

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH



BÁO CÁO ĐỒ ÁN MÔN HỌC

LẬP TRÌNH WEB

Giảng viên hướng dẫn: Ths. Nguyễn Trung Hiếu

Đề tài: Website bán thiết bị điện gia dụng

Nhóm 10:

N20DCCN023 - Phan Xuân Huynh

N20DCCN037 - Phan Văn Lực

N20DCCN065 - Nguyễn Trần Trọng Tín

LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến “Học viện Công nghệ Bưu chính viễn thông” đã đưa môn học *Lập trình website* vào chương trình giảng dạy.

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giáo viên bộ môn - thầy giáo Nguyễn Trung Hiếu đã miệt mài dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Lập trình website, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn Lập trình website là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của nhóm chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn.

MỤC LỤC

| | |
|--|-----------|
| I. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG | 5 |
| Tên đề tài | 5 |
| Sử dụng | 5 |
| Công cụ sử dụng..... | 5 |
| Phân quyền và chức năng | 5 |
| Phân tích hệ thống | 6 |
| II. GIAO DIỆN | 12 |
| 1. Giao diện của khách hàng..... | 12 |
| 1.1. Trang chủ..... | 12 |
| 1.2. Trang đăng nhập..... | 13 |
| 1.3. Trang đăng ký..... | 13 |
| 1.4. Trang nhập mã xác minh | 14 |
| 1.5. Trang quên mật khẩu | 14 |
| 1.6. Trang lọc tìm theo loại sản phẩm..... | 14 |
| 1.7. Trang thông tin sản phẩm..... | 15 |
| 1.8. Trang tìm kiếm sản phẩm..... | 16 |
| 1.9. Trang giỏ hàng | 16 |
| 1.10. Trang đặt hàng..... | 16 |
| 1.11. Trang lịch sử đơn hàng..... | 17 |
| 1.12. Trang thông tin khách hàng..... | 17 |
| 1.13. Trang đổi mật khẩu | 17 |
| 2. Giao diện của tài khoản người quản lý | 18 |
| 2.1. Giao diện trang thống kê..... | 18 |
| 2.2. Giao diện trang quản lý đơn hàng | 19 |
| 2.3. Giao diện trang quản lý sản phẩm | 20 |
| 2.4. Giao diện trang thêm sản phẩm | 21 |

| | |
|---|-----------|
| 2.5. Giao diện trang sửa sản phẩm | 22 |
| 2.6. Giao diện trang thêm thương hiệu | 23 |
| 2.7. Giao diện trang sửa thương hiệu..... | 23 |
| 2.8. Giao diện trang quản lý tài khoản khách hàng | 24 |
| 2.9. Giao diện trang thông tin cá nhân | 24 |
| 2.10.Giao diện trang đổi mật khẩu..... | 25 |
| 3. Giao diện trang của tài khoản chủ (Owner) | 25 |
| III. TÍNH NĂNG | 27 |
| 1. Đăng kí-Đăng nhập-Lấy mật khẩu | 27 |
| 2. Thay đổi thông tin cá nhân, thay đổi mật khẩu (người dùng, admin,owner)..... | 36 |
| Người dùng: | 36 |
| Admin và owner:..... | 38 |
| 3. Tìm kiếm sản phẩm | 40 |
| 4. Bộ lọc sản phẩm | 42 |
| 5. Xem chi tiết sản phẩm | 43 |
| 6. Thêm sản phẩm vào giỏ hàng | 43 |
| 7. Mua hàng -Huỷ đơn hàng- xem lịch sử đơn hàng | 45 |
| Mua hàng | 45 |
| Xem lịch sử đơn hàng | 46 |
| 8. Đánh giá sản phẩm | 50 |
| 9. Xem thống kê doanh thu, đơn hàng..... | 50 |
| 10. Quản lý đơn hàng..... | 53 |
| 11. Quản lý sản phẩm, thương hiệu (thêm, xóa, sửa thông tin, sửa trạng thái) | 54 |
| 12. Xem thông tin và quản lý tài khoản khách hàng (khóa, mở khóa)..... | 62 |
| 13. Quản lý tài khoản admin(thêm mới, khóa, mở khóa) | 63 |

I. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

1. **Tên đề tài:** Xây dựng website bán thiết bị điện gia dụng

2. Sử dụng

-Spring MVC, SQL Server

-Hibernate, Validation, UploadFile, Interceptor, Email

-HTML, CSS, Bootstrap và một vài thư viện được thiết kế sẵn.

3. Công cụ sử dụng

- Microsoft SQL Server 2019

- Eclispe IDE.

4. Phân quyền và chức năng

Có 4 nhóm phân quyền: người dùng chưa có tài khoản, người dùng có tài khoản, người quản lý (admin), người chủ (owner),

- Người dùng chưa có tài khoản:

- ✓ Xem sản phẩm
- ✓ Tìm kiếm, lọc các sản phẩm
- ✓ Đăng ký tài khoản

- Người dùng đã đăng ký tài khoản

- ✓ Có các quyền như người dùng chưa có tài khoản
- ✓ Có thể thêm sản phẩm vào giỏ hàng và tạo đơn hàng
- ✓ Nhập mã khuyến mãi
- ✓ Xem lịch sử đơn hàng
- ✓ Đánh giá, bình luận về sản phẩm đã mua
- ✓ Chính sửa thông tin cá nhân

- Người quản lý (admin)

- ✓ Chính sửa thông tin cá nhân
- ✓ Xem thống kê doanh số, đơn hàng
- ✓ Thêm, sửa, xóa, chuyển trạng thái sản phẩm
- ✓ Thêm, sửa, xoá, chuyển trạng thái loại hàng
- ✓ Thêm, sửa, xoá, chuyển trạng thái thương hiệu
- ✓ Xem, quản lý đơn hàng
- ✓ Quản lý tài khoản khách hàng

- Người chủ (owner)

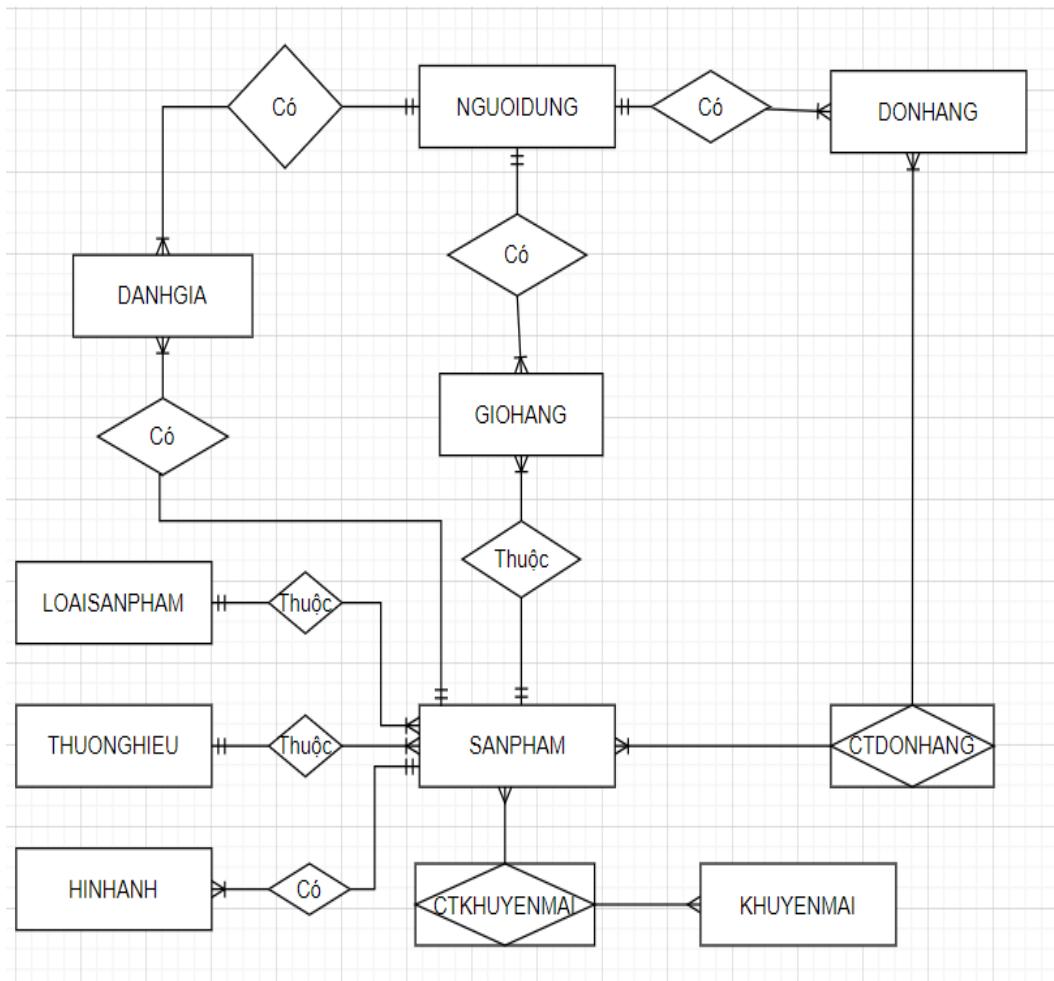
- ✓ Có các quyền như người quản lý
- ✓ Có thể thêm xóa tài khoản người quản lý.

5. Phân tích hệ thống

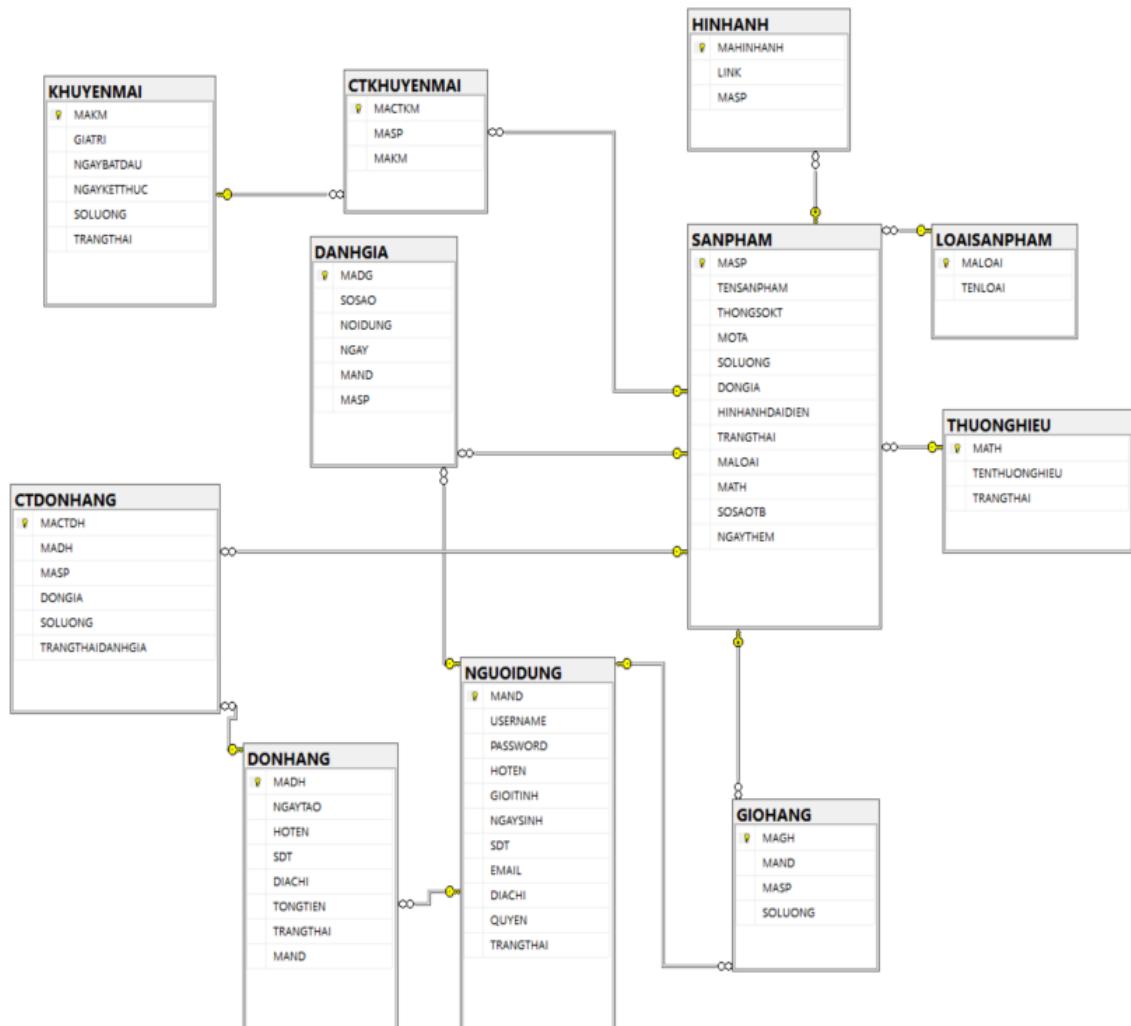
Gồm 11 bảng:

- NGUOIDUNG
- SANPHAM
- THUONGHIEU
- LOAISANPHAM
- HINHANH
- GIOHANG
- DONHANG
- CTDONHANG
- DANHGIA
- KHUYENMAI
- CTKHUYENMAI

ERD



Mô hình Diagram



Chi tiết bảng:**NGUOIDUNG**

| Column Name | Data Type | Allow Nulls |
|-------------|---------------|-------------------------------------|
| MAND | int | <input type="checkbox"/> |
| USERNAME | varchar(30) | <input type="checkbox"/> |
| PASSWORD | varchar(100) | <input type="checkbox"/> |
| HOTEN | nvarchar(50) | <input type="checkbox"/> |
| GIOITINH | bit | <input checked="" type="checkbox"/> |
| NGAYSINH | date | <input checked="" type="checkbox"/> |
| SDT | varchar(15) | <input type="checkbox"/> |
| EMAIL | varchar(30) | <input type="checkbox"/> |
| DIACHI | nvarchar(200) | <input checked="" type="checkbox"/> |
| QUYEN | int | <input type="checkbox"/> |
| TRANGTHAI | bit | <input type="checkbox"/> |
| | | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

SANPHAM

| Column Name | Data Type | Allow Nulls |
|----------------|----------------|--------------------------|
| MASP | varchar(20) | <input type="checkbox"/> |
| TENSANPHAM | nvarchar(50) | <input type="checkbox"/> |
| THONGSOKT | nvarchar(500) | <input type="checkbox"/> |
| MOTA | nvarchar(1000) | <input type="checkbox"/> |
| SOLUONG | int | <input type="checkbox"/> |
| DONGIA | money | <input type="checkbox"/> |
| HINHANHDAIDIEN | varchar(200) | <input type="checkbox"/> |
| TRANGTHAI | bit | <input type="checkbox"/> |
| MALOAI | varchar(10) | <input type="checkbox"/> |
| MATH | varchar(10) | <input type="checkbox"/> |
| SOSAOTB | float | <input type="checkbox"/> |
| NGAYTHEM | date | <input type="checkbox"/> |
| | | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

THUONGHIEU

| Column Name | Data Type | Allow Nulls |
|---------------|--------------|--------------------------|
| MATH | varchar(10) | <input type="checkbox"/> |
| TENTHUONGHIEU | nvarchar(50) | <input type="checkbox"/> |
| TRANGTHAI | bit | <input type="checkbox"/> |

LOAISANPHAM

| Column Name | Data Type | Allow Nulls |
|-------------|--------------|--------------------------|
| MALOAI | varchar(10) | <input type="checkbox"/> |
| TENLOAI | nvarchar(50) | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

HINHANH

| Column Name | Data Type | Allow Nulls |
|-------------|--------------|--------------------------|
| MAHINHANH | int | <input type="checkbox"/> |
| LINK | varchar(100) | <input type="checkbox"/> |
| MASP | varchar(20) | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

GIOHANG

| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| MAGH | int | <input type="checkbox"/> |
| MAND | int | <input type="checkbox"/> |
| MASP | varchar(20) | <input type="checkbox"/> |
| SOLUONG | int | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

DONHANG

| Column Name | Data Type | Allow Nulls |
|-------------|---------------|--------------------------|
| MADH | int | <input type="checkbox"/> |
| NGAYTAO | date | <input type="checkbox"/> |
| HOTEN | nvarchar(40) | <input type="checkbox"/> |
| SDT | varchar(15) | <input type="checkbox"/> |
| DIACHI | nvarchar(300) | <input type="checkbox"/> |
| TONGTIEN | money | <input type="checkbox"/> |
| TRANGTHAI | int | <input type="checkbox"/> |
| MAND | int | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

CTDONHANG

| Column Name | Data Type | Allow Nulls |
|------------------|-------------|--------------------------|
| MACTDH | int | <input type="checkbox"/> |
| MADH | int | <input type="checkbox"/> |
| MASP | varchar(20) | <input type="checkbox"/> |
| DONGIA | money | <input type="checkbox"/> |
| SOLUONG | int | <input type="checkbox"/> |
| TRANGTHAIDANHgia | bit | <input type="checkbox"/> |

DANHGIA

| Column Name | Data Type | Allow Nulls |
|-------------|----------------|-------------------------------------|
| MADG | int | <input type="checkbox"/> |
| SOSAO | float | <input type="checkbox"/> |
| NOIDUNG | nvarchar(1000) | <input checked="" type="checkbox"/> |
| NGAY | date | <input type="checkbox"/> |
| MAND | int | <input type="checkbox"/> |
| MASP | varchar(20) | <input type="checkbox"/> |

KHUYENMAI

| Column Name | Data Type | Allow Nulls |
|-------------|-------------|-------------------------------------|
| MAKM | varchar(10) | <input type="checkbox"/> |
| GIATRI | int | <input type="checkbox"/> |
| NGAYBATDAU | date | <input checked="" type="checkbox"/> |
| NGAYKETTHUC | date | <input checked="" type="checkbox"/> |
| SOLUONG | int | <input checked="" type="checkbox"/> |
| TRANGTHAI | bit | <input type="checkbox"/> |

CTKHUYENMAI

| Column Name | Data Type | Allow Nulls |
|-------------|-------------|--------------------------|
| MACTKM | int | <input type="checkbox"/> |
| MASP | varchar(20) | <input type="checkbox"/> |
| MAKM | varchar(10) | <input type="checkbox"/> |

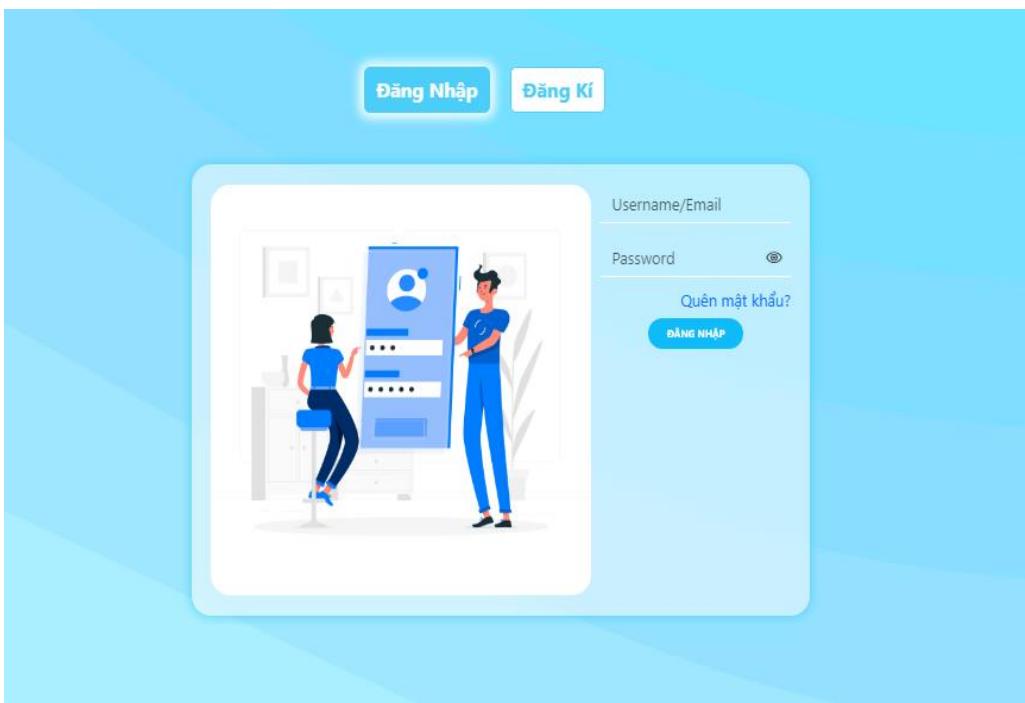
II. GIAO DIỆN

1. Giao diện của khách hàng

1.1. Trang chủ

The screenshot shows the homepage of 'The Gioi Dien May' (The World of Electronics) website. At the top, there's a banner for a promotion: 'Mua Sắm Máy Lạnh TẶNG TẶNG CẤP 2 + 2 NĂM BẢO HÀNH + 2 LẦN VỆ SINH'. Below the banner, there's a large advertisement for a washing machine with a woman in traditional Vietnamese clothing, offering 'Mua máy giặt TRUNG MỐI NGÀY 30 TRIỆU' (Buy a washing machine every day 30 million). It highlights three offers: 'TRẢ GÓP 0%' (0% installment), 'QUÀ ĐẾN 1.2 TRIỆU' (Gift up to 1.2 million), and '1 THÁNG 1 ĐỔI 1' (1 month 1 exchange). The main navigation menu includes categories like Tủ Lạnh, Máy Lạnh, Máy Giặt, Tủ Đông, Quạt, and Nồi Cơm Điện. Below the main content, there's a section titled 'GỢI Ý CHO BẠN' (Suggestions for you) featuring various products with their prices: Quạt đứng Senko DH1600 (550.000đ), Tủ đông SANAKY 208 Lit TD.VH255HY2 (4.790.000đ), Quạt treo Senko TC1622 (535.000đ), Máy lạnh LG Inverter 1 HP V10WIN (11.000.000đ), Tủ lạnh Toshiba Inverter 511 lit GR-RF610WWE (19.990.000đ), and Máy lạnh Midea Inverter 1 HP MSAFA-10CRDNB (6.500.000đ). A section titled 'SẢN PHẨM MỚI NHẤT' (Newest products) displays more items: Tủ lạnh Panasonic Inverter 170 lit NR-BA190PPVN (5.000.000đ), Máy lạnh Samsung Inverter 1 HP ARD9TYHQASINSV (9.390.000đ), Máy giặt Samsung Inverter 9.5kg WW95T4040CE/SV (11.090.000đ), Tủ đông SANAKY 208 Lit TD.VH255HY2 (4.790.000đ), Máy lạnh Midea Inverter 1 HP MSAFA-10CRDNB (6.500.000đ), and Máy giặt Samsung Inverter 9 kg WW90T634DLN (11.500.000đ). The footer contains contact information: Địa chỉ: Đường Mạc Thiên, P. Hiệp Phước, Q.9, TP.Hồ Chí Minh; Hotline: 0937696969; Email: thegioedienmay@gmail.com. It also features a map showing the store's location in District 9, Ho Chi Minh City.

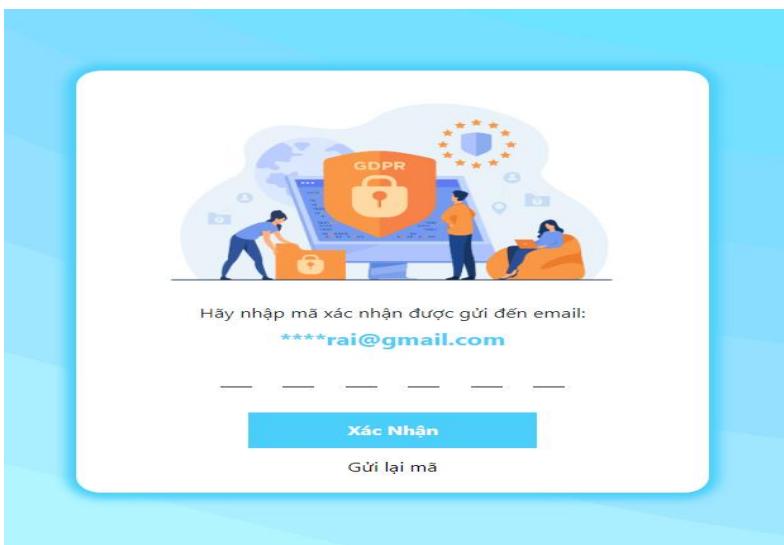
1.2. Trang đăng nhập



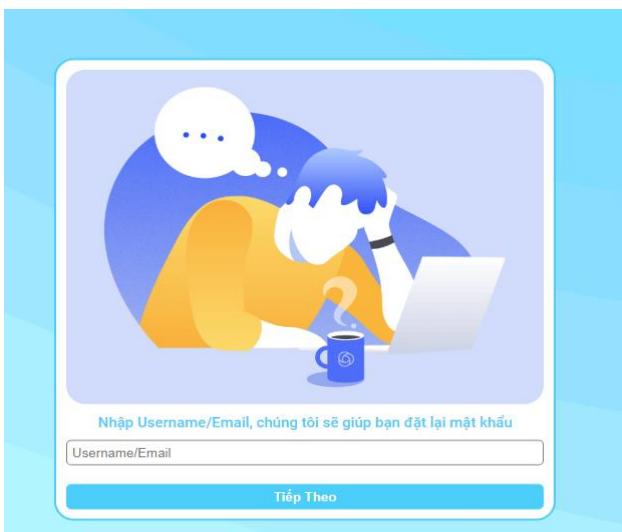
1.3. Trang đăng kí

A screenshot of a registration page with a light blue background. At the top center are two buttons: "Đăng Nhập" (Login) in a blue box and "Đăng Kí" (Register) in a white box. Below these buttons is a large central form box. The form contains the following fields: "Email" with the placeholder "xxx@gmail.com", "Username", "Password (tối thiểu 8 ký tự)" (Minimum 8 characters), "Nhập lại mật khẩu" (Re-enter password), "Họ Tên" (Last name), "Giới Tính:" (Gender) with radio buttons for "Nam" (Male) and "Nữ" (Female) where "Nữ" is selected, "Ngày Sinh" (Birth date) with a placeholder "dd/mm/yyyy" and a calendar icon, "SĐT" (Phone number), and "Địa Chỉ" (Address). At the bottom of the form is a blue "ĐĂNG KÍ" button.

1.4. Trang nhập mã xác minh



1.5. Trang quên mật khẩu



1.6. Trang lọc tìm theo loại sản phẩm

1.7. Trang thông tin sản phẩm

Mua Sắm Máy Lạnh TẶNG TẶNG GẤP 2 + 2 NĂM BẢO HÀNH + 2 LẦN VỆ SINH

THE GIỚI ĐIỆN MÁY

Tìm kiếm...

Khuyến mãi Giảm Giá Lịch sử đơn hàng Lỗi PTT Đăng Xuất

Tủ Lạnh Máy lạnh Máy Giặt Tủ Đông Quạt Nồi Cơm Điện

Thương hiệu: Panasonic
Loại: TỦ LẠNH
Tên sản phẩm:
Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN
Đánh giá: 4.7/5 ★
Còn lại: 45
5.000.000đ

Mô tả

Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN sử dụng công nghệ biến tần Inverter có khả năng điều chỉnh vòng quay của máy nén giúp duy trì làm lạnh ổn định, hoạt động êm ái, ít gây tiếng ồn hơn và tiết kiệm được điện năng hiệu quả.

Ngoài ra, tủ lạnh Panasonic còn tích hợp cảm biến Econavi với: cảm biến cửa, cảm biến nhiệt độ bên

Nhờ đó, tủ lạnh có thể nhận biết được tần suất các thao tác đóng mở và sự thay đổi trong thói quen sinh hoạt, sau đó tự động điều chỉnh nhiệt độ làm mát phù hợp, giúp tiết kiệm năng lượng tiêu thụ.

Các thông số của sản phẩm

| Tổng quan | |
|-----------------------------|---|
| Kiểu tủ: | Ngăn đá trên - 2 cánh |
| Dung tích tổng: | 190 lít |
| Dung tích sử dụng: | 170 lít - 2 - 3 người |
| Dung tích ngăn đá: | 53 lít |
| Dung tích ngăn lạnh: | 117 lít |
| Dung tích ngăn chuyển đổi: | Không có |
| Chất liệu cửa tủ lạnh: | Kính loai phủ sơn tĩnh điện |
| Chất liệu khay ngăn lạnh: | Kính chịu lực |
| Chất liệu ống dẫn gas, dân: | Ống dẫn gas bằng Đồng và Sắt - Là tản nhiệt bằng Nhôm |
| Năm ra mắt: | 2020 |
| Sản xuất tại: | Việt Nam |
| Mức tiêu thụ điện năng | |
| Công suất tiêu thụ công bố | ~ 0.92 kW/ngày |
| theo TCVN: | |
| Công nghệ tiết kiệm điện: | Cảm biến thông minh Econavi, Inverter |

ĐÁNH GIÁ SẢN PHẨM

Có 3 đánh giá
Đánh giá trung bình: 4.7/5 ★

| |
|--|
| Nguyễn Tin ★★★★★ 2023-05-11 Phù hợp giá tiền, nói chung đáng mua |
| Văn Lực ★★★★★ 2023-04-30 Sản phẩm như hình, chất lượng tốt |
| Xuân Huynh ★★★★★ 2023-04-01 Sản phẩm chất lượng |

Xem các tủ lạnh khác

1.8. Trang tìm kiếm sản phẩm

| Tên Sản Phẩm | Đơn giá | Số Lượng | Thao Tác |
|--|-------------|----------|-------------------------|
| Tủ lạnh Panasonic Inverter 170 lít NR-BA190FPVN | 5.000.000đ | 1 | Buy Now |
| Máy giặt Panasonic Inverter 9.5 Kg NA-V95FC1LVT | 12.690.000đ | 2 | Buy Now |
| Máy lạnh Panasonic Inverter 1 HP CU/CS-PU9KKH-8M | 11.900.000đ | | |

1.9. Trang giỏ hàng

| Tên Sản Phẩm | Đơn giá | Số Lượng | Thao Tác |
|--|-------------|----------|-------------------------|
| Nồi cơm điện Kangaroo 1.2 lít KG822 đỏ | 1.020.000đ | 1 | Buy Now |
| Tủ lạnh LG Inverter 394 lít GN-D392BLA | 13.500.000đ | 2 | Buy Now |

Tổng cộng: 28.020.000đ

[Buy Now](#)

1.10. Trang đặt hàng

| Tên Sản Phẩm | Đơn giá | Số Lượng | Thành Tiền |
|--|-------------|----------|-------------|
| Nồi cơm điện Kangaroo 1.2 lít KG822 đỏ | 1.020.000đ | 1 | 1.020.000đ |
| Tủ lạnh LG Inverter 394 lít GN-D392BLA | 13.500.000đ | 2 | 27.000.000đ |

Tổng cộng: 28.020.000đ

[Place Order](#)

1.11. Trang lịch sử đơn hàng

The screenshot shows a successful order page titled "ĐƠN HÀNG THÀNH CÔNG". It displays three recent orders with details like date, total amount, and links to view or print the order.

| # | Ngày đặt | Tổng Tiền | Chi tiết đơn | Danh Giá |
|---|------------|-------------|------------------------------|--------------------------|
| 1 | 2023-05-26 | 26.269.000đ | Chi tiết đơn | Danh Giá |
| 2 | 2023-05-25 | 1.020.000đ | Chi tiết đơn | Danh Giá |
| 3 | 2023-05-22 | 6.510.000đ | Chi tiết đơn | Danh Giá |

1.12. Trang thông tin khách hàng

The screenshot shows a customer profile page titled "Hồ Sơ Của Tôi". It lists personal information such as Username, Name, Email, Phone number, Gender, Birthdate, and Address, each with an edit field and a "Lưu" (Save) button.

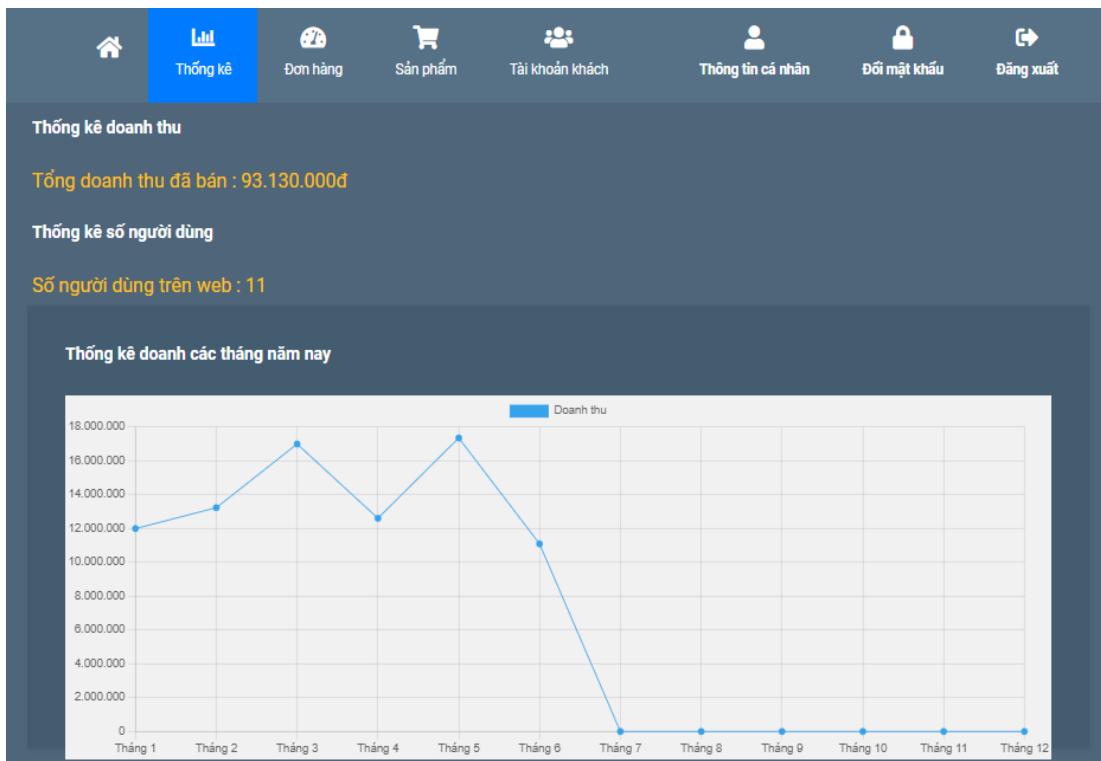
| User Name | ptit123 |
|---------------|---|
| Họ Tên | Lê Pi Tít |
| Email | vanluccmc@gmail.com |
| Số điện thoại | 0359490251 |
| Giới Tính | <input checked="" type="radio"/> Nam <input type="radio"/> Nữ |
| Ngày Sinh | 25/04/2002 |
| Địa chỉ | Bình Định |

1.13. Trang đổi mật khẩu

The screenshot shows a password change page titled "Đổi Mật Khẩu". It has fields for Old Password, New Password, and Confirm New Password, along with a "Lưu" (Save) button.

2. Giao diện của tài khoản người quản lý

2.1. Giao diện trang thống kê



2.2. Giao diện trang quản lý đơn hàng

The screenshot shows a dashboard with a navigation bar at the top containing links for Home, Statistics, Orders (highlighted in blue), Products, Customer Accounts, Personal Information, Change Password, and Log Out.

1. Đơn hàng chờ xác nhận (Pending Confirmation):

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết | Xác nhận duyệt | Hủy đơn này |
|--------|----------------|------------|-------------|--------------|------------------------|----------------------|
| #1047 | Nguyễn Văn Sơn | 2023-06-04 | 5.000.000đ | | <button>Duyệt</button> | <button>Hủy</button> |
| #1049 | Đặng Văn Nam | 2023-06-04 | 10.990.000đ | | <button>Duyệt</button> | <button>Hủy</button> |

2. Đơn hàng đang được giao (Being Delivered):

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|-----------------|------------|-------------|--------------|
| #1044 | Nguyễn Văn Hùng | 2023-06-04 | 11.900.000đ | |
| #1048 | Nguyễn Văn Sơn | 2023-04-04 | 6.535.000đ | |
| #1043 | Nguyễn Văn Hùng | 2023-02-04 | 1.070.000đ | |

3. Đơn hàng đã giao thành công (Delivered Successfully):

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|-----------------|------------|-------------|--------------|
| #1045 | Nguyễn Văn Hùng | 2023-06-04 | 11.090.000đ | |
| #26 | Nguyễn Văn Hùng | 2023-05-24 | 5.000.000đ | |
| #1046 | Nguyễn Văn Sơn | 2023-05-04 | 12.345.000đ | |
| #1050 | Đặng Văn Nam | 2023-04-04 | 550.000đ | |
| #1051 | Trần Văn B | 2023-04-04 | 5.550.000đ | |
| #1052 | Trần Văn R | 2023-04-04 | 6.500.000đ | |

4. Đơn hàng đã bị hủy (Canceled):

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|--------------|------------|------------|--------------|
| #1053 | Đặng Văn Nam | 2023-06-04 | 5.466.000đ | |

© 2023 PTITHOM

2.3. Giao diện trang quản lý sản phẩm

The screenshot shows the product management section of the website. At the top, there is a navigation bar with icons for Home, Statistics, Orders, Products (highlighted in blue), Customers, Personal Information, Change Password, and Logout.

Các sản phẩm đang kinh doanh

| Ảnh | Tên sản phẩm | Giá | Còn | Trạng thái | Sửa | Xóa |
|-----|---|-------------|-----|------------|-----|-----|
| | Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN | 5.000.000đ | 21 | Đang bán | | |
| | Máy lạnh Samsung Inverter 1 HP AR09TYQASINSV | 9.390.000đ | 12 | Đang bán | | |
| | Máy giặt Samsung Inverter 9.5kg WW95T4040CE/SV | 11.090.000đ | 29 | Đang bán | | |
| | Tủ đông SANAKY 208 Lít TD.VH255HY2 | 4.790.000đ | 29 | Đang bán | | |
| | Máy lạnh Midea Inverter 1 HP MSAFA-10CRDN8 | 6.500.000đ | 11 | Đang bán | | |

THÊM SẢN PHẨM

Các thương hiệu đang bán

| Tên thương hiệu | Sửa | Xóa |
|-----------------|-----|-----|
| Casper | | |
| DAIKIN | | |
| KANGAROO | | |
| LG | | |
| MIDEA | | |
| Panasonic | | |

THÊM THƯƠNG HIỆU

The screenshot shows the discontinued products section of the website. At the top, there is a navigation bar with icons for Home, Statistics, Orders, Products (highlighted in blue), Customers, Personal Information, Change Password, and Logout.

Các sản phẩm đã tạm ngừng kinh doanh

| Ảnh | Tên sản phẩm | Giá | Còn | Trạng thái | Sửa | Xóa |
|-----|--|-------------|-----|------------|-----|-----|
| | Tủ đông Sanaky Inverter 530 lít VH-6699HY3 | 16.240.000đ | 28 | Ngừng bán | | |

© 2023 PTITHCM

2.4. Giao diện trang thêm sản phẩm

Thêm sản phẩm mới

Mã sản phẩm

Tên sản phẩm

Mô tả

Hình ảnh đại diện(không được để trống)

Chọn ảnh

Hình ảnh khác(tối đa 4 hình)

Chọn tệp | Không có tệp nào được chọn

Ảnh thông số kỹ thuật

Chọn tệp | Không có tệp nào được chọn

Loại sản phẩm

MÁY GIẶT

Thương hiệu

Casper

Giá

0

Số lượng

1

THÊM SẢN PHẨM

2.5. Giao diện trang sửa sản phẩm

Sửa sản phẩm

Mã sản phẩm
BA190PPVN

Tên sản phẩm
Tủ lạnh Panasonic Inverter 170 lít NR-BA190F

Mô tả

Tùy chỉnh mô tả (WYSIWYG editor):

Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN sử dụng công nghệ biến tần Inverter có khả năng điều chỉnh vòng quay của máy nén giúp duy trì làm lạnh ổn định, hoạt động êm ái, ít gây tiếng ồn hơn và tiết kiệm được điện năng hiệu quả. Ngoài ra, tủ lạnh Panasonic còn tích hợp cảm biến Econavi với: cảm biến cửa, cảm biến nhiệt độ bên trong, cảm biến nhiệt độ phòng, cảm biến

Đổi hình ảnh đại diện

Chọn tệp | Chưa có tệp nào được chọn



Đổi hình ảnh khác(tối đa 4 hình)

Chọn tệp | Chưa có tệp nào được chọn



Đổi ảnh thông số kỹ thuật

Chọn tệp | Chưa có tệp nào được chọn

Loại sản phẩm: TỦ LẠNH

TÚ LẠNH

Thương hiệu: Panasonic

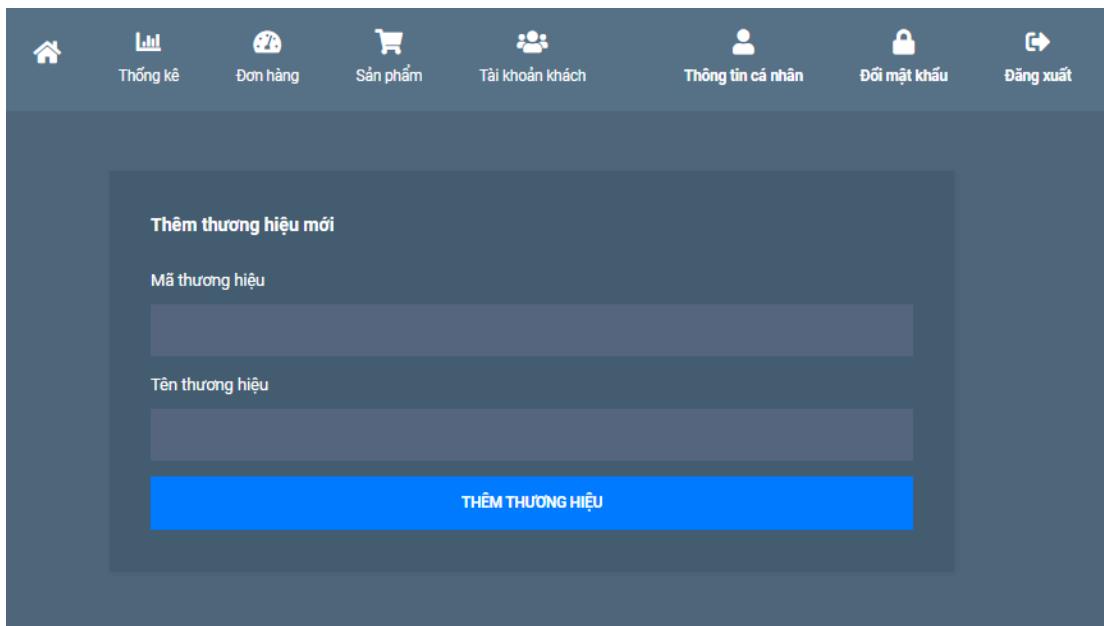
Panasonic

Giá Số lượng

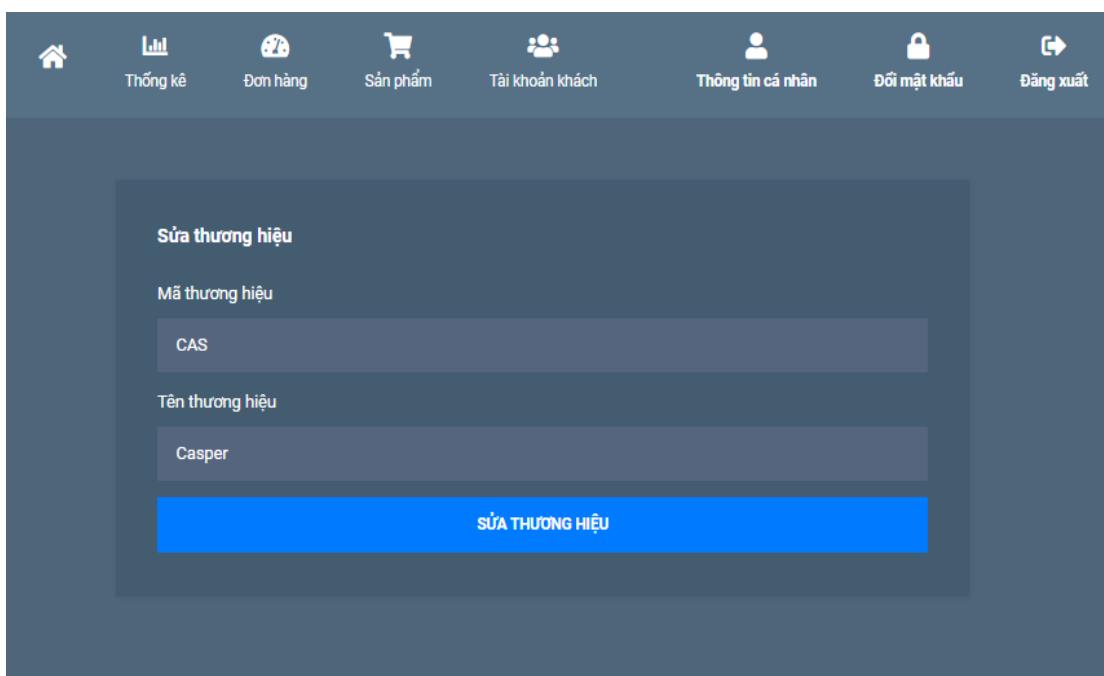
5000000 21

CẬP NHẬT SẢN PHẨM

2.6. Giao diện trang thêm thương hiệu



2.7. Giao diện trang sửa thương hiệu



2.8. Giao diện trang quản lý tài khoản khách hàng

| ID | HỌ TÊN | SĐT | THAO TÁC | Chi tiết |
|----|-------------------|------------|----------|--------------------------|
| 6 | Lê Tèo | 0931234567 | | Chi tiết |
| 7 | Phan Xuân Huỳnh | 0934561234 | | Chi tiết |
| 8 | Nguyễn Tín | 0937123456 | | Chi tiết |
| 9 | Văn Lực | 0937123456 | | Chi tiết |
| 10 | Trần Văn Hùng | 0123456789 | | Chi tiết |
| 19 | Trần Văn Mười Hai | 0937567567 | | Chi tiết |

| ID | HỌ TÊN | SĐT | THAO TÁC | Chi tiết |
|----|-------------------|------------|----------|--------------------------|
| 18 | Nguyễn Văn Tèo Em | 0937121212 | | Chi tiết |

2.9. Giao diện trang thông tin cá nhân

THÔNG TIN

| | |
|--------------------------------------|---------------------|
| Họ Tên | Ngày Sinh |
| Nguyễn Tín | 27/05/2002 |
| Giới Tính: | SĐT |
| <input checked="" type="radio"/> Nam | 0937681469 |
| <input type="radio"/> Nữ | |
| Địa Chỉ | Bình Thạnh |
| Email | tinnguyen@gmail.com |

CẬP NHẬT THÔNG TIN

2.10. Giao diện trang đổi mật khẩu

3. Giao diện trang của tài khoản chủ (Owner)

Các trang của tài khoản owner đều giống với tài khoản Admin chỉ có thêm phần quản lý tài khoản Admin

| ID | USERNAME | HỌ TÊN | TRẠNG THÁI | THAO TÁC |
|----|-------------|-----------------|-----------------|---|
| 22 | xuanhuynh | Phan Xuân Huỳnh | Hoạt động | <input checked="" type="radio"/> Chi tiết |
| 24 | vanluc | Phan Văn Lực | Hoạt động | <input checked="" type="radio"/> Chi tiết |
| 25 | HungNguyenn | Nguyễn Văn Tèo | Hoạt động | <input checked="" type="radio"/> Chi tiết |
| 27 | nguyenson | Nguyễn Văn Sơn | Ngừng hoạt động | <input type="radio"/> Chi tiết |
| 28 | vantran | Trần Văn B | Ngừng hoạt động | <input type="radio"/> Chi tiết |

-Trang thêm tài khoản Admin

The screenshot shows a dark-themed administrative interface with a top navigation bar containing icons for Home, Statistics, Orders, Products, Customer Accounts, Admin Accounts, Personal Information, Password Change, and Logout.

The main content area is titled "Thêm Admin mới" (Add New Admin). It contains the following fields:

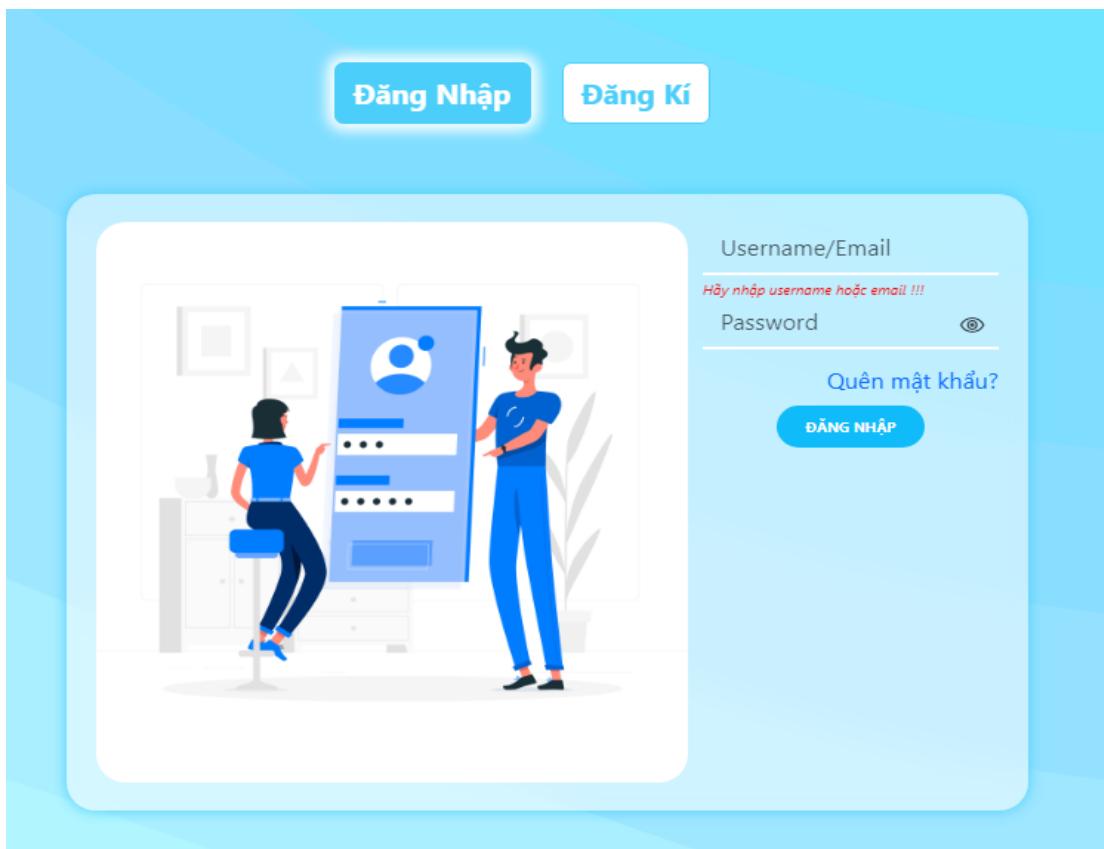
- Họ Tên (Last Name): [Input field]
- Ngày Sinh (Date of Birth): [Input field] (dd/mm/yyyy)
- Giới Tính (Gender):
 - Nam
 - Nữ
- SDT (Phone Number): [Input field]
- Địa Chỉ (Address): [Input field]
- Email: [Input field] (xxx@gmail.com)
- Username: [Input field]
- Password (tối thiểu 8 kí tự) (Minimum 8 characters): [Input field]
- Nhập lại mật khẩu (Re-enter password): [Input field]

A blue button at the bottom center is labeled "THÊM ADMIN" (Add Admin).

III. TÍNH NĂNG

1. Đăng kí-Đăng nhập-Lấy mật khẩu

-Đăng kí, đăng nhập: Sử dụng validation để kiểm tra



```

@RequestMapping(value = "form/login", params = "login", method = RequestMethod.POST)
public String login(ModelMap model, HttpServletRequest request, @ModelAttribute("user") NguoiDungEntity user,
                     BindingResult errors) {

    Boolean loi = Boolean.TRUE;

    if (user.getUserName().isEmpty()) {
        errors.rejectValue("userName", "user", "Hãy nhập username hoặc email !!!");
        return "user/login";
    } else if (user.getPassword().isEmpty()) {
        errors.rejectValue("passWord", "user", "Hãy nhập mật khẩu !!!");
        return "user/login";
    }

    NguoiDungEntity check = userService.findUserByNameAndEmail(user.getUserName(), user.getUserName());

    if (check == null) {

        errors.rejectValue("userName", "user", "Tài khoản không tồn tại !!!");
        loi = Boolean.FALSE;

    } else if ((user.getUserName().equals(check.getUserName()) || user.getUserName().equals(check.getEmail()))
               && !userService.kiemTraMatKhau(user.getPassword(), check.getPassword())) {
        errors.rejectValue("passWord", "user", "Sai mật khẩu !!!");
        loi = Boolean.FALSE;
    }
    else if (check.isTrangThai()==false) {
        errors.rejectValue("userName", "user", "Tài khoản của bạn đã bị khoá !!!");
        loi = Boolean.FALSE;
    }
    if (loi == Boolean.FALSE)
        return "user/login";
    Hibernate.initialize(check.getGioHangs());
    HttpSession session0 = request.getSession();
    session0.setAttribute("USER", check);
    if (check.getQuyen()!=0) { // Kiểm tra quyền người dùng là admin
        return "redirect:/admin/index.htm"; // Trả về trang quản trị admin
    }
    String maSp= (String) session0.getAttribute("SANPHAM");
    if(maSp!=null)
        {session0.removeAttribute("SANPHAM");
        return "redirect:/sanpham/"+maSp+".htm";}
    session0.removeAttribute("SANPHAM");
    return "redirect:/";
}

```

Email

Hãy nhập email !!!

Username

Hãy nhập username !!!

Password (tối thiểu 8 ký tự)

Hãy nhập mật khẩu !!!

Nhập lại mật khẩu

Họ Tên

Hãy nhập họ tên !!!

Giới Tính: Nam Nữ

Ngày Sinh

 □

Hãy nhập ngày sinh !!!

SĐT

Hãy nhập sdt !!!

Địa Chỉ

Hãy nhập địa chỉ !!!

ĐĂNG KÍ

```

@RequestMapping(value = "form/signup", params = "signup", method = RequestMethod.POST)
public String signUp(HttpServletRequest request, ModelMap model, @ModelAttribute("user") NguoiDungEntity user,
                     BindingResult errors) throws ParseException {

    Boolean loi = Boolean.TRUE;
    String rePassword = request.getParameter("re-passWord");
    NguoiDungEntity userCheck;

    if (user.getEmail().isEmpty()) {
        errors.rejectValue("email", "user", "Hãy nhập email !!!");
        loi = Boolean.FALSE;
    } else if (!user.getEmail().endsWith("@gmail.com")) {
        errors.rejectValue("email", "user", "Hãy nhập email đúng định dạng !!!");
        loi = Boolean.FALSE;
    }
    if (user.getUserName().isEmpty()) {
        errors.rejectValue("userName", "user", "Hãy nhập username !!!");
        loi = Boolean.FALSE;
    } else if (user.getUserName().contains(" ")) {
        errors.rejectValue("userName", "user", "UserName không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (user.getUserName().length() > 30) {
        errors.rejectValue("userName", "user", "UserName không được dài quá 30 kí tự !!!");
        loi = Boolean.FALSE;
    }

    userCheck = userService.findUserByNameAndEmail(user.getUserName(), user.getEmail());
    if (userCheck != null) {
        if (userCheck.getEmail().equals(user.getEmail())) {
            errors.rejectValue("email", "user", "Email đã được sử dụng !!!");
            loi = Boolean.FALSE;
        }
        if (userCheck.getUserName().equals(user.getUserName())) {
            errors.rejectValue("userName", "user", "Username đã được sử dụng !!!");
            loi = Boolean.FALSE;
        }
    }

    if (user.getPassWord().isEmpty()) {
        errors.rejectValue("passWord", "user", "Hãy nhập mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (user.getPassWord().length() < 8) {
        errors.rejectValue("passWord", "user", "Password tối thiểu 8 kí tự !!!");
        loi = Boolean.FALSE;
    } else if (user.getPassWord().contains(" ")) {
        errors.rejectValue("passWord", "user", "Password không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (rePassword.isEmpty()) {
        errors.rejectValue("passWord", "user", "Hãy nhập lại mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (!rePassword.equals(user.getPassWord())) {
        errors.rejectValue("passWord", "user", "Xác nhận mật khẩu không đúng !!!");
        loi = Boolean.FALSE;
    }
}

```

```

if (user.getHoTen().isEmpty()) {
    errors.rejectValue("hoTen", "user", "Hãy nhập họ tên !!!");
    loi = Boolean.FALSE;
} else if (user.getHoTen().length() > 50) {
    errors.rejectValue("hoTen", "user", "Họ tên quá dài !!!");
    loi = Boolean.FALSE;
} else if (!user.getHoTen().matches("[\\p{L} ]+")) {
    errors.rejectValue("hoTen", "user", "Họ tên không được chứa số !!!");
    loi = Boolean.FALSE;
}

if (user.getNgaySinh() == null) {
    errors.rejectValue("ngaySinh", "user", "Hãy nhập ngày sinh !!!");
    loi = Boolean.FALSE;
} else if (!isValidAndOver18(user.getNgaySinh())) {
    errors.rejectValue("ngaySinh", "user", "Bạn cần lớn hơn 18 tuổi để tạo tài khoản !!!");
    loi = Boolean.FALSE;
}
if (user.getSdt().isEmpty()) {
    errors.rejectValue("sdt", "user", "Hãy nhập sdt !!!");
    loi = Boolean.FALSE;
} else if (!user.getSdt().matches("[0-9]+")) {
    errors.rejectValue("sdt", "user", "SDT không hợp lệ !!!");
    loi = Boolean.FALSE;
} else if (user.getSdt().length() != 10 && user.getSdt().length() != 11) {
    errors.rejectValue("sdt", "user", "SDT không hợp lệ !!!");
    loi = Boolean.FALSE;
}
if (user.getDiaChi().isEmpty()) {
    errors.rejectValue("diaChi", "user", "Hãy nhập địa chỉ !!!");
    loi = Boolean.FALSE;
}

if (loi == Boolean.FALSE)
    return "/user/signup";

String NGAYSINH = request.getParameter("ngaySinh");
java.sql.Date ns = java.sql.Date.valueOf(NGAYSINH);
user.setPassWord(userService.maHoaMatKhau(user.getPassWord()));
user.setNgaySinh(ns);
user.setHoTen(capitalizeString(user.getHoTen()));
user.setTrangThai(true);
user.setQuyen(0);
HttpSession session = request.getSession();
session.setAttribute("USERSIGNUP", user);
String otp = taoOTP();
session.setAttribute("OTP", otp);
mailer.sendMailAsync("THEGIOIDIENMAY", user.getEmail(), "OTP", "Mã OTP của bạn là: " + otp);
return "/user/verify";
}

```

Mật khẩu của các tài khoản sẽ được bảo vệ bằng cách mã hoá và lưu vào cơ sở dữ liệu bằng thuật toán Bcrypt.

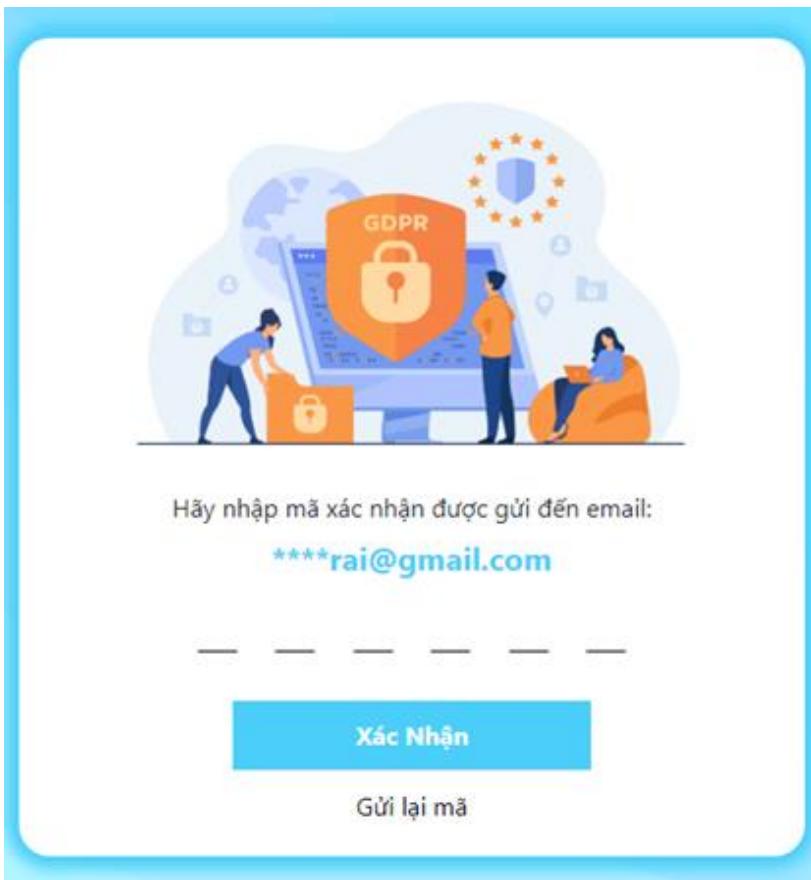
```

@Override
public String maHoaMatKhau(String str) {
    String hashedPassword = BCrypt.hashpw(str, BCrypt.gensalt());
    return hashedPassword;
}

@Override
public boolean kiemTraMatKhau(String password, String hashedPassword) {
    return BCrypt.checkpw(password, hashedPassword);
}

```

Sau khi điền đủ các thông tin hợp lệ sẽ chuyển đến trang nhập mã xác minh Email



Hãy nhập mã xác nhận được gửi đến email:

****rai@gmail.com

Xác Nhận

Gửi lại mã

```
@RequestMapping(value = "form/verify", params = "verify", method = RequestMethod.POST)
public String verify(HttpServletRequest request, ModelMap model) {
    HttpSession session = request.getSession();
    String a = request.getParameter("a");
    String b = request.getParameter("b");
    String c = request.getParameter("c");
    String d = request.getParameter("d");
    String e = request.getParameter("e");
    String f = request.getParameter("f");

    String otp = session.getAttribute("OTP").toString();
    String temp = a + b + c + d + e + f;

    if (otp.equals(temp)) {
        NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USERSIGNUP");
        userService.addUser(user);
        model.addAttribute("user", new NguoiDungEntity()); //nếu nhập đúng OTP thì sẽ chuyển đến trang login
        return "/user/login";
    }
    NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USERSIGNUP");
    model.addAttribute("email", "****"+user.getEmail().substring(user.getEmail().length()-13));

    model.addAttribute("messenger", "OTP bạn nhập không đúng !!!");
    return "/user/verify";
}
```

Sau khi đăng nhập sẽ lưu account vào HttpSession. Sử dụng Interceptor để phân quyền đăng nhập:

```
<!-- Spring MVC Annotation -->
<context:annotation-config />
<mvc:annotation-driven />
<mvc:interceptors>
<mvc:interceptor>
    <mvc:mapping path="/admin/adminAccount.htm" />
    <bean class="ptithcm.interceptor.ownerInterceptor" />
</mvc:interceptor>
<mvc:interceptor>
    <mvc:mapping path="/admin/**" />
    <bean class="ptithcm.interceptor.adminInterceptor" />
</mvc:interceptor>
</mvc:interceptors>



---


package ptithcm.interceptor;

import javax.servlet.http.HttpServletRequest;

public class adminInterceptor extends HandlerInterceptorAdapter
{
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception {
        NguoiDungEntity user=(NguoiDungEntity) request.getSession().getAttribute("USER");

        if(user==null|| user.getQuyen()==0)
        {
            response.sendRedirect(request.getContextPath() + "/khongCoQuyen.htm");
        }
        return true;
    }



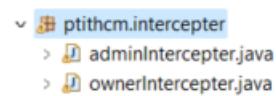
---


package ptithcm.interceptor;

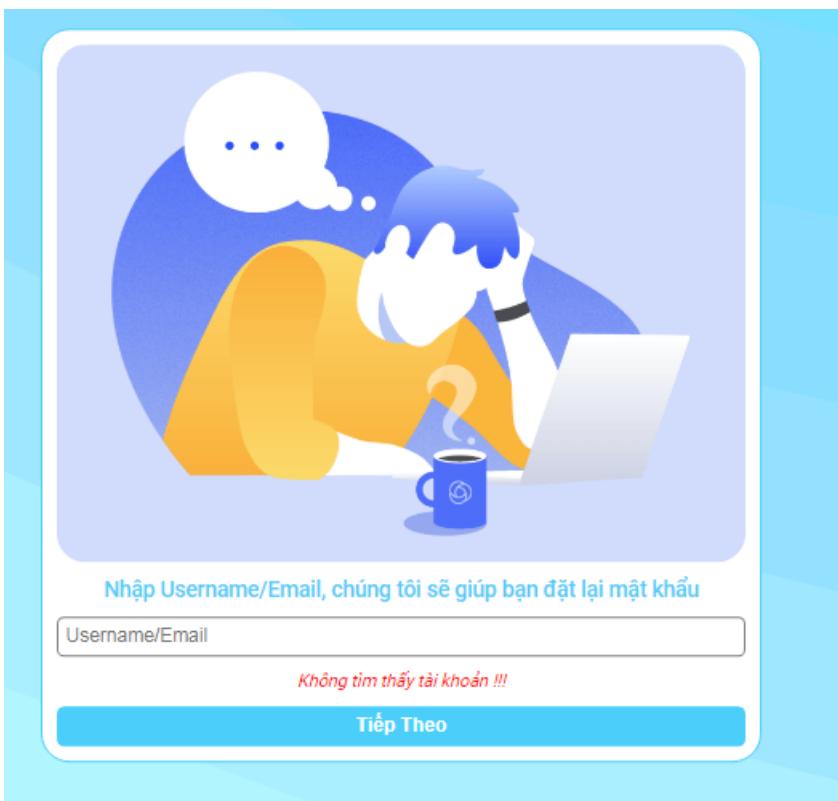
import javax.servlet.http.HttpServletRequest;

public class ownerInterceptor extends HandlerInterceptorAdapter
{
    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
        throws Exception {
        NguoiDungEntity user=(NguoiDungEntity) request.getSession().getAttribute("USER");

        if(user==null|| user.getQuyen()!=2)
        {
            response.sendRedirect(request.getContextPath() + "/khongCoQuyen.htm");
        }
        return true;
    }
}
```



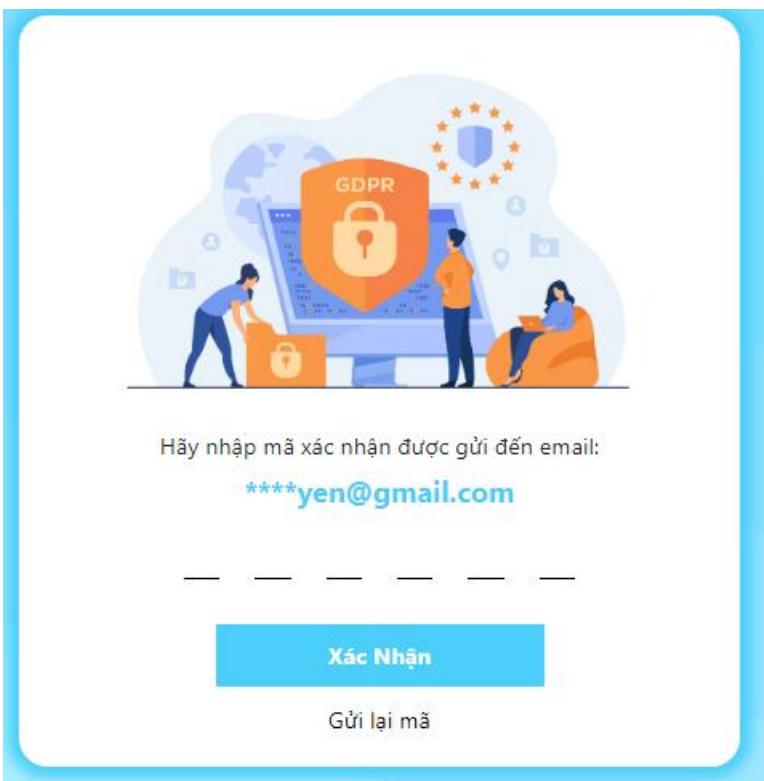
-Lấy lại mật khẩu



```
@RequestMapping(value = "form/forgotpass", params = "tieptheo", method = RequestMethod.POST)
public String forgotPass(HttpServletRequest request, ModelMap model) {
    String userName = request.getParameter("username");
    System.out.print(userName);

    if (userName.equals(null)) {
        model.addAttribute("messenger", "Hãy nhập Username/Email của bạn !!!");
        return "/user/forgotpass";
    }
    NguoiDungEntity user = userService.findUserByNameAndEmail(userName, userName);
    if (user == null) {
        model.addAttribute("messenger", "Không tìm thấy tài khoản !!!");
        return "/user/forgotpass";
    }
    HttpSession session = request.getSession();
    session.setAttribute("USERFORGOT", user);
    String otp = taoOTP();
    session.setAttribute("OTP", otp);
    mailer.sendMailAsync("THEGIOIDIENMAY", user.getEmail(), "OTP", "Mã OTP của bạn là: " + otp);
    model.addAttribute("email", "*****"+user.getEmail().substring(user.getEmail().length()-13));
    return "/user/verify2";
}
```

Khi nhập đúng Username/Email sẽ chuyển đến trang xác thực:



Khi nhập đúng OTP sẽ chuyển đến trang tạo mật khẩu mới:

| | |
|---|--|
| Đặt mật khẩu mới của bạn | |
| <input type="text" value="Mật khẩu mới"/> | <input type="text" value="Nhập lại mật khẩu"/> |
| Mật khẩu phải hơn 8 kí tự !!! | |
| Hoàn Tất | |

```

@RequestMapping(value = "form/newpass", params = "hoanTat", method = RequestMethod.POST)
public String newPass(HttpServletRequest request, ModelMap model) {
    HttpSession session = request.getSession();

    NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USERFORGOT");
    String pass = request.getParameter("password");
    String confirmPass = request.getParameter("confirmPass");

    if (pass.isEmpty()) {
        model.addAttribute("messenger", "Mật khẩu không được trống !!!");
        return "/user/newpass";
    }
    if (!pass.equals(confirmPass)) {
        model.addAttribute("messenger", "Xác nhận mật khẩu không khớp !!!");
        return "/user/newpass";
    }
    if (pass.length() < 8) {
        model.addAttribute("messenger", "Mật khẩu phải hơn 8 kí tự !!!");
        return "/user/newpass";
    }
    if (pass.equals(confirmPass)) {

        user.setPassWord(userService.maHoaMatKhau(pass));

        userService.updateUser(user);

        model.addAttribute("user", new NguoiDungEntity());
        return "/user/login";
    }
}
return "/user/newpass";
}

```

2. Thay đổi thông tin cá nhân, thay đổi mật khẩu(người dùng, admin,owner)

Người dùng:

- Cho phép thay đổi họ tên, giới tính, ngày sinh, địa chỉ.

Hồ Sơ Của Tôi

| | |
|---------------|---|
| UserName | tinnguyen |
| Họ Tên | Nguyễn Tín |
| Email | tinnguyen3523@gmail.com |
| Số điện thoại | 0937681469 |
| Giới Tính | <input checked="" type="radio"/> Nam <input type="radio"/> Nữ |
| Ngày Sinh | 27/05/2002 |
| Địa chỉ | Bình Thạnh |

Lưu

Cập nhật thông tin thành công

```

@RequestMapping(value = "form/info", params = "save", method = RequestMethod.POST)
public String infoUpdate(HttpServletRequest request, ModelMap model, @ModelAttribute("user") NguoiDungEntity user,
        BindingResult errors) {
    Boolean loi = Boolean.TRUE;
    HttpSession session = request.getSession();
    NguoiDungEntity userSave = (NguoiDungEntity) session.getAttribute("USER");
    user.setUserName(userSave.getUserName());
    user.setEmail(userSave.getEmail());
    user.setSdt(userSave.getSDT());

    String NGAYSINH = request.getParameter("ngaySinh");
    java.sql.Date ns = java.sql.Date.valueOf(NGAYSINH);
    if (user.getHoTen().isEmpty()) {
        model.addAttribute("loiHoTen", "Họ tên không được để trống !!!");
        loi = Boolean.FALSE;
    } else if (user.getHoTen().length() > 50) {
        model.addAttribute("loiHoTen", "Họ tên quá dài !!!");
        loi = Boolean.FALSE;
    } else if (!user.getHoTen().matches("[\\p{L} ]+")) {
        model.addAttribute("loiHoTen", "Họ tên không được chứa số !!!");
        loi = Boolean.FALSE;
    }
    if (user.getNgaySinh() == null) {
        model.addAttribute("loiNgaySinh", "Hãy nhập ngày sinh !!!");
        loi = Boolean.FALSE;
    } else if (!isValidAndOver18(user.getNgaySinh())) {
        model.addAttribute("loiNgaySinh", "Bạn cần lớn hơn 18 tuổi để tạo tài khoản !!!");
        loi = Boolean.FALSE;
    }
    if (user.getDiaChi().isEmpty()) {
        model.addAttribute("loiDiaChi", "Địa chỉ không được trống!!!");
        loi = Boolean.FALSE;
    }
    if (loi == Boolean.FALSE) {
        model.addAttribute("user", userSave);
        return "/user/user-info";
    }
    userSave.setHoTen(user.getHoTen());
    userSave.setGioiTinh(user.isGioiTinh());
    userSave.setNgaySinh(ns);
    userSave.setDiaChi(user.getDiaChi());
    session.setAttribute("USER", userSave);

    userService.updateUser(userSave);
    model.addAttribute("user", userSave);
    model.addAttribute("successMessage", "Cập nhật thông tin thành công");
    return "/user/user-info";
}

```

- Đổi mật khẩu: khi đổi mật khẩu mới thì phải nhập lại đúng mật khẩu cũ

```

@RequestMapping(value = "form/changePass.htm", params = "save", method = RequestMethod.POST)
public String changePass(HttpServletRequest request, ModelMap model) {
    Boolean loi = Boolean.TRUE;
    HttpSession session = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USER");
    String pass = request.getParameter("password");
    String newPassword = request.getParameter("newPassword");
    String reNewPass = request.getParameter("reNewPassword");

    if (pass.isEmpty()) {
        model.addAttribute("loiPassword", "Hãy nhập mật khẩu cũ !!!");
        loi = Boolean.FALSE;
    } else if (!userService.kiemTraMatKhau(pass, user.getPassWord())) {
        model.addAttribute("loiPassword", "Mật khẩu cũ không đúng !!!");
        return "/user/changePass";
    }

    if (newPass.isEmpty()) {
        model.addAttribute("loiNewPassword", "Hãy nhập mật khẩu mới !!!");
        loi = Boolean.FALSE;
    } else if (newPass.length() < 8) {
        model.addAttribute("loiNewPassword", "Mật khẩu tối thiểu 8 kí tự !!!");
        loi = Boolean.FALSE;
    } else if (newPass.contains(" ")) {
        model.addAttribute("loiNewPassword", "Mật khẩu không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (reNewPass.isEmpty()) {
        model.addAttribute("loiRePassword", "Hãy nhập lại mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (!reNewPass.equals(newPass)) {
        model.addAttribute("loiRePassword", "Xác nhận mật khẩu không đúng !!!");
        loi = Boolean.FALSE;
    }

    if (loi == Boolean.TRUE) {
        user.setPassWord(userService.maHoaMatKhau(newPass));

        userService.updateUser(user);
        model.addAttribute("thanhCong", "Đổi mật khẩu thành công !!!");
    }
}

return "/user/changePass";
}

```

Admin và owner:

-Cho phép sửa tất cả thông tin cá nhân

The screenshot shows a dark-themed web interface with a navigation bar at the top containing links for 'Thống kê', 'Đơn hàng', 'Sản phẩm', 'Tài khoản khách', 'Tài khoản Admin', 'Thông tin cá nhân', 'Đổi mật khẩu', and 'Đăng xuất'. Below the navigation bar is a form titled 'THÔNG TIN' (Information). The form fields include:

- Họ Tên (Name): Nguyễn Admin
- Ngày Sinh (Date of Birth): 27/05/2002
- Giới Tính (Gender):
 - Nam
 - Nữ
- SĐT (Phone Number): 0937681469
- Địa Chỉ (Address): Bình Thạnh
- Email: tinnguyen3523@gmail.com

At the bottom of the form is a blue button labeled 'CẬP NHẬT THÔNG TIN' (Update Information).

```

@RequestMapping(value = "changeInfo", params = "save", method = RequestMethod.POST)
public String infoUpdate(HttpServletRequest request, ModelMap model, @ModelAttribute("admin") NguoiDungEntity admin,
        BindingResult errors) {
    Boolean loi = Boolean.TRUE;
    HttpSession session = request.getSession();
    NguoiDungEntity adminSave = (NguoiDungEntity) session.getAttribute("USER");
    admin.setUserName(adminSave.getUserName());

    String NGAYSINH = request.getParameter("ngaySinh");
    java.sql.Date ns = java.sql.Date.valueOf(NGAYSINH);
    if (admin.getHoTen().isEmpty()) {
        model.addAttribute("loiHoTen", "Họ tên không được để trống !!!");
        loi = Boolean.FALSE;
    } else if (admin.getHoTen().length() > 50) {
        model.addAttribute("loiHoTen", "Họ tên quá dài !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getHoTen().matches("[\\p{L} ]+")) {
        model.addAttribute("loiHoTen", "Họ tên không được chứa số !!!");
        loi = Boolean.FALSE;
    }

    if (admin.getNgaySinh() == null) {
        model.addAttribute("loiNgaySinh", "Hãy nhập ngày sinh !!!");
        loi = Boolean.FALSE;
    } else if (!isValidAndOver18(admin.getNgaySinh())) {
        model.addAttribute("loiNgaySinh", "Bạn cần lớn hơn 18 tuổi để tạo tài khoản !!!");
        loi = Boolean.FALSE;
    }
    if (admin.getDiaChi().isEmpty()) {
        model.addAttribute("loiDiaChi", "Địa chỉ không được trống!!!");
        loi = Boolean.FALSE;
    }

    if (admin.getSdt().isEmpty()) {
        model.addAttribute("loiSdt", "Hãy nhập sdt !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getSdt().matches("[0-9]+")) {
        model.addAttribute("loiSdt", "SDT không hợp lệ !!!");
        loi = Boolean.FALSE;
    } else if (admin.getSdt().length() != 10 && admin.getSdt().length() != 11) {
        model.addAttribute("loiSdt", "SDT không hợp lệ !!!");
        loi = Boolean.FALSE;
    }

    if (admin.getEmail().isEmpty()) {
        model.addAttribute("loiEmail", "Hãy nhập email !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getEmail().endsWith("@gmail.com")) {
        model.addAttribute("loiEmail", "Hãy nhập email đúng định dạng !!!");
        loi = Boolean.FALSE;
    }
}

```

```

if (loi == Boolean.FALSE) {
    model.addAttribute("admin", adminSave);
    return "/admin/me";
}
adminSave.setHoTen(admin.getHoTen());
adminSave.setGioiTinh(admin.isGioiTinh());
adminSave.setNgaySinh(ns);
adminSave.setDiaChi(admin.getDiaChi());
admin.setEmail(admin.getEmail());
admin.setSdt(admin.getSdt());
session.setAttribute("USER", adminSave);

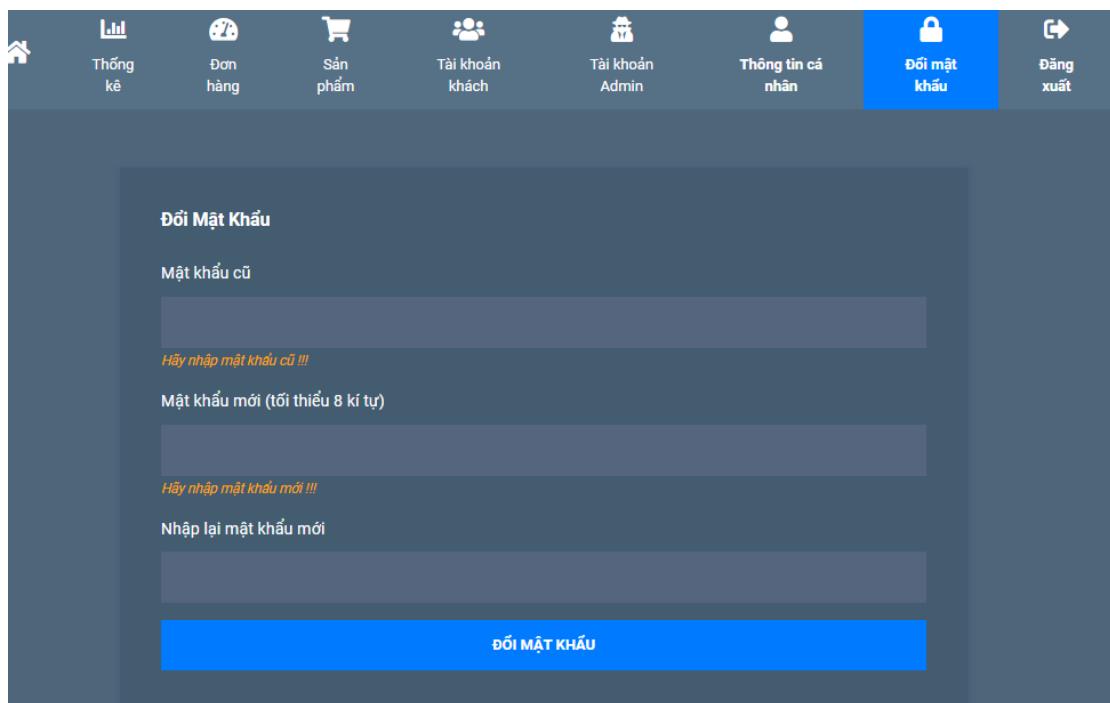
nguoIDungService.updateUser(adminSave);

model.addAttribute("admin", adminSave);
model.addAttribute("successMessage", "Cập nhật thông tin thành công");

return "/admin/me";
}

```

- Đổi mật khẩu: khi đổi mật khẩu mới thì phải nhập lại đúng mật khẩu cũ



```

@RequestMapping(value = "changePass.htm", params = "save", method = RequestMethod.POST)
public String changePass(HttpServletRequest request, ModelMap model) {
    Boolean loi = Boolean.TRUE;
    HttpSession session = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USER");
    String pass = request.getParameter("pass");
    String newPass = request.getParameter("newPass");
    String reNewPass = request.getParameter("renewPass");

    if (pass.isEmpty()) {
        model.addAttribute("loiPass", "Hãy nhập mật khẩu cũ !!!");
        loi = Boolean.FALSE;
    }

    if (!nguoIDungService.kiemTraMatKhau(pass, user.getPassword())) {
        model.addAttribute("loiPass", "Mật khẩu cũ không đúng !!!");
        return "/admin/changePass";
    }

    if (newPass.isEmpty()) {
        model.addAttribute("loiNewPass", "Hãy nhập mật khẩu mới !!!");
        loi = Boolean.FALSE;
    } else if (newPass.length() < 8) {
        model.addAttribute("loiNewPass", "Mật khẩu tối thiểu 8 kí tự !!!");
        loi = Boolean.FALSE;
    } else if (newPass.contains(" ")) {
        model.addAttribute("loiNewPass", "Mật khẩu không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (reNewPass.isEmpty()) {
        model.addAttribute("loiRePass", "Hãy nhập lại mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (!reNewPass.equals(newPass)) {
        model.addAttribute("loiRePass", "Xác nhận mật khẩu không đúng !!!");
        loi = Boolean.FALSE;
    }

    if (loi == Boolean.TRUE) {
        user.setPassword(nguoiDungService.maHoaMatKhau(newPass));
        nguoIDungService.updateUser(user);
        model.addAttribute("successMessage", "Đổi mật khẩu thành công !!!");
    }
}

return "/admin/changePass";
}

```

3. Tìm kiếm sản phẩm

Khách hàng có thể tìm kiếm sản phẩm bằng cách nhập tên, loại hoặc mã sản phẩm mình muốn tìm.

```

<div id="seacrh-box">
    <input type="text" name="key" id="search" placeholder="Bạn tìm gì..." />
    <button id="search-icon" type="submit" name="btnSearch">
        <i class="ti-search "></i>
    </button>
</div>

```



Xử lý tìm kiếm sản phẩm theo từ khóa nhập vào trong Controller:

```

@RequestMapping(params = "btnSearch")
public String timSanPham(ModelMap model, @RequestParam(value = "key", required = false) String key,
    HttpServletRequest request) {
    List<SanPhamEntity> listSP=SanPhamService.laySanPhamTheoMa(key);
    model.addAttribute("key", key);
    model.addAttribute("listSP", listSP);
    int so = listSP.size();
    model.addAttribute("soLuong", so);
    String mess ="";
    if (listSP.isEmpty()) mess = "Không có sản phẩm này";
    else {
        mess = "Tìm thấy "+so+" sản phẩm";
    }
    model.addAttribute("message", mess);
    return "sanPham/timKiem";
}

```

Hàm lấy sản phẩm theo từ khóa tìm kiếm:

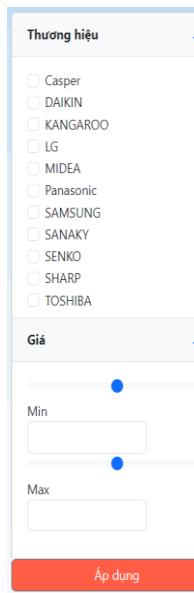
```

@Override
public List<SanPhamEntity> laySanPhamTheoMa(String key) {
    Session session = sessionFactory.getCurrentSession();
    String hql = "FROM SanPhamEntity sp WHERE sp.maSP LIKE :key " +
        "OR sp.tenSanPham LIKE :key " +
        "OR sp.thuongHieu.tenThuongHieu LIKE :key " +
        "OR sp.loaiSanPham.maLoai LIKE :key";
    Query query = session.createQuery(hql);
    query.setParameter("key", "%" + key + "%");
    return query.list();
}

```

View “SanPham/TimKiem” trả về danh sách sản phẩm liên quan đến từ khóa tìm kiếm và hiển thị số sản phẩm trên view là 6 sản phẩm và một nút “Xem thêm”. Khi ấn vào nút “Xem thêm” sẽ hiển thị 6 sản phẩm tiếp theo hoặc số sản phẩm còn lại trong danh sách nếu ít hơn 6. Khi hiển thị hết sản phẩm nút “Xem thêm” sẽ ẩn đi.

4. Bộ lọc sản phẩm



Trong từng trang loại sản phẩm, người dùng có thể lọc sản phẩm theo thương hiệu và giá.

Người dùng có thể tìm sản phẩm theo thương hiệu mình muốn bằng cách chọn vào các ô có tên thương hiệu đó, và giá trị sản phẩm tối thiểu và tối đa có thể lọc theo ô bên dưới, có thể nhập giá trị vào hoặc sử dụng thanh trượt để điều chỉnh giá theo mong muốn.

Khi đã điều chỉnh thông tin theo yêu cầu, bấm vào nút áp dụng để trả về view chứa các sản phẩm theo yêu cầu. Khi hiển thị chỉ hiển thị tối đa 6 sản phẩm, ấn vào nút “Xem thêm” để xem các sản phẩm phía sau. Nếu không có sản phẩm thuộc yêu cầu trả về thông

Xử lý bộ lọc trong Controller:

```

@RequestMapping(value = "/{loaiSp}", params = "btnApply", method = RequestMethod.POST)
public String locSanPham(@PathVariable("loaiSp") String loaiSp, ModelMap model, HttpServletRequest request,
    @RequestParam(value = "brands", required = false) List<String> brandsList,
    @RequestParam(value = "minPrice", defaultValue = "0") int minPrice,
    @RequestParam(value = "maxPrice", defaultValue = "99999999") int maxPrice,
    @RequestParam(defaultValue = "0") int page)
{
    List<SanPhamEntity> categoryList = sanPhamService.locSanPhamTheoThuongHieuVaGia(loaiSp, brandsList, minPrice,
        maxPrice);
    List<ThuongHieuEntity> thuongHieulist = thuongHieuService.layThuongHieu();
    model.addAttribute("thuongHieuList", thuongHieuList);
    model.addAttribute("categoryList", categoryList);
    model.addAttribute("selectedBrands", brandsList);
    model.addAttribute("minPrice", minPrice);
    model.addAttribute("maxPrice", maxPrice);
    if (categoryList.isEmpty()) {
        model.addAttribute("message", "Hiện không có sản phẩm cần tìm");
        return "sanPham/locSanPham";
    }
    return "sanPham/locSanPham";
}

```

Hàm trả về danh sách sản phẩm từ database có thông tin yêu cầu từ bộ lọc:

```

@Override
public List<SanPhamEntity> locSanPhamTheoThuongHieuVaGia(String loai, List<String> brandsList, int minPrice, int maxPrice) {
    Session session = sessionFactory.getCurrentSession();
    String hql = "FROM SanPhamEntity sp WHERE sp.loaiSanPham.maLoai = :loai and trangThai=True ";
    if (brandsList != null && !brandsList.isEmpty()) {
        hql += "AND sp.thuongHieu.tenThuongHieu IN :brandsList ";
    }
    if (minPrice >= 0 && maxPrice >=0) {
        hql += "AND sp.donGia >= :minPrice AND sp.donGia <= :maxPrice ";
    }

    Query query = session.createQuery(hql).setParameter("loai", loai);
    if (brandsList != null && !brandsList.isEmpty()) {
        query.setParameterList("brandsList", brandsList);
    }
    if (minPrice >=0 && maxPrice >=0) {
        query.setParameter("minPrice", minPrice).setParameter("maxPrice", maxPrice);
    }
    List<SanPhamEntity> categoryList = query.list();
    return categoryList;
}

```

5. Xem chi tiết sản phẩm

Khi click chuột vào một khối chứa sản phẩm bất kì trả về view chứa thông tin chi tiết về sản phẩm đó.

Bên dưới thông tin về sản phẩm sẽ có thể có các đánh giá của khách hàng đã mua sản phẩm.

Hàm trả về một sản phẩm trong Controller:

```

@RequestMapping("/{maSp}")
public String sanPham(@PathVariable("maSp") String maSp, ModelMap model, HttpServletRequest request) {
    SanPhamEntity sanPham=sanPhamService.laySanPham(maSp);
    Hibernate.initialize(sanPham.getHinhAnhs());
    List<SanPhamEntity> sanPhamCungLoai = sanPhamService.laySanPhamCungLoai(maSp);
    List<DanhGiaEntity> listDanhGia = danhGiaService.layDanhGiaSanPham(maSp);
    int count = listDanhGia.size();
    float soSaoTB = sanPhamService.tinhSoSaoTB(sanPham);
    sanPham.setSoSaoTB(soSaoTB);
    sanPhamService.updateSanPham(sanPham);

    model.addAttribute("sanPham", sanPham);
    model.addAttribute("sanPhamCungLoai", sanPhamCungLoai);
    model.addAttribute("danhGiaList", listDanhGia);
    model.addAttribute("count", count);

    return "/sanPham/sanPham";
}

```

6. Thêm sản phẩm vào giỏ hàng

Người dùng sau khi đăng nhập sẽ được thêm sản phẩm vào giỏ hàng

Thương hiệu: **TOSHIBA**

Loại: **MÁY GIẶT**

Tên sản phẩm:

Máy giặt Toshiba Inverter 10.5 Kg TW-BK115G4V

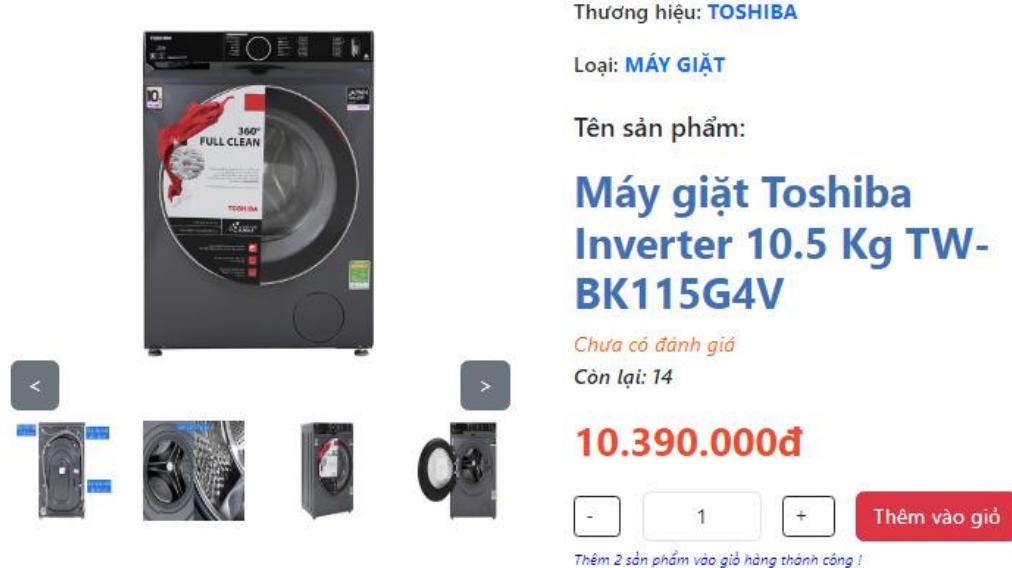
Chưa có đánh giá

Còn lại: 14

10.390.000đ

- 1 + Thêm vào giỏ

Thêm 2 sản phẩm vào giỏ hàng thành công!



Trong giỏ hàng có thể tùy chỉnh lại số lượng hoặc xoá sản phẩm khỏi giỏ hàng

| GIỎ HÀNG CỦA BẠN | | | | | |
|---|-------------|---|--------------------|---|---------------------------------|
| Tên Sản Phẩm | Đơn giá | Số Lượng | Số Tiền | Thao Tác | |
|  Máy giặt Toshiba Inverter 10.5 Kg TW-BK115G4V | 10.390.000đ | <input type="button"/> 2 Cập Nhật | 20.780.000đ |  | |
|  Tủ lạnh Toshiba Inverter 511 lít GR-RF610WE | 19.990.000đ | <input type="button"/> 3 Cập Nhật | 59.970.000đ |  | |
| Tổng cộng: 80.750.000đ | | | | | <input type="button"/> Mua Hàng |

```

@RequestMapping(value = "/themVaoGio/{maSP}", params = "them", method = RequestMethod.POST)
public String them(@PathVariable("maSP") String maSP, HttpServletRequest request, ModelMap model) {

    HttpSession session0 = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session0.getAttribute("USER");

    SanPhamEntity sanPham = laySanPhamTheoMa(maSP);

    if(user==null)
    {
        model.addAttribute("user", new NguoiDungEntity());
        session0.setAttribute("SANPHAM", maSP);
        return "/user/login";
    }

    GioHangEntity gioHang = gioHangService.layGioHangTheoMaNvSanPham(user.getMaNv(), maSP);
    int soLuong = Integer.parseInt(request.getParameter("soLuong"));

    if (gioHang == null) {
        gioHang = new GioHangEntity();

        gioHang.setNguoiDung(user);
        gioHang.setSanPham(sanPham);
        gioHang.setSoLuong(soLuong);

        gioHangService.addGioHang(gioHang);

        String mes = "Thêm " + String.valueOf(soLuong) + " sản phẩm vào giỏ hàng thành công !";
        model.addAttribute("messenger", mes);
    }

    else {
        gioHang.setSoLuong(gioHang.getSoLuong() + soLuong); //nếu sản phẩm đã có trong giỏ hàng
        // thì sẽ cộng thêm số lượng vào

        gioHangService.updateGioHang(gioHang);
        String mes = "Thêm " + String.valueOf(soLuong) + " sản phẩm vào giỏ hàng thành công !";
        model.addAttribute("messenger", mes);
    }

    model.addAttribute("sanPham", sanPham);
    List<SanPhamEntity> sanPhamCungLoai = laySanPhamCungLoai(sanPham.getLoaiSanPham(), maSP);
    model.addAttribute("sanPhamCungLoai", sanPhamCungLoai);
    return "/sanPham/sanPham";
}

```

7. Mua hàng -Huỷ đơn hàng- xem lịch sử đơn hàng

Mua hàng

- Người dùng sau khi thêm sản phẩm vào giỏ có thể tiến hành mua hàng
- Đơn hàng có thể chọn địa chỉ mặc định hoặc chọn địa chỉ khác

| Tên Sản Phẩm | Đơn giá | Số Lượng | Thành Tiền |
|---|-------------|----------|-------------|
| Máy giặt Toshiba Inverter 10.5 Kg TW-BK115G4V | 10.390.000đ | 2 | 20.780.000đ |
| Tủ lạnh Toshiba Inverter 511 lít GR-RF610WE | 19.990.000đ | 3 | 59.970.000đ |

Tổng cộng: 80.750.000đ

Đặt Hàng

```

@RequestMapping(value = "donHang/datHang", method = RequestMethod.POST, params = "datHang")
public String datHang(HttpServletRequest request, HttpSession session) {
    NguoiDungEntity newInfo = (NguoiDungEntity) session.getAttribute("NEWINFO"); // lấy địa chỉ mới
    NguoiDungEntity user = (NguoiDungEntity) session.getAttribute("USER");
    List<GioHangEntity> gioHangList =
        gioHangService.layGioHangCuaUser(user.getMaND());
    int tongTien = Integer.parseInt(request.getParameter("tongTien"));
    DonHangEntity donHang = new DonHangEntity();
    Date currentDate = new Date();
    donHang.setNgayTao(currentDate);
    donHang.setTongTien(tongTien);
    donHang.setTrangThai(1);
    if (newInfo == null) {
        donHang.setHoTen(user.getHoTen());
        donHang.setSdt(user.getSdt());
        donHang.setDiaChi(user.getDiaChi());
    }
    donHang.setNguoiDung(user);

    } else {
        donHang.setHoTen(newInfo.getHoTen());
        donHang.setSdt(newInfo.getSdt());
        donHang.setDiaChi(newInfo.getDiaChi());
        donHang.setNguoiDung(user); }
    DonHangService.luuDonHang(donHang);

    GioHangEntity gh = new GioHangEntity();
    SanPhamEntity sp=new SanPhamEntity();

    for (int i = 0; i < gioHangList.size(); i++)
    { gh = gioHangList.get(i);
    CTDonHangEntity ctdh = new CTDonHangEntity();
    ctdh.setDonHang(donHang);
    ctdh.setSanPham(gh.getSanPham());
    ctdh.setDonGia(gh.getSanPham().getDonGia());
    ctdh.setSoLuong(gh.getSoLuong());
    ctdh.setTrangThaiDanhGia(false);
    ctDonHangService.luuCtdh(ctdh);

    sp=sanPhamService.laySanPham(gh.getSanPham().getMaSP());
    sp.setSoLuong(sp.getSoLuong()-gh.getSoLuong());
    sanPhamService.updateSanPham(sp);
    gioHangService.deleteGioHang(gh.getMaGH());
    }

    return "/donHang/thanhCong";
}

```

Xem lịch sử đơn hàng

- Sau khi đặt hàng có thể vào xem trạng thái đơn hàng hoặc lịch sử các đơn hàng đã đặt

-Đơn hàng vừa đặt sẽ có trạng thái chờ xác nhận và có thể huỷ

| Thành công | Đang giao | Chờ xác nhận | Đã huỷ |
|------------------------------|------------|--------------|---|
| ĐƠN HÀNG CHỜ XÁC NHẬN | | | |
| # | Ngày đặt | Tổng Tiền | |
| 1 | 2023-06-04 | 11.900.000đ | Chi tiết đơn X Huỷ đơn hàng |
| 2 | 2023-06-04 | 11.090.000đ | Chi tiết đơn X Huỷ đơn hàng |

Hàm xác nhận hủy:

```

@RequestMapping("huyDonHang/{maDh}")
public String huyDonHang(@PathVariable("maDh")int maDh )
{
    DonHangEntity donHang=DonHangService.timDonHangTheoMa(maDh);
    donHang.setTrangThai(0);
    DonHangService.updateDonHang(donHang);

    List<CTDonHangEntity> ctDonHangList=ctDonHangService.timctdhTheoMaDh(maDh);
    SanPhamEntity sp=new SanPhamEntity();
    for (int i=0; i<ctDonHangList.size();i++)
    {
        sp=sanPhamService.laySanPham(ctDonHangList.get(i).getSanPham().getMaSP());
        sp.setSoLuong(sp.getSoLuong()+ctDonHangList.get(i).getSoLuong());
        sanPhamService.updateSanPham(sp);
    }

    return "redirect:/lichSuDonHang.htm";
}

```

-Đơn hàng đang giao

| Thành công | Dang giao | Chờ xác nhận | Đã huỷ |
|------------|------------|--------------|--|
| # | Ngày đặt | Tổng Tiền | |
| 1 | 2023-02-04 | 1.070.000đ | Chi tiết đơn ✓ Đã Nhận Hàng |

```
@RequestMapping("daNhanHang/{maDh}")
public String daNhanHang(@PathVariable("maDh")int maDh )
{
    DonHangEntity donHang=DonHangService.timDonHangTheoMa(maDh);
    donHang.setTrangThai(3);
    DonHangService.updateDonHang(donHang);

    return "redirect:/lichSuDonHang.htm";
}
```

-Đơn hàng thành công:

| ĐƠN HÀNG THÀNH CÔNG | | | | |
|---------------------|------------|-------------|------------------------------|--------------------------|
| # | Ngày đặt | Tổng Tiền | | |
| 1 | 2023-06-05 | 95.730.000đ | Chi tiết đơn | Đánh Giá |
| 2 | 2023-02-24 | 535.000đ | Chi tiết đơn | Đánh Giá |

-Đơn hàng đã huỷ:

| ĐƠN HÀNG ĐÃ HUỶ | | | | |
|-----------------|------------|------------|------------------------------|--|
| # | Ngày đặt | Tổng Tiền | | |
| 1 | 2023-05-24 | 5.000.000đ | Chi tiết đơn | |

8. Đánh giá sản phẩm

- Sau khi nhận hàng thành công thì có thể đánh giá các sản phẩm trong đơn hàng, gồm có số sao và nội dung.
- Đánh giá sẽ không thể sửa hoặc xoá.

Máy giặt Toshiba Inverter 10.5 Kg TW-BK115G4V

5 ★ - Tuyệt Vời

Hãy chia sẻ những điều bạn thích về sản phẩm này

Lưu Đánh Giá

```

@RequestMapping("luuDanhGia/{maSp}/{maCTDH}")
public String luuDanhGia(@PathVariable("maSp")String maSp,@PathVariable("maCTDH") int maCTDH ,HttpServletRequest request ,HttpSession session)
{
    Date currentDate = new Date();
    DanhGiaEntity danhGia=new DanhGiaEntity();
    NguoiDungEntity user=(NguoiDungEntity) session.getAttribute("USER");
    SanPhamEntity sanPham=sanPhamService.laySanPham(maSp);
    CTDonHangEntity ctdh=ctDonHangService.timCtdhTheoMaCtdh(maCTDH);

    float soSao=Float.parseFloat(request.getParameter("soSao"));
    String noiDung=request.getParameter("noiDung");
    danhGia.setNguoiDung(user);
    danhGia.setNgay(currentDate);
    danhGia.setSoSao(soSao);
    danhGia.setNoiDung(noitDung);
    danhGia.setSanPham(sanPham);
    danhGiaService.saveDanhGia(danhGia);

    ctdh.setTrangThaiDanhGia(true);
    ctDonHangService.updateCtdh(ctdh);

    int maDh=(int) session.getAttribute("MADH");
    session.removeAttribute("MADH");

    return "redirect:/danhGia/"+maDh+".htm";
}

```

9. Xem thống kê doanh thu, đơn hàng

Khi đăng nhập vào tài khoản Admin hoặc Owner thì sẽ tự động được chuyển đến trang thống kê của trang web. Trang này bao gồm một số thống kê chính về trang web như:

- Doanh thu (tổng cộng số tiền đã bán trên web)
- Số người dùng trên web

Thống kê doanh thu

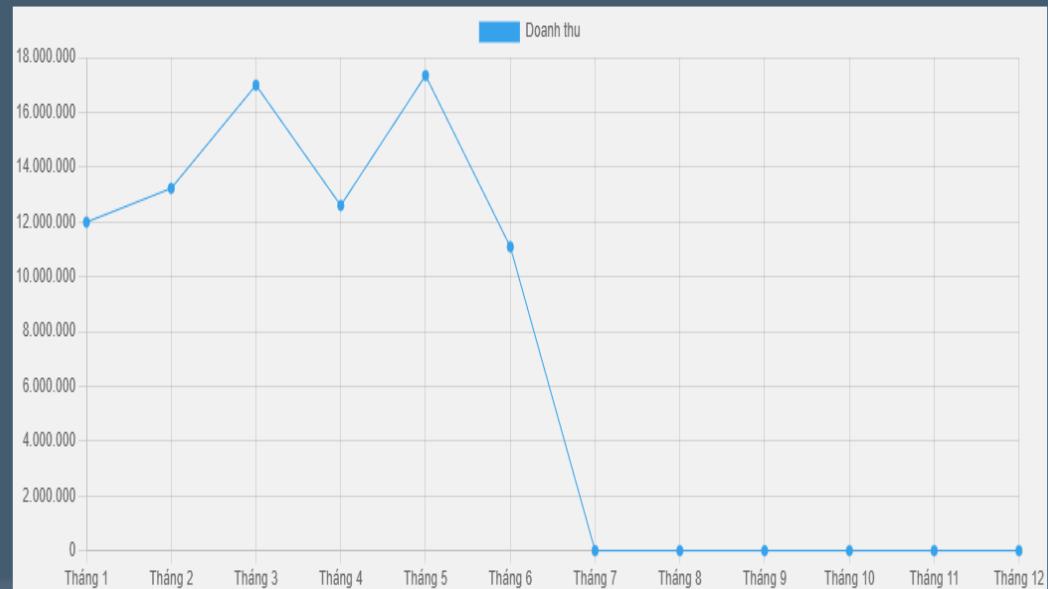
Tổng doanh thu đã bán : 93.130.000đ

Thống kê số người dùng

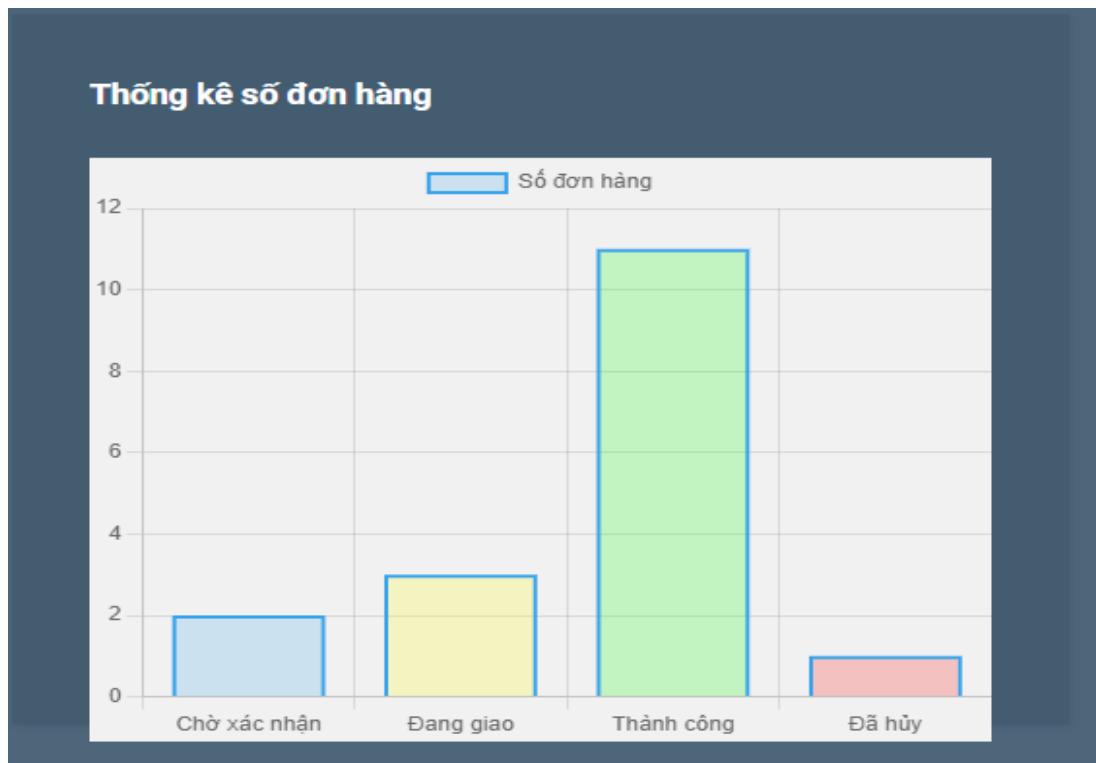
Số người dùng trên web : 11

- Doanh thu các tháng của năm hiện tại(được so sánh bằng một biểu đồ đường)

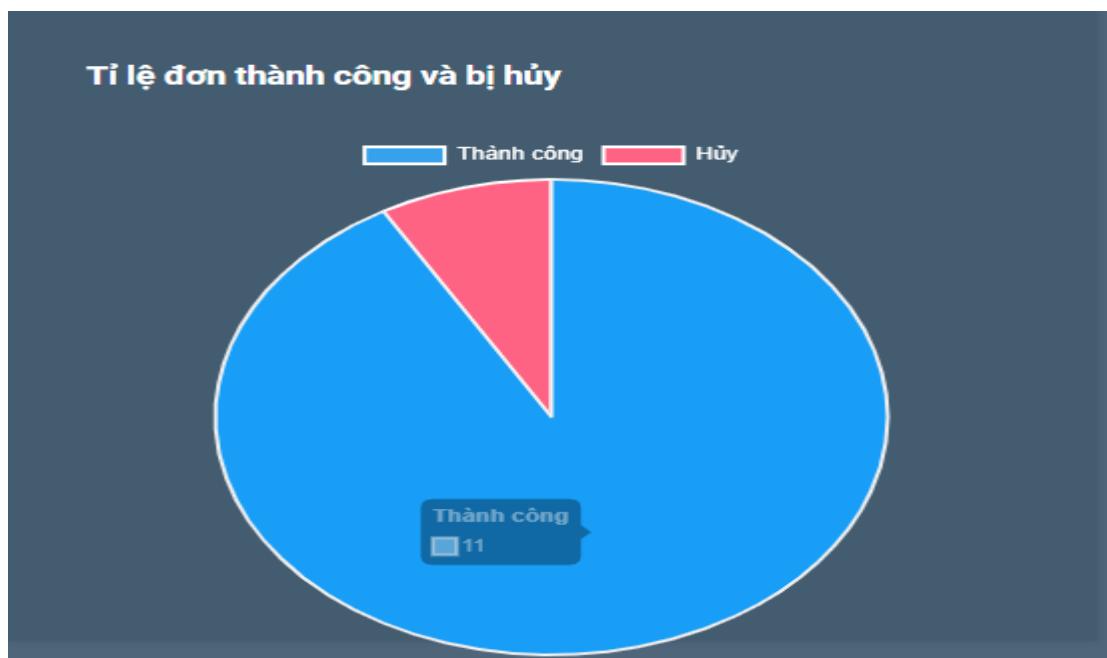
Thống kê doanh thu các tháng năm nay



- Các đơn hàng theo từng trạng thái(được so sánh bằng một biểu đồ cột)



- Tỉ lệ giữa đơn hàng thành công và đơn bị hủy(được so sánh bằng một biểu đồ tròn)



```

    @RequestMapping("index")
    public String index(HttpServletRequest request, ModelMap model) {
        HttpSession session0 = request.getSession();
        NguoiDungEntity user = (NguoiDungEntity) session0.getAttribute("USER");
        List<DonHangEntity> donThanhCong = donHangService.layDonHangTheoTrangThai(3);
        int tongDoanhThu = 0;
        for (DonHangEntity donHang : donThanhCong) {
            tongDoanhThu += donHang.getTongTien();
        }
        model.addAttribute("tongDoanhThu", tongDoanhThu);
        // Tính tổng doanh thu theo từng tháng
        List<Long> monthlyRevenues = new ArrayList<>();
        for (int i = 1; i <= 12; i++) {
            long totalRevenue = donHangService.tinhTongDoanhThuTheoThang(i);
            monthlyRevenues.add(totalRevenue);
        }
        model.addAttribute("monthlyRevenues", monthlyRevenues);
        // Thống kê số người dùng
        List<NguoiDungEntity> listNguoiDung = nguoiDungService.getAllUserByRole(0);
        int tongSoNguoiDung=listNguoiDung.size();
        model.addAttribute("tongSoNguoiDung", tongSoNguoiDung);
        // Thống kê số đơn hàng
        int tongDonChoXacNhan= donHangService.layDonHangTheoTrangThai(1).size();
        int tongDonDangGiao= donHangService.layDonHangTheoTrangThai(2).size();
        int tongDonThanhCong= donHangService.layDonHangTheoTrangThai(3).size();
        int tongDonDaHuy= donHangService.layDonHangTheoTrangThai(0).size();

        model.addAttribute("pendingOrders", tongDonChoXacNhan);
        model.addAttribute("deliveringOrders", tongDonDangGiao);
        model.addAttribute("completedOrders", tongDonThanhCong);
        model.addAttribute("cancelledOrders", tongDonDaHuy);
        return "admin/index";
    }
}

```

10. Quản lý đơn hàng

Trang này cho phép xem và chi tiết số đơn hàng theo từng trạng thái (Chờ xác nhận, Đang giao, Thành công, Đã hủy) và cho phép người quản lý duyệt hoặc hủy đơn hàng phù hợp.

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết | Xác nhận duyệt | Hủy đơn này |
|--------|----------------|------------|-------------|--------------|------------------------|----------------------|
| #1047 | Nguyễn Văn Sơn | 2023-06-04 | 5.000.000đ | | <button>Duyệt</button> | <button>Hủy</button> |
| #1049 | Đặng Văn Nam | 2023-06-04 | 10.990.000đ | | <button>Duyệt</button> | <button>Hủy</button> |

Đơn hàng đang được giao

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|-----------------|------------|-------------|--------------|
| #1044 | Nguyễn Văn Hùng | 2023-06-04 | 11.900.000đ | |
| #1048 | Nguyễn Văn Sơn | 2023-04-04 | 6.535.000đ | |
| #1043 | Nguyễn Văn Hùng | 2023-02-04 | 1.070.000đ | |

Đơn hàng đã giao thành công

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|-----------------|------------|-------------|--------------|
| #1045 | Nguyễn Văn Hùng | 2023-06-04 | 11.090.000đ | |
| #26 | Nguyễn Văn Hùng | 2023-05-24 | 5.000.000đ | |
| #1046 | Nguyễn Văn Sơn | 2023-05-04 | 12.345.000đ | |
| #1050 | Đặng Văn Nam | 2023-04-04 | 550.000đ | |
| #1051 | Trần Văn B | 2023-04-04 | 5.550.000đ | |
| #1052 | Trần Văn B | 2023-04-04 | 6.500.000đ | |

Đơn hàng đã bị hủy

| Mã đơn | Người nhận | Ngày đặt | Tổng tiền | Xem chi tiết |
|--------|--------------|------------|------------|--------------|
| #1053 | Đặng Văn Nam | 2023-06-04 | 5.466.000đ | |

```

@RequestMapping(value = "admin/order", method = RequestMethod.GET)
public String product(ModelMap model, HttpServletRequest request) {
    HttpSession session0 = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session0.getAttribute("USER");

    List<DonHangEntity> listDonHang = donHangService.layAllDonHang();
    model.addAttribute("listDonHang", listDonHang);

    return "admin/order";
}

@RequestMapping(value = "admin/order", params = "changeStatus2", method = RequestMethod.POST)
public String chuyenDonHangSangDangGiao(HttpServletRequest request) {
    int maDonHang = Integer.parseInt(request.getParameter("maDonHangDuyet"));
    int trangThai = Integer.parseInt(request.getParameter("trangThaiDonDuyet"));

    if (trangThai == 1) { // Nếu trạng thái là "Chờ xác nhận"
        DonHangEntity donHang = donHangService.timDonHangTheoMa(maDonHang);
        donHang.setTrangThai(2); // Chuyển trạng thái đơn hàng sang "Đang giao"
        donHangService.updateDonHang(donHang);
    }

    return "redirect:/admin/order.htm";
}

@RequestMapping(value = "admin/order", params = "changeStatus1", method = RequestMethod.POST)
public String chuyenDonHangSangHuy(HttpServletRequest request) {
    int maDonHang = Integer.parseInt(request.getParameter("maDonHangHuy"));
    int trangThai = Integer.parseInt(request.getParameter("trangThaiDonHuy"));

    if (trangThai == 1) { // Nếu trạng thái là "Chờ xác nhận"
        DonHangEntity donHang = donHangService.timDonHangTheoMa(maDonHang);
        donHang.setTrangThai(0); // Chuyển trạng thái đơn hàng sang "Đang giao"
        donHangService.updateDonHang(donHang);
    }

    return "redirect:/admin/order.htm";
}

```

11. Quản lý sản phẩm, thương hiệu (thêm, xóa, sửa thông tin, sửa trạng thái)

Tài khoản quyền Admin trỏ lên có quyền thêm, sửa, xóa, sản phẩm hoặc thương hiệu để cập nhật vào cơ sở dữ liệu.

- Mỗi sản phẩm có nút chuyển đổi trạng thái để dễ dàng chuyển giữa đang bán hoặc ngừng kinh doanh

The screenshot shows a web application for managing products and brands. On the left, there's a section titled "Các sản phẩm đang kinh doanh" (Products currently for sale) displaying five items with columns for Image, Product Name, Price, Stock, Status, Repair, and Delete. Below this is a blue button labeled "THÊM SẢN PHẨM" (Add Product). On the right, there's a section titled "Các thương hiệu đang bán" (Brands currently for sale) listing brands like Casper, DAIKIN, KANGAROO, LG, MIDEA, and Panasonic, each with a修 (Repair) icon and a delete icon. Below this is a blue button labeled "THÊM THƯƠNG HIỆU" (Add Brand). At the bottom, there's another section titled "Các sản phẩm đã tạm ngừng kinh doanh" (Products temporarily stopped for sale) showing one item with a status of "Ngừng bán" (Not for sale).

-Khi ngừng kinh doanh thì bên trang thông tin sản phẩm sẽ như thế này



Và nếu sản phẩm đang ở trong giỏ hàng của khách thì sẽ hiện thế này

GIỎ HÀNG CỦA BẠN

| Tên Sản Phẩm | Đơn giá | Số Lượng | Số Tiền | Thao Tác |
|---|------------|-------------------------------|------------|---|
|  Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN <small>Sản phẩm này đã ngừng kinh doanh, vui lòng xoá khỏi giỏ hàng!</small> | 5.000.000đ | 1 Cập Nhật | 5.000.000đ |  |
| Tổng cộng: 5.000.000đ | | | | |
| Mua Hàng | | | | |

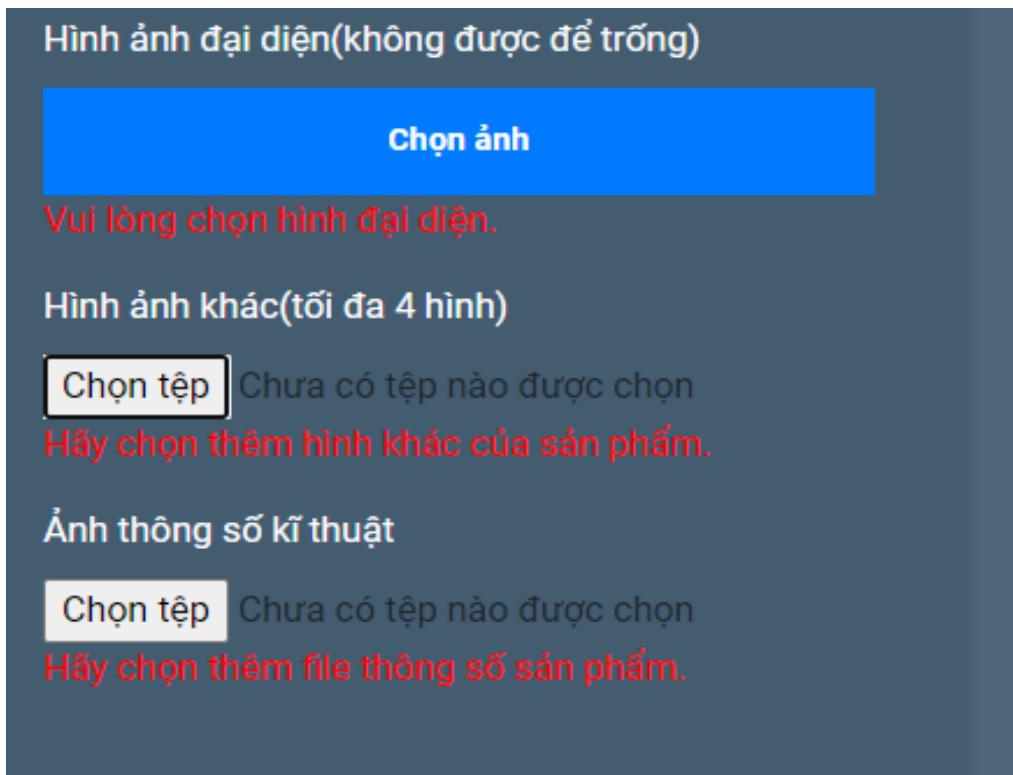
-Khi bấm vào nút thêm sản phẩm sẽ đến trang để thêm sản phẩm

Các sản phẩm đang kinh doanh

| Ảnh | Tên sản phẩm | Giá | Còn | Trạng thái | Sửa | Xóa |
|---|---|-------------|-----|--|---|---|
|  | Tủ lạnh Panasonic Inverter 170 lít NR-BA190PPVN | 5.000.000đ | 21 | Đang bán |  |  |
|  | Máy lạnh Samsung Inverter 1 HP AR09TYHQASINSV | 9.390.000đ | 12 | Đang bán |  |  |
|  | Máy giặt Samsung Inverter 9.5kg WW95T4040CE/SV | 11.090.000đ | 29 | Đang bán |  |  |
|  | Tủ đông SANAKY 208 Lít TD.VH255HY2 | 4.790.000đ | 29 | Đang bán |  |  |
|  | Máy lạnh Midea Inverter 1 HP MSAFA-10CRDN8 | 6.500.000đ | 11 | Đang bán |  |  |

[THÊM SẢN PHẨM](#)

- Bắt lỗi nếu người dùng không nhập đầy đủ thông tin sản phẩm hay không thêm đủ các hình.



```

@RequestMapping(value = "/admin/product/add", method = RequestMethod.GET)
public String viewAddProduct(ModelMap model, HttpServletRequest request) {
    HttpSession session0 = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session0.getAttribute("USER");
    List<ThuongHieuEntity> listThuongHieu = thuongHieuService.layThuongHieu();
    model.addAttribute("listThuongHieu", listThuongHieu);
    List<LoaiSanPhamEntity> listLoai = loaiService.layLoai();
    model.addAttribute("listLoai", listLoai);
    model.addAttribute("productForm", new SanPhamEntity());
    return "admin/addProduct";
}

@RequestMapping(value = "/admin/product/add", params = "add", method = RequestMethod.POST)
public String addProduct(@ModelAttribute("productForm") SanPhamEntity product,
    @RequestParam("avatar") MultipartFile avatar,
    @RequestParam("images") MultipartFile[] images,
    @RequestParam("thongSo") MultipartFile thongSo,
    ModelMap model) throws IOException {

    //Lấy ngày tháng công vào tên file để khỏi bị trùng file
    Date today = new Date();
    SimpleDateFormat formatter = new SimpleDateFormat("yyyyMMddHHmmss");
    String now = formatter.format(today);

    String avatarFileName = now + "-" + avatar.getOriginalFilename();
    String avatarFilePath = filePath + avatarFileName;
    File avatarFile = new File(avatarFilePath);
    avatar.transferTo(avatarFile);
    product.setHinhAnhDaiDien("assets/img/sanPham/" + avatarFileName);
}

```

```

List<HinhAnhEntity> hinhAnhKhacs = new ArrayList<>();
int count = 0;
for (MultipartFile file : images) {

    String fileName = now + "-" + file.getOriginalFilename();
    String otherImagefilePath = filePath + fileName;
    File dest = new File(otherImagefilePath);
    file.transferTo(dest);

    HinhanhEntity hinhanhKhac = new HinhanhEntity();
    hinhanhKhac.setLink("assets/img/sanPham/" + fileName);
    hinhanhKhac.setSanPham(product);
    hinhanhKhacs.add(hinhanhKhac);
}
model.addAttribute("count", count);

product.setHinhAnhs(hinhAnhKhacs);

String thongSoFileName = now + "-" + thongSo.getOriginalFilename();
String thongSoFilePath = filePath + thongSoFileName;
File thongSoFile = new File(thongSoFilePath);
thongSo.transferTo(thongSoFile);
product.setThongSoKt("assets/img/sanPham/" + thongSoFileName);

product.setNgayThem(today);

try {
    sanPhamService.themSanPham(product);
    sanPhamService.themHinhSanPham(hinhAnhKhacs);
    model.addAttribute("successMessage", "Thêm sản phẩm thành công.");
} catch (Exception e) {
    model.addAttribute("errorMessage", "Có lỗi xảy ra khi thêm sản phẩm. " + e.getMessage());
    return "/admin/addProduct";
}

return "admin/addProduct";
}

```

-Sửa sản phẩm

```

@RequestMapping(value = "/admin/product/edit/{masp}", method = RequestMethod.GET)
public String viewEditProduct(@PathVariable("masp") String masp, ModelMap model, HttpServletRequest request) {
    HttpSession session0 = request.getSession();
    NguoiDungEntity user = (NguoiDungEntity) session0.getAttribute("USER");

    SanPhamEntity sanPham = sanPhamService.laySanPham(masp);
    Hibernate.initialize(sanPham.getHinhAnhs());
    List<ThuongHieuEntity> listThuongHieu = thuongHieuService.layThuongHieu();
    List<LoaiSanPhamEntity> listLoai = loaiService.layLoai();
    model.addAttribute("sanPham", sanPham);
    model.addAttribute("listThuongHieu", listThuongHieu);
    model.addAttribute("listLoai", listLoai);
    model.addAttribute("product", sanPham);
    return "admin/editProduct";
}

@RequestMapping(value = "/admin/product/edit/{masp}", params = "update", method = RequestMethod.POST)
public String editProduct(@PathVariable("masp") String masp,
                           @ModelAttribute("product") SanPhamEntity product,
                           @RequestParam("avatar") MultipartFile avatar,
                           @RequestParam("images") MultipartFile[] images,
                           @RequestParam("thongSo") MultipartFile thongSo,
                           ModelMap model) throws IOException {
    SanPhamEntity sp= sanPhamService.laySanPham(masp);

    //Lấy ngày tháng cộng vào tên file để khởi tạo tên file
    Date today = new Date();
    SimpleDateFormat formatter = new SimpleDateFormat("yyyyMMddHHmmss");
    String now = formatter.format(today);

    if (!avatar.isEmpty()) {
        // Xóa hình đại diện cũ
        String hinhanhdaiDien = sp.getHinhAnhDaiDien();
        if (hinhanhdaiDien != null) {
            xoatepTinHinhAnh(hinhanhdaiDien);
        }
    }
}

```

```

if (!avatar.isEmpty()) {
    // Xóa hình đại diện cũ
    String hinhAnhDaiDien = sp.getHinhAnhDaiDien();
    if (hinhAnhDaiDien != null) {
        xotepTinHinhAnh(hinhAnhDaiDien);
    }

    String avatarFileName = now + "-" + avatar.getOriginalFilename();
    String avatarFilePath = filePath + avatarFileName;
    File avatarFile = new File(avatarFilePath);
    avatar.transferTo(avatarFile);
    product.setHinhAnhDaiDien("assets/img/sanPham/" + avatarFileName);
} else {
    // Giữ nguyên ảnh đại diện cũ
    product.setHinhAnhDaiDien(sp.getHinhAnhDaiDien());
}

// Kiểm tra xem có thay đổi hình ảnh khác hay không
if (images.length > 0 && !images[0].isEmpty()) {

    // Xóa các hình ảnh khác cũ
    List<HinhAnhEntity> danhSachHinhAnh = sp.getHinhAnhs();
    if (danhSachHinhAnh != null && !danhSachHinhAnh.isEmpty()) {
        for (HinhAnhEntity hinhAnh : danhSachHinhAnh) {
            String tenTep = hinhAnh.getLink();
            xotepTinHinhAnh(tenTep);
        }
        sanPhamService.xoaHinhAnhSanPham(danhSachHinhAnh); //xóa trong csdl
    }

    List<HinhAnhEntity> hinhAnhKhacs = new ArrayList<>();
    for (MultipartFile file : images) {
        String fileName = now + file.getOriginalFilename();
        String otherImgfilePath = filePath + fileName;
        File dest = new File(otherImgfilePath);
        file.transferTo(dest);

        HinhanhEntity hinhAnhKhac = new HinhanhEntity();
        hinhAnhKhac.setLink("assets/img/sanPham/" + fileName);
        hinhAnhKhac.setSanPham(product);
        hinhAnhKhacs.add(hinhAnhKhac);

        ...
    }
    product.setHinhAnhs(hinhAnhKhacs);
    sanPhamService.updateSanPham(product);
    sanPhamService.themHinhAnhSanPham(hinhAnhKhacs);
}
}

if (!thongSo.isEmpty()) {

    // Xóa hình tskt cũ
    String anhThongSoKT = sp.getThongSoKT();
    if (anhThongSoKT != null) {
        xotepTinHinhAnh(anhThongSoKT);
    }

    String thongSoFileName = now + "-" + thongSo.getOriginalFilename();
    String thongSoFilePath = filePath + thongSoFileName;
    File thongSoFile = new File(thongSoFilePath);
    thongSo.transferTo(thongSoFile);
    product.setThongSoKT("assets/img/sanPham/" + thongSoFileName);
} else {
    // Giữ nguyên ảnh đại diện cũ
    product.setThongSoKT(sp.getThongSoKT());
}

try {
    sanPhamService.updateSanPham(product);
    model.addAttribute("successMessage", "Cập nhật sản phẩm thành công.");
} catch (Exception e) {
    model.addAttribute("errorMessage", "Có lỗi xảy ra khi cập nhật sản phẩm. " + e.getMessage());
    return "admin/editProduct";
}
return "admin/editProduct";
}

```

- Xóa sản phẩm(chỉ có thể xóa sản phẩm không nằm trong đơn hàng hoặc giỏ hàng của khách hàng, nếu không sẽ hiển thị thông báo khi xóa)

localhost:8080 cho biết

Không thể xóa sản phẩm vì đang nằm trong giỏ hàng!

OK

Hoặc

localhost:8080 cho biết

Không thể xóa sản phẩm vì đang nằm đơn hàng!

OK

```
@RequestMapping(value="admin/product", params="deleteSP", method = RequestMethod.POST)
public String deleteSanPham(HttpServletRequest request, ModelMap model) {
    String maSp = request.getParameter("maSPXoa");
    SanPhamEntity sanPham = sanPhamService.laySanPham(maSp);

    boolean coGioHangLienKet = sanPhamService.kiemTraSanPhamCoNamTrongGioHang(maSp);
    boolean coDonHangLienKet = sanPhamService.kiemTraSanPhamCoNamTrongDonHang(maSp);

    if (coGioHangLienKet) {
        model.addAttribute("errorMessageXoaSP", "Không thể xóa sản phẩm vì đang nằm trong giỏ hàng!");
        return "admin/product";
    }
    else if (coDonHangLienKet) {
        model.addAttribute("errorMessageXoaSP", "Không thể xóa sản phẩm vì đang nằm đơn hàng!");
        return "admin/product";
    }
    else {
        // Xóa sản phẩm trong cơ sở dữ liệu
        sanPhamService.xoaSanPham(sanPham);

        // Xóa hình đại diện
        String hinhAnhDaiDien = sanPham.getHinhAnhDaiDien();
        if (hinhAnhDaiDien != null) {
            xoaTepTinHinhAnh(hinhAnhDaiDien);
        }

        // Xóa hình tskt
        String anhThongSoKT = sanPham.getThongSoKT();
        if (anhThongSoKT != null) {
            xoaTepTinHinhAnh(anhThongSoKT);
        }

        // Xóa các hình ảnh khác
        List<HinhAnhEntity> danhSachHinhAnh = sanPham.getHinhAnhs();
        if (danhSachHinhAnh != null && !danhSachHinhAnh.isEmpty()) {
            for (HinhAnhEntity hinhAnh : danhSachHinhAnh) {
                String tenTep = hinhAnh.getLink();
                xoaTepTinHinhAnh(tenTep);
            }
        }
    }
}

return "redirect:/admin/product.htm";
}
```

-Thêm thương hiệu

```

@RequestMapping(value = "admin/brands/add", method = RequestMethod.GET)
public String viewAddBrand(@ModelAttribute("brandForm") ThuongHieuEntity brand, ModelMap model){
    return "admin/addBrand";
}

@RequestMapping(value = "admin/brands/add", params = "add", method = RequestMethod.POST)
public String addBrand(@ModelAttribute("brandForm") ThuongHieuEntity brand, ModelMap model) throws IOException {
    try {
        thuongHieuService.themThuongHieu(brand);
        model.addAttribute("successMessage", "Thêm thành công.");
    } catch (Exception e) {
        model.addAttribute("errorMessage", "Có lỗi xảy ra." + e.getMessage());
        return "admin/addBrand";
    }
    return "admin/addBrand";
}

```

-Sửa thương hiệu

```

@RequestMapping(value = "admin/brands//edit/{math}", method = RequestMethod.GET)
public String viewEditBrand(@PathVariable("math") String math, ModelMap model, HttpServletRequest request) {
    ThuongHieuEntity thuongHieu = thuongHieuService.layThuongHieuTheoMa(math);
    model.addAttribute("thuongHieu", thuongHieu);
    return "admin/editBrand";
}

@RequestMapping(value = "admin/brands/edit/{math}", params = "update", method = RequestMethod.POST)
public String editBrand(@PathVariable("math") String math, ModelMap model, HttpServletRequest request) throws IOException {
    String mathuongHieu=request.getParameter("mathuongHieu");
    String tenthuongHieu=request.getParameter("tenth");
    ThuongHieuEntity thuongHieu = thuongHieuService.layThuongHieuTheoMa(math);
    model.addAttribute("thuongHieu", thuongHieu);
    thuongHieu.setMath(mauthuongHieu);
    thuongHieu.setTenThuongHieu(tenthuongHieu);

    try {
        thuongHieuService.updateThuongHieu(thuongHieu);
        model.addAttribute("successMessage", "Cập nhật thành công.");
    } catch (Exception e) {
        model.addAttribute("errorMessage", "Có lỗi xảy ra khi cập nhật." + e.getMessage());
        return "admin/editBrand";
    }
    return "admin/editBrand";
}

```

-Xóa thương hiệu(chỉ xóa được những thương hiệu chưa liên kết với sản phẩm nào), nếu thương hiệu có liên kết với sản phẩm thì sẽ báo lỗi khi bấm xóa

```

@RequestMapping(value = "admin/product", params = "deleteTH", method = RequestMethod.POST)
public String deleteThuongHieu(HttpServletRequest request, ModelMap model) {
    String maTH = request.getParameter("maTHXoa");
    ThuongHieuEntity thuongHieu = thuongHieuService.layThuongHieuTheoMa(maTH);

    boolean coSanPhamLienKet = thuongHieuService.kiemTraSanPhamTheoThuongHieu(maTH);

    if (coSanPhamLienKet) {
        model.addAttribute("errorMessageTH", "Không thể xóa thương hiệu vì có sản phẩm đang liên kết!");
        return "admin/product";
    } else
        thuongHieuService.xoaThuongHieu(thuongHieu);

    return "redirect:/admin/product.htm";
}

```

12. Xem thông tin và quản lý tài khoản khách hàng (khóa, mở khóa)

Tài khoản admin, owner có thể xem thông tin và quản lý tài khoản khách hàng

| ID | HỌ TÊN | SDT | THAO TÁC |
|----|-------------------|------------|--|
| 6 | Lê Tèo | 0931234567 | <input checked="" type="button"/> Chi tiết |
| 7 | Phan Xuân Huỳnh | 0934561234 | <input checked="" type="button"/> Chi tiết |
| 8 | Nguyễn Tín | 0937123456 | <input checked="" type="button"/> Chi tiết |
| 9 | Văn Lực | 0937123456 | <input checked="" type="button"/> Chi tiết |
| 10 | Trần Văn Hùng | 0123456789 | <input checked="" type="button"/> Chi tiết |
| 19 | Trần Văn Mười Hai | 0937567567 | <input checked="" type="button"/> Chi tiết |

| ID | HỌ TÊN | SDT | THAO TÁC |
|----|-------------------|------------|--|
| 18 | Nguyễn Văn Tèo Em | 0937121212 | <input checked="" type="button"/> Chi tiết |
| 20 | Đặng Văn Nam | 0937567123 | <input checked="" type="button"/> Chi tiết |

```

@RequestMapping("on/{maNd}")
public String on(@PathVariable("maNd") int maNd, HttpServletRequest request, ModelMap model) {
    HttpSession session0 = request.getSession();
    NguoiDungEntity user = nguoiDungService.findUserById(maNd);
    user.setTrangThai(true);
    nguoiDungService.updateUser(user);

    if(user.getQuyen()==0)
        return "redirect:/admin/customerAccount.htm";

    return "redirect:/admin/adminAccount.htm";
}

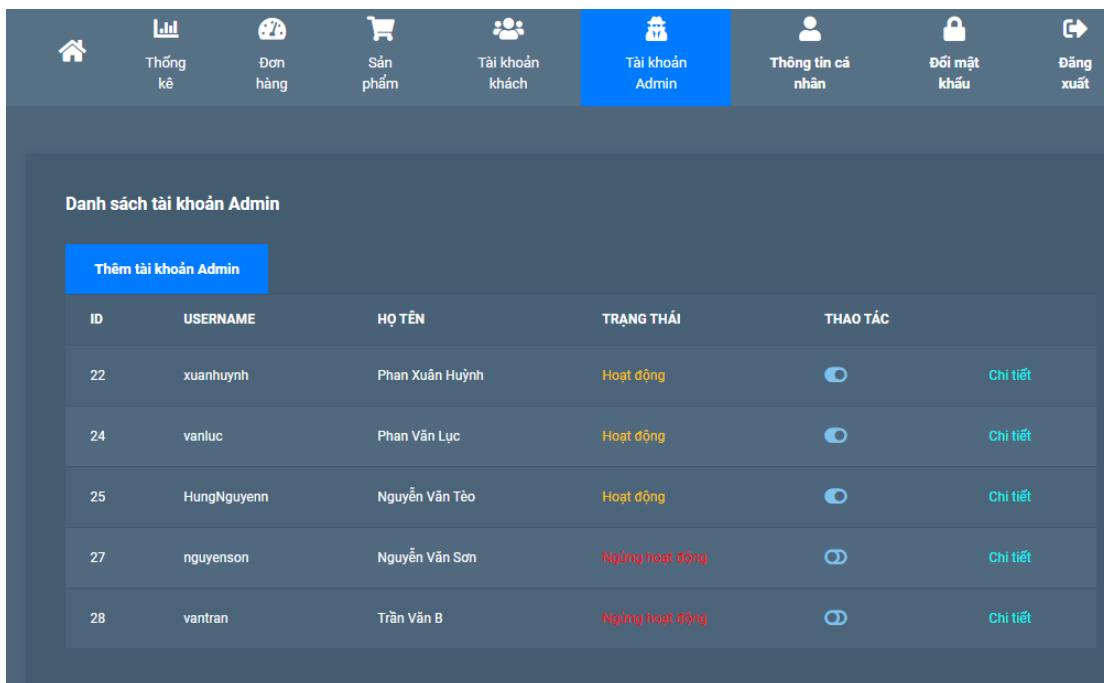
@RequestMapping("off/{maNd}")
public String off(@PathVariable("maNd") int maNd, HttpServletRequest request, ModelMap model) {
    HttpSession session0 = request.getSession();
    NguoiDungEntity user = nguoiDungService.findUserById(maNd);
    user.setTrangThai(false);
    nguoiDungService.updateUser(user);
    if(user.getQuyen()==0)
        return "redirect:/admin/customerAccount.htm";

    return "redirect:/admin/adminAccount.htm";
}

```

13. Quản lý tài khoản admin(Thêm mới, khóa, mở khóa)

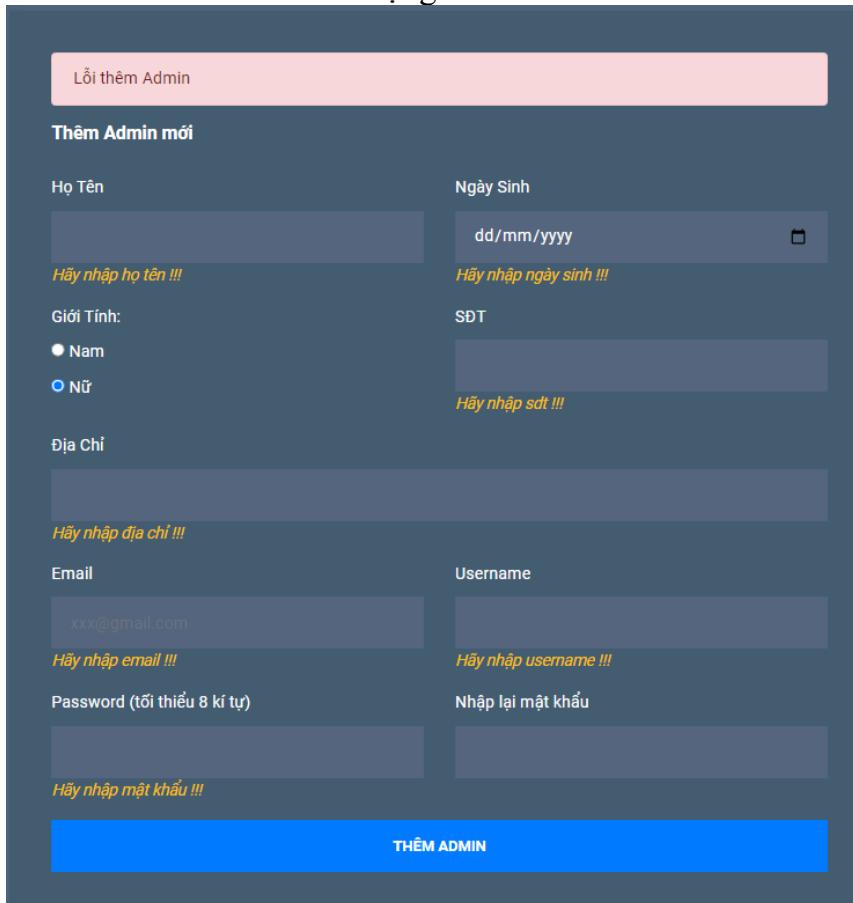
Tài khoản owner sẽ có quyền cao nhất trong trang web, có thể thêm mới, khoá, mở khoá tài khoản admin



The screenshot shows a dark-themed administrative interface. At the top, there is a navigation bar with several icons and their corresponding labels: Thống kê (Statistics), Đơn hàng (Orders), Sản phẩm (Products), Tài khoản khách (Customer Accounts), Tài khoản Admin (Admin Accounts) [highlighted in blue], Thông tin cá nhân (Personal Information), Đổi mật khẩu (Change Password), and Đăng xuất (Logout). Below the navigation bar, the main content area has a title 'Danh sách tài khoản Admin' (List of Admin Accounts). Underneath this, there is a table titled 'Thêm tài khoản Admin' (Add New Admin Account). The table has columns for ID, USERNAME, HỌ TÊN (Name), TRẠNG THÁI (Status), and THAO TÁC (Actions). The data in the table is as follows:

| ID | USERNAME | HỌ TÊN | TRẠNG THÁI | THAO TÁC |
|----|--------------|-----------------|-----------------|----------|
| 22 | xuanhuynh | Phan Xuân Huỳnh | Hoạt động | Chi tiết |
| 24 | vanluc | Phan Văn Lực | Hoạt động | Chi tiết |
| 25 | HungNguyenn | Nguyễn Văn Tèo | Hoạt động | Chi tiết |
| 27 | nguyensonson | Nguyễn Văn Sơn | Ngừng hoạt động | Chi tiết |
| 28 | vantran | Trần Văn B | Ngừng hoạt động | Chi tiết |

Form thêm mới admin sử dụng validation để bắt lỗi



The screenshot shows a modal dialog titled 'Thêm Admin mới' (Add New Admin). The form contains several input fields with validation error messages displayed below them:

- Họ Tên (Name): 'Hãy nhập họ tên !!!'
- Ngày Sinh (Birth Date): 'Hãy nhập ngày sinh !!!'
- Giới Tính (Gender):
 - Nam
 - Nữ
- SĐT (Phone Number): 'Hãy nhập sdt !!!'
- Địa Chỉ (Address): 'Hãy nhập địa chỉ !!!'
- Email: 'Hãy nhập email !!!'
- Username: 'Hãy nhập username !!!'
- Password (tối thiểu 8 kí tự) (Password): 'Hãy nhập mật khẩu !!!'
- Nhập lại mật khẩu (Re-enter Password): 'Nhập lại mật khẩu'

At the bottom of the form is a large blue button labeled 'THÊM ADMIN' (Add Admin).

```

@RequestMapping(value = "/form/addAcc", params = "add", method = RequestMethod.POST)
public String addAcc(HttpServletRequest request, ModelMap model, @ModelAttribute("admin") NguoiDungEntity admin,
                      BindingResult errors) throws ParseException {

    Boolean loi = Boolean.TRUE;
    String rePassword = request.getParameter("re-passWord");
    NguoiDungEntity userCheck;

    if (admin.getEmail().isEmpty()) {
        errors.rejectValue("email", "user", "Hãy nhập email !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getEmail().endsWith("@gmail.com")) {
        errors.rejectValue("email", "user", "Hãy nhập email đúng định dạng !!!");
        loi = Boolean.FALSE;
    }
    if (admin.getUserName().isEmpty()) {
        errors.rejectValue("userName", "user", "Hãy nhập username !!!");
        loi = Boolean.FALSE;
    } else if (admin.getUserName().contains(" ")) {
        errors.rejectValue("userName", "user", "UserName không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (admin.getUserName().length() > 30) {
        errors.rejectValue("userName", "user", "UserName không được dài quá 30 kí tự !!!");
        loi = Boolean.FALSE;
    }

    userCheck = nguoiDungService.findUserByNameAndEmail(admin.getUserName(), admin.getEmail());
    if (userCheck != null) {
        if (userCheck.getEmail().equals(admin.getEmail())) {
            errors.rejectValue("email", "user", "Email đã được sử dụng !!!");
            loi = Boolean.FALSE;
        }
        if (userCheck.getUserName().equals(admin.getUserName())) {
            errors.rejectValue("userName", "user", "Username đã được sử dụng !!!");
            loi = Boolean.FALSE;
        }
    }

    if (admin.getPassWord().isEmpty()) {
        errors.rejectValue("passWord", "user", "Hãy nhập mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (admin.getPassWord().length() < 8) {
        errors.rejectValue("passWord", "user", "Password tối thiểu 8 kí tự !!!");
        loi = Boolean.FALSE;
    } else if (admin.getPassWord().contains(" ")) {
        errors.rejectValue("passWord", "user", "Password không được chứa khoảng trắng !!!");
        loi = Boolean.FALSE;
    } else if (rePassword.isEmpty()) {
        errors.rejectValue("passWord", "user", "Hãy nhập lại mật khẩu !!!");
        loi = Boolean.FALSE;
    } else if (!rePassword.equals(admin.getPassWord())) {
        errors.rejectValue("passWord", "user", "Xác nhận mật khẩu không đúng !!!");
        loi = Boolean.FALSE;
    }
}

```

```

    if (admin.getHoTen().isEmpty()) {
        errors.rejectValue("hoTen", "user", "Hãy nhập họ tên !!!");
        loi = Boolean.FALSE;
    } else if (admin.getHoTen().length() > 50) {
        errors.rejectValue("hoTen", "user", "Họ tên quá dài !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getHoTen().matches("[\\p{L} ]+")) {
        errors.rejectValue("hoTen", "user", "Họ tên không được chứa số !!!");
        loi = Boolean.FALSE;
    }

    if (admin.getNgaySinh() == null) {
        errors.rejectValue("ngaySinh", "user", "Hãy nhập ngày sinh !!!");
        loi = Boolean.FALSE;
    } else if (!isValidAndOver18(admin.getNgaySinh())) {
        errors.rejectValue("ngaySinh", "user", "Admin cần lớn hơn 18 tuổi để tạo tài khoản !!!");
        loi = Boolean.FALSE;
    }

    if (admin.getSdt().isEmpty()) {
        errors.rejectValue("sdt", "user", "Hãy nhập sdt !!!");
        loi = Boolean.FALSE;
    } else if (!admin.getSdt().matches("[0-9]+")) {
        errors.rejectValue("sdt", "user", "SDT không hợp lệ !!!");
        loi = Boolean.FALSE;
    } else if (admin.getSdt().length() != 10 && admin.getSdt().length() != 11) {
        errors.rejectValue("sdt", "user", "SDT không hợp lệ !!!");
        loi = Boolean.FALSE;
    }

    if (admin.getDiaChi().isEmpty()) {
        errors.rejectValue("diaChi", "user", "Hãy nhập địa chỉ !!!");
        loi = Boolean.FALSE;
    }

    if (loi == Boolean.FALSE)
        { model.addAttribute("errorMessage","Lỗi thêm Admin"); }

    return "/admin/createAcc";}

String NGAYSINH = request.getParameter("ngaySinh");
java.sql.Date ns = java.sql.Date.valueOf(NGAYSINH);
admin.setPassWord(nguoIDungService.maHoaMatKhau(admin.getPassword()));
admin.setNgaySinh(ns);
admin.setHoTen(capitalizeString(admin.getHoTen()));
admin.setTrangThai(true);
admin.setQuyen(1);
nguoIDungService.addUser(admin);
model.addAttribute("successMessage","Thêm Admin mới thành công");
return "/admin/createAcc";
}

```

Tổng kết các chức năng của trang web:

-Trang web này cung cấp một loạt chức năng đáng chú ý, nhằm mang lại trải nghiệm mua sắm trực tuyến thuận tiện và hấp dẫn cho người dùng.

-Người dùng có thể tận hưởng các tính năng đăng nhập và đăng ký, giúp họ tạo tài khoản và xác minh thông tin cá nhân một cách dễ dàng. Họ có thể dễ dàng tìm kiếm và lọc sản phẩm, xem thông tin chi tiết và đánh giá của từng sản phẩm. Ngoài ra, người dùng có thể thêm sản phẩm vào giỏ hàng, tạo đơn hàng và theo dõi trạng thái đơn hàng của mình.

-Với tài khoản quản lý, có thể quản lý đơn hàng, sản phẩm và tài khoản người dùng. Họ có quyền thêm, sửa và xóa thông tin sản phẩm và duyệt đơn hàng một cách thuận tiện. Nếu cần thiết, tài khoản quản lý cũng có khả năng khóa tài khoản người dùng.

Bên cạnh đó, trang web cung cấp các tính năng thống kê và báo cáo, giúp quản lý xem thông tin doanh thu, số lượng người dùng, số lượng đơn hàng và tỉ lệ đơn hàng thành công và bị hủy. Thông tin này được hiển thị một cách trực quan và giúp quản lý đánh giá hiệu quả hoạt động của trang web.