**PHÁT TRIỂN ỨNG DỤNG WEB 2**

**BÁO CÁO DỰ ÁN MÔN HỌC**

**ĐỀ TÀI QUẢN LÝ NHÂN VIÊN**

**Sinh viên thực hiện: Võ Đại Hiệp\_63132946**

**MỤC LỤC**

[**BÁO CÁO 2**](#_Toc167631844)

[**I. Trang Home 2**](#_Toc167631845)

[**II. Trang Login 2**](#_Toc167631846)

[**1. Login với vai trò Admin 3**](#_Toc167631847)

[**2. Hiện thị trang Admin 3**](#_Toc167631848)

[**3. Login với vai trò Nhân Viên 4**](#_Toc167631849)

[**4. Hiện thị trang Nhân Viên 4**](#_Toc167631850)

[**III. Hiển thị danh sách nhân viên 5**](#_Toc167631851)

[**IV. Hiển thị danh sách phòng ban 6**](#_Toc167631852)

[**V. Xem chi tiết nhân viên 6**](#_Toc167631853)

[**VI. Sửa thông tin nhân viên 7**](#_Toc167631854)

[**VII. Xóa nhân viên 7**](#_Toc167631855)

[**VIII. Thêm nhân viên 8**](#_Toc167631856)

[**IX. Gửi mail 8**](#_Toc167631857)

BÁO CÁO

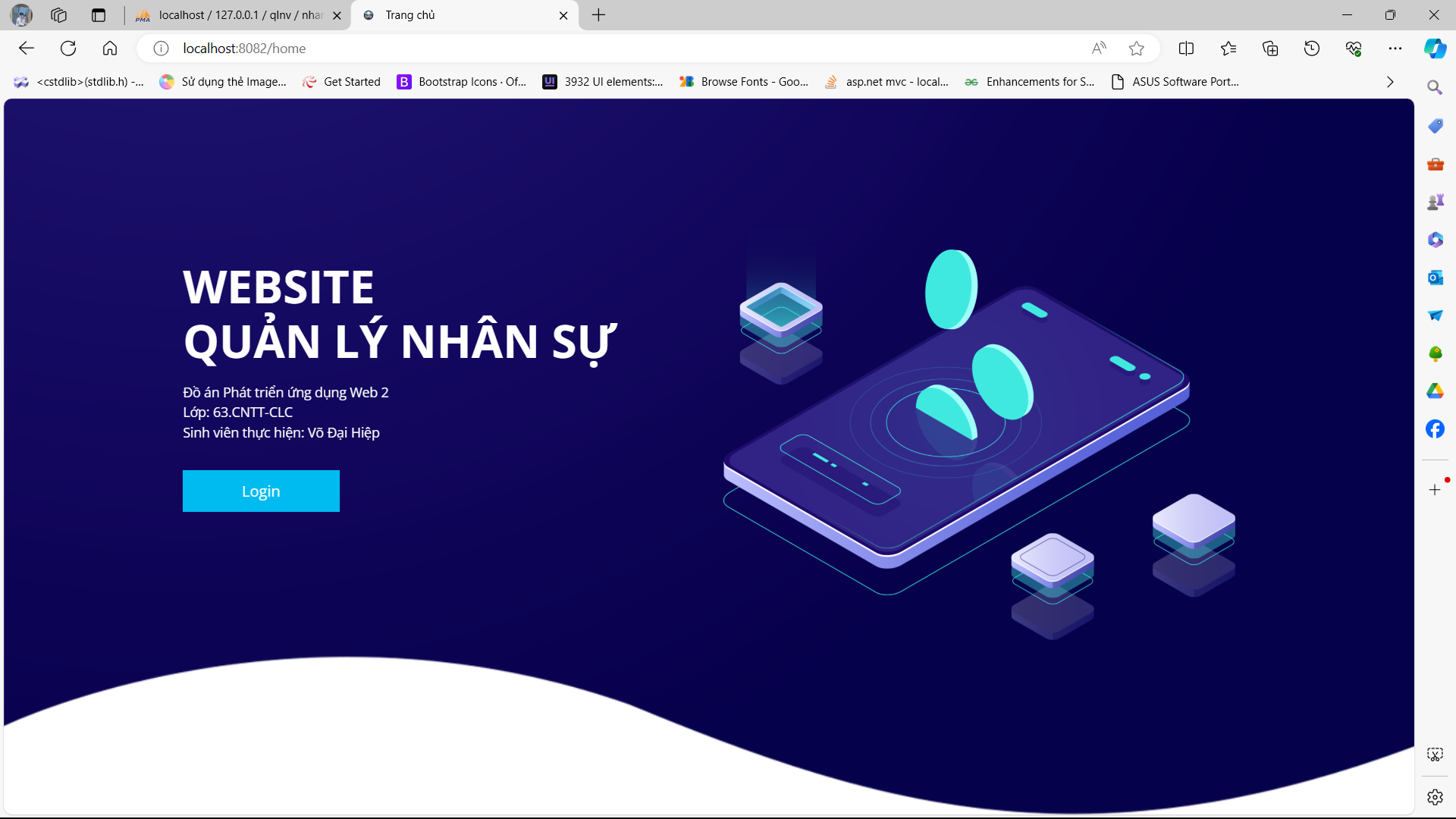
1. Trang Home

Để tạo trang chủ này, em sử dụng HTML và CSS, cùng với Bootstrap và Thymeleaf.

HTML và CSS để tạo cấu trúc và trình bày giao diện của trang web. Bootstrap sử dụng để tạo giao diện đáp ứng và các thành phần giao diện như nút, tiêu đề, v.v. Thymeleaf sử dụng để tạo các trang web động trong ứng dụng Spring Boot. Trong mã HTML, em sử dụng các thuộc tính như th:href để thêm các đường dẫn động cho các tài nguyên như CSS và hình ảnh.

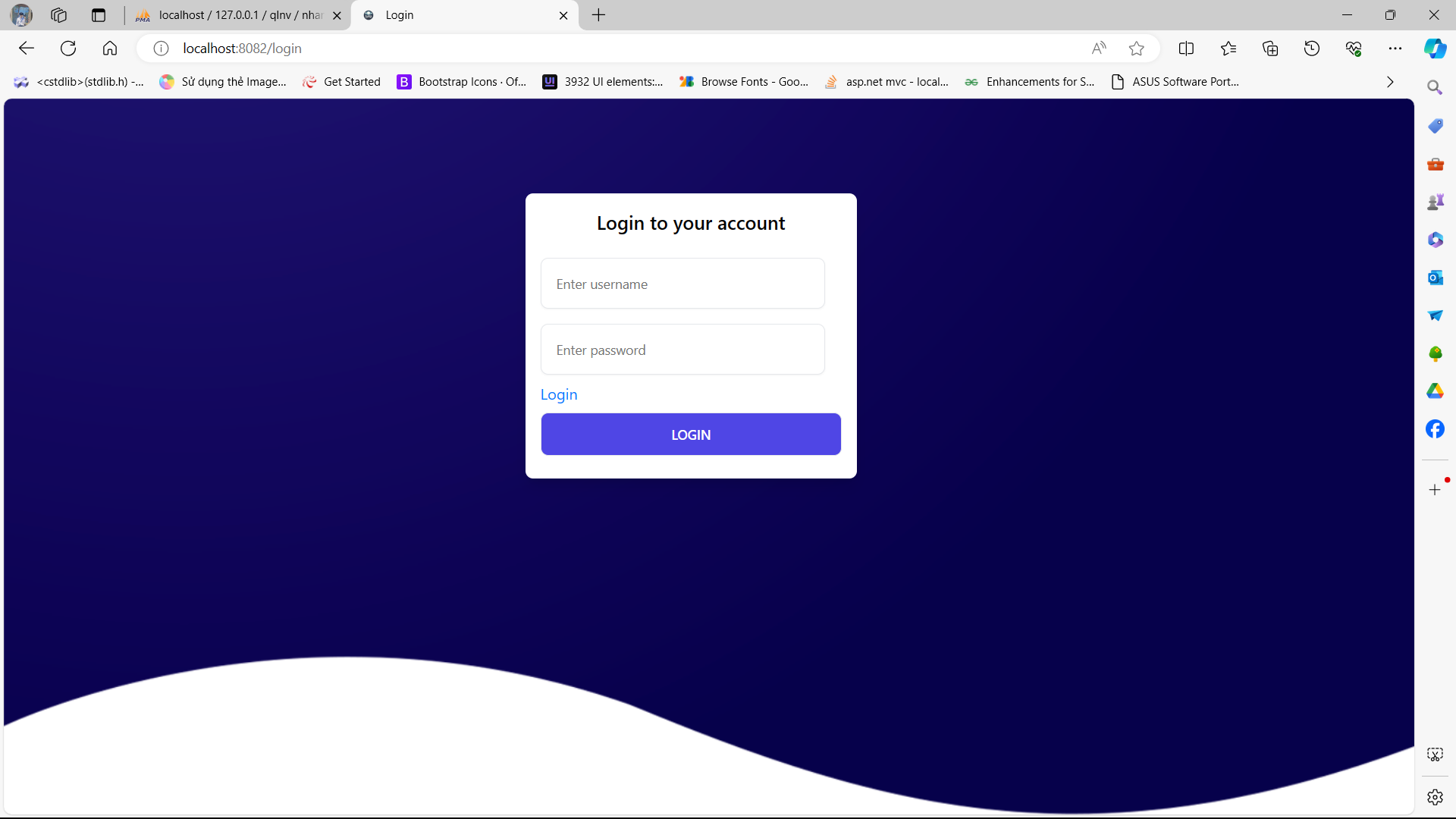
Em tái sử dụng các lớp và id CSS để tái sử dụng các phần của giao diện như các hộp, nút, và hình ảnh.

Kết quả là một trang chủ đơn giản nhưng có giao diện hấp dẫn.



1. Trang Login

Để tạo trang đăng nhập này, em sử dụng HTML và CSS để tạo giao diện, và tích hợp Bootstrap để làm cho trang web trở nên responsive và có giao diện đẹp hơn.

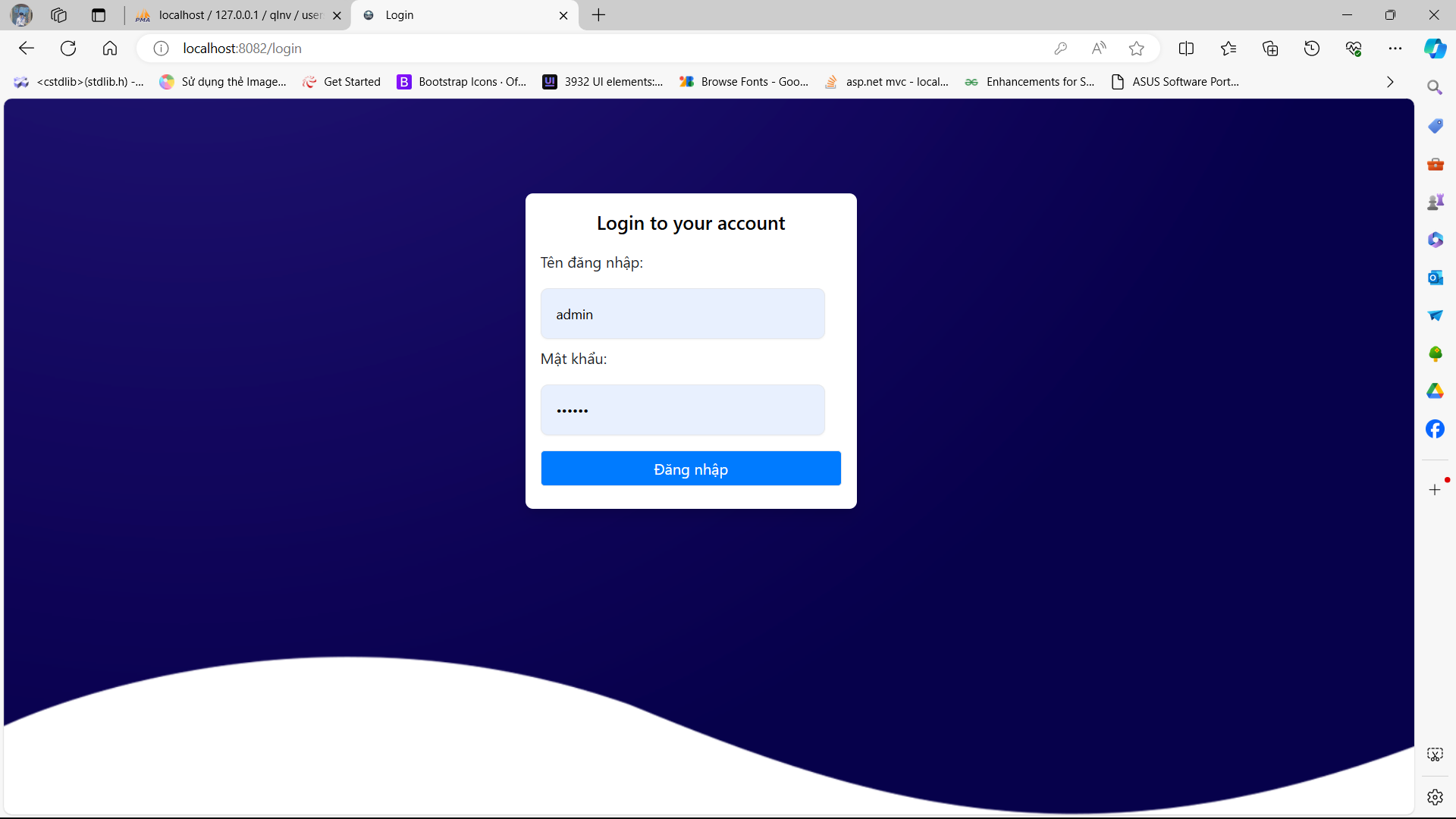


1. **Login với vai trò Admin**

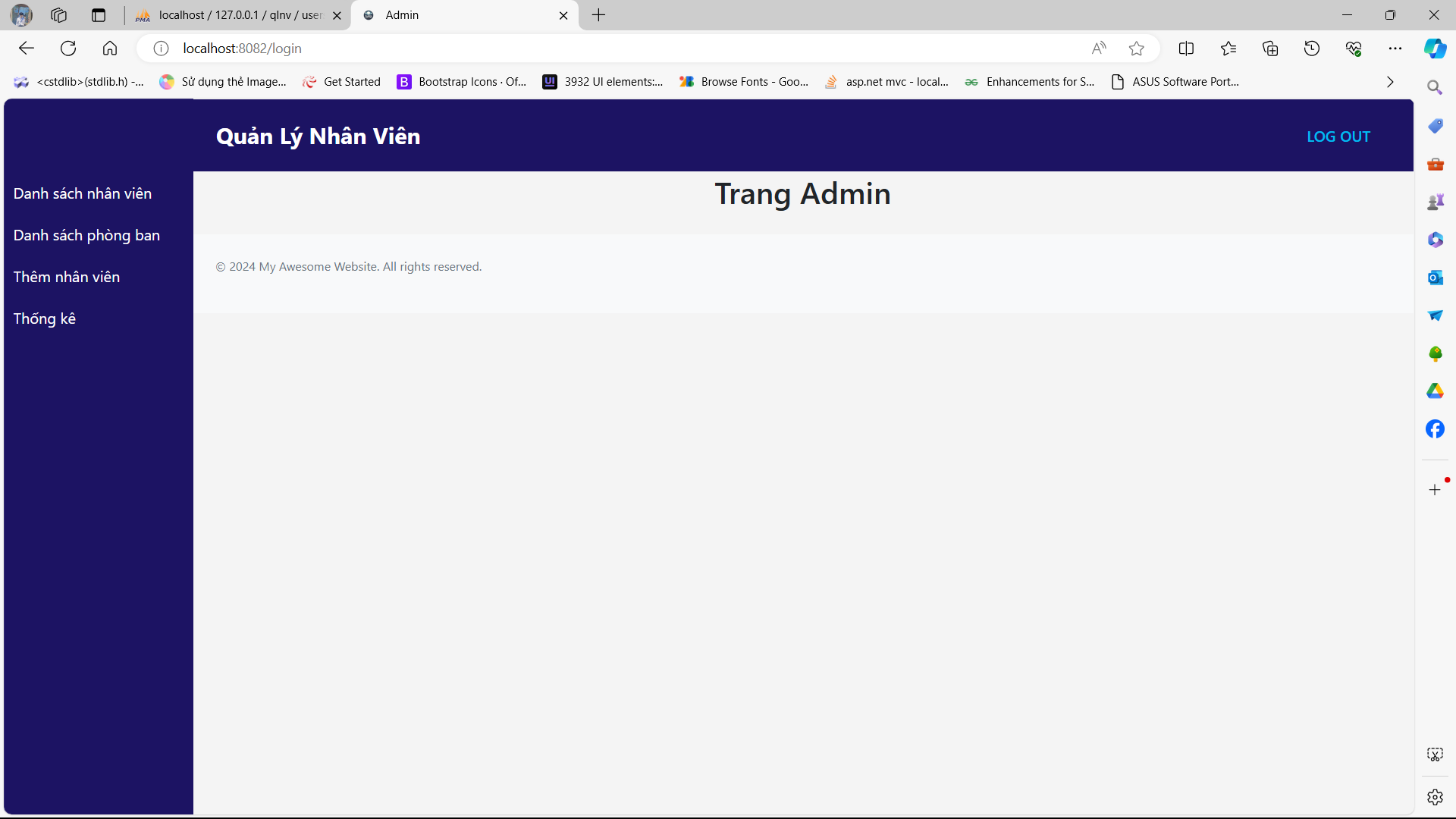
Em dùng xử lý yêu cầu POST khi người dùng đăng nhập. Kiểm tra thông tin đăng nhập, nếu hợp lệ, lưu thông tin người dùng vào session và định tuyến đến trang tương ứng với vai trò của người dùng ("admin" hoặc "nhanvien"). Nếu không hợp lệ, trả về trang đăng nhập.

Em phân quyền bằng cách kiểm tra vai trò của người dùng để định tuyến đến trang tương ứng ("admin" hoặc "nhanvien"). Sau đó sử dụng HttpSession để lưu trữ thông tin đăng nhập của người dùng.

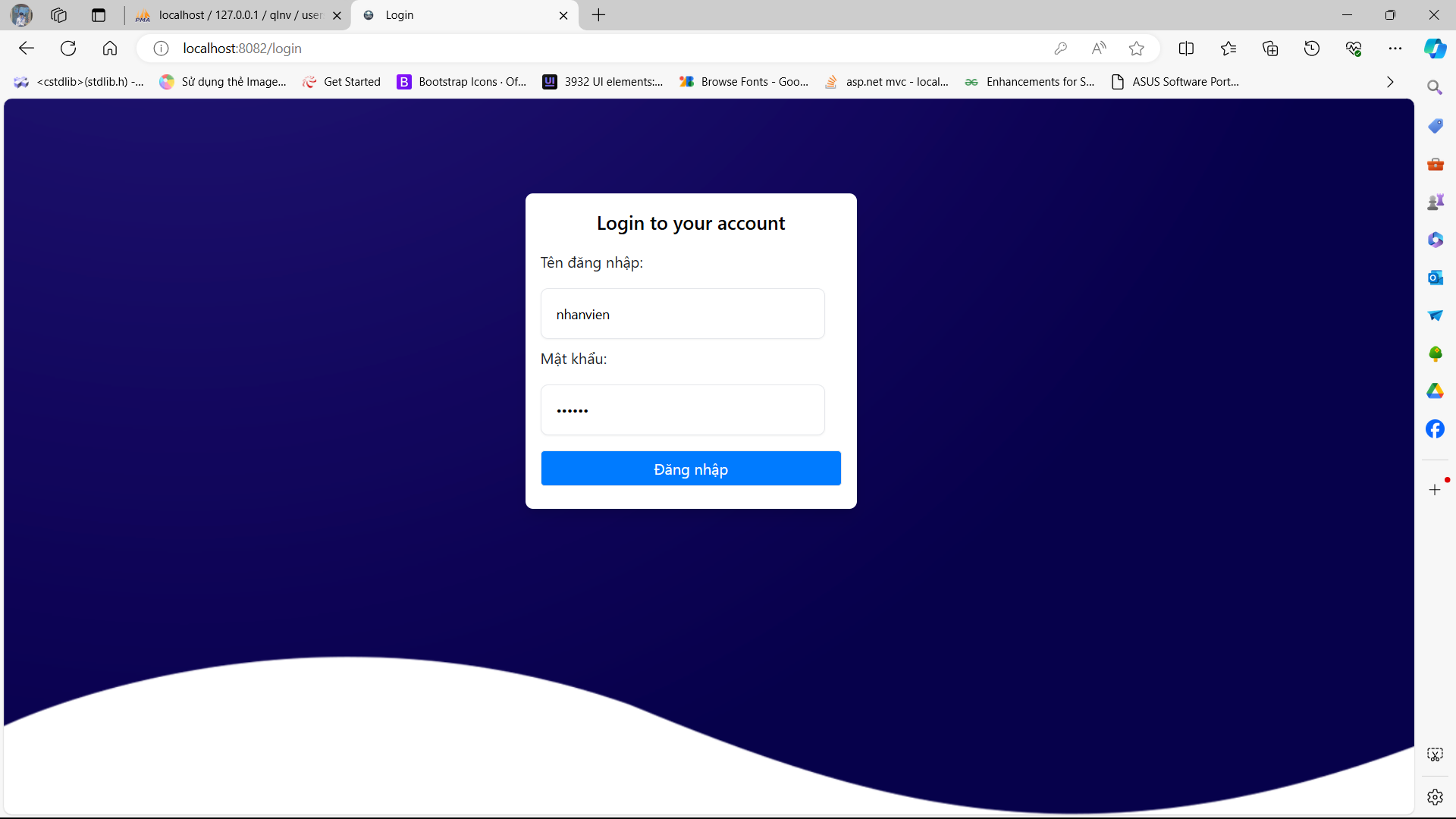
Kết quả là có thể đăng nhập vào dưới 2 vai trò là quản trị viên (admin) và nhân viên (nhân viên).



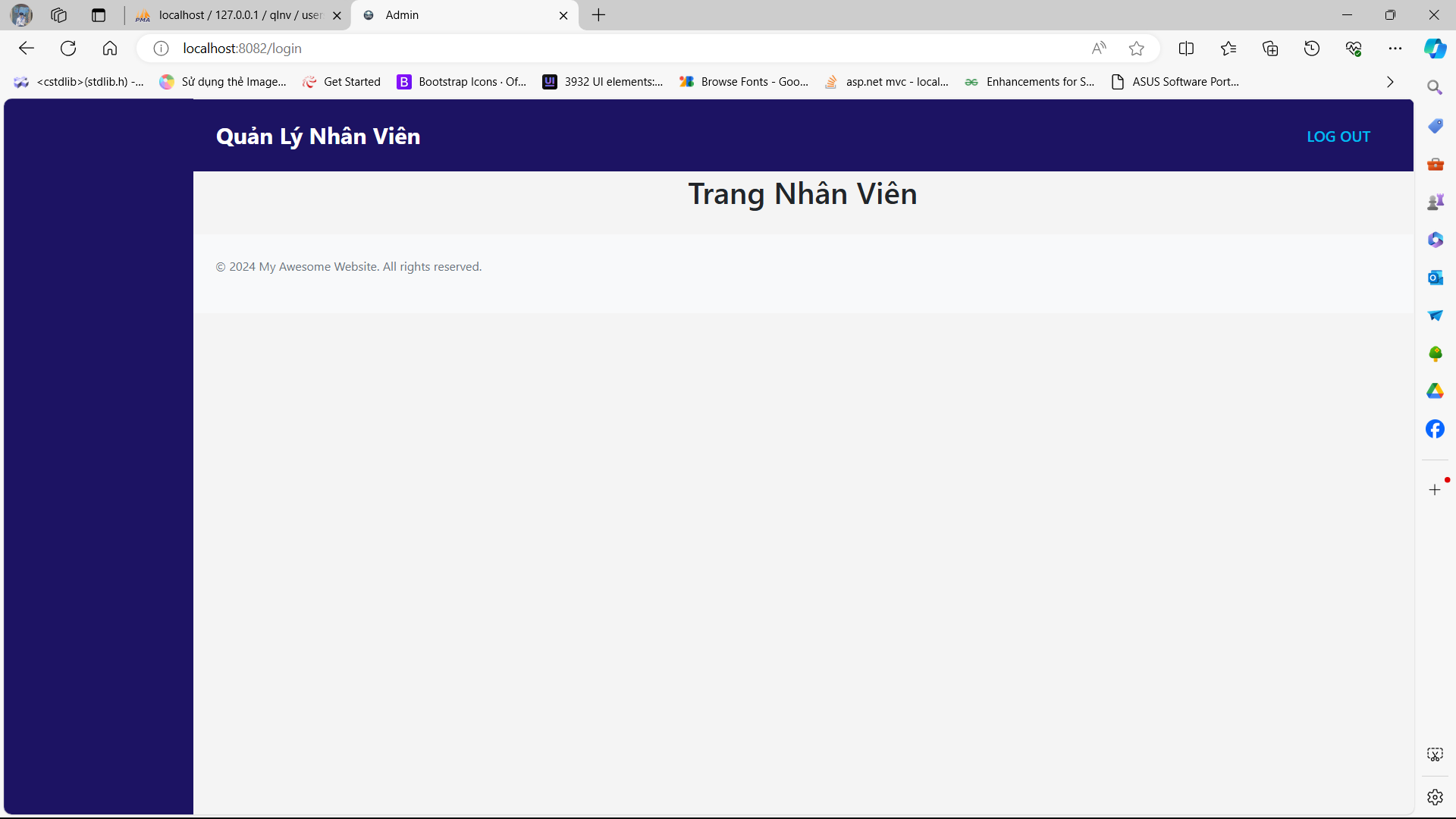
1. **Hiện thị trang Admin**



1. **Login với vai trò Nhân Viên**

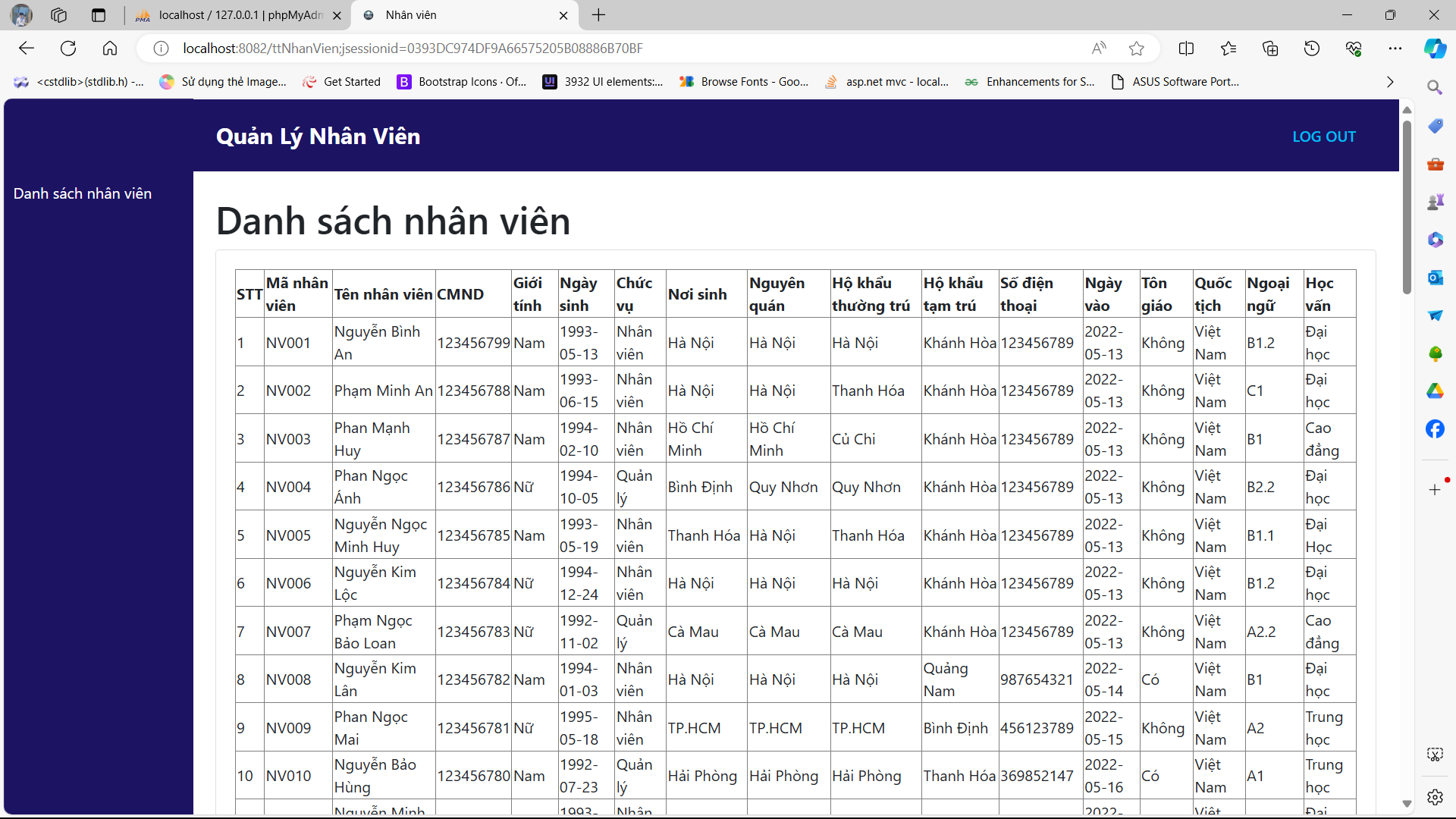


1. **Hiện thị trang Nhân Viên**



Để hiển thị danh sách nhân viên trong trang web, em sử dụng Spring Boot để xử lý yêu cầu và Thymeleaf để render HTML. Trong phương thức getAllTTNhanViens của UserController, em sử dụng TTNhanVienService và NhanVienService để lấy thông tin về tất cả nhân viên và thông tin nhân viên. Sau đó, em add các thông tin này vào model và trả về view "ttnhanvien". Trong file HTML, em sử dụng Thymeleaf để lặp qua danh sách nhân viên và hiển thị thông tin chi tiết về mỗi nhân viên trong một bảng.

Kết quả là em có một trang web hiển thị danh sách nhân viên một cách dễ đọc và dễ quản lý.

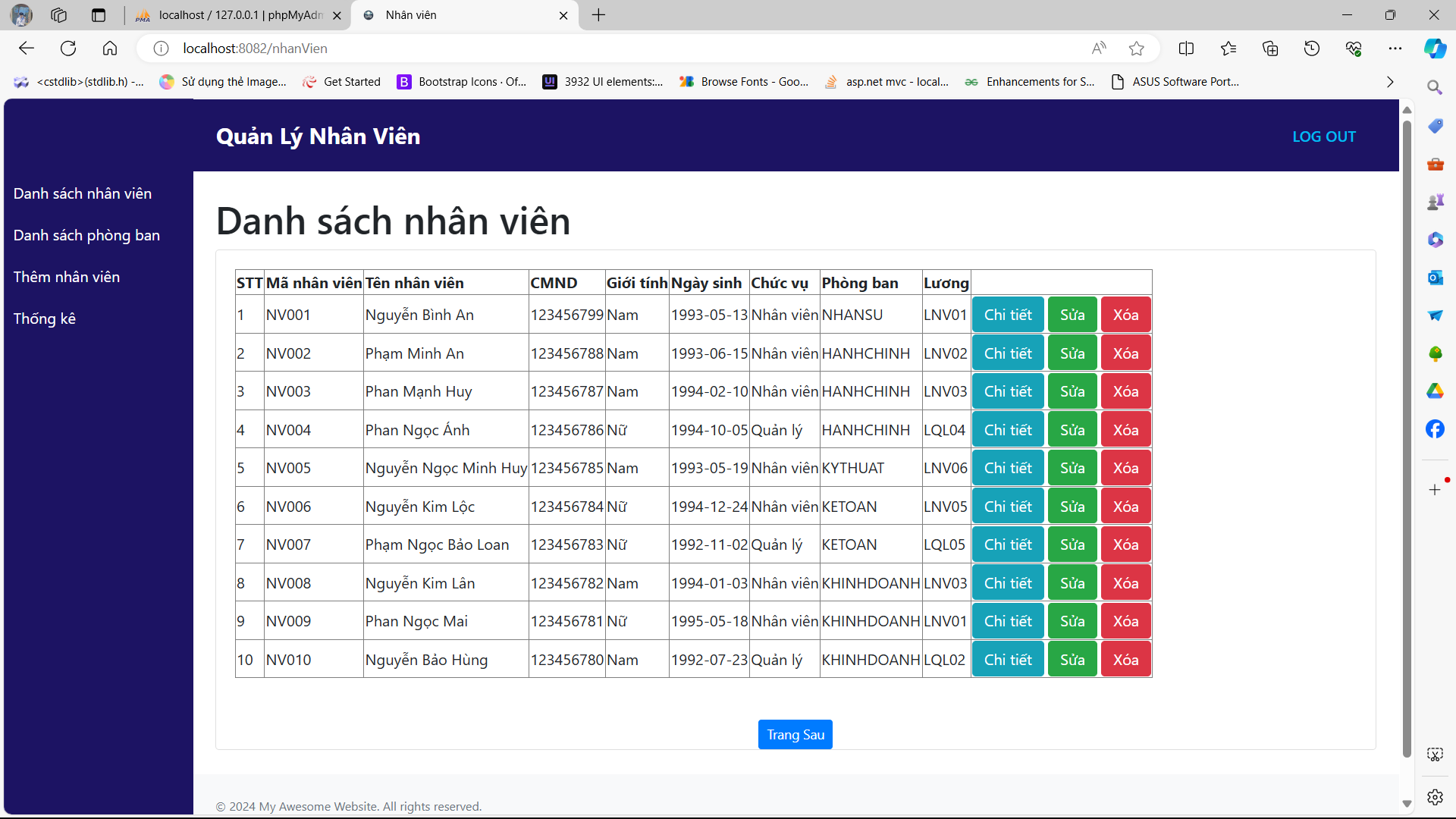


1. Hiển thị danh sách nhân viên

Để xây dựng trang danh sách nhân viên trong ứng dụng Quản Lý Nhân Viên, em sử dụng Spring Boot và Thymeleaf. Trong NhanVienController, em sử dụng các phương thức @GetMapping để lấy danh sách nhân viên và hiển thị form thêm, sửa, chi tiết và xóa nhân viên. Để hiển thị danh sách nhân viên, em sử dụng phân trang bằng Pageable, với mỗi trang chứa 10 nhân viên.

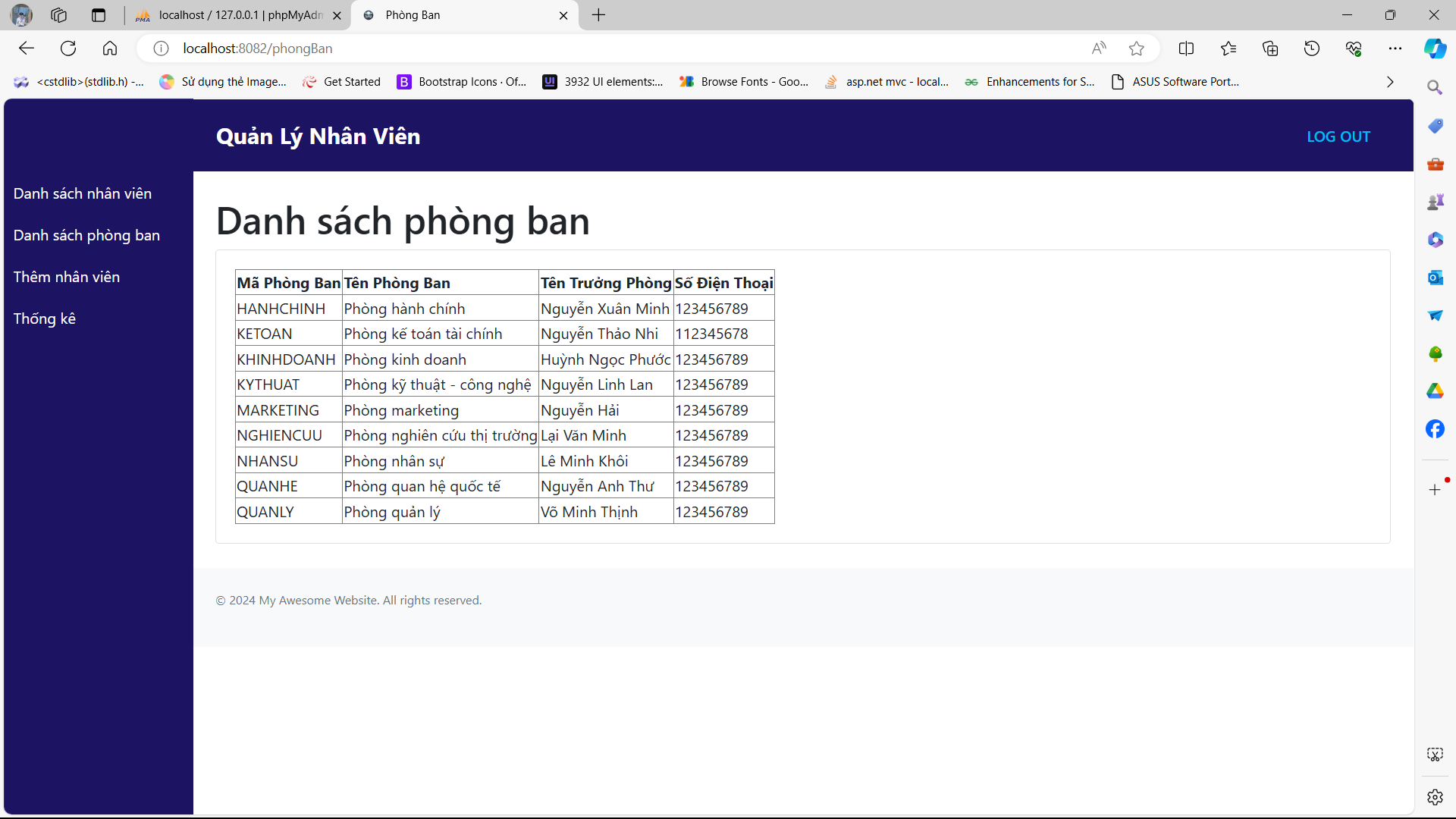
Em cũng tích hợp các dịch vụ như NhanVienService, PhongBanService, và LuongService để quản lý dữ liệu. Trong file HTML, em sử dụng Thymeleaf để hiển thị thông tin nhân viên và thực hiện các hành động như sửa, xóa, và xem chi tiết.

Kết quả là em có một trang web linh hoạt, dễ dàng quản lý và tương tác với dữ liệu nhân viên.



1. Hiển thị danh sách phòng ban

Em đã tái sử dụng lại cách hiển thị của thông tin nhân viên để hiển thị được danh sách phòng ban với các trường mã phòng, tên phòng, tên trưởng phòng,...

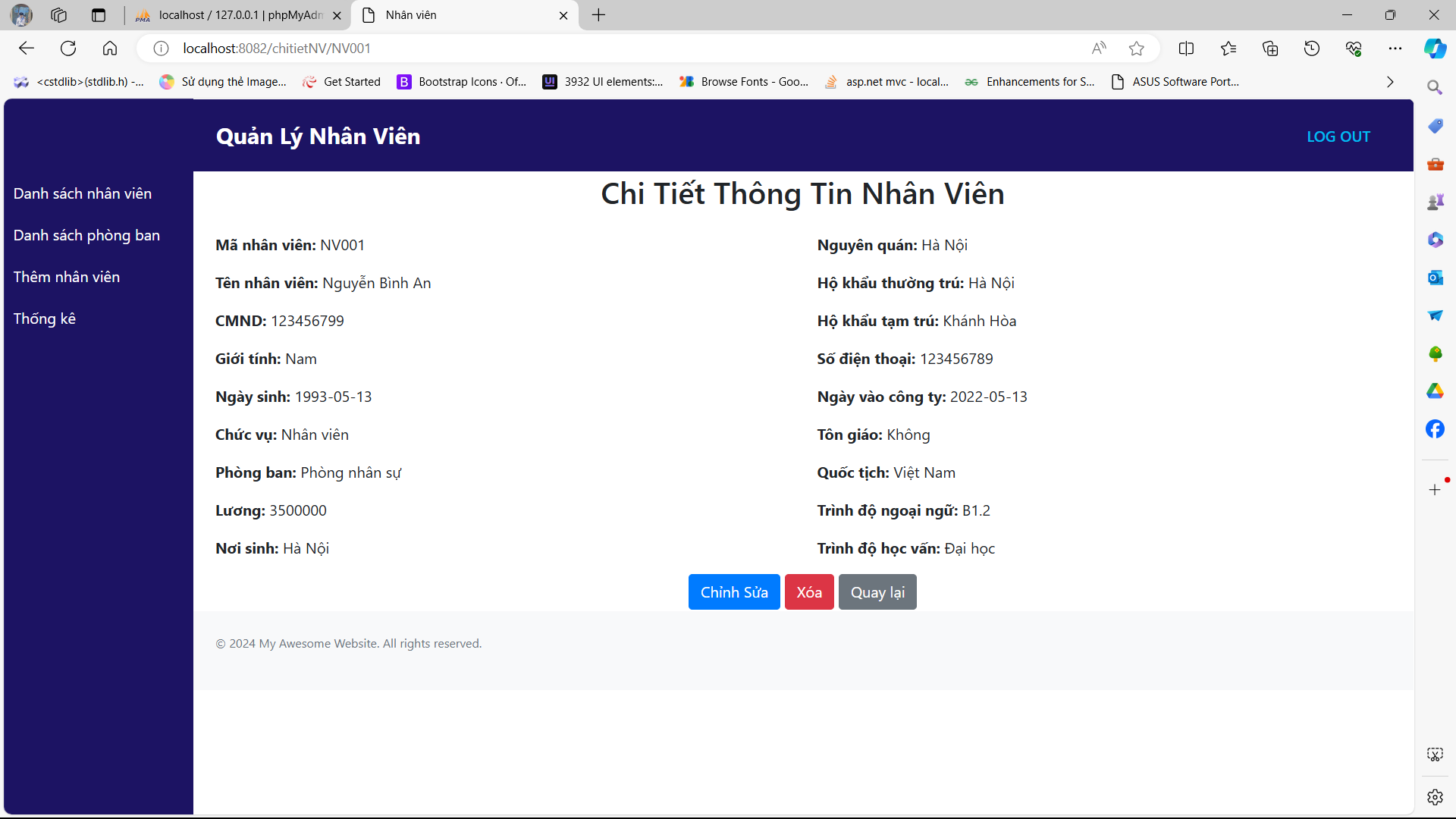


1. Xem chi tiết nhân viên

Để hiển thị trang chi tiết thông tin nhân viên, em sử dụng HTML và Thymeleaf. Trong file HTML này, em tạo các phần tử để hiển thị thông tin chi tiết về nhân viên, bao gồm mã nhân viên, tên, CMND, giới tính, ngày sinh, chức vụ, phòng ban, lương, và các thông tin cá nhân khác. Em sử dụng Thymeleaf để gán dữ liệu từ backend vào các thẻ HTML tương ứng.

Đồng thời, em cũng tích hợp các nút chức năng như chỉnh sửa và xóa nhân viên, kèm theo các câu hỏi xác nhận trước khi thực hiện các hành động này.

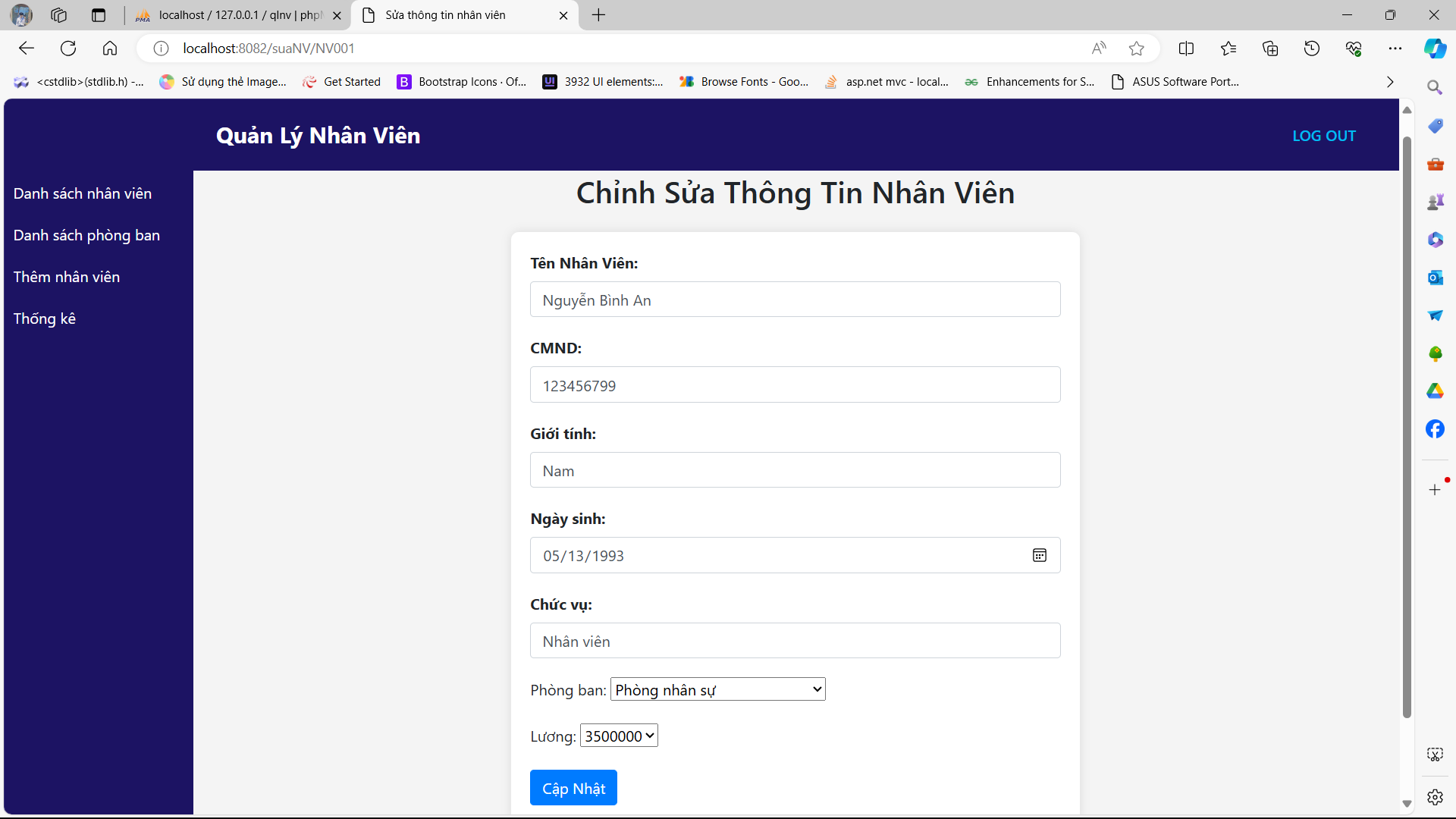
Kết quả là em có một trang web hiển thị thông tin chi tiết nhân viên một cách rõ ràng và dễ dàng tương tác với dữ liệu.



1. Sửa thông tin nhân viên

Để tạo trang sửa thông tin nhân viên, em sử dụng HTML và Thymeleaf để tạo giao diện và tích hợp dữ liệu từ backend. Em thiết kế một form để người dùng nhập liệu và cập nhật thông tin nhân viên. Dữ liệu được gán vào các trường nhập tự động từ thông tin nhân viên được truyền từ backend. Em sử dụng CSS để tạo giao diện thân thiện với người dùng.

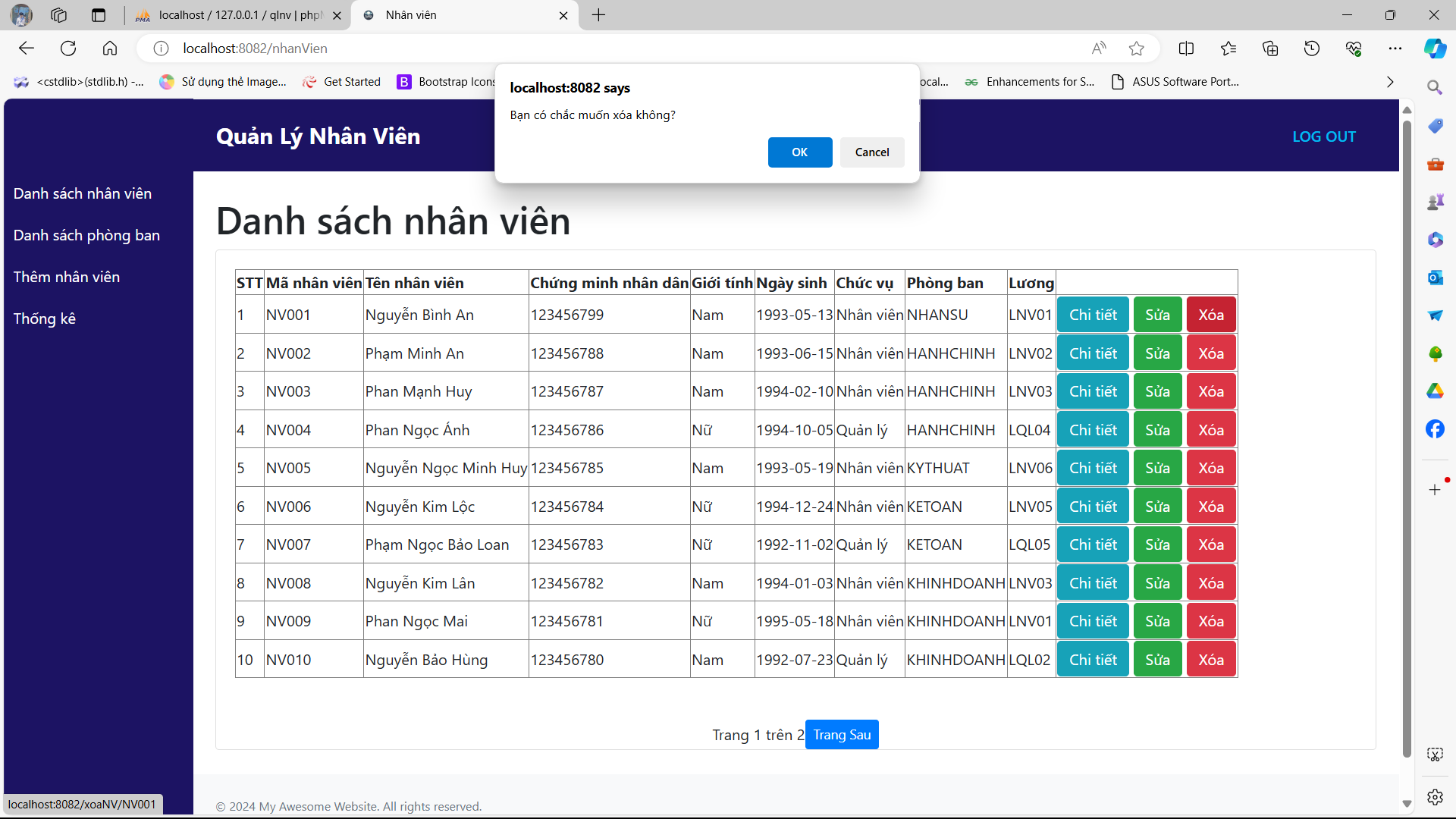
Kết quả là một trang web cho phép người dùng dễ dàng chỉnh sửa thông tin nhân viên và cập nhật vào cơ sở dữ liệu.



1. Xóa nhân viên

Để xóa một nhân viên khỏi cơ sở dữ liệu, em sử dụng một phương thức trong controller được gắn với đường dẫn "/xoaNV/{maNV}". Trong phương thức này, em trước tiên xóa thông tin chi tiết của nhân viên, sau đó xóa thông tin nhân viên từ bảng NhanVien.

Kết quả là thông tin nhân viên được xóa và người dùng được chuyển hướng trở lại trang danh sách nhân viên.

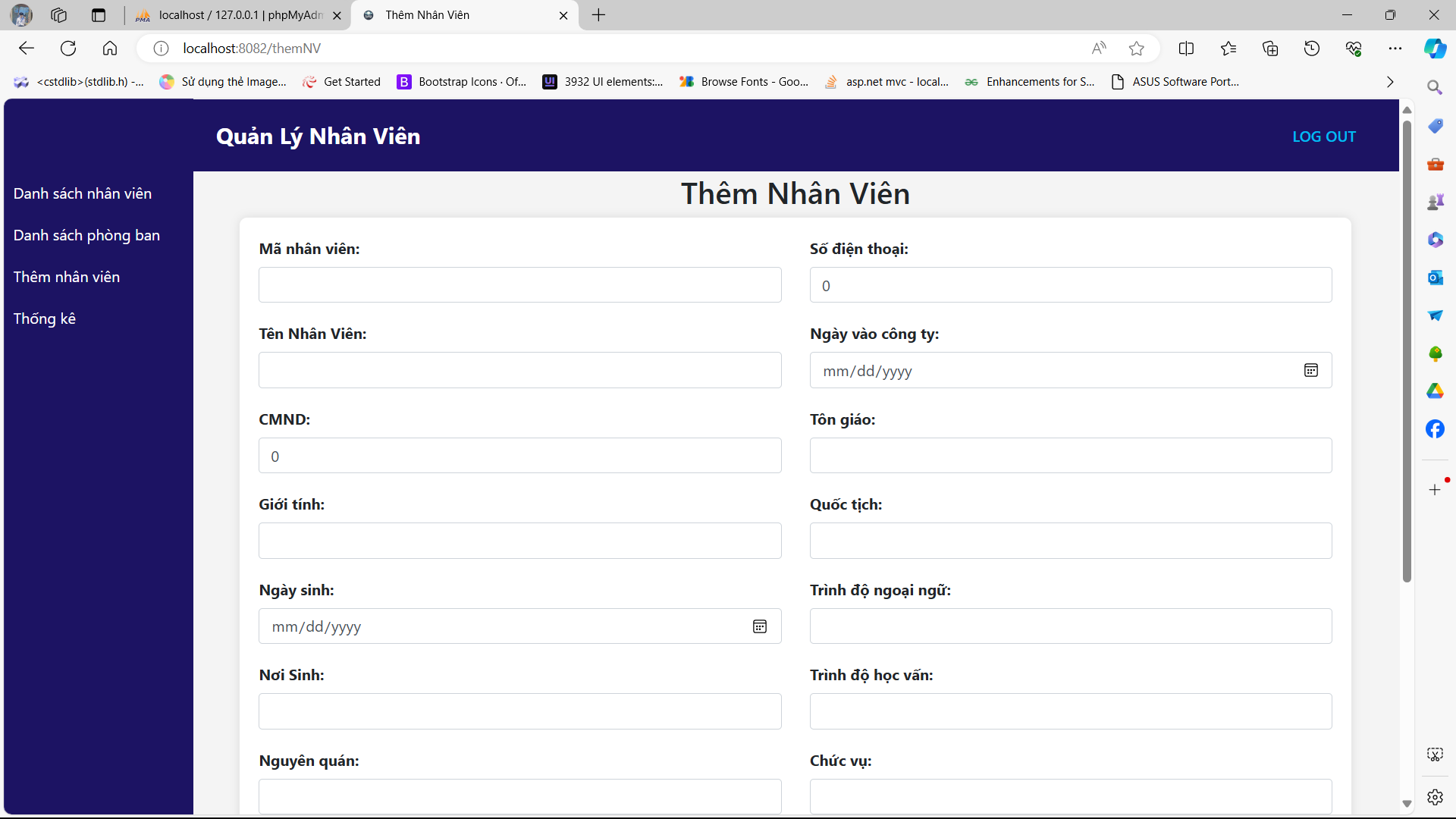


1. Thêm nhân viên

Em sử dụng Thymeleaf để các trường dữ liệu và hiển thị danh sách phòng ban, lương mà nhân viên có thể được gán vào.

Khi người dùng ấn nút "Thêm Nhân Viên", dữ liệu từ form sẽ được gửi đến phương thức themNhanVien thông qua phương thức POST. Trong phương thức này, em kiểm tra xem dữ liệu nhập vào có hợp lệ không thông qua BindingResult. Nếu có lỗi, em hiển thị form lại với thông báo lỗi. Nếu không có lỗi, em lưu thông tin nhân viên và thông tin liên quan, sau đó chuyển hướng người dùng đến trang danh sách nhân viên.

Kết quả đạt được là một giao diện đẹp, tiện lợi cho người dùng thêm nhân viên và tích hợp các chức năng cần thiết để quản lý nhân viên trong hệ thống.



1. Gửi mail

Để gửi email từ ứng dụng web, em đã sử dụng Spring Framework và tích hợp dịch vụ email bằng cách sử dụng EmailService.

Trong form HTML, em sử dụng thuộc tính th:action của Thymeleaf để chỉ định đường dẫn cho việc gửi dữ liệu form. Người dùng có thể nhập địa chỉ email, tiêu đề và nội dung của email để gửi đi. Khi gửi thành công, thông báo "Email đã được gửi thành công!" sẽ được hiển thị trên trang "result".

Kết quả đạt được là khả năng gửi email từ giao diện web và hiển thị thông báo thành công khi email được gửi đi.

