọn hoặc nhập mã				
<input checked="" type="checkbox"/> Chọn tất cả		Xóa tất cả	Tổng thanh toán 900.00 VND	Mua ngay

- Mục đích của chức năng "Giỏ hàng":
 - Quản lý sản phẩm muốn mua: Cho phép người dùng lưu trữ tạm thời các sản phẩm mà họ quan tâm để tiện kiểm tra trước khi thanh toán.
 - Hỗ trợ quá trình mua sắm: Tạo trải nghiệm liền mạch khi người dùng thêm, xóa, hoặc thay đổi số lượng sản phẩm mà không cần thao tác phức tạp.
 - Hỗ trợ khách hàng quên hoặc phân vân: Giỏ hàng giữ các sản phẩm trong một khoảng thời gian nhất định, cho phép khách hàng quay lại và tiếp tục mua sắm.
 - Trải nghiệm mua sắm tiện lợi: Cung cấp đầy đủ thông tin như tên sản phẩm, giá, số lượng, phí vận chuyển (nếu có), giúp người dùng tính toán chi phí dễ dàng.

4.12 Chức năng tìm kiếm.

- Khi người dùng nhập nội dung nhấn enter hoặc biểu tượng kính lúp chức năng sẽ trả về kết quả tương ứng



- Cách hoạt động
 - Người dùng sẽ được điều hướng đến trang tìm kiếm với kết quả tương ứng

```
def find_page(request):
    query = request.POST.get('input-find', '').lower() # lấy dữ liệu từ input search
    products = Plant.objects.filter(plant_name__icontains=query) # _icontains cho phép tìm kiếm các kết quả chứa hoặc khớp với dữ liệu tìm
    paginator = Paginator(products, 6) # Mỗi trang sẽ có 6 sản phẩm
    page_number = request.GET.get('page') # Lấy số trang từ URL
    page_obj = paginator.get_page(page_number) # Lấy trang hiện tại
    cartItems = {}
    if request.user.is_authenticated:
        cartItems = CartItem.objects.filter(user = request.user)
    return render(request, 'find_page.html', {'page_obj': page_obj, 'cart_items': cartItems, 'has_products': products.exists()})
# has_product trả về kết quả true hoặc false về việc tìm kiếm có kết quả hay không ho
```

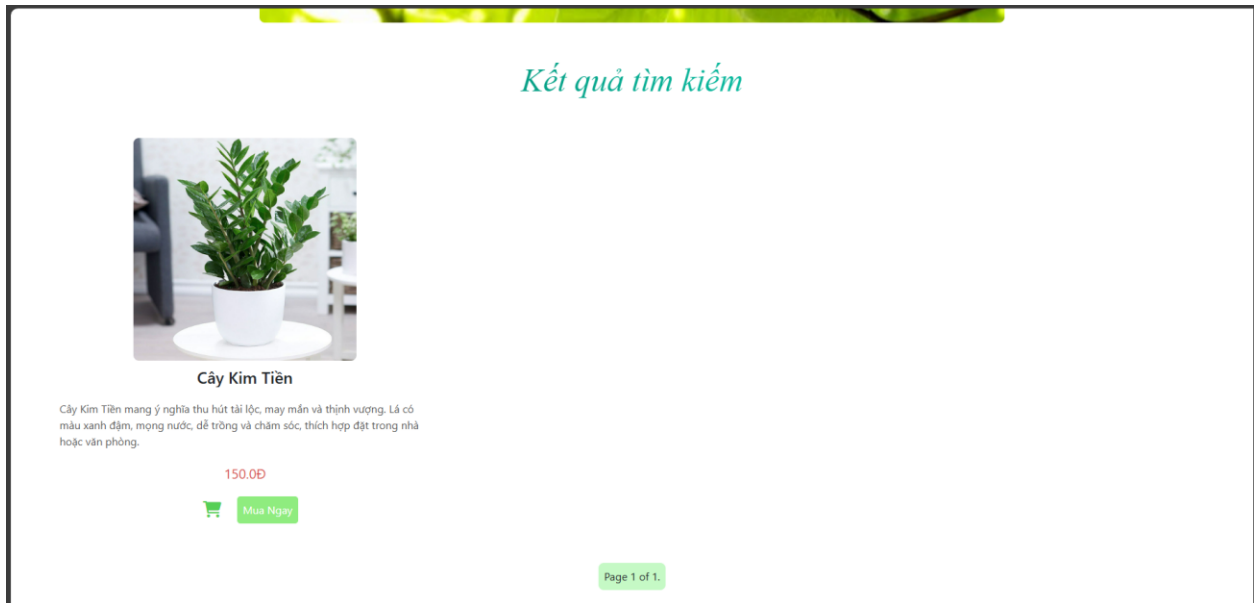
- Hiện thị

```

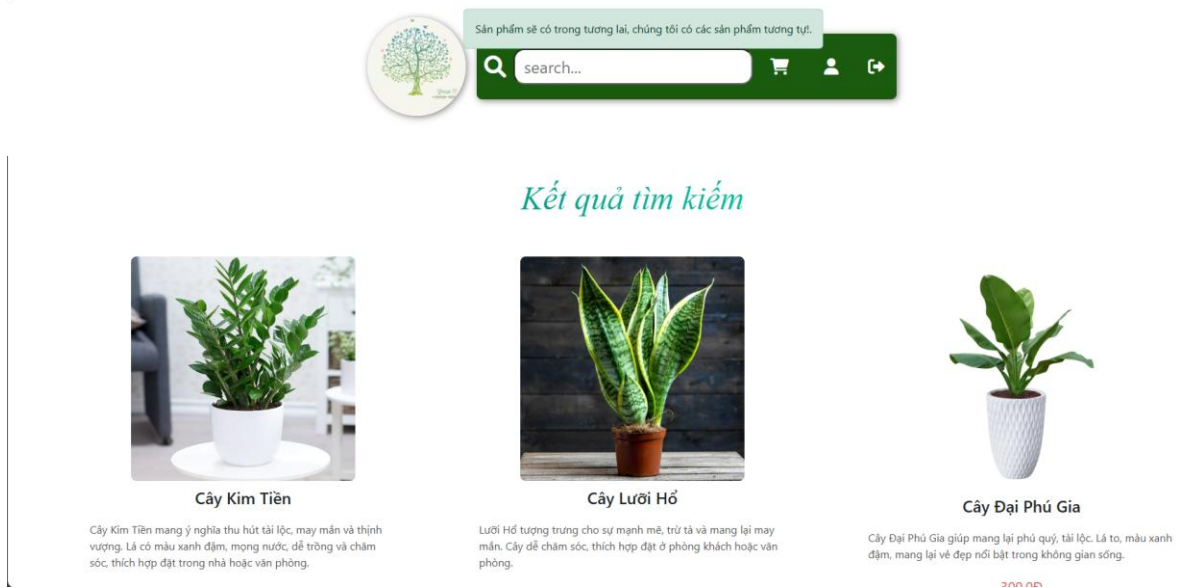
<body>
  {% block title %} Kết quả tìm kiếm {% endblock %}
  {% if has_products == False %} <!-- kiểm tra xem có kết quả không -->
    {% block content %}
      <div class="error_find_message">Không tìm thấy kết quả. Vui Lòng Thử Lại!</div>
    {% endblock %}
  {% endif %}
</body>

```

- Kết quả
- Khi có kết quả:

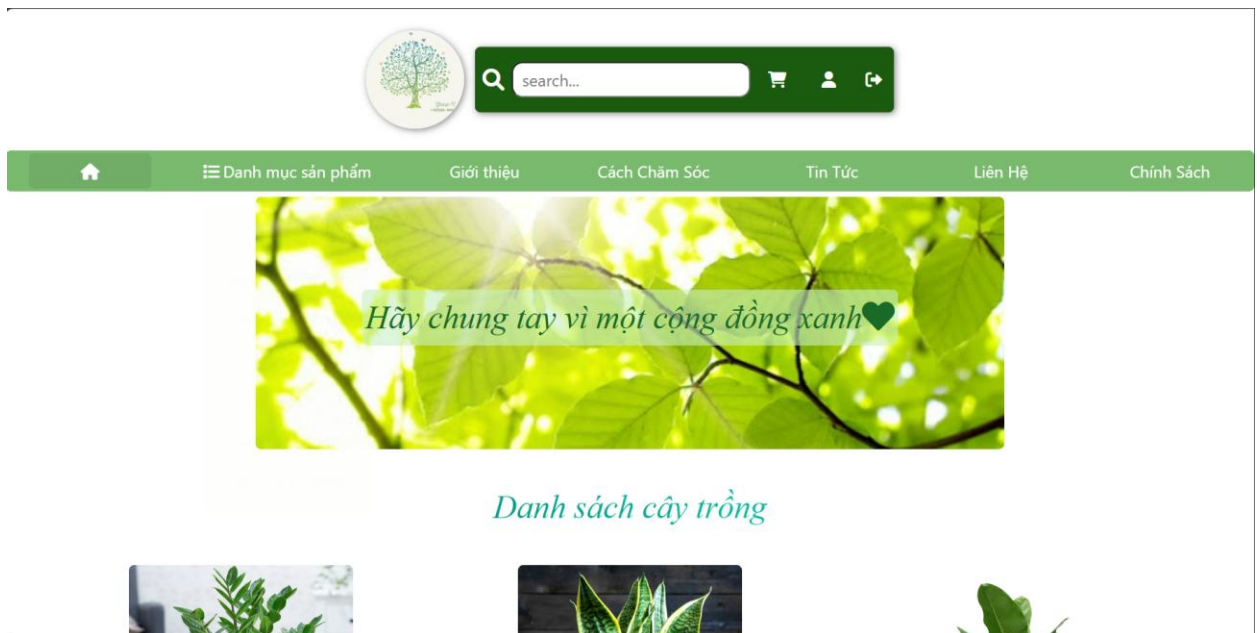


- Khi không có kết quả, trang web sẽ hiển thị sản phẩm đề xuất và trả về thông báo



- Mục đích của chức năng "Tìm kiếm":
 - Cải thiện trải nghiệm người dùng : Cho phép người dùng nhanh chóng tìm thấy thông tin hoặc sản phẩm họ cần mà không phải duyệt qua nhiều trang hay danh mục.
 - Tăng tốc quá trình mua sắm: Rút ngắn thời gian tìm kiếm sản phẩm cụ thể, giúp người dùng đưa ra quyết định mua sắm nhanh hơn.
 - Hướng người dùng tới nội dung phù hợp: Kết hợp với các từ khóa phổ biến, tính năng này có thể gợi ý nội dung hoặc sản phẩm phù hợp với nhu cầu tìm kiếm.
 - Tăng trải nghiệm người dùng : Người dùng có thể nhanh chóng tìm được nội dung mong muốn thay vì rời khỏi trang do không tìm thấy sản phẩm phù hợp.

4.13 Trang chủ.



- Giỏ hàng:
 - Hiển thị biểu tượng giỏ hàng.

- Nếu người dùng đã đăng nhập và có sản phẩm, hiển thị danh sách sản phẩm đã thêm vào giỏ.

– Đăng nhập/Đăng xuất:

- Nút đăng nhập chuyển hướng đến trang Login.
- Nếu người dùng đã đăng nhập, hiển thị nút đăng xuất.



– Hiển thị sản phẩm: danh sách sản phẩm

- Hiển thị tất cả các sản phẩm với hình ảnh, tên, mô tả và giá của từng sản phẩm.

4.14 Danh mục sản phẩm.



– Mục đích của danh mục sản phẩm cây trồng

- Hỗ trợ quá trình mua sắm: Tạo ra một trải nghiệm mua sắm trực tuyến tiện lợi, giúp khách hàng dễ dàng lọc và tìm kiếm các sản phẩm cây trồng theo các tiêu chí như loại cây, kích thước, giá cả, hoặc đặc điểm môi trường.
- Tăng khả năng tương tác và gắn kết với khách hàng: Danh mục sản phẩm có thể bao gồm các tính năng như đánh giá, nhận xét từ khách hàng, giúp những người mua sau có thêm thông tin để ra quyết định. Đồng thời, website có thể đưa ra các gợi ý cây trồng phù hợp cho các không gian sống cụ thể.
- Tăng cường trải nghiệm người dùng: Thiết kế danh mục sản phẩm thân thiện và dễ điều hướng giúp khách hàng dễ dàng duyệt qua các loại cây trồng, nâng cao sự hài lòng và sự trung thành với website.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận.

- Web mang lại nhiều tiện ích cho người dùng trong việc truy cập thông tin và thực hiện các giao dịch cây trồng trực tuyến. Tuy nhiên, websites vẫn còn những nhược điểm cần được nâng cấp, cải thiện.
- Ưu điểm của websites:
 - Tiện lợi và dễ tiếp cận: Web giúp người dùng có thể dễ dàng truy cập và sử dụng mọi lúc, mọi nơi chỉ với một thiết bị kết nối internet. Điều này mang lại sự tiện lợi trong việc truy cập thông tin và thực hiện các hoạt động trực tuyến.
 - Mua sắm trực tuyến: Web cung cấp một nền tảng thuận tiện cho người dùng trong việc mua sắm trực tuyến. Người tiêu dùng có thể lựa chọn sản phẩm, đặt hàng, và thanh toán mà không cần phải đến cửa hàng trực tiếp, tiết kiệm thời gian và công sức.
 - Có thể truy cập thông tin mọi lúc, mọi nơi: Internet giúp người dùng truy cập thông tin liên tục, bất kể thời gian và địa điểm. Việc này rất hữu ích cho những người cần dùng ngay khi có yêu cầu.
 - Dễ dàng so sánh và lựa chọn sản phẩm phù hợp: Web cung cấp nhiều nền tảng cho phép người dùng so sánh các sản phẩm và dịch vụ khác nhau. Điều

này giúp người tiêu dùng đưa ra quyết định mua sắm thông minh, chọn lựa sản phẩm tốt nhất với giá cả hợp lý nhất, phù hợp với nhu cầu cá nhân.

– Nhược điểm của web:

- Giao diện còn đơn giản, khá đơn điệu: Một số trang vẫn còn sử dụng giao diện thiết kế đơn giản, thiếu sự sáng tạo và tính thẩm mỹ. Điều này có thể làm giảm trải nghiệm người dùng, khiến họ cảm thấy nhàm chán và không muốn quay lại website.
- Các loại cây còn hạn chế, chưa đa dạng: các loại cây có thể còn hạn chế, chưa đa dạng so với nhu cầu của người dùng. Việc này khiến người dùng khó tìm thấy các loại cây mà họ mong muốn, dẫn đến việc giảm khả năng cạnh tranh của website.

5.2. Hướng phát triển.

- Để nâng cao trải nghiệm người dùng và tối ưu hóa hiệu quả kinh doanh, web bán cây trồng có thể phát triển theo các hướng sau:
 - Sử dụng công nghệ AI: Công nghệ trí tuệ nhân tạo có thể giúp phân tích thói quen và sở thích của người dùng, từ đó đưa ra các gợi ý cá nhân hóa về cây trồng phù hợp với điều kiện môi trường và nhu cầu của khách hàng. AI cũng có thể giúp cải thiện khả năng tìm kiếm sản phẩm, giúp người dùng dễ dàng tìm được những loại cây trồng mà họ đang tìm kiếm.
 - Tối ưu hóa giao diện: Một giao diện đẹp, dễ sử dụng và thân thiện với người dùng cũng là yếu tố quan trọng. Tối ưu hóa giao diện giúp khách hàng dễ

dàng duyệt sản phẩm, tìm kiếm và mua hàng một cách nhanh chóng. Các tính năng như tìm kiếm nhanh, phân loại sản phẩm rõ ràng, hình ảnh sắc nét và thông tin chi tiết sẽ làm tăng sự hài lòng của người dùng.

- Thêm các gợi ý mua sắm: Việc tích hợp các gợi ý mua sắm thông minh dựa trên lịch sử tìm kiếm hoặc sản phẩm đã xem sẽ giúp người dùng khám phá thêm các cây trồng khác mà họ có thể yêu thích. Điều này không chỉ nâng cao trải nghiệm người dùng mà còn thúc đẩy doanh thu bằng cách khuyến khích khách hàng mua thêm sản phẩm.
- Mở thêm các dịch vụ sản phẩm: Để tạo ra giá trị gia tăng, website có thể mở rộng các dịch vụ như cung cấp dịch vụ chăm sóc cây trồng, tư vấn về cách trồng và bảo vệ cây, thiết kế vườn hoặc bán các sản phẩm phụ trợ như chậu, đất trồng, phân bón. Những dịch vụ này sẽ giúp khách hàng có trải nghiệm toàn diện hơn và tạo ra cơ hội kinh doanh mới cho website.
- Với những hướng phát triển này, web bán cây trồng không chỉ nâng cao chất lượng mà còn gia tăng sự hài lòng của người sử dụng đối với trang web.

TÀI LIỆU THAM KHẢO

1. Django Tutorial:

<https://www.w3schools.com/django/?gidzl=18e3KEAm35yDd0TIuAiENWoG15FygWnR49u5KQsg2GWIg1NhVe834F9L5BxhmfP79G3KMRU0ZewwReBKm>

2. Working with forms: <https://docs.djangoproject.com/en/5.1/topics/forms/>

3. User authentication in Django:

<https://docs.djangoproject.com/en/5.1/topics/auth/>

4. Django - Reset Password: https://dev.to/earthcomfy/django-reset-password-3k0l?utm_source=zalo&utm_medium=zalo&utm_campaign=zalo

5. Handling Ajax requests in Django: <https://www.geeksforgeeks.org/handling-ajax-request-in-django/>

6. Getting started with Django: <https://www.djangoproject.com/start/>