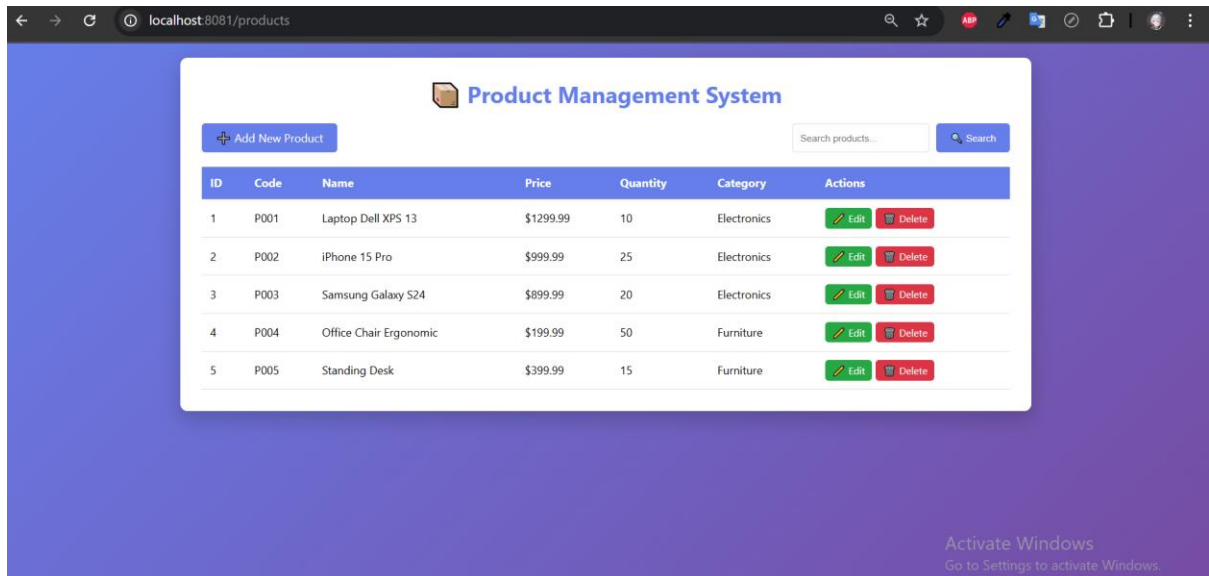


Lab 7 Practice

Name: Nguyễn Hữu Hoàng Nam

StudentID: ITCSIU23028

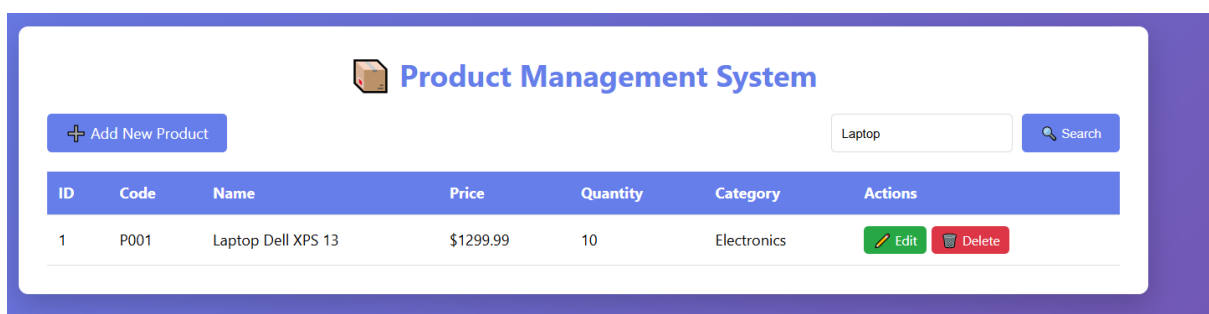
Product Page



Flow:

1. User accesses `/products`
2. Browser sends GET request to ProductController
3. Controller calls `productService.getAllProducts()`
4. Service calls `productRepository.findAll()`
5. Repository queries database and returns List<Product>
6. Controller adds products to Model
7. Returns "product-list" view
8. Thymeleaf renders product-list.html with product data
9. User sees table with all products

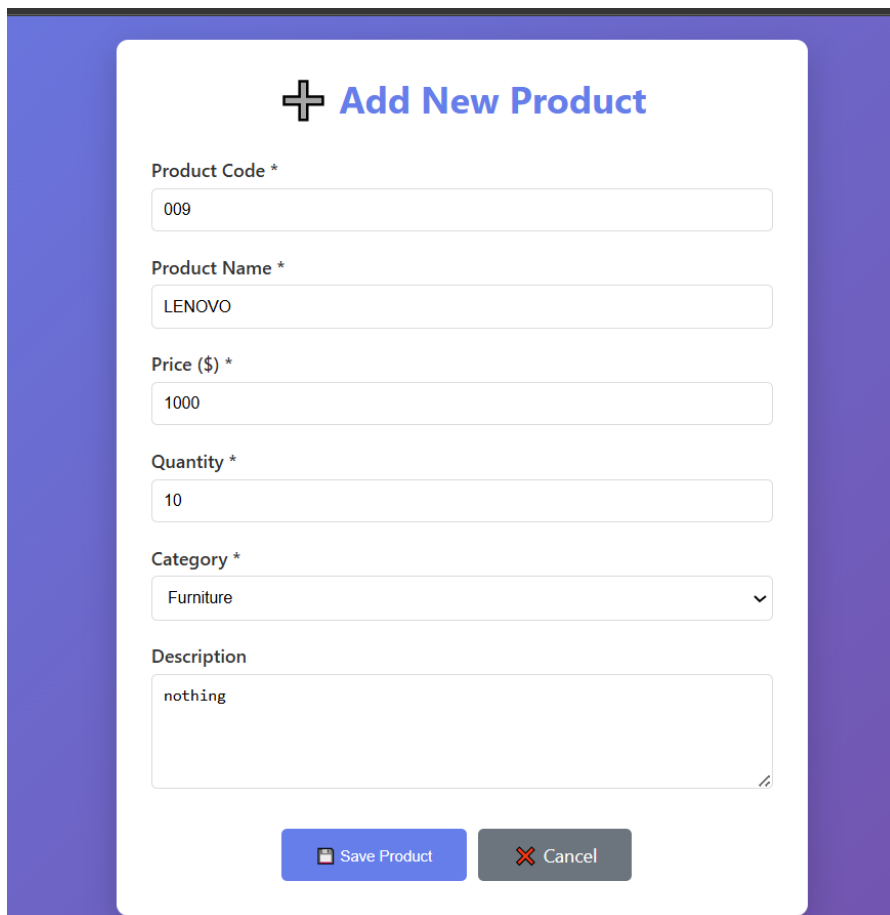
Search Function



Flow:

1. User types keyword in search box and clicks Search button
2. Browser sends GET request to `/products/search?keyword=laptop`
3. Controller receives keyword parameter
4. Controller calls `productService.searchProducts(keyword)`
5. Service calls `productRepository.findByNameContainingIgnoreCase(keyword)`
6. Repository searches database for products with name containing keyword
8. Controller adds filtered products and keyword to Model
9. Returns "product-list" view
10. User sees only matching products

Add new Product feature



+ Add New Product

Product Code *
009



Product Name *
LENOVO

Price (\$) *
1000

Quantity *
10

Category *
Furniture

Description
nothing

 Save Product  Cancel

<div>  Product Management System </div>						
<div> <div>+</div> Add New Product </div>			<div> <input type="text" value="Search products..."/> <div>  Search </div> </div>			
ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	10	Electronics	<div>  Edit  Delete </div>
2	P002	iPhone 15 Pro	\$999.99	25	Electronics	<div>  Edit  Delete </div>
3	P003	Samsung Galaxy S24	\$899.99	20	Electronics	<div>  Edit  Delete </div>
4	P004	Office Chair Ergonomic	\$199.99	50	Furniture	<div>  Edit  Delete </div>
5	P005	Standing Desk	\$399.99	15	Furniture	<div>  Edit  Delete </div>
6	009	LENOVO	\$1000.00	10	Furniture	<div>  Edit  Delete </div>

Flow:

1. User fills in product details (code, name, price, quantity, category, description)
2. User clicks Save button
3. Browser sends POST request to `/products/save` with form data
4. Controller receives Product object with form data
5. Controller calls `productService.saveProduct(product)`
6. Service calls `productRepository.save(product)`
7. Repository inserts new product into database
9. Controller adds success message
10. Redirects to `/products`
11. User sees updated product list with new product

Edit Product



Edit Product

Product Code *

P001

Product Name *

Laptop Dell XPS 13

Price (\$) *

1299.99

Quantity *

1

Category *

Electronics



Description

High-performance laptop with Intel i7



Save Product



Cancel

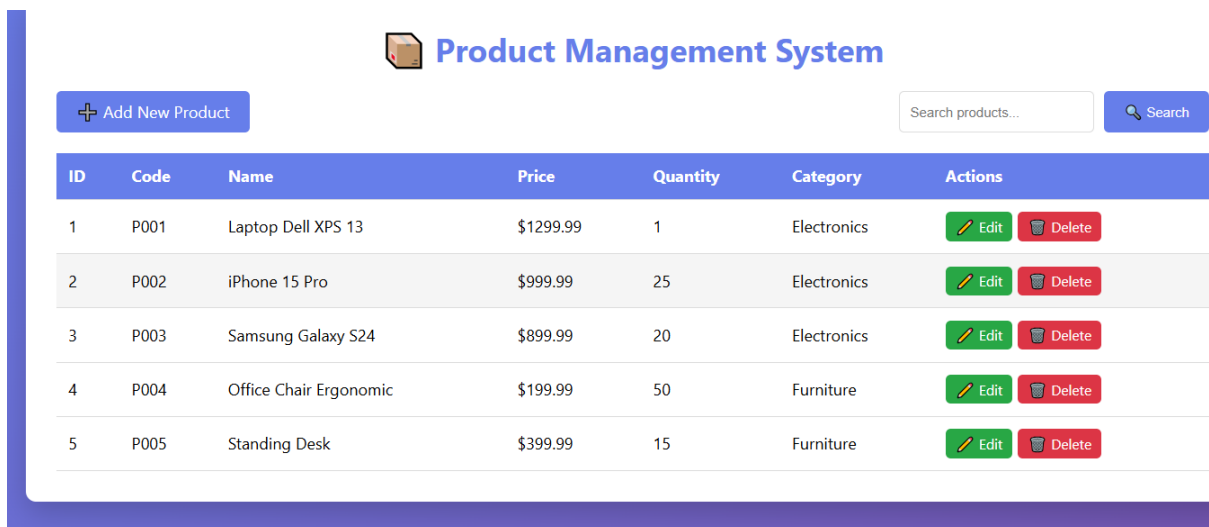
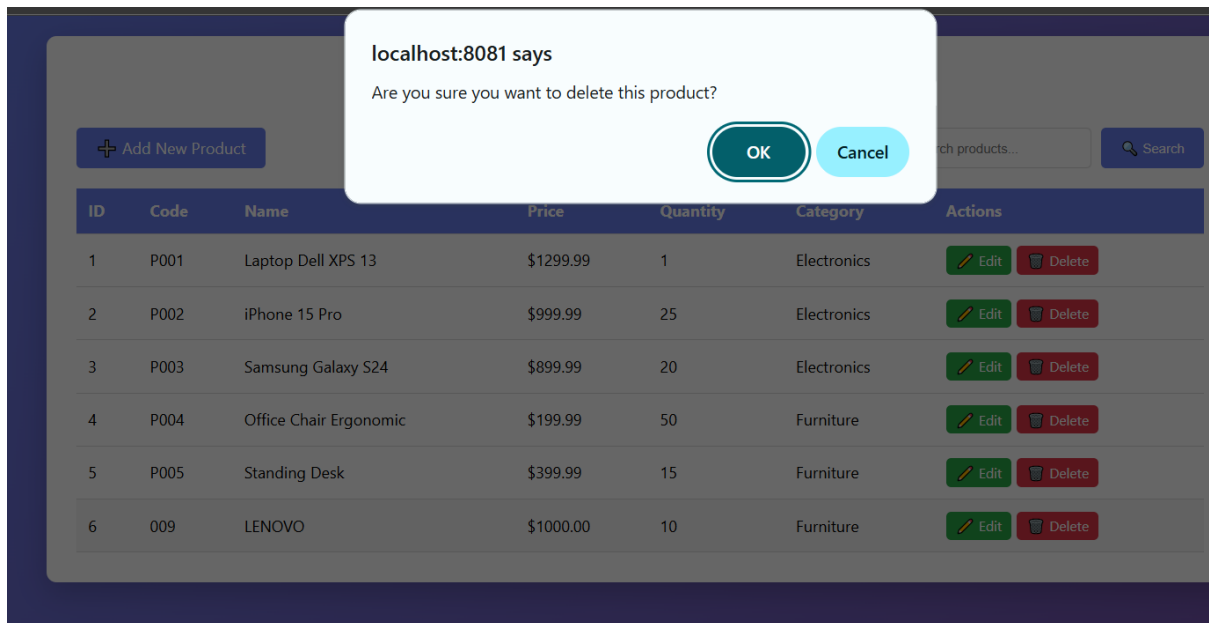
ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	1	Electronics	Edit Delete

Flow:

1. User modifies product details in the form
2. User clicks Save button
3. Browser sends POST request to `/products/save` with updated data
4. Controller receives Product object with id and updated data
5. Controller calls `productService.saveProduct(product)`
6. Service calls `productRepository.save(product)`

- Repository updates existing product in database (because id exists)
- Controller adds success message
- Redirects to `/products`
- User sees updated product list

Delete Product



Flow:

- User clicks "Delete" button next to a product in the list
- Browser sends GET request to `/products/delete/1` (where 1 is the product ID)
- Controller receives id from URL path
- Controller calls `productService.deleteProduct(id)`
- Service calls `productRepository.deleteById(id)`

6. Repository deletes product from database
7. Controller adds success message "Product deleted successfully!"
8. Redirects to `/products`
9. User sees updated product list without the deleted product