

## Lab 7 Homework

Name: Nguyễn Hữu Hoàng Nam

StudentID: ITCSIU23028

### Exercise 5:

[+ Add New Product](#)  [Search](#)

[Advanced Search](#)

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	1	Electronics	<a href="#">Edit</a> <a href="#">Delete</a>
2	P002	iPhone 15 Pro	\$999.99	25	Electronics	<a href="#">Edit</a> <a href="#">Delete</a>
3	P003	Samsung Galaxy S24	\$899.99	20	Electronics	<a href="#">Edit</a> <a href="#">Delete</a>
4	P004	Office Chair Ergonomic	\$199.99	50	Furniture	<a href="#">Edit</a> <a href="#">Delete</a>
5	P005	Standing Desk	\$399.99	15	Furniture	<a href="#">Edit</a> <a href="#">Delete</a>

[Activate Windows](#)  
[Go to Settings to activate](#)

[Product Management System](#)

[+ Add New Product](#)  [Search](#)

[Advanced Search](#)

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	1	Electronics	<a href="#">Edit</a> <a href="#">Delete</a>
2	P002	iPhone 15 Pro	\$999.99	25	Electronics	<a href="#">Edit</a> <a href="#">Delete</a>
4	P004	Office Chair Ergonomic	\$199.99	50	Furniture	<a href="#">Edit</a> <a href="#">Delete</a>
5	P005	Standing Desk	\$399.99	15	Furniture	<a href="#">Edit</a> <a href="#">Delete</a>

[Activate Windows](#)  
[Go to Settings to activate](#)

## Product Management System

[+ Add New Product](#)

Search products...

Search

Advanced Search

Product Name:

E

Category:

Electronics

Min Price:

0.09

Max Price:

Max price

Advanced Search

Clear Filters

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	1	Electronics	Edit  Delete
2	P002	iPhone 15 Pro	\$999.99	25	Electronics	Edit  Delete

## Product Management System

[+ Add New Product](#)

Search products...

Search

Advanced Search

Product Name:

Search by name...

Category:

All Categories

Min Price:

Min price

Max Price:

Max price

Advanced Search

Clear Filters

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

[« Previous](#) [1](#) [2](#) [3](#) [Next »](#)

Showing page 1 of 5 (Total: 5 products)

### Flow(Advanced Search):

1. User fills in advanced search form with criteria.
2. User clicks "Advanced Search" button.
3. Browser sends GET request to `/products/advanced-search` with all parameters.
4. Controller receives parameters (name, category, minPrice, maxPrice).
5. Controller calls `productService.advancedSearch`.
6. Service calls `productRepository.searchProducts`.
7. Repository executes custom JPQL query .
8. Repository returns filtered List<Product>.
9. Controller adds filtered products to Model.
10. Returns "product-list" view.
11. User sees only products matching ALL criteria.

**Flow(Search with pagination):**

1. User types keyword in search box and clicks Search
2. Browser sends GET request to `/products/search?keyword=laptop&page=0&size=10`
3. Controller receives parameters:
5. Controller calls `productService.searchProducts(keyword, pageable)`
6. Service calls `productRepository.findByNameContainingIgnoreCase(keyword, pageable)`
7. Repository executes paginated query:
9. Controller extracts data from Page object and adds to Model:
10. Returns "product-list" view

**Exercise 6:**

## Add New Product

Product Code \*

1

Product code must be 3-20 characters

Product code must start with P followed by numbers

Product Name \*

2

Name must be 3-100 characters

Price (\$) \*

3

Quantity \*

4

Category \*

Electronics



Description

4

 Save Product

 Cancel

### Flow:

1. User fills in product form with valid data:
2. User clicks Save button
3. Browser sends POST request to `/products/save` with form data
4. Controller receives Product object
5. `@Valid` annotation triggers validation

6. Spring Validator checks all constraints - ALL PASS
7. BindingResult contains no errors
8. Controller checks: `if (result.hasErrors())` → FALSE
9. Controller proceeds to save:
  - Calls `productService.saveProduct(product)`
  - Service calls `productRepository.save(product)`
  - Repository saves to database
10. Controller adds success message: `"Product saved successfully!"`
11. Redirects to `/products`
12. User sees product list with success message

### Exercise 7:

The screenshot shows a web application titled "Product Management System". At the top, there is a navigation bar with a "Dashboard" button, an "Add New Product" button, a search input field, and a "Search" button. Below the navigation bar, there is a "Filter by Category" section with buttons for "All Categories", "Electronics", "Clothing", "Books", "Furniture", "Food", and "Sports". There is also an "Advanced Search" section with fields for "Product Name", "Category", "Min Price", and "Max Price", along with "Advanced Search" and "Clear Filters" buttons. The main content area displays a table of products with columns: ID, Code, Name, Price, Quantity, Category, and Actions. The table contains four rows of data.

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	\$1299.99	1	Electronics	<button>Edit</button> <button>Delete</button>
2	P002	iPhone 15 Pro	\$999.99	25	Electronics	<button>Edit</button> <button>Delete</button>
3	P003	Samsung Galaxy S24	\$899.99	20	Electronics	<button>Edit</button> <button>Delete</button>
7	1	2	\$3.00	4	Electronics	<button>Edit</button> <button>Delete</button>

### Flow

1. User selects category "Electronics" AND sorts by "Price" ascending
2. Browser sends GET request with both parameters
3. Controller receives category, sortBy, sortDir
4. If category provided:
  - Calls `productService.getProductsByCategory(category)`
  - Applies manual sorting using Stream API and Comparator
5. If no category (all products):

- Creates Sort object
  - Calls `productService.getAllProducts(sort)`
6. Returns sorted and filtered List<Product>
7. Controller adds products, category, sortBy, sortDir to Model
8. Returns "product-list" view

### Exercise 8:

The screenshot shows a dashboard for a Product Management System. At the top, there's a header with a bar chart icon and the title "Dashboard - Product Management System". Below the header are four cards: "Total Products" (6 items in inventory), "Total Inventory Value" (\$60310.89 total worth), "Average Product Price" (\$633.82 per item), and "Low Stock Alerts" (2 items need restock). Underneath these cards is a section titled "Products by Category" with two buttons: "Electronics" and "Furniture". Below this is a "Low Stock Alerts" section with a table showing two items: Laptop Dell XPS 13 (1 unit) and Office Chair Ergonomic (4 units). Finally, there's a "Recently Added Products" section with a table listing five products: Laptop Dell XPS 13, iPhone 15 Pro, Samsung Galaxy S24, Office Chair Ergonomic, and another entry for Laptop Dell XPS 13.

PRODUCT CODE	NAME	CATEGORY	QUANTITY	PRICE
P001	Laptop Dell XPS 13	Electronics	1	\$1299.99
1	2	Electronics	4	\$3.00

PRODUCT CODE	NAME	CATEGORY	PRICE	QUANTITY	ADDED DATE
1	2	Electronics	\$3.00	4	2025-12-07 17:03
P001	Laptop Dell XPS 13	Electronics	\$1299.99	1	2025-12-01 16:08
P002	iPhone 15 Pro	Electronics	\$999.99	25	2025-12-01 16:08
P003	Samsung Galaxy S24	Electronics	\$899.99	20	2025-12-01 16:08
P004	Office Chair Ergonomic	Furniture	\$199.99	50	2025-12-01 16:08

### Flow:

1. User clicks "Dashboard" button from product list.
2. Browser sends GET request to `/dashboard` .
3. Controller gathers 6 types of statistics from ProductService.
4. Service calls repository methods with @Query annotations
5. Repository executes SQL queries .
6. Controller adds all statistics to Model.
7. Returns "dashboard" view.