

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA: CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN

**Xây dựng Hệ thống Quản lý Cửa hàng Cafe
"CoffeeCup" bằng Spring Boot và Kiến trúc
Microservices**

MÔN: CÔNG NGHỆ PHÁT TRIỂN PHẦN MỀM MỚI

Ngành: **KỸ THUẬT PHẦN MỀM**

Chuyên ngành **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **THẠC SĨ TRẦN QUỐC TRƯỜNG**

Sinh viên thực hiện: **NGUYỄN TRẦN HOÀNG THỊNH**

MSSV: **22140341**

Lớp học phần: **020100137002**

TP. Hồ Chí Minh, 04, 2025

Khoa: CÔNG NGHỆ THÔNG TIN

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN

TIỂU LUẬN MÔN: CÔNG NGHỆ PHÁT TRIỂN PHẦN MỀM MỚI

1. Họ và tên sinh viên: Nguyễn Trần Hoàng Thịnh

2. Tên đề tài: Xây dựng Hệ thống Quản lý Cửa hàng Cafe "CoffeeCup" bằng Spring Boot và Kiến trúc Microservices

3. Nhận xét:

a) Những kết quả đạt được:

.....
.....
.....
.....

b) Những hạn chế:

.....
.....
.....
.....

4. Điểm đánh giá (theo thang điểm 10, làm tròn đến 0.5):

Sinh viên: Nguyễn Trần Hoàng Thịnh MSSV: 22140341

Điểm số:..... Điểm chữ:

TP. HCM, ngày ... tháng 04 năm 2025

Giảng viên chấm thi

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đề tài tiểu luận: “Xây dựng Hệ thống Quản lý Cửa hàng Cafe "CoffeeCup" bằng Spring Boot và Kiến trúc Microservices” do Nguyễn Trần Hoàng Thịnh tìm hiểu và thực hiện.

Kết quả bài làm của đề tài “Xây dựng Hệ thống Quản lý Cửa hàng Cafe "CoffeeCup" bằng Spring Boot và Kiến trúc Microservices” là trung thực và không sao chép từ bất kỳ bài tập của các cá nhân khác.

TP. HCM, ... tháng 04 năm 2025.

Sinh viên cam đoan

(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1: MÔ TẢ ĐỀ TÀI THỰC HIỆN.....	1
1.1. Giới thiệu đề tài.....	1
1.2. Mục tiêu của đề tài.....	1
1.3. Phạm vi thực hiện	2
<i>1.3.1. Chức năng chính.....</i>	<i>2</i>
<i>1.3.2. Công nghệ áp dụng.....</i>	<i>3</i>
1.4. Bộ cục báo cáo	4
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	5
2.1. Phân tích yêu cầu	5
<i>2.1.1. Yêu cầu chức năng.....</i>	<i>5</i>
<i>2.1.2. Yêu cầu phi chức năng.....</i>	<i>6</i>
2.2. Thiết kế Use Case	7
<i>2.2.1. Xác định Actors.....</i>	<i>7</i>
2.2.2. Sơ đồ Use Case tổng quát.....	7
2.2.3. Mô tả chi tiết một số Use Case chính	8
2.3. Thiết kế Kiến trúc hệ thống	11
2.3.1. Sơ đồ kiến trúc Microservices (Tổng quan)	11
2.4. Thiết kế Cơ sở dữ liệu.....	12
2.4.1. Sơ đồ quan hệ thực thể (ERD).....	12
CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG	13

3.1. Tổng quan về các công nghệ	13
3.2. Backend.....	13
3.2.1. <i>Java & Spring Boot</i>	13
3.2.2. <i>Spring Data JPA & Hibernate.....</i>	14
3.2.3. <i>Spring Security.....</i>	14
3.2.4. <i>MySQL</i>	14
3.3. Frontend.....	15
3.3.1. <i>HTML5, CSS3, JavaScript.....</i>	15
3.3.2. <i>Thymeleaf & Thymeleaf Layout Dialect.....</i>	15
3.4. Microservices.....	15
3.4.1. <i>Spring Cloud Netflix Eureka.....</i>	15
3.4.2. <i>Spring Cloud Gateway.....</i>	16
3.5. Công cụ khác	16
3.5.1. <i>Maven.....</i>	16
3.5.2. <i>IntelliJ IDEA</i>	16
3.5.3. <i>XAMPP</i>	16
CHƯƠNG 4: KẾT QUẢ THỰC HIỆN	17
4.1. Cấu trúc thư mục project.....	17
4.2. Giao diện và Chức năng Public Site (coffeecup-public-site)	18
4.2.1. <i>Trang chủ (Home - /).....</i>	18
4.2.2. <i>Trang Giới thiệu (About - /about)</i>	19
4.2.3. <i>Trang Liên hệ (Contact - /contact).....</i>	20

4.2.4. Trang Menu (/menu)	21
4.2.5. Trang Blog (/blog)	22
4.2.6. Trang Thông tin Cửa hàng (Shop - /shop)	23
4.2.7. Trang Đăng nhập / Đăng ký User (/login, /register).....	24
4.2.8. Trang Danh sách Sản phẩm (/products)	25
4.2.9. Trang Chi tiết Sản phẩm (/products/{id})	26
4.3. Giao diện và Chức năng Admin/System Site (coffeecup-admin-system)	27
4.3.1. Trang Đăng nhập Admin/System (/manage/login).....	27
4.3.2. Dashboard chính (/).....	28
4.3.3. Quản lý Khách hàng (Customers - /admin/customers/**)	29
4.3.4. Quản lý Sản phẩm (Products - /admin/products/**)	31
4.3.5. Quản lý Người dùng (Users - /system/users/**)	31
4.3.6. Quản lý Vai trò (Authorities - /system/authorities/**).....	32
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	34
5.1. Kết quả đạt được	34
5.1.1. Các chức năng đã hoàn thành	34
5.1.2. Ưu điểm của hệ thống	35
5.2. Hạn chế của đề tài	35
5.3. Hướng phát triển trong tương lai	36
5.3.1. Hoàn thiện chức năng.....	36
5.3.2. Tách Database	36

5.3.3. <i>Giao tiếp Liên Service</i>	36
5.3.4. <i>Nâng cao Microservices</i>	37
5.3.5. <i>Frontend Framework</i>	37
5.3.6. <i>Deployment & CI/CD</i>	37
TÀI LIỆU THAM KHẢO	38

MỤC LỤC HÌNH ẢNH

Chương 2: Hình 2.1: Sơ đồ Use Case tổng quát.....	7
Chương 2: Hình 2.2: Sơ đồ Use Case đăng nhập tổng quát	8
Chương 2: Hình 2.3: Sơ đồ Use Case quản lý Sản phẩm (Admin)	9
Chương 2: Hình 2.4: Sơ đồ Use Case quản lý Người dùng (System)	10
Chương 2: Hình 2.4: . Sơ đồ kiến trúc Microservices (Tổng quan)	11
Chương 2: Hình 2.5: . Sơ đồ quan hệ thực thể (ERD).....	12
Chương 4: Hình 4.1: Cấu trúc thư mục project	17
Chương 4: Hình 4.2: Trang chủ	19
Chương 4: Hình 4.3: Trang about.....	20
Chương 4: Hình 4.4: Trang contact	21
Chương 4: Hình 4.5: Trang menu	22
Chương 4: Hình 4.6: Trang blog.....	23
Chương 4: Hình 4.7: Trang shop	24
Chương 4: Hình 4.8: Trang đăng nhập/đăng ký	25
Chương 4: Hình 4.9: Trang Danh sách Sản phẩm	26
Chương 4: Hình 4.10: Trang Chi tiết Sản phẩm.....	27
Chương 4: Hình 4.11: Trang Đăng nhập Admin/System	28
Chương 4: Hình 4.12: Trang Dashboard chính	29
Chương 4: Hình 4.13: Trang Quản lý Khách	30
Chương 4: Hình 4.14: Trang Quản lý Sản phẩm.....	31
Chương 4: Hình 4.15: Trang quản lý người dùng	32
Chương 4: Hình 4.16: Trang quản lý vai trò	33

CHƯƠNG 1: MÔ TẢ ĐỀ TÀI THỰC HIỆN

1.1. Giới thiệu đề tài

Trong bối cảnh thương mại điện tử ngày càng phát triển, các cửa hàng kinh doanh truyền thống như cửa hàng cà phê cần thích nghi với xu hướng công nghệ để đáp ứng nhu cầu của khách hàng và tối ưu hóa quản lý nội bộ. Đề tài " Xây dựng Hệ thống Quản lý Cửa hàng Cafe "CoffeeCup" bằng Spring Boot và Kiến trúc Microservices " được thực hiện nhằm phát triển một trang web cho phép khách hàng xem thông tin và sản phẩm của cửa hàng cà phê CoffeeCup, đồng thời cung cấp một hệ thống quản trị nội bộ hiệu quả dành cho quản trị viên và quản lý hệ thống.

Một giao diện công cộng (Public Site) cho phép khách hàng truy cập các trang thông tin tĩnh, xem danh sách sản phẩm, chi tiết sản phẩm từ cơ sở dữ liệu, cũng như đăng ký và đăng nhập tài khoản.

Một hệ thống quản trị (Admin/System Site) hỗ trợ quản lý dữ liệu cốt lõi như khách hàng, sản phẩm, người dùng và vai trò, với các chức năng tạo, đọc, cập nhật và xóa (CRUD).

Điểm nổi bật của đề tài là việc áp dụng kiến trúc Microservices để tách biệt rõ ràng giữa giao diện người dùng (Public Site) và phần quản trị (Admin/System Site). Kiến trúc này không chỉ đảm bảo tính độc lập giữa các thành phần mà còn tăng cường khả năng mở rộng, bảo trì và triển khai hệ thống trong tương lai. Đề tài kết hợp các công nghệ hiện đại như Spring Boot, Spring Cloud và MySQL để xây dựng một giải pháp toàn diện, đáp ứng yêu cầu thực tiễn của cửa hàng CoffeeCup.

1.2. Mục tiêu của đề tài

Đề tài hướng đến việc đạt được các mục tiêu cụ thể sau:

- Xây dựng giao diện người dùng (Frontend - Public Site): Phát triển giao diện công cộng dựa trên template có sẵn, cung cấp trải nghiệm thân thiện cho khách hàng khi truy cập các trang tĩnh và xem thông tin sản phẩm.

- Xây dựng hệ thống quản trị (Backend - Admin/System Site): Triển khai hệ thống quản lý dữ liệu cốt lõi, hỗ trợ quản trị viên và quản lý hệ thống thực hiện các thao tác CRUD trên các đối tượng chính.
- Triển khai chức năng CRUD: Đảm bảo các đối tượng chính như Customers (Khách hàng), Products (Sản phẩm), Users (Người dùng) và Authorities (Vai trò) đều có đầy đủ chức năng tạo, đọc, cập nhật và xóa, đi kèm với validation dữ liệu đầu vào để đảm bảo tính chính xác và an toàn.
- Áp dụng validation dữ liệu đầu vào: Kiểm tra và xác thực dữ liệu ở cả phía client và server để nâng cao chất lượng dữ liệu nhập vào hệ thống.
- Triển khai phân quyền truy cập dựa trên vai trò: Thiết lập cơ chế phân quyền linh hoạt với các vai trò User (Người dùng công cộng), Admin (Quản trị viên) và System (Quản lý hệ thống), đảm bảo mỗi nhóm chỉ truy cập được các chức năng phù hợp.
- Xây dựng hệ thống theo mô hình Microservices cơ bản: Ứng dụng Service Discovery (Eureka) để quản lý các dịch vụ và API Gateway làm điểm vào duy nhất, tạo nền tảng cho kiến trúc phân tán.

1.3. Phạm vi thực hiện

1.3.1. Chức năng chính

Public Site (Project 1)

- Hiển thị các trang tĩnh: Bao gồm Home (Trang chủ), About (Giới thiệu), Menu (Thực đơn), Blog (Bài viết), Shop Info (Thông tin cửa hàng) và Contact (Liên hệ), dựa trên template cung cấp.
- Hiển thị danh sách sản phẩm từ database: Truy xuất và hiển thị danh sách sản phẩm được lưu trữ trong cơ sở dữ liệu.
- Hiển thị chi tiết sản phẩm từ database: Cung cấp trang chi tiết sản phẩm với thông tin đầy đủ và nút "Mua ngay" (chức năng demo, chưa tích hợp thanh toán).

- Đăng ký tài khoản User: Cho phép khách hàng tạo tài khoản mới với các thông tin cơ bản.
- Đăng nhập/Đăng xuất tài khoản User: Hỗ trợ đăng nhập bằng tài khoản đã đăng ký và đăng xuất khi cần.

Admin/System Site (Project 2)

- Đăng nhập tài khoản Admin/System: Xác thực tài khoản quản trị viên và quản lý hệ thống với Spring Security.
- (Admin) Quản lý Khách hàng: Thực hiện CRUD (tạo, đọc, cập nhật, xóa) trên danh sách khách hàng, kèm validation dữ liệu đầu vào.
- (Admin) Quản lý Sản phẩm: Thực hiện CRUD trên danh sách sản phẩm, hỗ trợ upload, hiển thị và xóa ảnh sản phẩm, đi kèm validation dữ liệu.
- (System) Quản lý Người dùng: Thực hiện CRUD trên danh sách người dùng, kèm validation, gán vai trò (Roles), bật/tắt trạng thái tài khoản (Enable/Disable) trực tiếp và yêu cầu nhập mật khẩu cũ khi thay đổi mật khẩu.
- (System) Quản lý Vai trò (Authorities): Xem danh sách vai trò, thêm mới vai trò và xóa vai trò (với điều kiện không ảnh hưởng đến người dùng hiện tại).

Microservices Infrastructure

- Eureka Server: Đóng vai trò khám phá dịch vụ (Service Discovery), quản lý và đăng ký các microservices trong hệ thống.
- API Gateway: Định tuyến các request từ client đến các dịch vụ tương ứng, hoạt động như điểm vào duy nhất của hệ thống.

1.3.2. Công nghệ áp dụng

- Java 21, Spring Boot 3.3.10, Spring Cloud 2023.0.3: Nền tảng chính để phát triển backend và kiến trúc Microservices.

- Spring Data JPA, Hibernate, MySQL (qua XAMPP): Công nghệ quản lý và truy xuất cơ sở dữ liệu quan hệ.
- Spring Security 6: Đảm bảo xác thực, phân quyền và bảo mật hệ thống.
- Thymeleaf, Thymeleaf Layout Dialect, Thymeleaf Security Extras: Template engine phía server để xây dựng giao diện động và tích hợp phân quyền.
- HTML5, CSS3, JavaScript: Sử dụng cho giao diện frontend và validation client-side.
- Spring Cloud Netflix Eureka Server & Client: Hỗ trợ khám phá dịch vụ trong kiến trúc Microservices.
- Spring Cloud Gateway (WebFlux hoặc MVC): Định tuyến request, tùy thuộc vào lựa chọn cuối cùng của bạn (WebFlux cho bất đồng bộ hoặc MVC cho đồng bộ).
- Maven: Công cụ quản lý thư viện và xây dựng dự án.
- IntelliJ IDEA: Môi trường phát triển tích hợp (IDE) chính để viết mã và triển khai hệ thống.

1.4. Bố cục báo cáo

Báo cáo được tổ chức thành 5 chương với nội dung như sau:

- Chương 1: Mô tả đề tài thực hiện: Giới thiệu đề tài, mục tiêu, phạm vi thực hiện và bố cục báo cáo.
- Chương 2: Phân tích và thiết kế hệ thống: Phân tích yêu cầu, thiết kế Use Case, kiến trúc Microservices và cơ sở dữ liệu.
- Chương 3: Công nghệ sử dụng: Tổng quan và chi tiết về các công nghệ áp dụng trong hệ thống.
- Chương 4: Kết quả thực hiện: Trình bày cấu trúc dự án, giao diện, chức năng của Public Site, Admin/System Site, Eureka Server và cơ sở dữ liệu.
- Chương 5: Kết luận và hướng phát triển: Đánh giá kết quả, nêu hạn chế và đề xuất hướng phát triển trong tương lai.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu

2.1.1. Yêu cầu chức năng

Hệ thống CoffeeCup được thiết kế với các chức năng chính, chia theo hai thành phần Public Site và Admin/System Site, cùng hạ tầng Microservices:

Public Site

- Hiển thị các trang tĩnh: Cung cấp các trang Home, About, Menu, Blog, Shop Info và Contact dựa trên template có sẵn, không yêu cầu truy xuất dữ liệu động.
- Hiển thị danh sách sản phẩm: Truy xuất danh sách sản phẩm từ cơ sở dữ liệu và hiển thị dưới dạng danh sách có thể xem được.
- Hiển thị chi tiết sản phẩm: Cho phép xem thông tin chi tiết của một sản phẩm cụ thể (tên, giá, mô tả, ảnh) kèm nút "Mua ngay" (chức năng demo).
- Đăng ký tài khoản User: Hỗ trợ khách hàng tạo tài khoản mới với thông tin cơ bản (username, password, email).
- Đăng nhập/Đăng xuất tài khoản User: Xác thực tài khoản người dùng và cho phép đăng xuất khi cần.

Admin/System Site

- Đăng nhập tài khoản Admin/System: Xác thực tài khoản quản trị viên và quản lý hệ thống dựa trên vai trò (Admin hoặc System).

Quản lý Khách hàng (Admin):

- Xem danh sách khách hàng.
- Thêm, sửa, xóa thông tin khách hàng với validation
- Quản lý Sản phẩm (Admin):
- Xem danh sách sản phẩm.
- Thêm, sửa, xóa sản phẩm, hỗ trợ upload ảnh, hiển thị ảnh và xóa ảnh, kèm validation

Quản lý Người dùng (System):

- Xem danh sách người dùng.
- Thêm, sửa, xóa người dùng với validation (ví dụ: username duy nhất).
- Gán vai trò (Roles) cho người dùng.
- Bật/tắt trạng thái tài khoản (Enable/Disable) trực tiếp.
- Yêu cầu nhập mật khẩu cũ khi thay đổi mật khẩu.

Quản lý Vai trò (Authorities) (System):

- Xem danh sách vai trò.
- Thêm vai trò mới.
- Xóa vai trò với điều kiện không ảnh hưởng đến người dùng hiện tại.

Microservices Infrastructure

- Eureka Server: Đăng ký và quản lý các dịch vụ trong hệ thống (Public Site, Admin/System Site, API Gateway).
- API Gateway: Định tuyến các yêu cầu từ client đến các dịch vụ tương ứng, đóng vai trò điểm vào duy nhất.

2.1.2. Yêu cầu phi chức năng

- Dễ bảo trì: Kiến trúc Microservices cho phép phát triển và bảo trì từng thành phần độc lập.
- Khả năng mở rộng: Hệ thống có thể dễ dàng thêm dịch vụ mới hoặc nâng cấp mà không ảnh hưởng đến các phần khác.
- Giao diện thân thiện: Public Site cung cấp trải nghiệm người dùng trực quan, dễ sử dụng dựa trên template có sẵn.
- Bảo mật: Sử dụng Spring Security để đảm bảo xác thực và phân quyền an toàn.
- Hiệu năng: Hệ thống đáp ứng nhanh các yêu cầu cơ bản trong phạm vi hiện tại.

2.2. Thiết kế Use Case

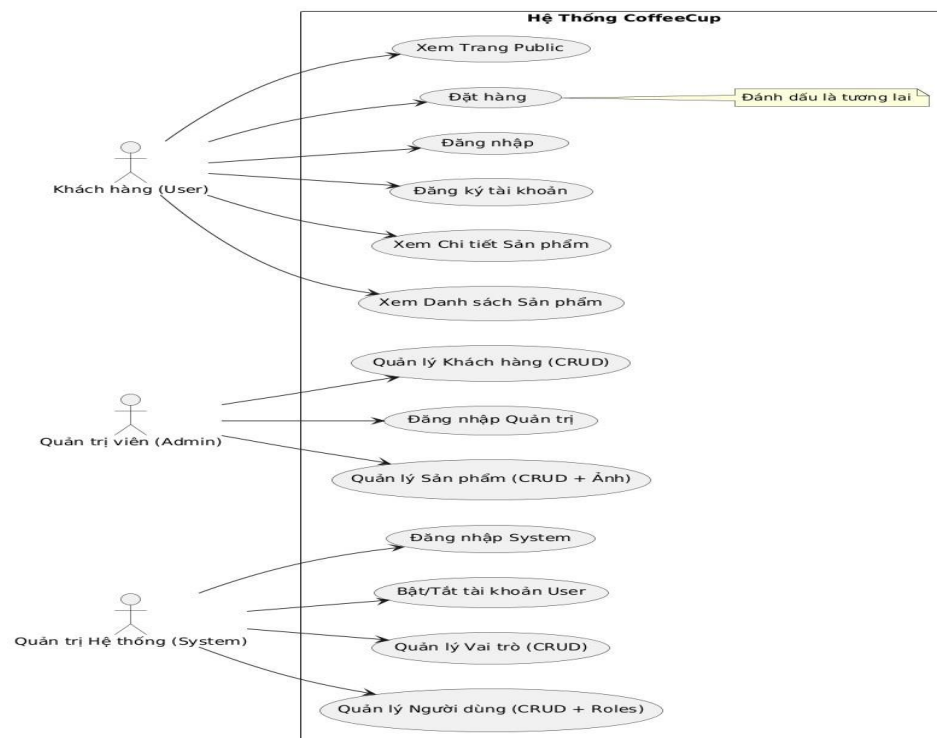
2.2.1. Xác định Actors

Hệ thống có 3 Actors chính:

- Khách hàng (User): Người dùng công cộng truy cập Public Site để xem thông tin, sản phẩm và quản lý tài khoản cá nhân.
- Quản trị viên (Admin): Người quản lý nội dung, thực hiện các thao tác CRUD trên khách hàng và sản phẩm trong Admin/System Site.
- Quản trị Hệ thống (System): Người quản lý người dùng và vai trò, đảm bảo cấu hình hệ thống hoạt động đúng.

2.2.2. Sơ đồ Use Case tổng quát

Sơ đồ Use Case tổng quát được thiết kế để thể hiện các tương tác chính giữa Actors và hệ thống.



Chương 2: Hình 2.1: Sơ đồ Use Case tổng quát

Khách hàng (User):

- Xem trang tĩnh (Home, About, Menu, Blog, Shop Info, Contact).
- Xem danh sách sản phẩm.
- Xem chi tiết sản phẩm.
- Đăng ký tài khoản.
- Đăng nhập/Đăng xuất.

Quản trị viên (Admin):

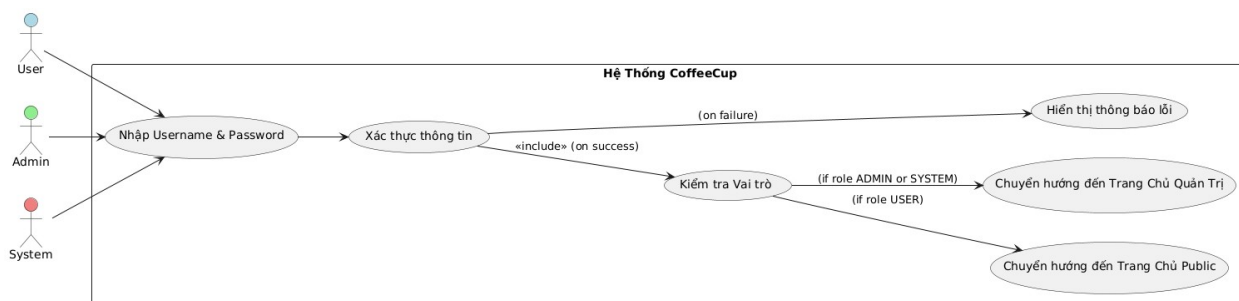
- Đăng nhập.
- Quản lý khách hàng (CRUD).
- Quản lý sản phẩm (CRUD + Upload ảnh).

Quản trị Hệ thống (System):

- Đăng nhập.
- Quản lý người dùng (CRUD + Gán Roles + Enable/Disable).
- Quản lý vai trò (Xem, Thêm, Xóa).

2.2.3. Mô tả chi tiết một số Use Case chính

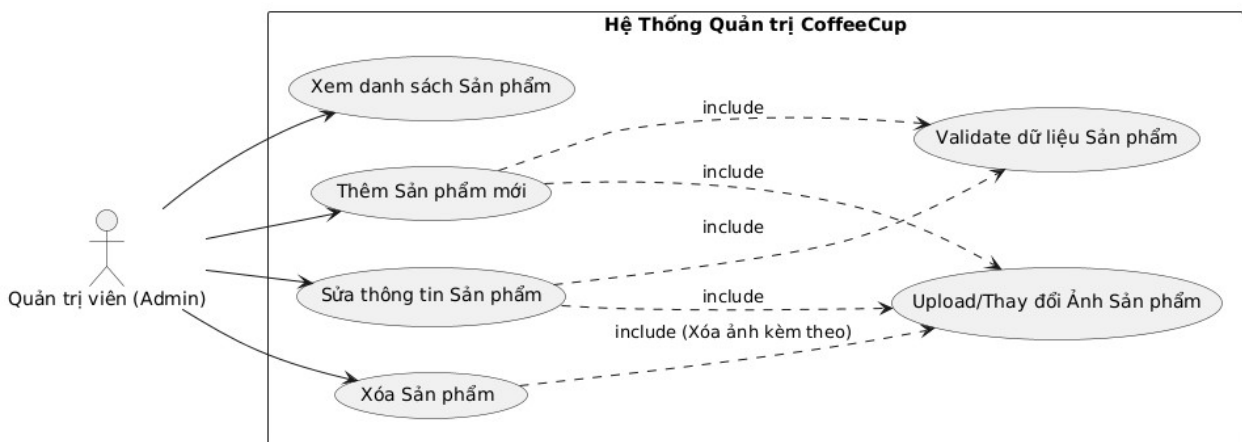
Use Case: Đăng nhập tổng quát (Cho cả User và Admin/System)



Chương 2: Hình 2.2: Sơ đồ Use Case đăng nhập tổng quát

- Actor: Khách hàng (User), Quản trị viên (Admin), Quản trị Hệ thống (System).
- Mục đích: Xác thực danh tính để truy cập các chức năng phù hợp với vai trò.
- Điều kiện tiên quyết: Người dùng đã có tài khoản trong hệ thống.
- Luồng chính:
 1. Người dùng truy cập trang đăng nhập.
 2. Nhập username và password.
 3. Hệ thống kiểm tra thông tin (validation client-side và server-side).
 4. Nếu đúng, chuyển hướng đến trang chính (Public Site với User, Dashboard với Admin/System).
- Nếu sai, hiển thị thông báo lỗi.

Use Case: Quản lý Sản phẩm (Admin)

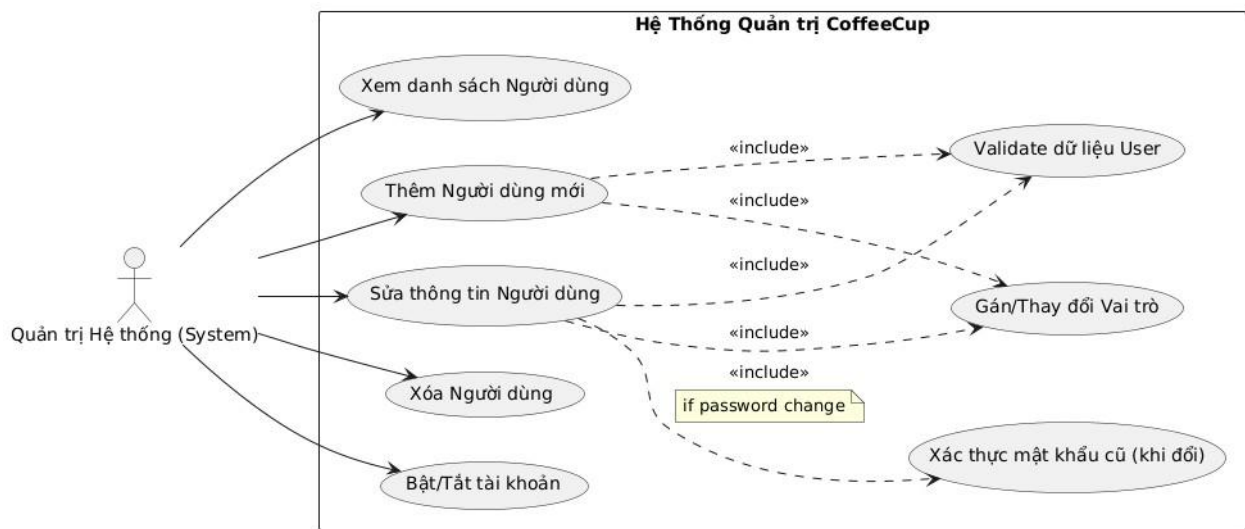


Chương 2: Hình 2.3: Sơ đồ Use Case quản lý Sản phẩm (Admin)

- Actor: Quản trị viên (Admin).
- Mục đích: Thêm, sửa, xóa và xem danh sách sản phẩm.
- Điều kiện tiên quyết: Admin đã đăng nhập.
- Luồng chính:
 1. Admin truy cập trang quản lý sản phẩm.
 2. Xem danh sách sản phẩm hiện có.

3. Chọn "Thêm mới": Nhập thông tin (tên, giá, mô tả), upload ảnh, gửi form.
 4. Hệ thống validate dữ liệu, lưu vào database và hiển thị sản phẩm mới.
 5. Chọn "Sửa": Cập nhật thông tin, thay ảnh nếu cần, lưu lại.
 6. Chọn "Xóa": Xóa sản phẩm và ảnh liên quan khỏi hệ thống.
- Ngoại lệ: Dữ liệu không hợp lệ (giá âm, tên trống), lỗi upload ảnh.

Use Case: Quản lý Người dùng (System)



Chương 2: Hình 2.4: Sơ đồ Use Case quản lý Người dùng (System)

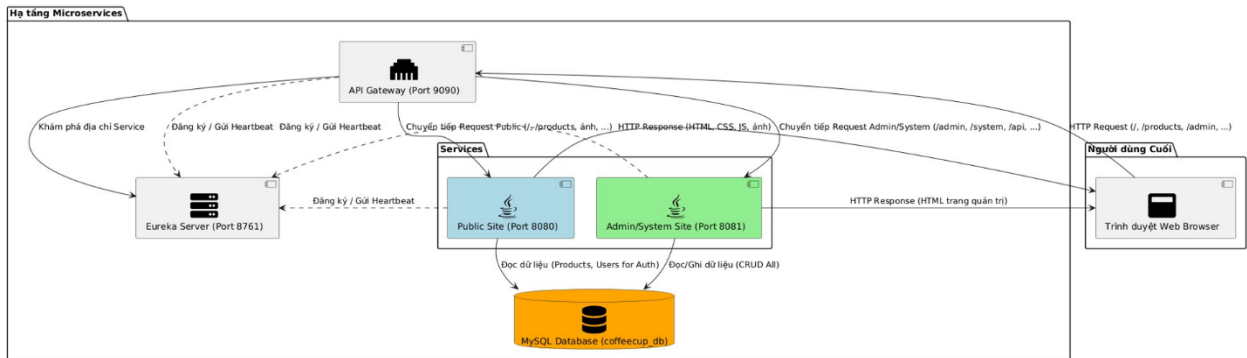
- Actor: Quản trị Hệ thống (System).
- Mục đích: Quản lý thông tin người dùng và vai trò.
- Điều kiện tiên quyết: System đã đăng nhập.
- Luồng chính:
 1. System truy cập trang quản lý người dùng.
 2. Xem danh sách người dùng hiện có.
 3. Chọn "Thêm mới": Nhập thông tin (username, password, email), gán vai trò, lưu.
 4. Chọn "Sửa": Cập nhật thông tin, thay đổi vai trò, bật/tắt trạng thái, yêu cầu mật khẩu cũ nếu đổi mật khẩu.

5. Chọn "Xóa": Xóa người dùng khỏi hệ thống.

- Ngoại lệ: Username trùng, mật khẩu cũ không đúng, dữ liệu không hợp lệ.

2.3. Thiết kế Kiến trúc hệ thống

2.3.1. Sơ đồ kiến trúc Microservices (Tổng quan)



Chương 2: Hình 2.4: . Sơ đồ kiến trúc Microservices (Tổng quan)

Hệ thống sử dụng kiến trúc Microservices với các thành phần chính:

- API Gateway: Điểm vào duy nhất, định tuyến yêu cầu từ client đến các dịch vụ (Public Site, Admin/System Site).
- Eureka Server: Quản lý và khám phá dịch vụ, đảm bảo các microservices đăng ký và hoạt động đồng bộ.
- Public Service: Xử lý yêu cầu từ Public Site (trang tĩnh, sản phẩm, đăng ký/đăng nhập).
- Admin Service: Xử lý yêu cầu từ Admin/System Site (quản lý khách hàng, sản phẩm, người dùng, vai trò).
- Database (MySQL): Lưu trữ dữ liệu chung cho các dịch vụ (tạm thời chưa tách riêng).

Luồng yêu cầu:

1. Client gửi yêu cầu đến API Gateway.

2. API Gateway kiểm tra và định tuyến đến dịch vụ tương ứng (Public Service hoặc Admin Service).
3. Dịch vụ truy xuất dữ liệu từ MySQL (nếu cần) và trả kết quả ngược lại qua Gateway.
4. Eureka Server theo dõi trạng thái của các dịch vụ để đảm bảo tính sẵn sàng.

2.4. Thiết kế Cơ sở dữ liệu

2.4.1. Sơ đồ quan hệ thực thể (ERD)

Sơ đồ ERD bao gồm các bảng chính:

- users: Lưu thông tin người dùng (username, password, email, enabled).
- authorities: Lưu danh sách vai trò (authority).
- users_authorities: Bảng trung gian liên kết users và authorities.
- customers: Lưu thông tin khách hàng (id, name, email, phone).
- products: Lưu thông tin sản phẩm (id, name, price, description, image).



Chương 2: Hình 2.5: . Sơ đồ quan hệ thực thể (ERD)

CHƯƠNG 3: CÔNG NGHỆ SỬ DỤNG

3.1. Tổng quan về các công nghệ

Hệ thống CoffeeCup được xây dựng dựa trên nền tảng công nghệ hiện đại, tập trung vào việc phát triển một hệ thống phân tán theo kiến trúc Microservices. Các công nghệ được chia thành ba nhóm chính: Backend, Frontend và Microservices, kết hợp với các công cụ hỗ trợ phát triển và triển khai. Cụ thể:

- Backend: Sử dụng Java với Spring Boot làm nền tảng chính, kết hợp Spring Data JPA, Hibernate, Spring Security và MySQL để xử lý logic nghiệp vụ, quản lý dữ liệu và bảo mật.
- Frontend: Sử dụng HTML5, CSS3, JavaScript và Thymeleaf để xây dựng giao diện người dùng động và thân thiện.
- Microservices: Ứng dụng Spring Cloud Netflix Eureka và Spring Cloud Gateway để quản lý dịch vụ và định tuyến yêu cầu trong kiến trúc phân tán.
- Công cụ khác: Maven, IntelliJ IDEA và XAMPP hỗ trợ quá trình phát triển, xây dựng và quản lý cơ sở dữ liệu.

Các công nghệ này được lựa chọn dựa trên tính phổ biến, khả năng tích hợp cao và sự phù hợp với yêu cầu của hệ thống CoffeeCup.

3.2. Backend

3.2.1. Java & Spring Boot

Vai trò: Java 21 là ngôn ngữ lập trình chính, kết hợp với Spring Boot 3.3.10 để phát triển backend của hệ thống. Spring Boot cung cấp một framework mạnh mẽ, giúp đơn giản hóa việc cấu hình và triển khai ứng dụng.

Lý do chọn: Java là ngôn ngữ ổn định, có cộng đồng hỗ trợ lớn, phù hợp cho các hệ thống doanh nghiệp. Spring Boot giảm thiểu thời gian phát triển nhờ các tính năng như dependency injection, auto-configuration và embedded server (Tomcat). Trong hệ thống

CoffeeCup, Spring Boot được dùng để xây dựng các microservices như Public Service và Admin Service.

3.2.2. *Spring Data JPA & Hibernate*

Vai trò: Spring Data JPA là một module của Spring, cung cấp cách tiếp cận đơn giản để tương tác với cơ sở dữ liệu quan hệ thông qua các repository. Hibernate là ORM (Object-Relational Mapping) bên dưới, ánh xạ các đối tượng Java (entities) với bảng trong database.

Ứng dụng: Trong CoffeeCup, Spring Data JPA được dùng để thực hiện các thao tác CRUD trên các bảng users, customers, products, authorities. Hibernate tự động tạo và quản lý các truy vấn SQL, giảm công sức viết mã thủ công.

3.2.3. *Spring Security*

Vai trò: Spring Security 6 cung cấp cơ chế bảo mật cho hệ thống, bao gồm xác thực (authentication) và phân quyền (authorization).

- SecurityFilterChain: Định nghĩa các quy tắc bảo mật, như yêu cầu đăng nhập cho các URL quản trị (/admin/, /system/).
- UserDetailsService: Tải thông tin người dùng từ bảng users và gán vai trò từ users_authorities.
- PasswordEncoder: Mã hóa mật khẩu để lưu trữ an toàn.

Trong CoffeeCup, Spring Security đảm bảo phân quyền giữa User, Admin và System, đồng thời yêu cầu mật khẩu cũ khi thay đổi mật khẩu người dùng.

3.2.4. *MySQL*

Vai trò: MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được sử dụng để lưu trữ dữ liệu của hệ thống, bao gồm thông tin người dùng, khách hàng, sản phẩm và vai trò.

Ứng dụng: MySQL được quản lý thông qua XAMPP, cung cấp giao diện phpMyAdmin để tạo schema và kiểm tra dữ liệu. Đây là giải pháp nhẹ, phù hợp cho giai đoạn phát triển ban đầu của CoffeeCup.

3.3. Frontend

3.3.1. HTML5, CSS3, JavaScript

Vai trò: HTML5 cung cấp cấu trúc cho các trang web, CSS3 định dạng giao diện, và JavaScript xử lý các tương tác phía client như validation dữ liệu đầu vào.

Ứng dụng: Trong Public Site, bộ ba này được dùng để hiển thị các trang tĩnh (Home, About, Menu, v.v.) dựa trên template có sẵn. JavaScript hỗ trợ kiểm tra dữ liệu form (ví dụ: email hợp lệ) trước khi gửi đến server, nâng cao trải nghiệm người dùng.

3.3.2. Thymeleaf & Thymeleaf Layout Dialect

Vai trò: Thymeleaf là template engine phía server, cho phép tích hợp dữ liệu động từ backend vào giao diện HTML. Thymeleaf Layout Dialect hỗ trợ tạo layout chung, tái sử dụng code giao diện. Thymeleaf Security Extras tích hợp với Spring Security để hiển thị nội dung dựa trên vai trò.

- Trong Public Site, Thymeleaf hiển thị danh sách và chi tiết sản phẩm từ database.
- Trong Admin/System Site, Thymeleaf xây dựng các form CRUD và dashboard, với layout chung cho giao diện quản trị.
- Ví dụ: Chỉ hiển thị menu "Quản lý người dùng" cho vai trò System bằng cú pháp `sec:authorize="hasRole('SYSTEM')"`.

3.4. Microservices

3.4.1. Spring Cloud Netflix Eureka

Vai trò: Spring Cloud Netflix Eureka (phiên bản 2023.0.3) cung cấp cơ chế Service Discovery, cho phép các microservices tự động đăng ký và khám phá lẫn nhau. Eureka Server là trung tâm quản lý, trong khi Eureka Client được tích hợp vào từng dịch vụ.

Ứng dụng: Trong CoffeeCup, Eureka Server chạy độc lập, theo dõi trạng thái của API Gateway, Public Service và Admin Service. Điều này đảm bảo hệ thống hoạt động linh hoạt và dễ mở rộng.

3.4.2. Spring Cloud Gateway

Vai trò: Spring Cloud Gateway đóng vai trò API Gateway, định tuyến các yêu cầu từ client đến các microservices phù hợp.

- Định tuyến /public/* đến Public Service và /admin/* hoặc /system/* đến Admin Service.
- Là điểm vào duy nhất, giúp quản lý tập trung và tăng tính bảo mật.

3.5. Công cụ khác

3.5.1. Maven

Vai trò: Maven là công cụ quản lý thư viện và xây dựng dự án, tự động tải các dependency (Spring Boot, Spring Cloud, v.v.) từ kho lưu trữ trung tâm.

Ứng dụng: CoffeeCup sử dụng file pom.xml để cấu hình các phiên bản cụ thể đảm bảo tính nhất quán trong quá trình phát triển.

3.5.2. IntelliJ IDEA

Vai trò: IntelliJ IDEA là IDE chính, cung cấp môi trường phát triển tích hợp với các tính năng như gợi ý mã, debug và quản lý microservices.

Ứng dụng: Hỗ trợ chạy từng microservice (Eureka Server, Gateway, Public, Admin) trong quá trình phát triển CoffeeCup.

3.5.3. XAMPP

Vai trò: XAMPP cung cấp môi trường localhost để chạy MySQL và phpMyAdmin, giúp quản lý cơ sở dữ liệu dễ dàng.

Ứng dụng: Dùng để tạo schema, thêm dữ liệu mẫu và kiểm tra kết nối từ Spring Data JPA trong CoffeeCup.

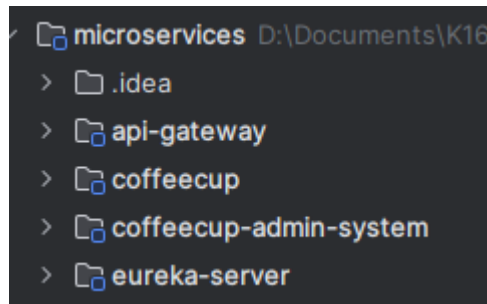
CHƯƠNG 4: KẾT QUẢ THỰC HIỆN

4.1. Cấu trúc thư mục project

Hệ thống CoffeeCup được tổ chức thành một dự án Microservices gồm bốn project con chính, mỗi project phục vụ một mục đích riêng biệt. Dưới đây là mô tả cấu trúc tổng quan:

Thư mục Microservices tổng quan:

- Chứa bốn project con: coffeecup-eureka-server, coffeecup-api-gateway, coffeecup-public-site, và coffeecup-admin-system.
- Mỗi project là một module Maven độc lập, được quản lý trong một thư mục cha (parent project).



Chương 4: Hình 4.1: Cấu trúc thư mục project

Cấu trúc project Eureka Server (coffeecup-eureka-server):

- src/main/java: Chứa mã nguồn chính, bao gồm class khởi động ứng dụng với annotation @EnableEurekaServer.
- src/main/resources: Chứa file cấu hình application.yml (cổng 8761, tắt client behavior).

Cấu trúc project Public Site (coffeecup-public-site):

- src/main/java: Bao gồm các package như controller, service, repository, entity để xử lý logic nghiệp vụ và truy xuất dữ liệu.

- src/main/resources/templates: Chứa các file Thymeleaf (.html) cho giao diện.
- src/main/resources/static: Chứa tài nguyên tĩnh (CSS, JS, hình ảnh).

Cấu trúc project Admin/System Site (coffeecup-admin-system):

- Tương tự Public Site, nhưng bổ sung các package quản lý khách hàng, sản phẩm, người dùng và vai trò.
- src/main/resources/templates: Chứa giao diện quản trị với layout riêng.

Cấu trúc project API Gateway (coffeecup-api-gateway):

- src/main/java: Chứa class cấu hình Gateway với các route đến Public Site và Admin/System Site.
- src/main/resources: File application.yml định nghĩa các tuyến đường (routes).

4.2. Giao diện và Chức năng Public Site (coffeecup-public-site)

Public Site (coffeecup-public-site) là thành phần giao diện người dùng chính của hệ thống CoffeeCup, được thiết kế để phục vụ khách hàng công cộng. Dưới đây là mô tả chi tiết các trang đã triển khai, bao gồm mục đích, nội dung chính và kết quả thực hiện.

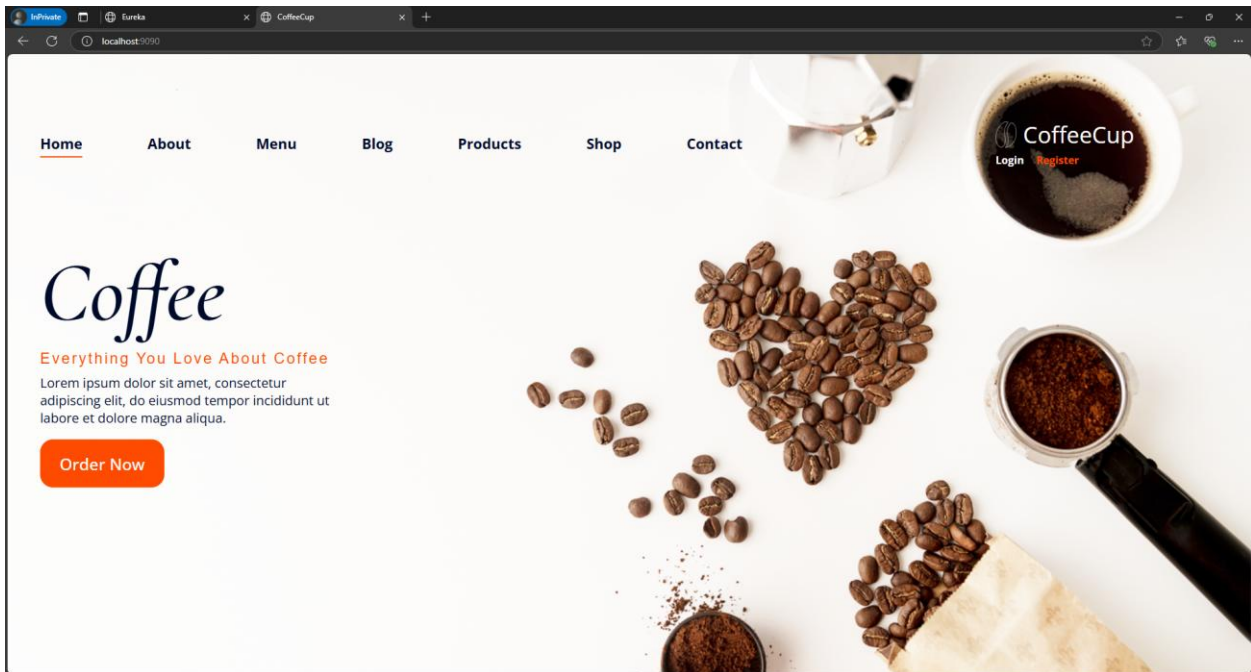
4.2.1. Trang chủ (Home - /)

Mục đích: Là trang đầu tiên khách hàng truy cập khi vào hệ thống, giới thiệu thương hiệu CoffeeCup và tạo ấn tượng ban đầu tích cực.

Nội dung chính:

- Hero Section: Một khu vực lớn với ảnh nền hấp dẫn liên quan đến cà phê, tiêu đề nổi bật như "CoffeeCup" và slogan ngắn gọn "Everything You Love About Coffee".
- Nút Call to Action (CTA): Nút "Order Now" trở đến trang danh sách sản phẩm (/products).
- Menu điều hướng: Liên kết đến các trang khác (About, Contact, Menu, Blog, Shop, Products).

- Trạng thái đăng nhập: Hiển thị "Login/Register" nếu chưa đăng nhập, hoặc "Hi [Username]/Logout" nếu đã đăng nhập.
- Kết quả thực hiện: Trang chủ được xây dựng dựa trên template, tích hợp Thymeleaf để hiển thị trạng thái đăng nhập động. Giao diện trực quan, thân thiện, thu hút người dùng ngay từ lần truy cập đầu tiên.



Chương 4: Hình 4.2: Trang chủ

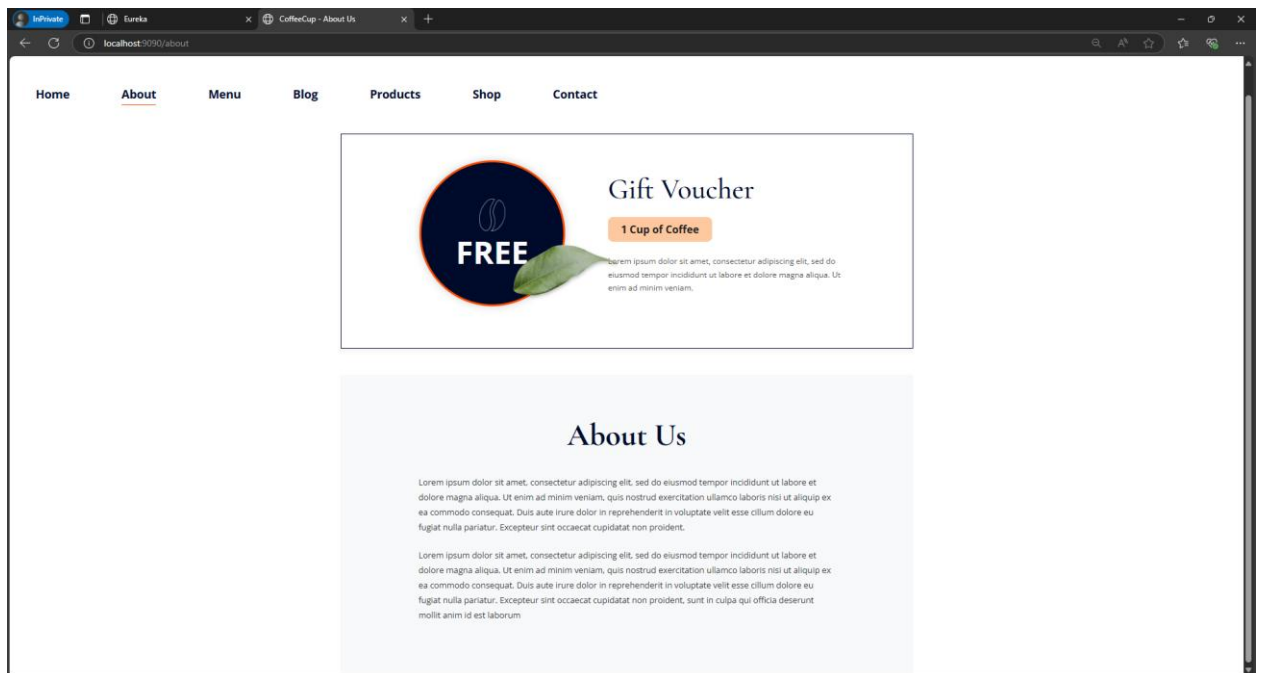
4.2.2. Trang Giới thiệu (About - /about)

Mục đích: Cung cấp thông tin về câu chuyện thương hiệu, giá trị cốt lõi và các chương trình đặc biệt của CoffeeCup.

Nội dung chính:

- Header điều hướng: Đánh dấu "About" là active, hiển thị logo và menu.
- Phần "Gift Voucher": Thông tin về chương trình khuyến mãi voucher (có trong template).

- Phần "About Us": Đoạn văn giới thiệu về lịch sử, sứ mệnh và đội ngũ của CoffeeCup, kèm hình ảnh minh họa về không gian cửa hàng hoặc sản phẩm đặc trưng.
- Kết quả thực hiện: Trang About sử dụng layout tĩnh từ template, bổ sung nội dung văn bản và hình ảnh để tăng tính sinh động. Thiết kế đơn giản nhưng truyền tải đầy đủ thông điệp thương hiệu.



Chương 4: Hình 4.3: Trang about

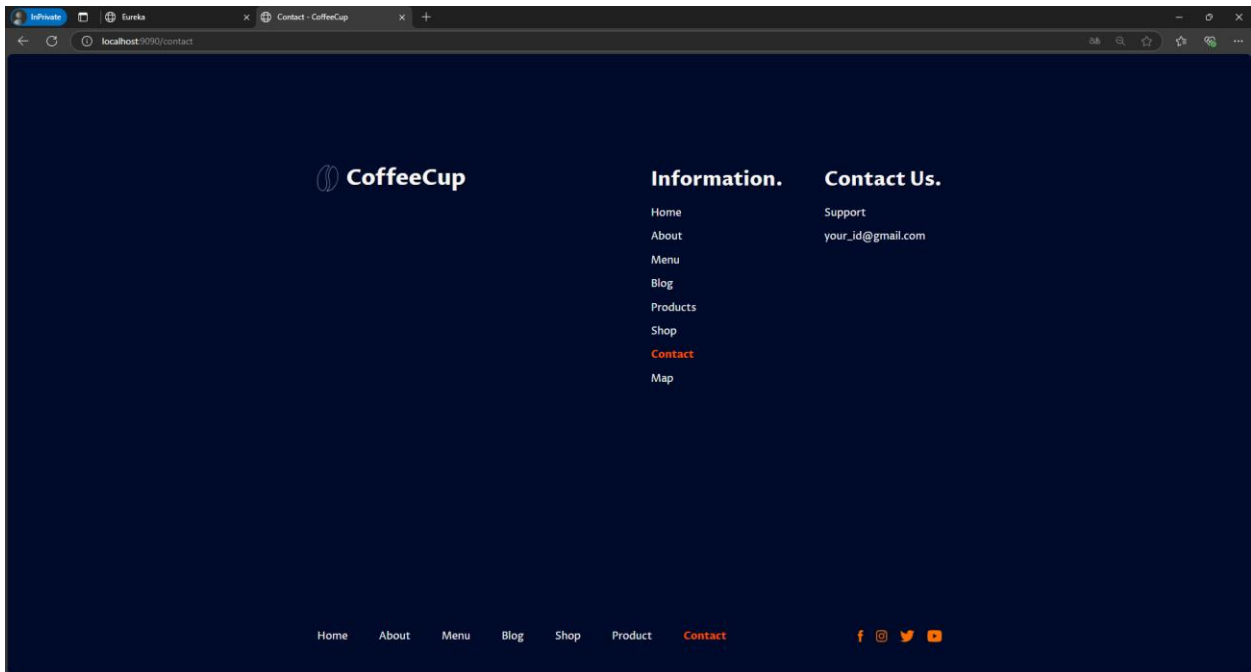
4.2.3. Trang Liên hệ (Contact - /contact)

Mục đích: Cung cấp thông tin liên lạc với cửa hàng và hiển thị các liên kết điều hướng/mạng xã hội.

Nội dung chính:

- Header: Logo, tên thương hiệu, thông tin liên hệ cơ bản (địa chỉ, email, số điện thoại - nếu có), và các liên kết nhanh đến các trang khác.

- Footer: Lặp lại liên kết điều hướng chính và hiển thị biểu tượng mạng xã hội (Facebook, Instagram, Twitter).
- Kết quả thực hiện: Trang Contact chủ yếu tái sử dụng header và footer từ template, không có form liên hệ động mà tập trung vào thông tin tĩnh. Giao diện gọn gàng, dễ tiếp cận.



Chương 4: Hình 4.4: Trang contact

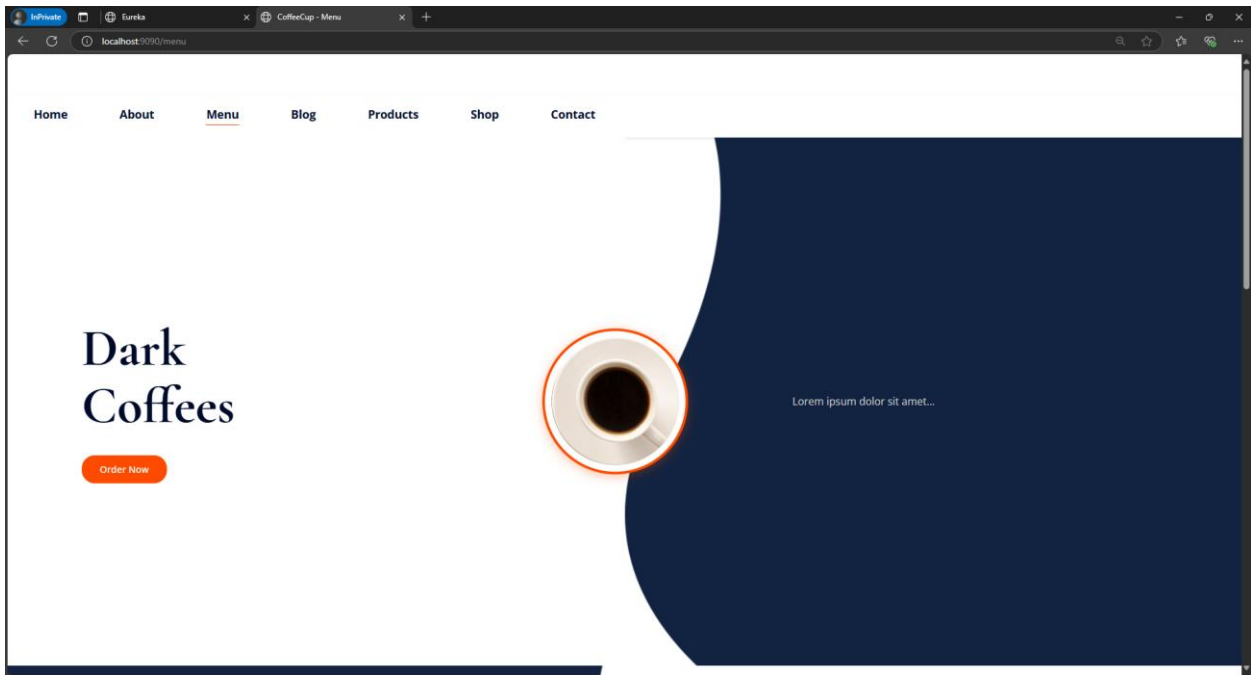
4.2.4. Trang Menu (/menu)

Mục đích: Trưng bày các loại đồ uống chính của CoffeeCup một cách trực quan và hấp dẫn.

Nội dung chính:

- Header điều hướng: Đánh dấu "Menu" là active.
- Các section: Chia thành ba khu vực chính: Dark Coffees, Cold Coffees, Hot Coffees.
- Nội dung mỗi section: Hình ảnh đại diện lớn, tên loại cà phê (ví dụ: "Dark Coffees"), mô tả ngắn gọn, và nút "Order Now" trở đến /products.

- Layout: Xen kẽ giữa ảnh và chữ (ảnh trái - chữ phải, rồi ngược lại) để tạo sự đa dạng.
- Kết quả thực hiện: Trang Menu được xây dựng tĩnh từ template, sử dụng HTML5 và CSS3 để tạo hiệu ứng bắt mắt. Chưa tích hợp dữ liệu động từ database.



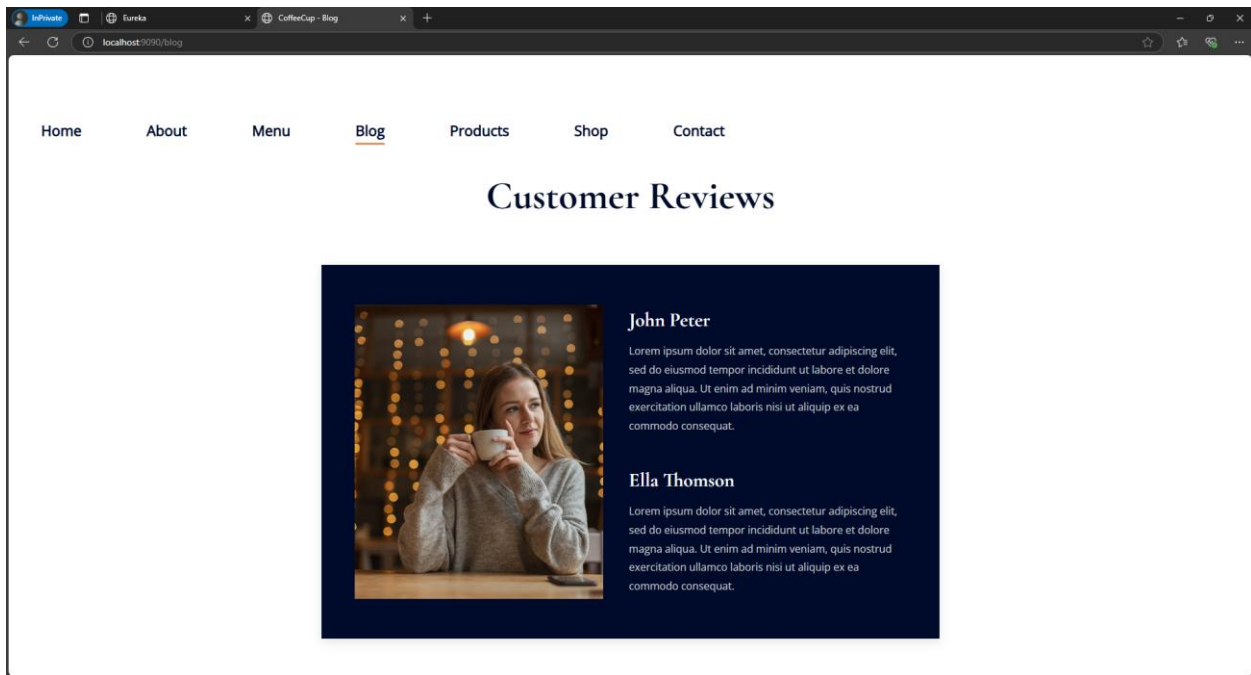
Chương 4: Hình 4.5: Trang menu

4.2.5. Trang Blog (/blog)

Mục đích: Hiện thị đánh giá của khách hàng (Customer Reviews) thay vì bài viết tin tức, tạo sự gần gũi với người dùng.

Nội dung chính:

- Header điều hướng: Đánh dấu "Blog" là active.
- Tiêu đề trang: "Customer Reviews".
- Layout chia cột: Một cột chứa hình ảnh minh họa (ví dụ: ảnh khách hàng uống cà phê), cột còn lại hiển thị các đoạn đánh giá với tên khách hàng và nội dung đánh giá.
- Kết quả thực hiện: Trang Blog sử dụng dữ liệu tĩnh từ template, thiết kế đơn giản nhưng hiệu quả trong việc truyền tải phản hồi tích cực từ khách hàng.



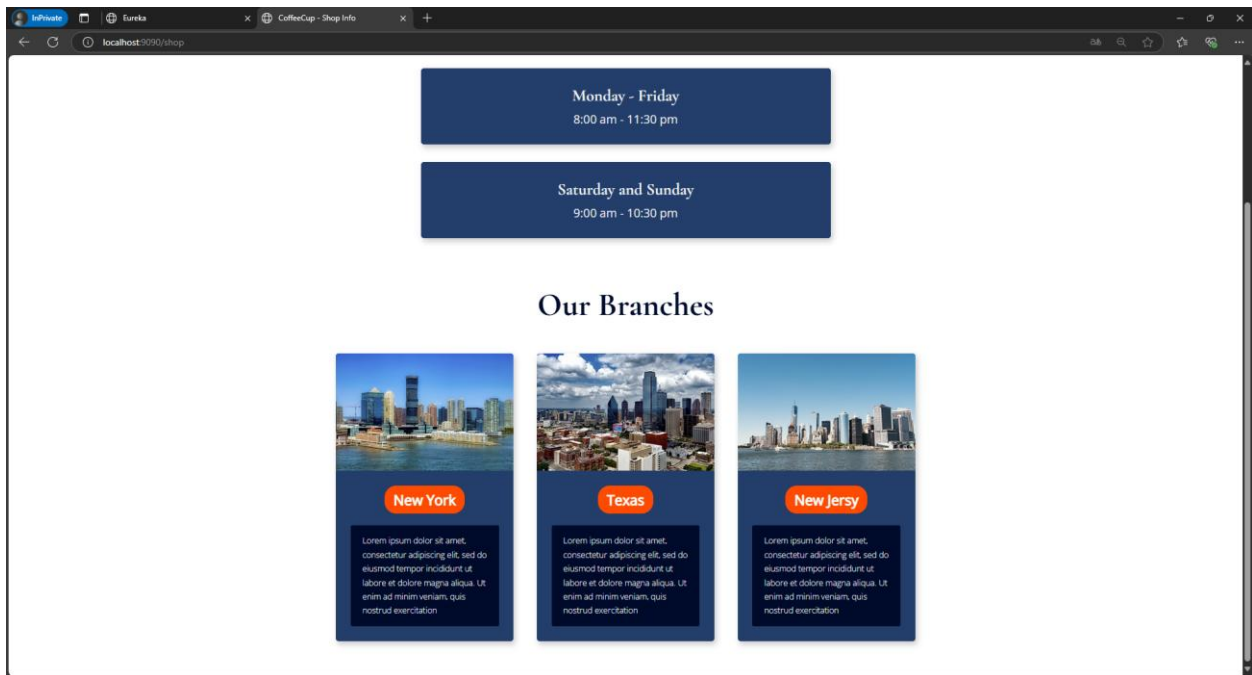
Chương 4: Hình 4.6: Trang blog

4.2.6. Trang Thông tin Cửa hàng (Shop - /shop)

Mục đích: Cung cấp thông tin thực tế về hoạt động của CoffeeCup, bao gồm giờ mở cửa và địa chỉ chi nhánh.

Nội dung chính:

- Header điều hướng: Đánh dấu "Shop" là active.
- Section Giờ mở cửa: Hiển thị giờ hoạt động (ví dụ: Thứ 2 - Thứ 6: 7:00 - 22:00, Thứ 7 - CN: 8:00 - 23:00).
- Section Chi nhánh: Sử dụng các "card" để hiển thị thông tin từng chi nhánh, mỗi card gồm ảnh đại diện, tên chi nhánh và địa chỉ.
- Kết quả thực hiện: Trang Shop được xây dựng tĩnh, tái sử dụng layout từ template, cung cấp thông tin rõ ràng và dễ đọc.



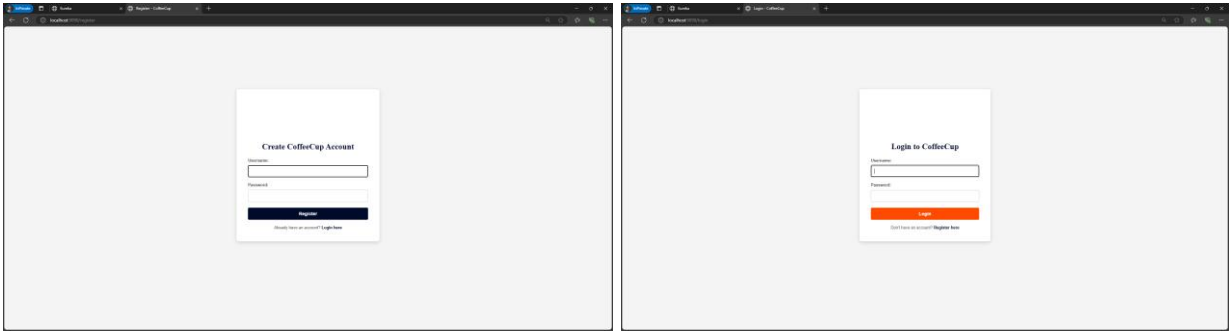
Chương 4: Hình 4.7: Trang shop

4.2.7. Trang Đăng nhập / Đăng ký User (/login, /register)

Mục đích: Cho phép khách hàng tạo tài khoản mới hoặc đăng nhập vào tài khoản hiện có.

Nội dung chính:

- Form Đăng ký (/register): Yêu cầu Username, Password với validation (username không trùng, password đủ dài). Sau khi thành công, chuyển hướng đến /login.
- Form Đăng nhập (/login): Yêu cầu Username, Password, hiển thị thông báo lỗi nếu sai. Thành công thì chuyển hướng về trang chủ (/) hoặc trang trước đó.
- Liên kết: Có link chuyển đổi giữa hai form (ví dụ: "Already have an account? Login").
- Kết quả thực hiện: Form được xây dựng bằng Thymeleaf, tích hợp Spring Security cho xác thực và JavaScript cho validation client-side. Giao diện đơn giản, dễ sử dụng.



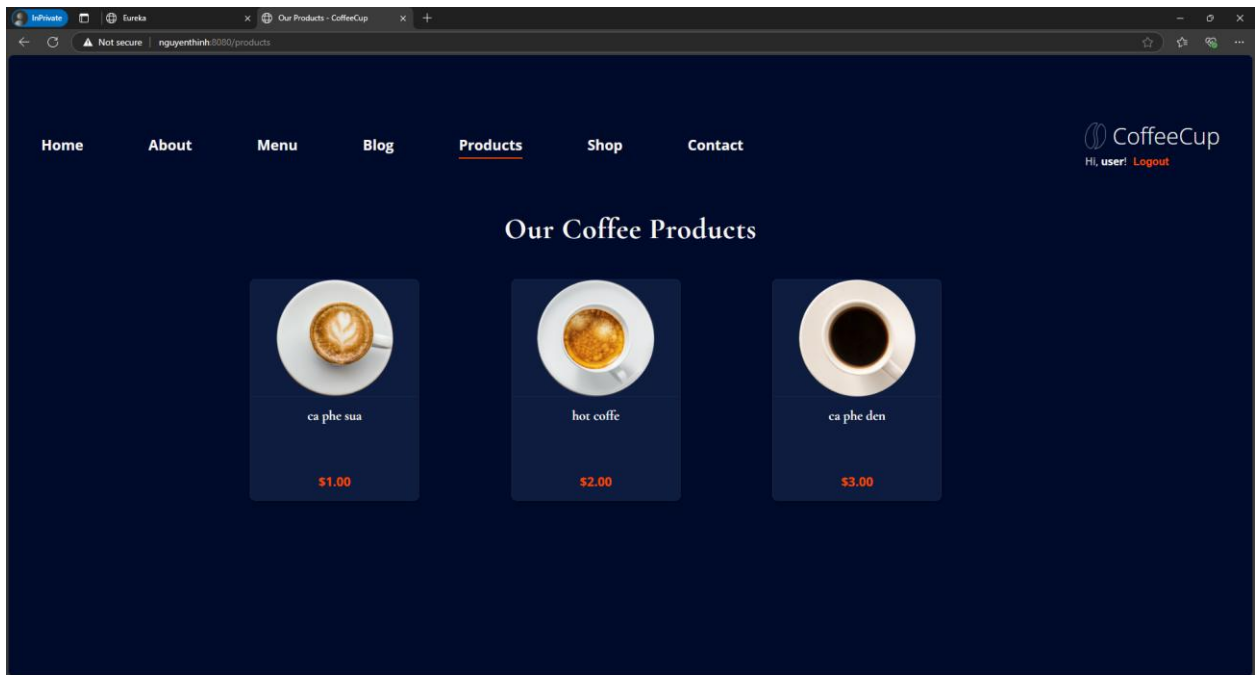
Chương 4: Hình 4.8: Trang đăng nhập/đăng ký

4.2.8. Trang Danh sách Sản phẩm (/products)

Mục đích: Hiển thị tất cả sản phẩm cà phê hiện có để khách hàng duyệt và chọn lựa, dữ liệu lấy động từ database.

Nội dung chính:

- Header điều hướng: Đánh dấu "Products" là active.
- Tiêu đề trang: "Our Coffee Products".
- Lưới sản phẩm (Product Grid): Hiển thị sản phẩm dưới dạng các thẻ (card), mỗi card là link đến /products/{id}, gồm ảnh thumbnail, tên sản phẩm, giá (mô tả ngắn tùy chọn).
- Thông báo: Hiển thị "No products available" nếu không có dữ liệu.
- Kết quả thực hiện: Dữ liệu sản phẩm được truy xuất từ bảng products qua Spring Data JPA, hiển thị bằng Thymeleaf. Giao diện lưới gọn gàng, dễ nhìn.



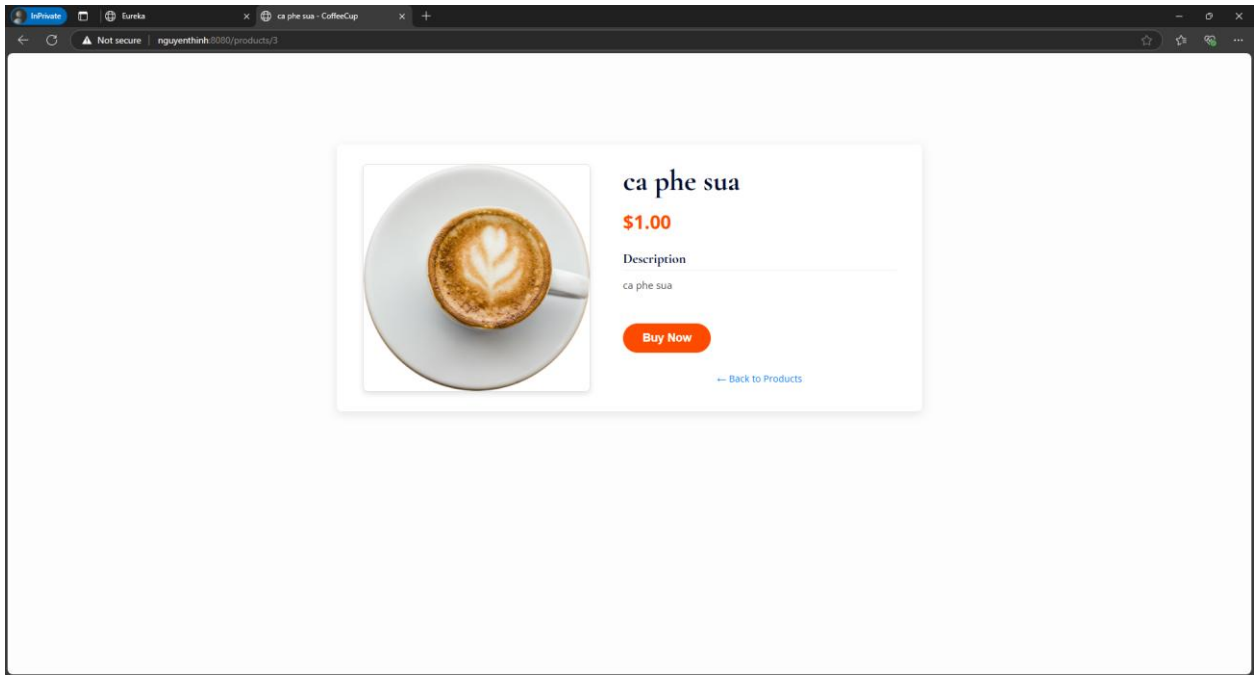
Chương 4: Hình 4.9: Trang Danh sách Sản phẩm

4.2.9. Trang Chi tiết Sản phẩm (/products/{id})

Mục đích: Cung cấp thông tin đầy đủ về một sản phẩm khi khách hàng chọn từ danh sách, dữ liệu lấy động từ database dựa trên ID.

Nội dung chính:

- Header điều hướng: Đánh dấu "Products" là active.
- Layout: Chia cột với ảnh sản phẩm lớn bên trái, thông tin chi tiết bên phải gồm tên sản phẩm (tiêu đề), giá, mô tả đầy đủ.
- Nút "Mua ngay": Hiện tại là nút demo, chưa tích hợp chức năng đặt hàng.
- Link "Back": Quay lại trang /products.
- Thông báo: "Product not found" nếu ID không tồn tại.
- Kết quả thực hiện: Trang chi tiết sử dụng Thymeleaf để hiển thị dữ liệu động từ bảng products. Thiết kế rõ ràng, tập trung vào thông tin sản phẩm.



Chương 4: Hình 4.10: Trang Chi tiết Sản phẩm

4.3. Giao diện và Chức năng Admin/System Site (coffeecup-admin-system)

Admin/System Site (coffeecup-admin-system) là thành phần quản trị nội bộ của hệ thống CoffeeCup, được thiết kế dành riêng cho người dùng có vai trò `ROLE_ADMIN` (Quản trị viên) hoặc `ROLE_SYSTEM` (Quản trị Hệ thống). Dưới đây là mô tả chi tiết các trang đã triển khai, bao gồm mục đích, nội dung chính và kết quả thực hiện.

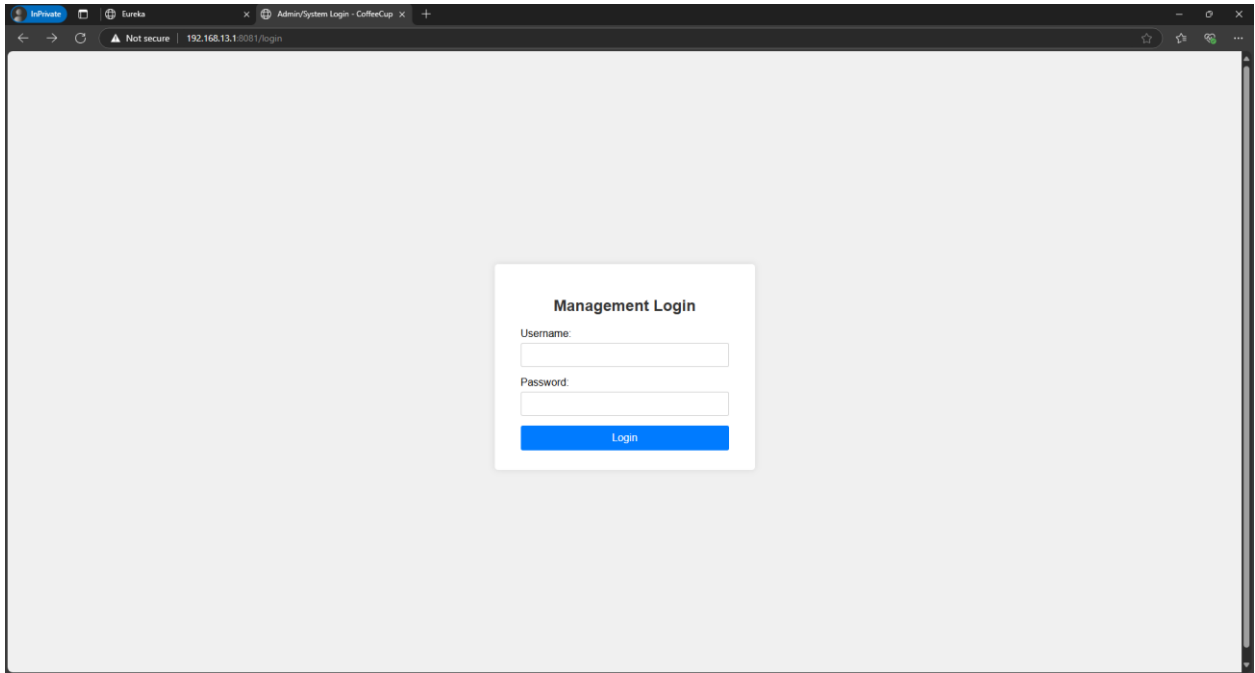
4.3.1. Trang Đăng nhập Admin/System (/manage/login)

Mục đích: Là cổng vào an toàn cho khu vực quản trị, chỉ cho phép người dùng có vai trò `ROLE_ADMIN` hoặc `ROLE_SYSTEM` truy cập.

Nội dung chính:

- Form đăng nhập: Thiết kế tối giản với hai trường Username và Password.
- Thông báo lỗi: Hiện thị thông báo nếu đăng nhập thất bại.
- Chuyển hướng: Sau khi đăng nhập thành công, chuyển đến Dashboard chính (/).

- Kết quả thực hiện: Trang đăng nhập được xây dựng bằng Thymeleaf, tích hợp Spring Security để xác thực và phân quyền. Giao diện đơn giản, tập trung vào chức năng, đảm bảo bảo mật với mã hóa mật khẩu (BCrypt).



Chương 4: Hình 4.11: Trang Đăng nhập Admin/System

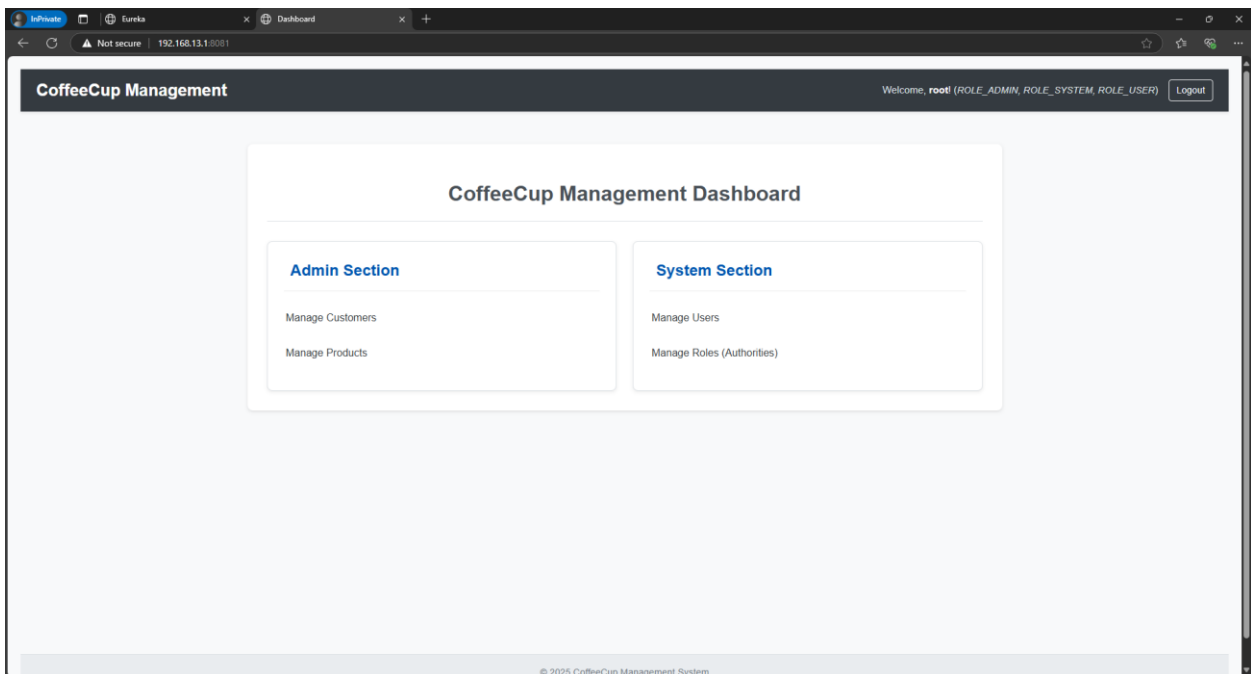
4.3.2. Dashboard chính (/)

Mục đích: Trang chào mừng sau khi đăng nhập, cung cấp các lối tắt truy cập nhanh đến các khu vực quản lý phù hợp với vai trò của người dùng.

Nội dung chính:

- Lời chào: "Welcome, [username]!" hiển thị tên người dùng đang đăng nhập.
- Vai trò: Hiển thị danh sách vai trò của người dùng (ví dụ: "Role: ADMIN, SYSTEM").
- Nút Logout: Cho phép đăng xuất khỏi hệ thống.
- Các khối điều hướng:

- Admin Section: Chỉ hiển thị với ROLE_ADMIN, chứa liên kết đến "Manage Customers" (/admin/customers) và "Manage Products" (/admin/products).
- System Section: Chỉ hiển thị với ROLE_SYSTEM, chứa liên kết đến "Manage Users" (/system/users) và "Manage Roles" (/system/authorities).
- Thông báo: Hiển thị thông báo nếu người dùng không có vai trò quản trị cụ thể (trường hợp lỗi).
- Kết quả thực hiện: Dashboard được xây dựng bằng Thymeleaf với điều kiện hiển thị động dựa trên vai trò (sử dụng sec:authorize). Giao diện rõ ràng, dễ điều hướng.



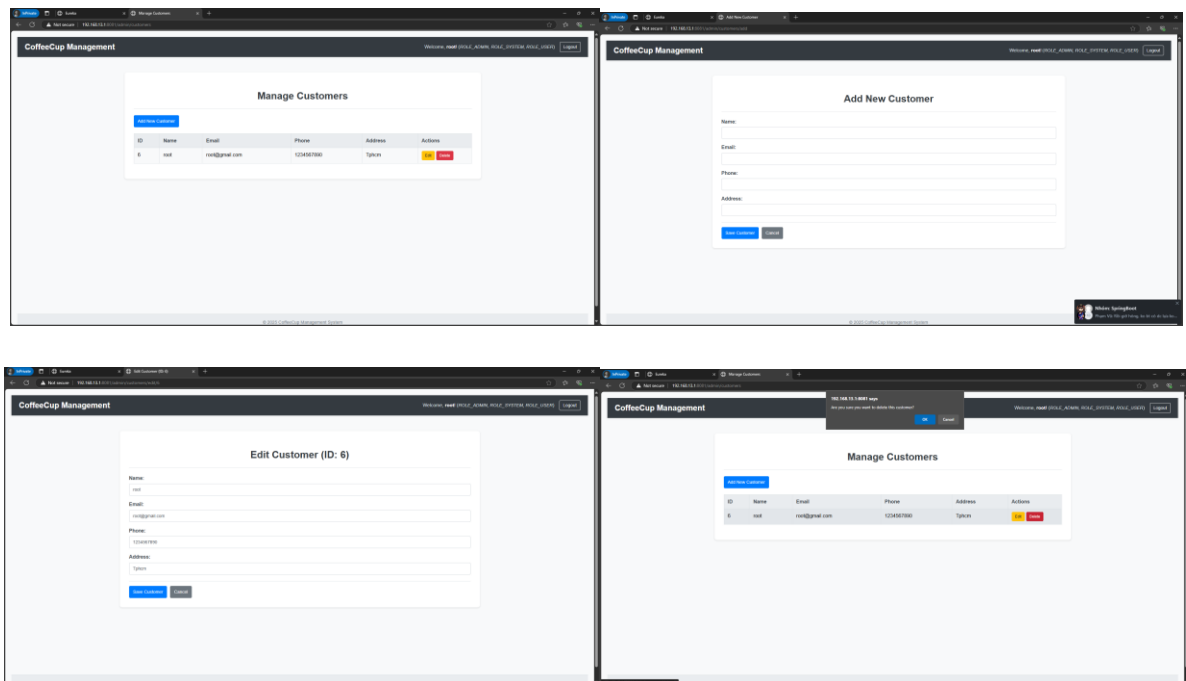
Chương 4: Hình 4.12: Trang Dashboard chính

*4.3.3. Quản lý Khách hàng (Customers - /admin/customers/**)*

Mục đích: Cho phép Admin quản lý thông tin của tất cả khách hàng trong hệ thống.

Chức năng chính (CRUD):

- Xem Danh sách (/admin/customers): Bảng liệt kê khách hàng với các cột ID, Name, Email, Phone, Address, kèm nút "Add New" và "Edit", "Delete" cho từng dòng.
- Thêm mới (/admin/customers/add -> POST /admin/customers/save): Form trống để nhập thông tin (Name, Email, Phone, Address), validation cả client-side (JavaScript) và server-side (Spring Validation) đảm bảo email hợp lệ, tên không trống. Lưu vào bảng customers.
- Sửa (/admin/customers/edit/{id} -> POST /admin/customers/save): Form điền sẵn thông tin khách hàng, cho phép chỉnh sửa, validation tương tự thêm mới, cập nhật dữ liệu trong database.
- Xóa (/admin/customers/delete/{id}): Hộp thoại xác nhận (JavaScript), xóa khách hàng khỏi bảng customers nếu xác nhận.
- Kết quả thực hiện: Chức năng CRUD hoàn thiện, bảng danh sách sử dụng Thymeleaf để hiển thị dữ liệu động, form có thông báo lỗi khi validation thất bại.



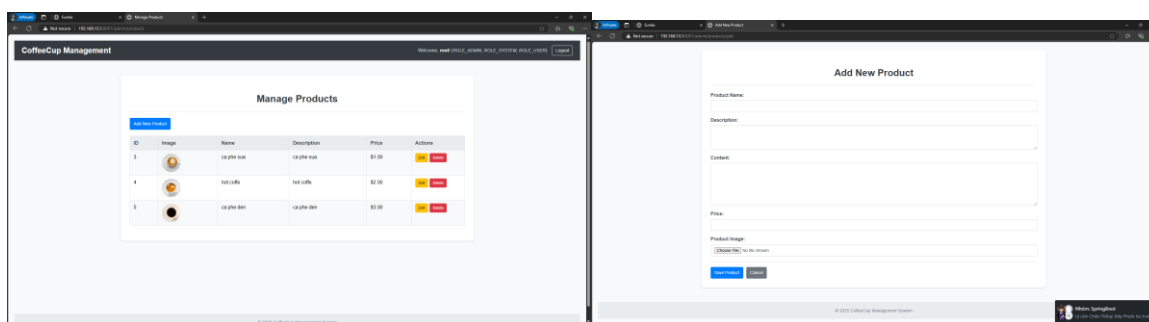
Chương 4: Hình 4.13: Trang Quản lý Khách

4.3.4. Quản lý Sản phẩm (Products - /admin/products/**)

Mục đích: Cho phép Admin quản lý danh mục sản phẩm của CoffeeCup.

Chức năng chính (CRUD + Ảnh):

- Xem Danh sách (/admin/products): Bảng liệt kê sản phẩm với cột ID, Ảnh thumbnail, Tên, Mô tả ngắn, Giá, kèm nút "Add New", "Edit", "Delete".
- Thêm mới (/admin/products/add -> POST /admin/products/save): Form trống để nhập Tên, Mô tả, Giá và upload file ảnh. Validation (giá > 0, tên không trống), lưu ảnh vào thư mục static/uploads và tên file vào bảng products.
- Sửa (/admin/products/edit/{id} -> POST /admin/products/save): Form điền sẵn thông tin, hiển thị ảnh hiện tại, cho phép upload ảnh mới (xóa ảnh cũ nếu thay), validation tương tự, cập nhật dữ liệu và ảnh trong database.
- Xóa (/admin/products/delete/{id}): Hộp thoại xác nhận, xóa sản phẩm và file ảnh khỏi hệ thống nếu xác nhận.
- Kết quả thực hiện: Quản lý sản phẩm hỗ trợ đầy đủ CRUD và upload ảnh, tích hợp Spring MVC để xử lý file, giao diện trực quan với thumbnail trong danh sách.



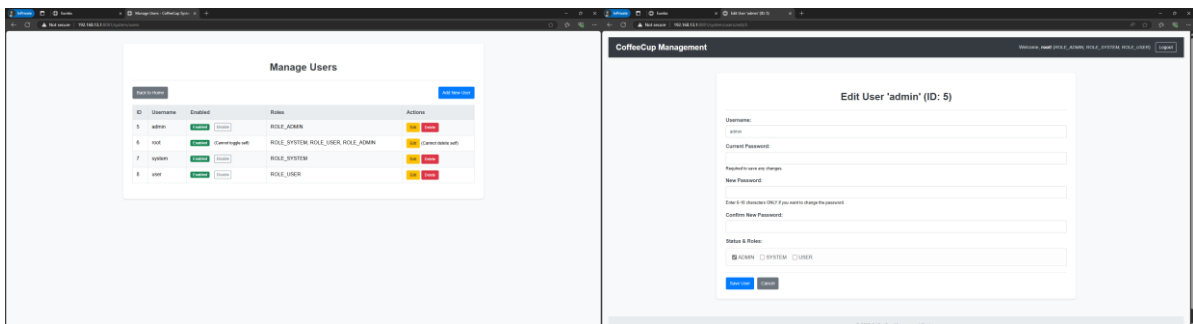
Chương 4: Hình 4.14: Trang Quản lý Sản phẩm

4.3.5. Quản lý Người dùng (Users - /system/users/**)

Mục đích: Cho phép System quản lý tất cả tài khoản người dùng trong hệ thống (bao gồm User, Admin, System).

Chức năng chính (CRUD + Roles + Status):

- Xem Danh sách (/system/users): Bảng liệt kê với cột ID, Username, Enabled/Disabled, Roles, nút "Add New", "Edit", "Delete" và "Enable/Disable" (ấn nút Delete/Enable cho chính mình).
- Thêm mới (/system/users/add -> POST /system/users/save): Form yêu cầu Username, Password, Confirm Password, chọn Roles (checkbox), validation (username không trùng, password khớp), mã hóa password bằng BCrypt, lưu vào bảng users và users_authorities.
- Sửa (/system/users/edit/{id} -> POST /system/users/save): Form điền sẵn thông tin, cho phép sửa Username, Roles, bật/tắt Enabled, yêu cầu mật khẩu hiện tại để lưu thay đổi, hỗ trợ đổi mật khẩu mới (validation), cập nhật database.
- Xóa (/system/users/delete/{id}): Xác nhận, xóa user và bản ghi liên quan trong users_authorities (không cho tự xóa).
- Bật/Tắt Trạng thái (/system/users/toggle/{id}): Thay đổi Enabled trực tiếp từ danh sách (không áp dụng cho bản thân).
- Kết quả thực hiện: Chức năng quản lý người dùng hoàn thiện, tích hợp Spring Security cho phân quyền và mã hóa, form có thông báo lỗi khi validation thất bại.



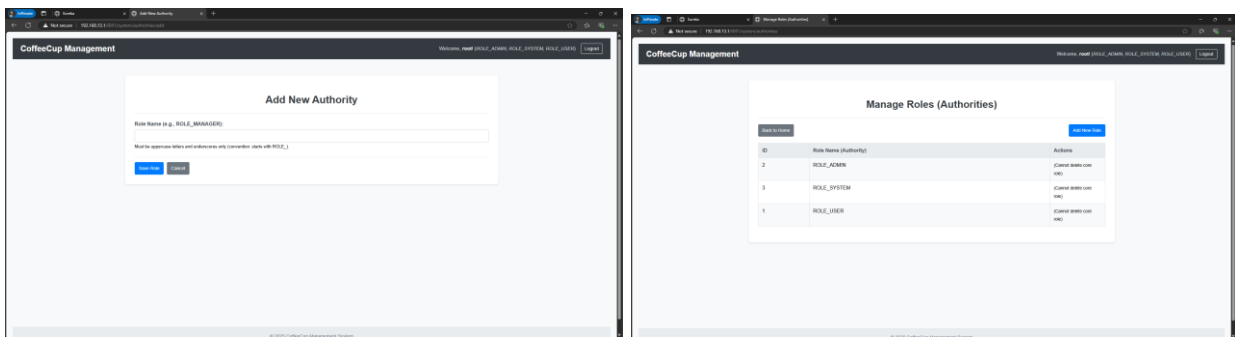
Chương 4: Hình 4.15: Trang quản lý người dùng

4.3.6. Quản lý Vai trò (Authorities - /system/authorities/**)

Mục đích: Cho phép System xem và quản lý các vai trò trong hệ thống.

Chức năng chính (CRUD cơ bản):

- Xem Danh sách (/system/authorities): Bảng liệt kê với cột ID, Tên Role (ví dụ: ROLE_ADMIN), nút "Add New" và "Delete" (ẩn Delete cho role cốt lõi).
- Thêm mới (/system/authorities/add -> POST /system/authorities/save): Form nhập tên Role (quy ước ROLE_...), validation (không trùng, đúng định dạng), lưu vào bảng authorities.
- Xóa (/system/authorities/delete/{id}): Xác nhận, chỉ cho xóa role không phải ROLE_ADMIN, ROLE_SYSTEM, ROLE_USER, kiểm tra không có user nào đang dùng role trước khi xóa.
- Kết quả thực hiện: Quản lý vai trò đơn giản nhưng hiệu quả, đảm bảo không xóa nhầm các role quan trọng, giao diện dễ sử dụng.



Chương 4: Hình 4.16: Trang quản lý vai trò

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

5.1.1. Các chức năng đã hoàn thành

Sau quá trình phát triển, hệ thống CoffeeCup đã hoàn thiện các chức năng chính như sau:

Public Site (coffeecup-public-site):

- Hiển thị các trang tĩnh (Home, About, Contact, Menu, Blog, Shop) dựa trên template, cung cấp giao diện thân thiện cho khách hàng.
- Hiển thị danh sách sản phẩm và chi tiết sản phẩm lấy động từ cơ sở dữ liệu, tích hợp nút "Mua ngay" (demo).
- Đăng ký và đăng nhập tài khoản User với validation cơ bản, chuyển hướng linh hoạt sau khi thành công.

Admin/System Site (coffeecup-admin-system):

- Đăng nhập an toàn cho Admin và System, phân quyền rõ ràng dựa trên vai trò (ROLE_ADMIN, ROLE_SYSTEM).
- Quản lý khách hàng (CRUD) với validation đầy đủ, hỗ trợ Admin theo dõi và cập nhật thông tin khách hàng.
- Quản lý sản phẩm (CRUD) kèm upload, hiển thị và xóa ảnh, đảm bảo quản lý danh mục sản phẩm hiệu quả.
- Quản lý người dùng (CRUD) với gán vai trò, bật/tắt trạng thái, yêu cầu mật khẩu cũ khi đổi, phù hợp cho System.
- Quản lý vai trò (xem, thêm, xóa có điều kiện), hỗ trợ System cấu hình quyền truy cập.

Hạ tầng Microservices:

- Triển khai Eureka Server để quản lý và khám phá dịch vụ, hiển thị trạng thái các microservices (Gateway, Public, Admin).
- Sử dụng API Gateway làm điểm vào duy nhất, định tuyến yêu cầu đến các dịch vụ tương ứng.

5.1.2. Ưu điểm của hệ thống

Hệ thống CoffeeCup mang lại nhiều ưu điểm nổi bật:

- Kiến trúc Microservices: Tách biệt Public Site và Admin/System Site thành các dịch vụ độc lập, dễ phát triển, bảo trì và triển khai riêng lẻ mà không ảnh hưởng toàn hệ thống.
- Công nghệ Spring phổ biến: Sử dụng Java Spring Boot, Spring Cloud, và Spring Security giúp hệ thống tận dụng được các tính năng mạnh mẽ, tài liệu phong phú và cộng đồng hỗ trợ lớn.
- Tính linh hoạt: Phân quyền rõ ràng giữa User, Admin và System, đảm bảo mỗi vai trò chỉ truy cập được chức năng phù hợp.
- Giao diện thân thiện: Public Site cung cấp trải nghiệm trực quan cho khách hàng, trong khi Admin/System Site tối ưu cho quản trị với các form và bảng dễ sử dụng.
- Bảo mật: Tích hợp Spring Security với mã hóa mật khẩu và kiểm soát truy cập, đảm bảo an toàn dữ liệu và hệ thống.

5.2. Hạn chế của đề tài

Mặc dù đã đạt được các mục tiêu cơ bản, hệ thống CoffeeCup vẫn tồn tại một số hạn chế:

- Chức năng còn cơ bản: Chưa triển khai các tính năng nâng cao như giỏ hàng, đặt hàng, thanh toán hay quản lý đơn hàng, khiến hệ thống chưa đáp ứng đầy đủ nhu cầu thương mại điện tử thực tế.
- Chưa tách database: Các microservices hiện dùng chung một cơ sở dữ liệu MySQL, không đạt được tính độc lập lý tưởng của kiến trúc Microservices.

- Chưa xử lý vấn đề phức tạp của Microservices: Thiếu giao tiếp liên dịch vụ (inter-service communication), quản lý transaction phân tán, hoặc các cơ chế xử lý lỗi nâng cao (Circuit Breaker, Retry).
- Giao diện chưa tối ưu hoàn toàn: Một số trang tĩnh (Menu, Blog) vẫn phụ thuộc vào template, chưa tích hợp dữ liệu động hoặc tối ưu hóa trải nghiệm người dùng (UI/UX).
- Hiệu năng: Chưa kiểm tra tải với lượng lớn người dùng đồng thời, có thể gặp hạn chế khi mở rộng quy mô.

5.3. Hướng phát triển trong tương lai

Để khắc phục hạn chế và nâng cao hệ thống, các hướng phát triển sau được đề xuất:

5.3.1. Hoàn thiện chức năng

- Thêm các tính năng thương mại điện tử như giỏ hàng, đặt hàng, thanh toán (tích hợp cổng thanh toán như PayPal, VNPay).
- Quản lý đơn hàng cho Admin với trạng thái (đang xử lý, đã giao, hủy).
- Hỗ trợ phân trang và tìm kiếm trên danh sách sản phẩm, khách hàng, người dùng để cải thiện hiệu quả quản lý.

5.3.2. Tách Database

- Mỗi microservice (Public Site, Admin/System Site) sử dụng cơ sở dữ liệu riêng, đảm bảo tính độc lập và giảm phụ thuộc.
- Đồng bộ dữ liệu giữa các dịch vụ bằng cơ chế sao chép (replication) hoặc sự kiện (event-driven).

5.3.3. Giao tiếp Liên Service

- Sử dụng RestTemplate hoặc WebClient để các microservices gọi API lẫn nhau (ví dụ: Public Site lấy danh sách sản phẩm từ Admin Service).
- Áp dụng Message Queue (Kafka, RabbitMQ) để xử lý sự kiện bất đồng bộ, như thông báo khi sản phẩm mới được thêm.

5.3.4. Nâng cao Microservices

- Circuit Breaker: Sử dụng Resilience4j để xử lý lỗi khi một dịch vụ không phản hồi.
- Distributed Tracing: Áp dụng Zipkin hoặc Micrometer Tracing để theo dõi luồng yêu cầu qua các microservices.
- Cấu hình tập trung: Sử dụng Spring Cloud Config để quản lý cấu hình từ một nguồn duy nhất.

5.3.5. Frontend Framework

- Chuyển đổi giao diện Public Site sang framework hiện đại như React, Angular hoặc Vue.js, gọi API từ backend để tăng tính tương tác và hiệu năng.
- Cải thiện UI/UX với thiết kế responsive, animation và tối ưu trải nghiệm trên thiết bị di động.

5.3.6. Deployment & CI/CD

- Đóng gói các microservices bằng Docker để triển khai nhất quán trên các môi trường.
- Triển khai hệ thống lên cloud (AWS, Google Cloud, Heroku) để tăng khả năng truy cập và mở rộng.
- Thiết lập quy trình CI/CD (Continuous Integration/Continuous Deployment) với GitHub Actions hoặc Jenkins để tự động hóa quá trình build, test và deploy.

TÀI LIỆU THAM KHẢO

1. Spring Official Documentation, Tác giả: Pivotal Software, Inc, Tài liệu chính thức của Spring, bao gồm Spring Boot, Spring Security, Spring Data JPA và Spring Cloud, được tham khảo để triển khai các thành phần backend và kiến trúc Microservices của hệ thống CoffeeCup
2. Spring Boot in Action, Tác giả: Craig Walls, Sách hướng dẫn chi tiết về Spring Boot, cung cấp kiến thức cơ bản và nâng cao để phát triển các microservices như Public Site và Admin/System Site, Nhà xuất bản: Manning Publications, 2018
3. Microservices Patterns: With examples in Java, Tác giả: Chris Richardson, Sách giới thiệu các mẫu thiết kế Microservices, được tham khảo để thiết kế kiến trúc hệ thống CoffeeCup với Eureka Server và API Gateway, Nhà xuất bản: Manning Publications, 2018
4. Bài giảng và tài liệu hướng dẫn từ giảng viên, Thạc Sĩ Trần Quốc Trường, Các bài giảng về Spring Boot, Microservices và lập trình web, được sử dụng làm cơ sở lý thuyết cho đề tài