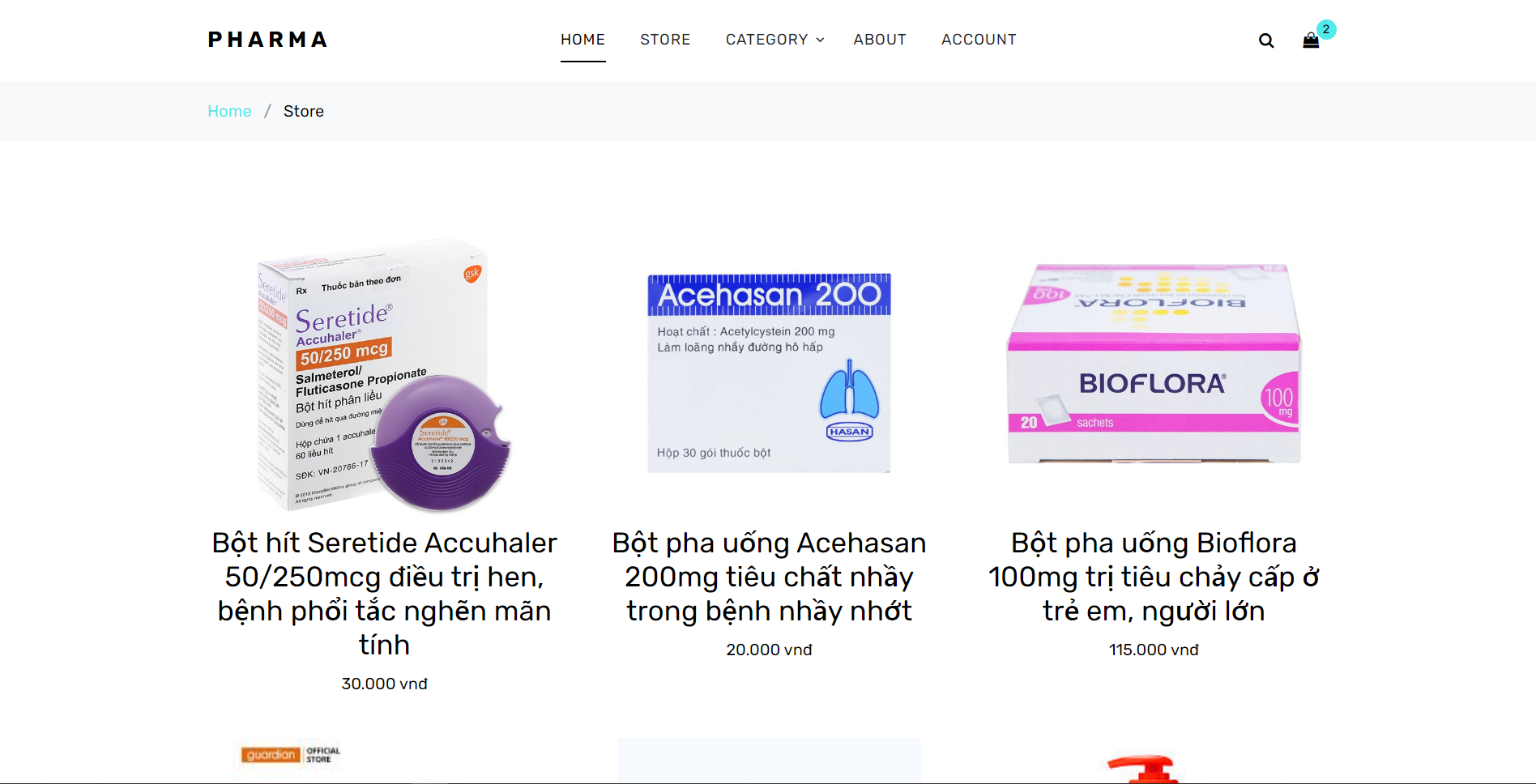
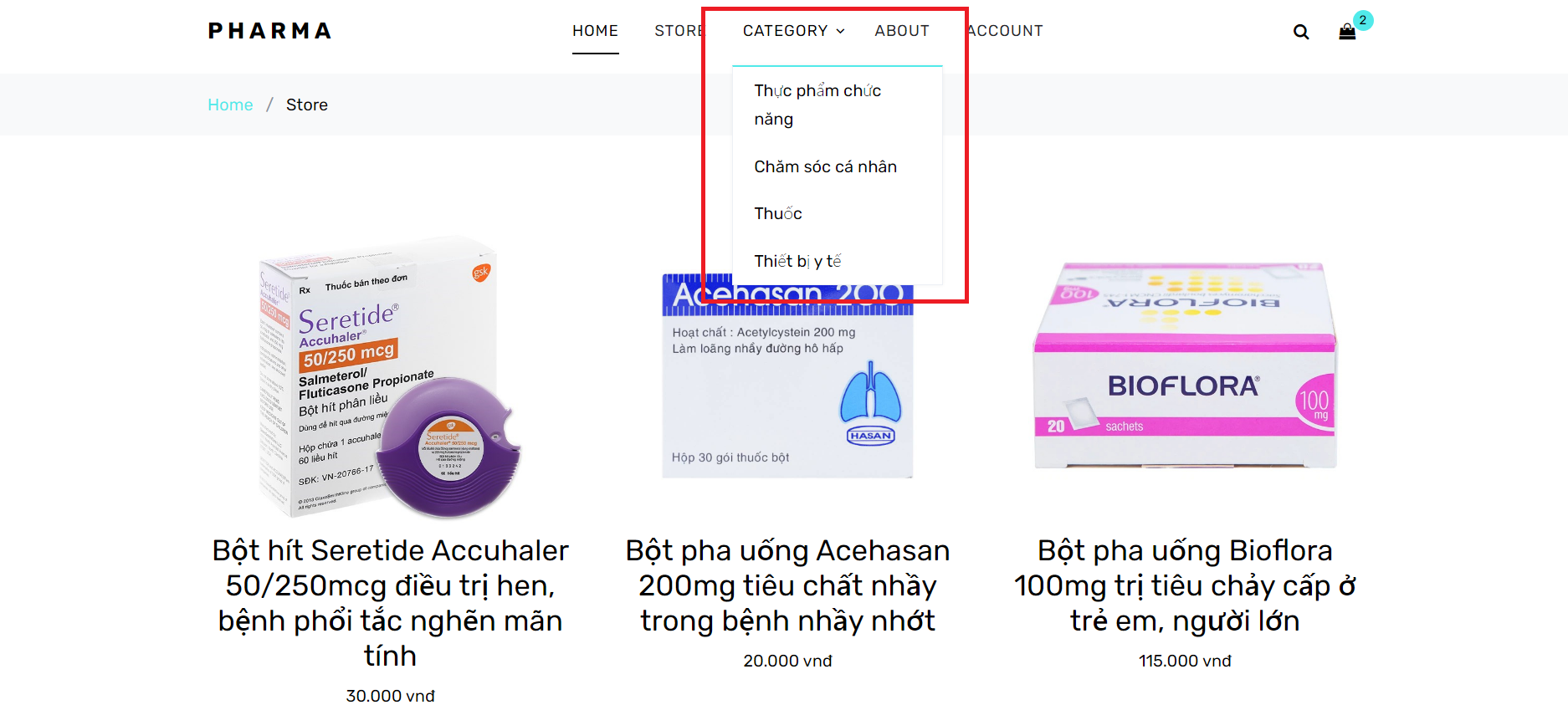


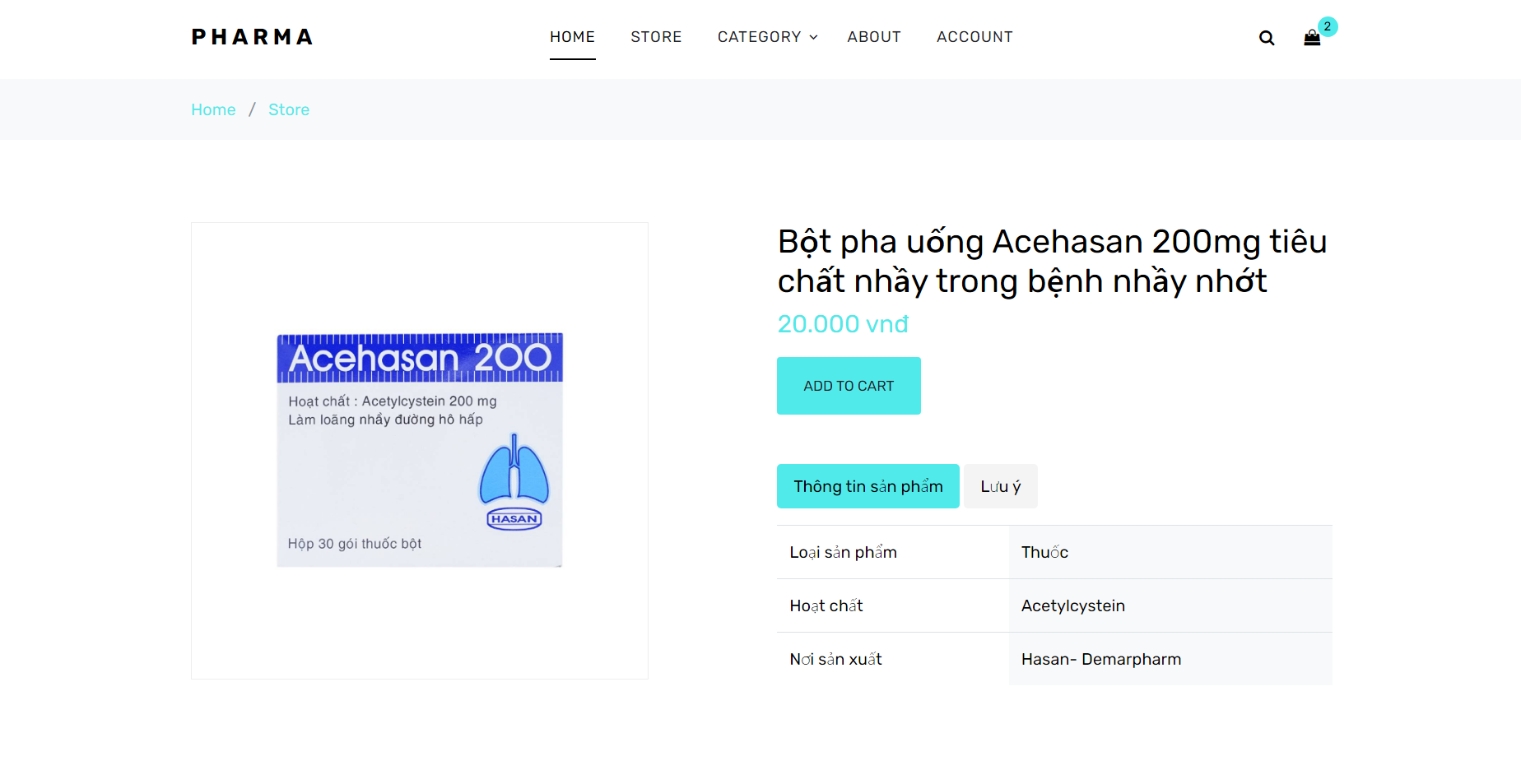
Hình 1 - 1. Trang Home



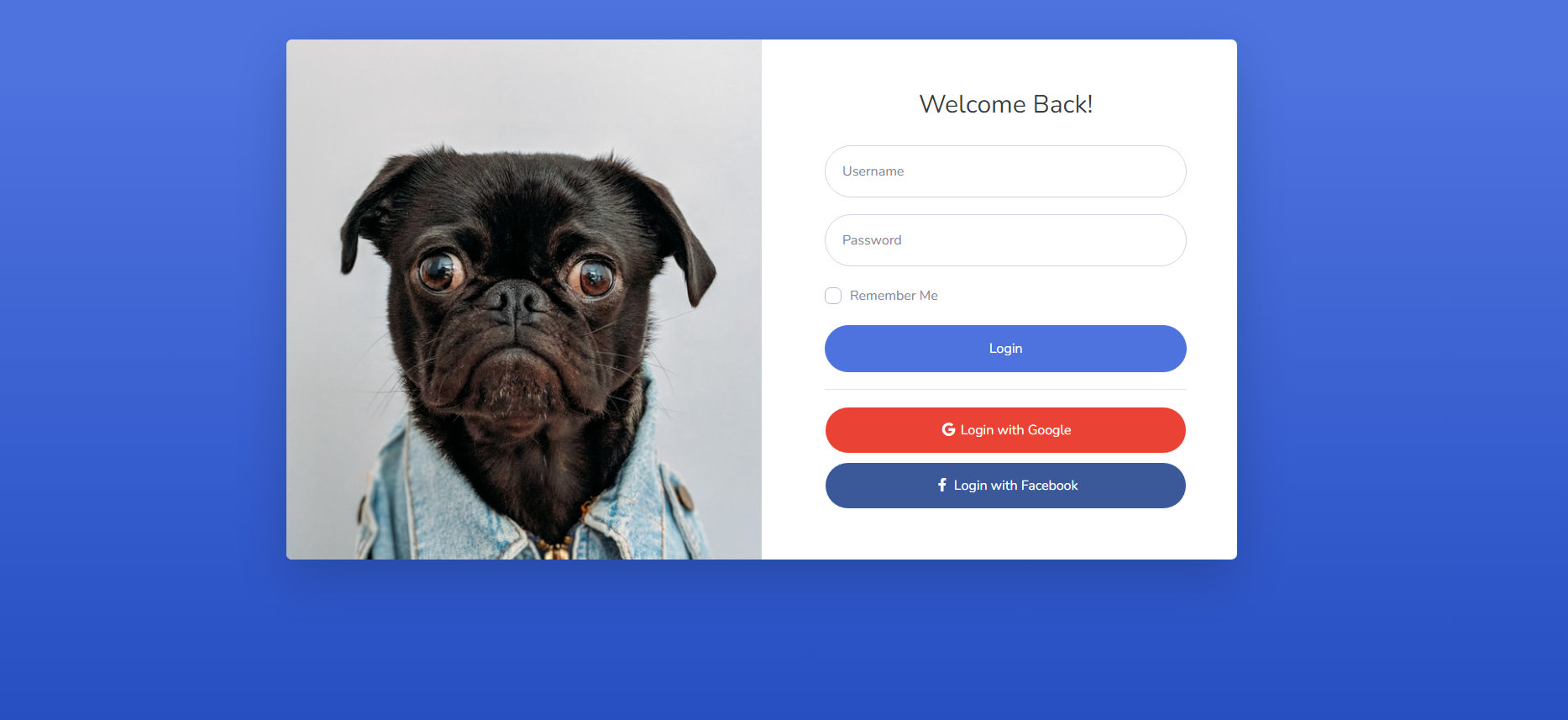
Hình 1 - 2. Trang Shop



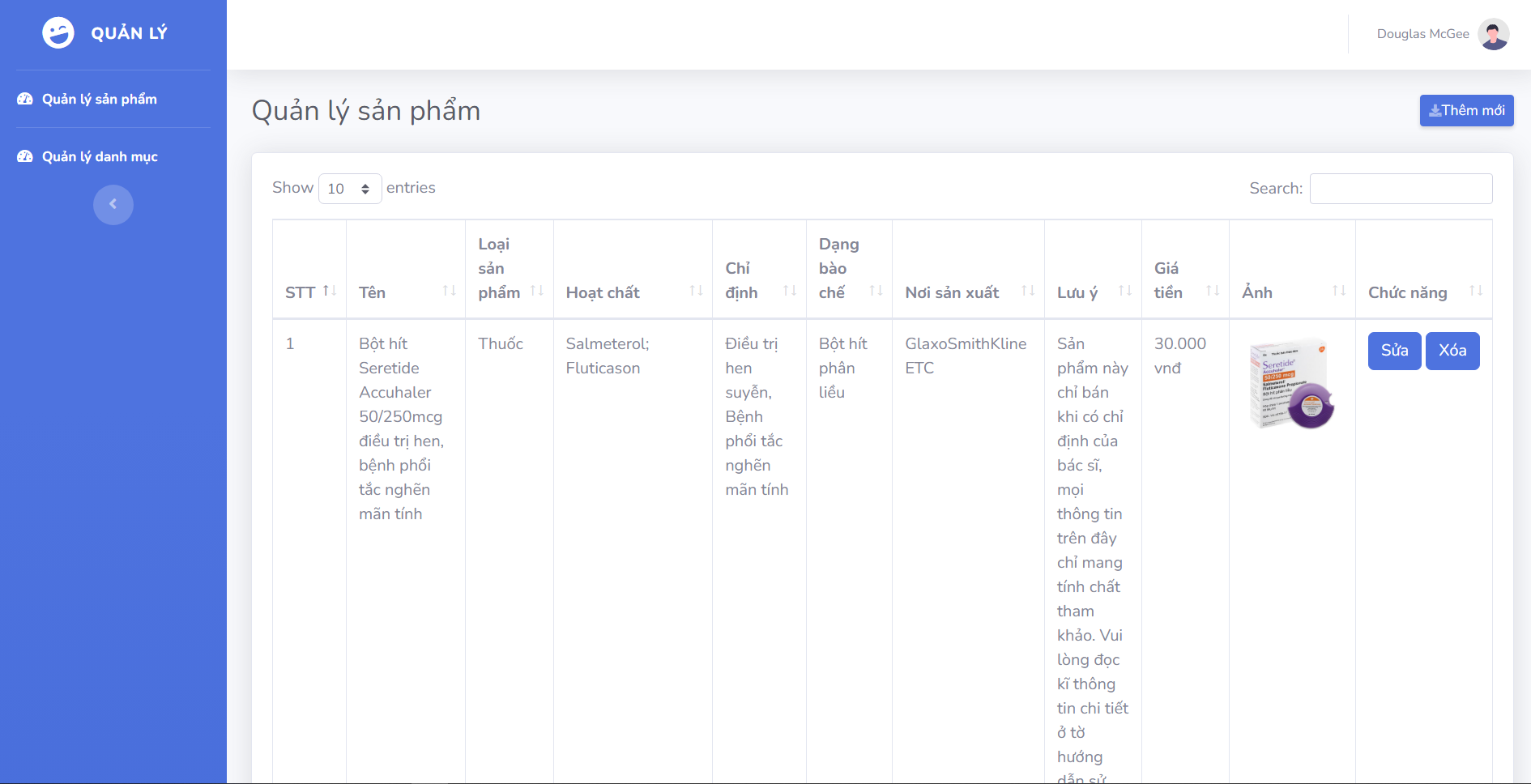
Hình 1 - 3- Danh mục sẽ hiển thị tương ứng với từng sản phẩm



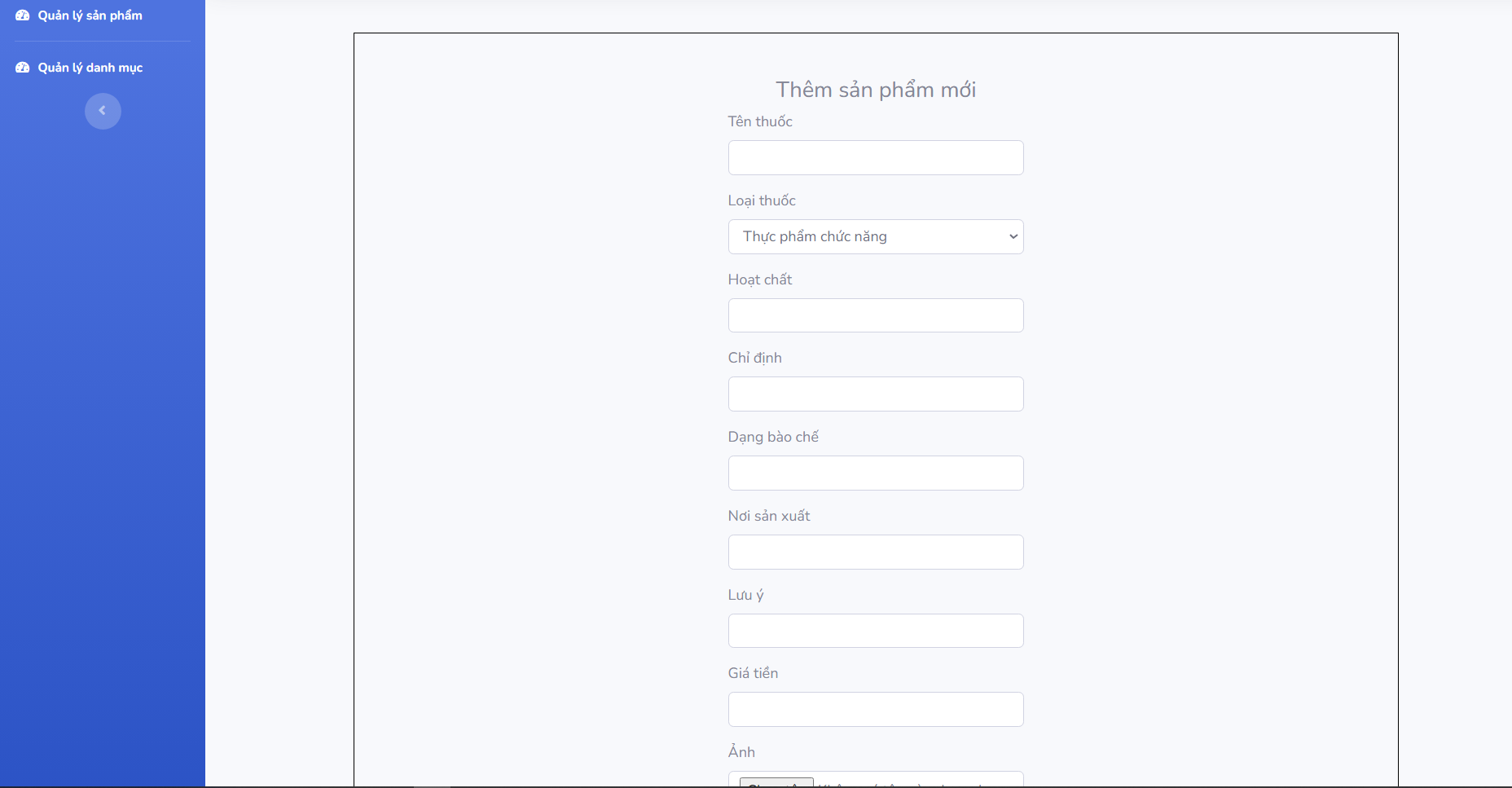
Hình 1 - 4 - Trang chi tiết sản phẩm



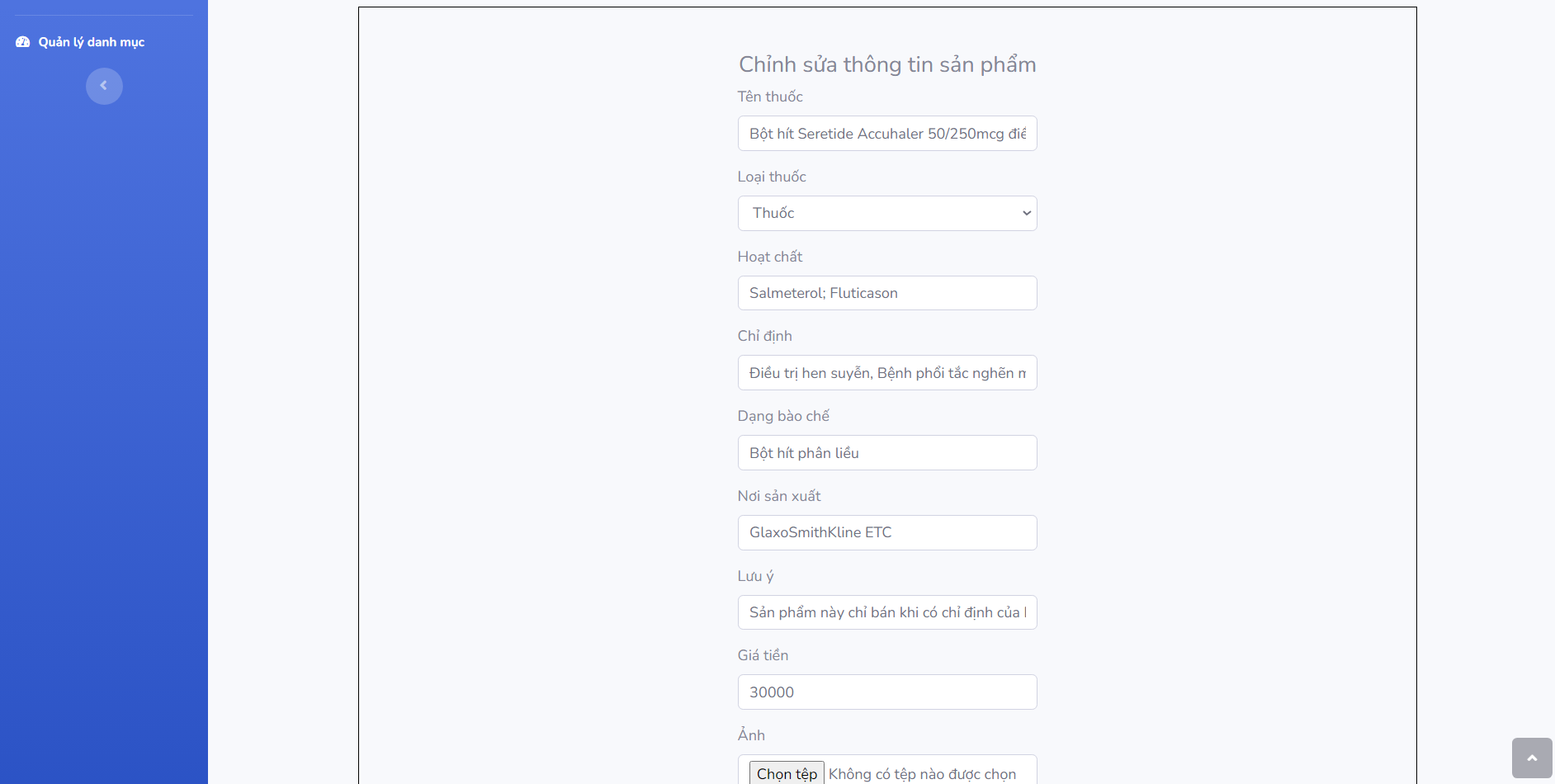
Hình 1 - 5 - Trang Login vào trang quản trị



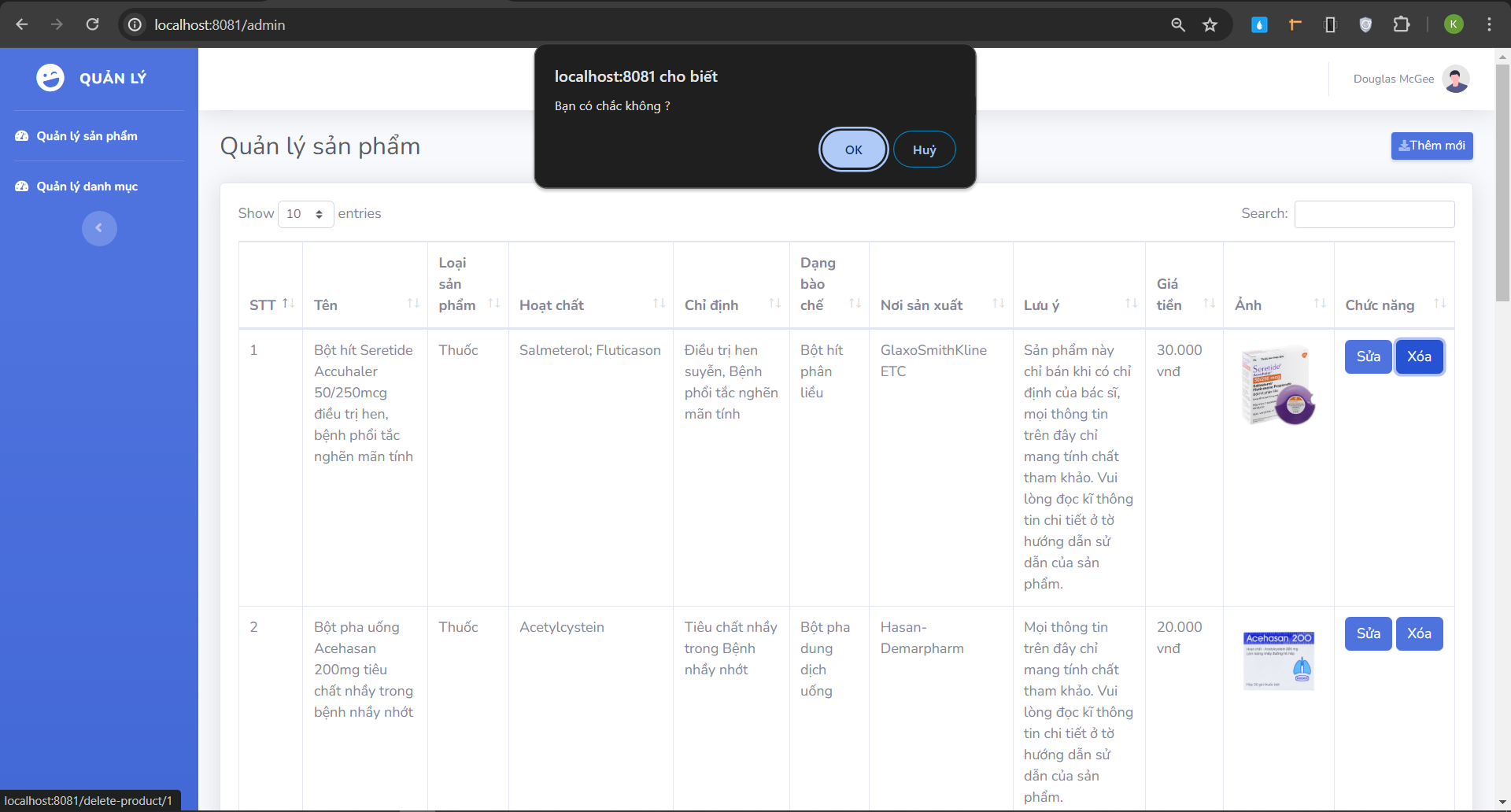
Hình 1 - 6- Trang Quản lý sản phẩm



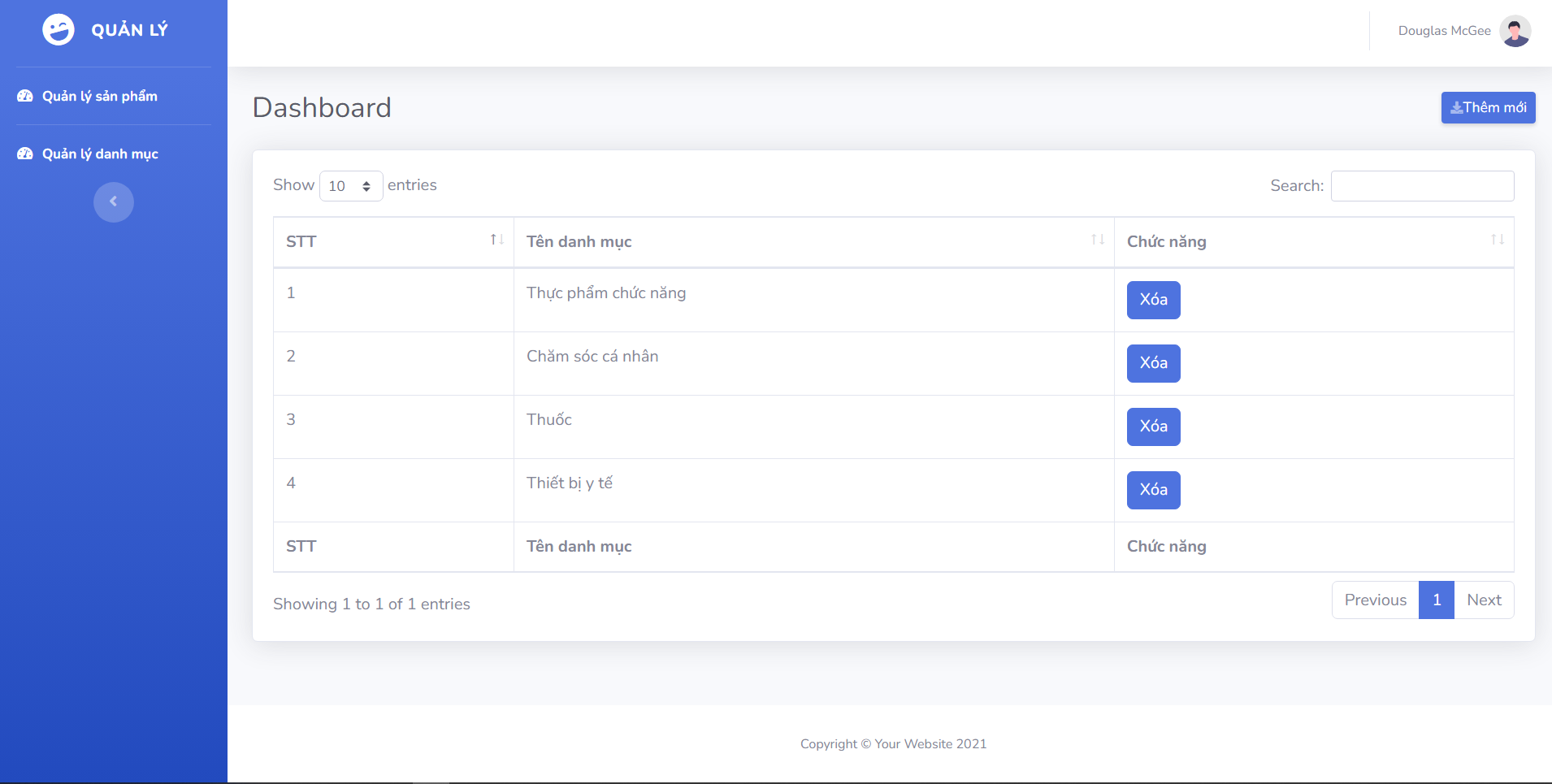
Hình 1 - 7- Trang Thêm sản phẩm



Hình 1 - 8- Trang sửa chi tiết sản phẩm



Hình 1 - 9- Chức năng xóa



Hình 1 - 10 - Trang quản lý danh mục

Ở phần Trang Users em đã áp dụng fragments để phân chia các nội dung giống nhau và tái sử dụng lại các phần nhằm dễ dàng chỉnh sửa. Em đã chia các phần như Header, footer và site-navbar

* Đối với phần hiển thị tất cả sản phẩm em đã áp dụng thuộc tính findAll kế thừa từ lớp repository và gọi lại vào trong controller kết hợp với model tương ứng truyền biến vào và gọi ra từ CSDL bằng lặp ra tất cả sản phẩm (th:each)
* Còn Phần hiển thị danh mục theo từng sản phẩm em sử dụng try-catch để xét và phương thức productService. getProductsByCategoryId (categoryId) được gọi để lấy thông tin sản phẩm với id tương ứng.

Còn ở phần trang quản trị em cũng áp dụng fragments để phân chia các nội dung giống nhau và tái sử dụng lại.

* Em sử dụng Data tables để hiển thị nó có đầy đủ các chức năng như search và phân trang.
* Ở phần edit em cũng lấy id của sản phẩm và sử dụng phương thức và dùng GET và POST, updateProduct ở phần Service để thực thi chức năng
* Ở phần delete em cũng lấy id của sản phẩm và sử dụng phương thức deleteProduct ở phần Service để thực thi chức năng

Ở phần chức năng Login em sử dụng cách làm đơn giản như sau:

* Tìm tài khoản theo trường username (UNIQUE)
* Kiểm tra xem tài khoản có tồn tại (account != null) và mật khẩu nhập vào có khớp với mật khẩu của tài khoản (password.equals(account.getPassword())).
* Nếu cả hai điều kiện đều đúng, tức là thông tin đăng nhập hợp lệ và sẽ tới trang quản trị.

Phần chưa làm được:

* Giỏ hàng
* Chi tiết đơn hàng