Trần Phúc Quý - ITITDK23027

LAB 7

FEATURE CODE FLOWS

1. LIST PRODUCTS FEATURE

Flow:

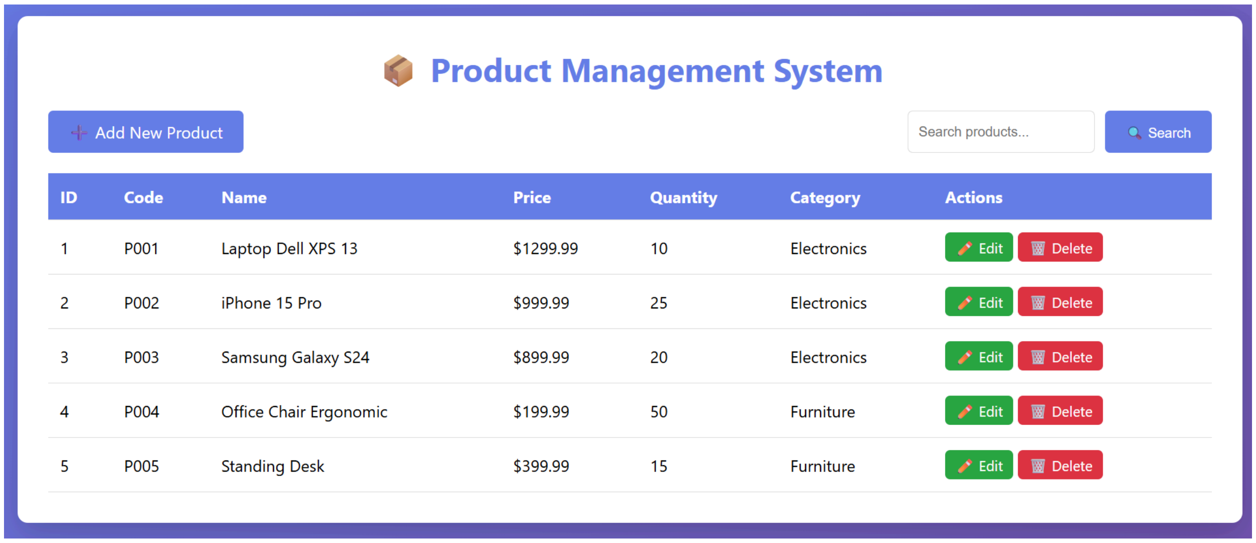
1. User accesses http://localhost:8080/products
2. ProductController.listProducts() method invoked (@GetMapping)
3. Calls productService.getAllProducts()
4. ProductService calls productRepository.findAll()
5. Spring Data JPA executes: SELECT * FROM products
6. Results added to Model: model.addAttribute("products", products)
7. Returns "product-list" view name
8. Thymeleaf renders product-list.html with data
9. Table displays all products using th:each loop

Code Snippet (Controller):

```
@GetMapping
public String listProducts(Model model) {
   List<Product> products = productService.getAllProducts();
   model.addAttribute("products", products);
   return "product-list";
}
```

View Rendering (Thymeleaf):

```
<tr th:each="product : ${products}">
   <td th:text="${product.id}"></td>
   <td th:text="${product.name}"></td>
   <td th:text="${product.price}"></td>
</tr>
```



ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	$1299.99	10	Electronics	Edit Delete
2	P002	iPhone 15 Pro	$999.99	25	Electronics	Edit Delete
3	P003	Samsung Galaxy S24	$899.99	20	Electronics	Edit Delete
4	P004	Office Chair Ergonomic	$199.99	50	Furniture	Edit Delete
5	P005	Standing Desk	$399.99	15	Furniture	Edit Delete

2. CREATE PRODUCT FEATURE

Flow:

1. User clicks “Add New Product” button

2. GET /products/new - Shows empty form - ProductController.showNewForm() creates new Product object - Returns product-form.html view

3. User fills form and clicks Submit

4. POST /products/save - Processes form data - @ModelAttribute binds form data to Product object - Calls productService.saveProduct(product) - Repository executes INSERT statement - RedirectAttributes adds success message - Redirects to /products (PRG pattern)

5. Success message displayed on list page

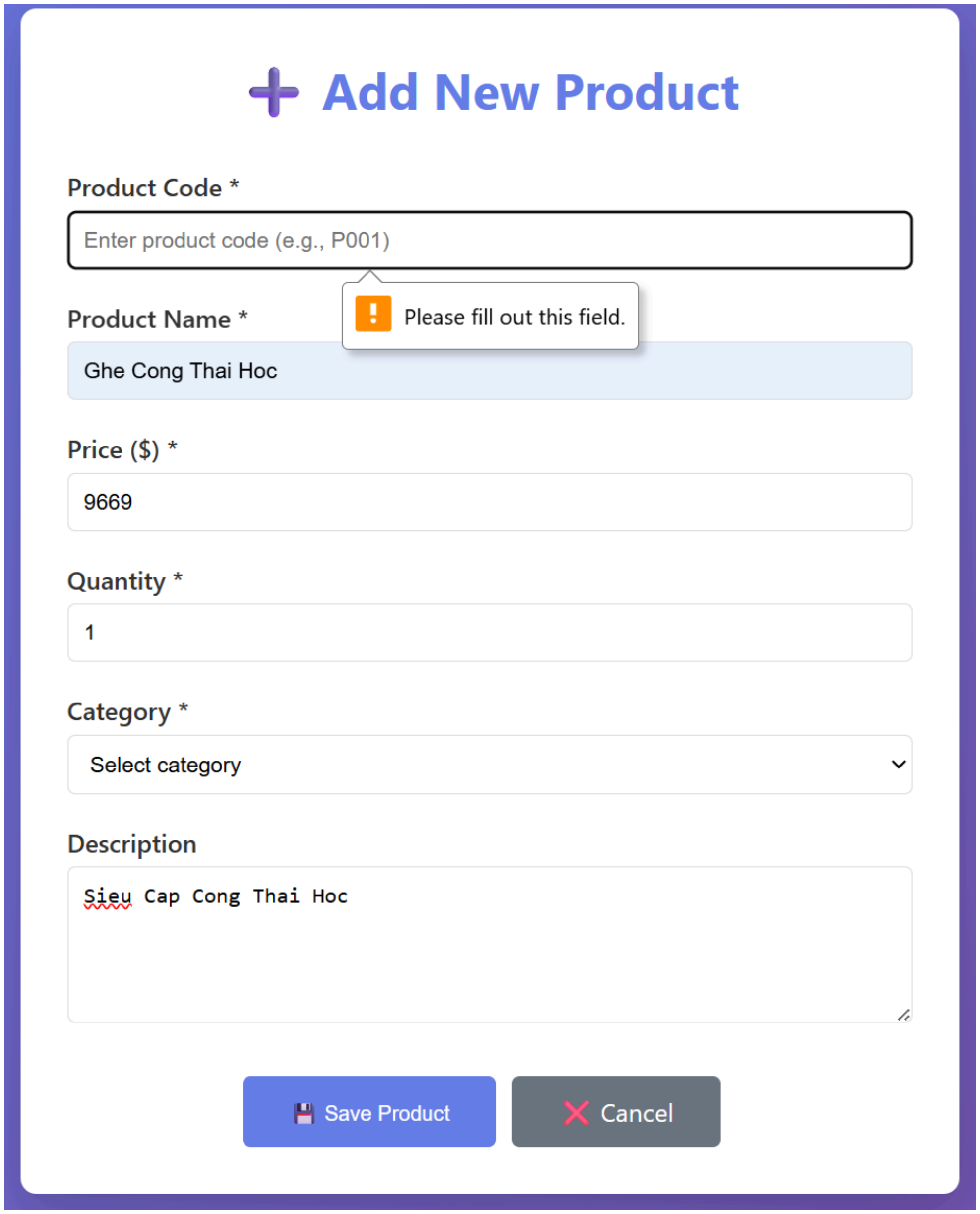
Code Snippet (Controller):

```
@GetMapping("/new")
public String showNewForm(Model model) {
    Product product = new Product();
    model.addAttribute("product", product);
    return "product-form";
}

@PostMapping("/save")
public String saveProduct(@ModelAttribute("product") Product product,
                         RedirectAttributes redirectAttributes) {
    productService.saveProduct(product);
    redirectAttributes.addFlashAttribute("message", "Product added successfully!");
    return "redirect:/products";
}
```

Form Binding (Thymeleaf):

```
<form th:action="@{/products/save}" th:object="${product}" method="post">
  <input type="text" th:field="*{name}" />
  <input type="number" th:field="*{price}" />
</form>
```



Add New Product

Product Code *

Enter product code (e.g., P001)

Please fill out this field.

Product Name *

Ghe Cong Thai Hoc

Price ($) *

9669

Quantity *

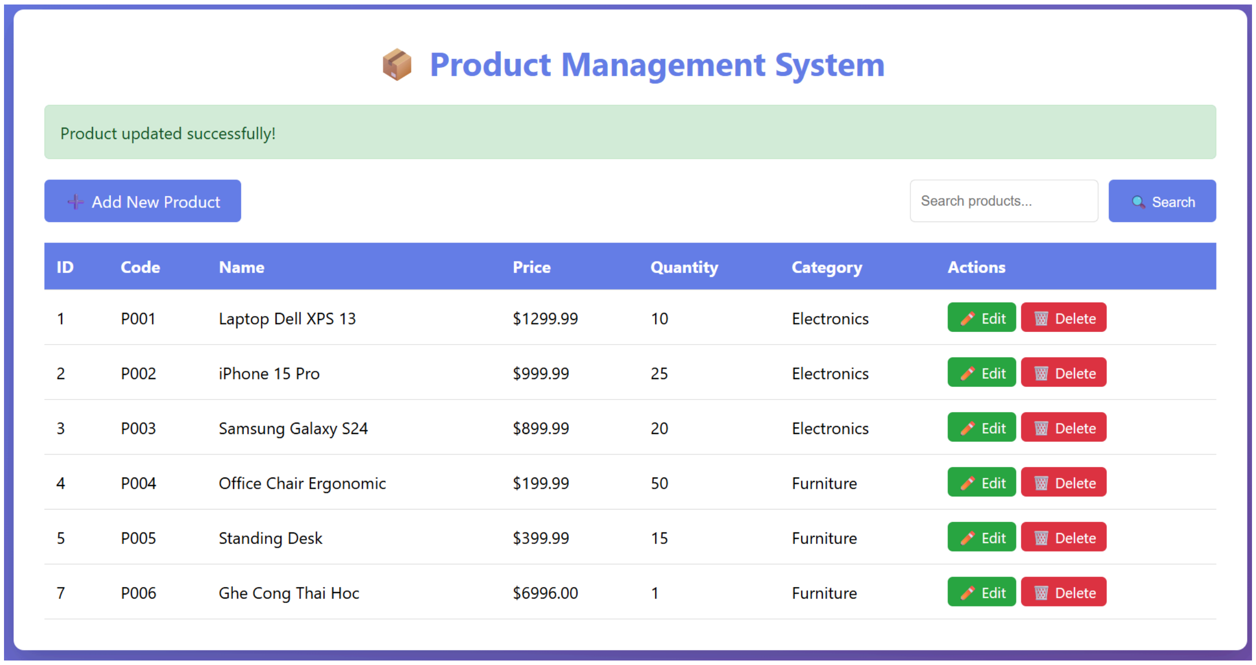
1

Category *

Select category

Description

Sieu Cap Cong Thai Hoc


Product Management System

Product updated successfully!

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	$1299.99	10	Electronics	Edit Delete
2	P002	iPhone 15 Pro	$999.99	25	Electronics	Edit Delete
3	P003	Samsung Galaxy S24	$899.99	20	Electronics	Edit Delete
4	P004	Office Chair Ergonomic	$199.99	50	Furniture	Edit Delete
5	P005	Standing Desk	$399.99	15	Furniture	Edit Delete
7	P006	Ghe Cong Thai Hoc	$6996.00	1	Furniture	Edit Delete

3. UPDATE PRODUCT FEATURE

Flow:

1. User clicks Edit button on product row

2. GET /products/edit/{id} - Shows pre-filled form - @PathVariable extracts id from URL - productService.getProductById(id) returns Optional - If found: adds product to model, returns form view - If not found: redirects with error message

3. Form displays existing product data (th:field auto-populates)

4. User modifies data and submits

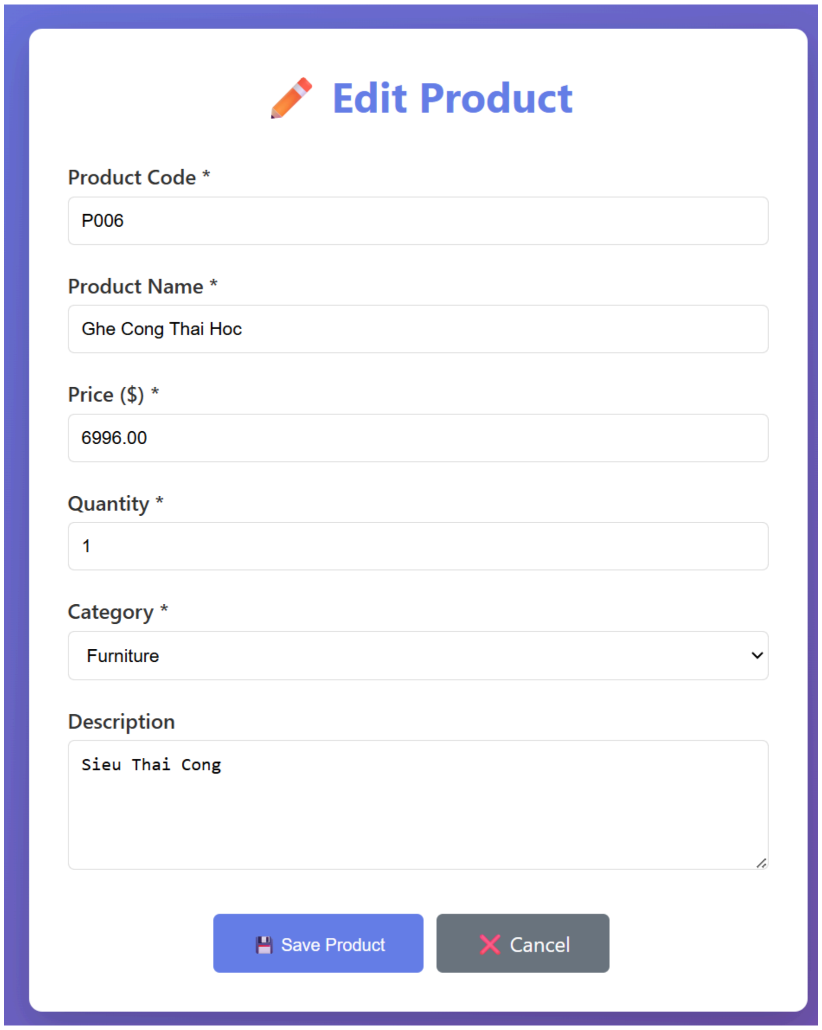
5. POST /products/save - Same endpoint as create - If product.id exists: Spring Data JPA performs UPDATE - If product.id is null: Spring Data JPA performs INSERT

6. Redirect to list with success message

Code Snippet (Controller):

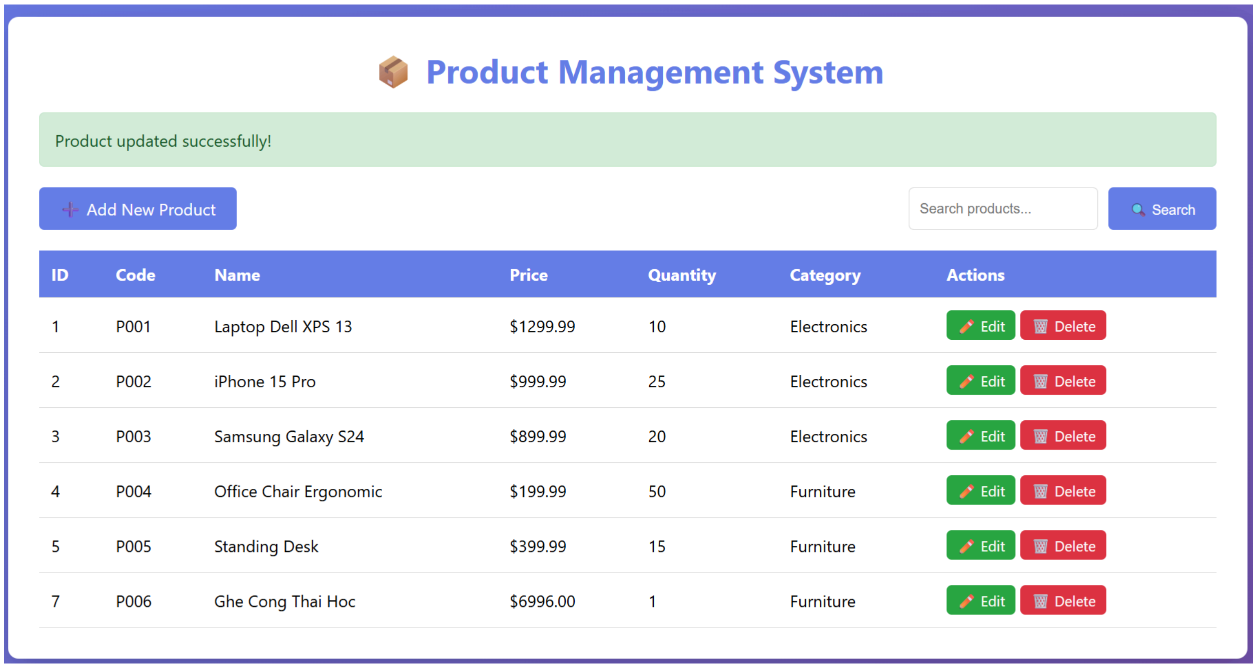
```
@GetMapping("/edit/{id}")
public String showEditForm(@PathVariable Long id, Model model,
                          RedirectAttributes redirectAttributes) {
    return productService.getProductById(id)
        .map(product -> {
            model.addAttribute("product", product);
            return "product-form";
        })
        .orElseGet(() -> {
            redirectAttributes.addFlashAttribute("error", "Product not found");
            return "redirect:/products";
        });
}
```

Key Concept: Spring Data JPA's save() method: - Checks if entity ID exists - If ID exists: UPDATE query - If ID is null: INSERT query - Single method handles both create and update



✏️ Edit Product

Product Code *	P006
Product Name *	Ghe Cong Thai Hoc
Price ($) *	6996.00
Quantity *	1
Category *	Furniture
Description	Sieu Thai Cong


Product Management System

Product updated successfully!

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	$1299.99	10	Electronics	Edit Delete
2	P002	iPhone 15 Pro	$999.99	25	Electronics	Edit Delete
3	P003	Samsung Galaxy S24	$899.99	20	Electronics	Edit Delete
4	P004	Office Chair Ergonomic	$199.99	50	Furniture	Edit Delete
5	P005	Standing Desk	$399.99	15	Furniture	Edit Delete
7	P006	Ghe Cong Thai Hoc	$6996.00	1	Furniture	Edit Delete

4. DELETE PRODUCT FEATURE

Flow:

1. User clicks Delete button (with confirmation dialog)

2. JavaScript confirms: “Are you sure?”

3. GET /products/delete/{id} - @PathVariable Long id extracts product ID - Calls productService.deleteProduct(id) - Service calls productRepository.deleteById(id) - Spring Data JPA executes: DELETE FROM products WHERE id = ?

4. Try-catch handles errors gracefully

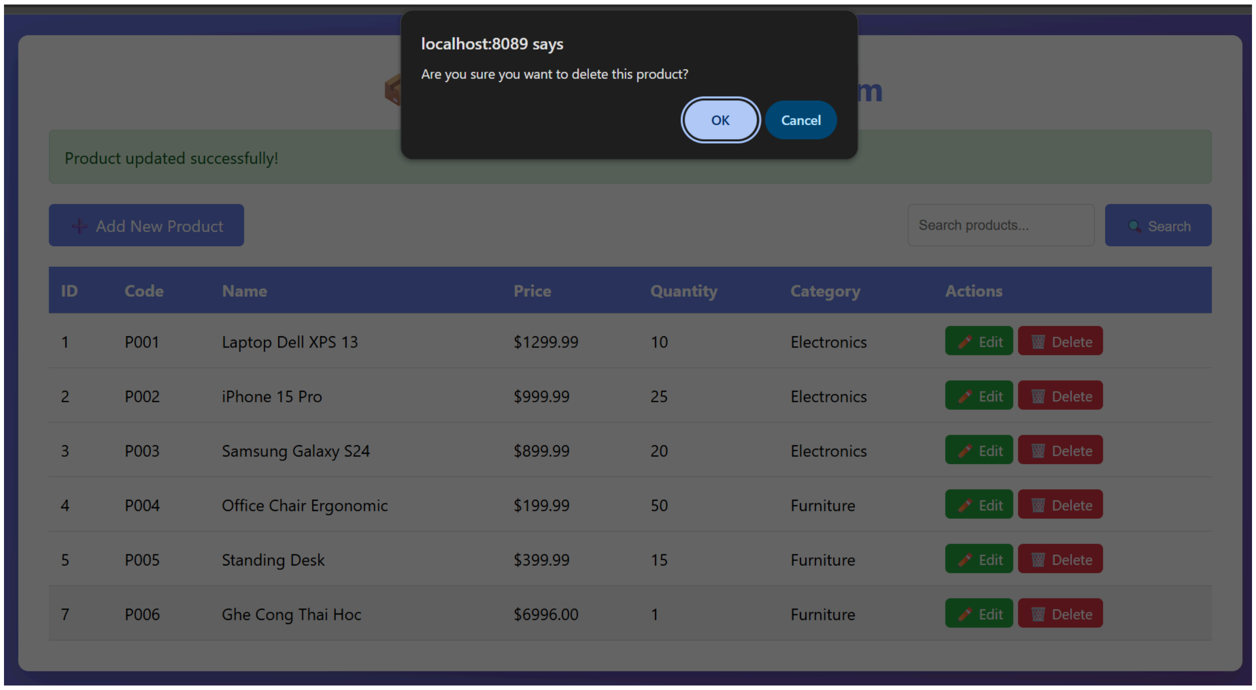
5. Flash message confirms deletion 6. Redirects back to product list

Code Snippet (Controller):

```
@GetMapping("/delete/{id}")
public String deleteProduct(@PathVariable Long id,
                           RedirectAttributes redirectAttributes) {
    try {
        productService.deleteProduct(id);
        redirectAttributes.addFlashAttribute("message", "Product deleted successfully!");
    } catch (Exception e) {
        redirectAttributes.addFlashAttribute("error", "Error deleting product");
    }
    return "redirect:/products";
}
```

Client-Side Confirmation (HTML):

```
<a th:href="@{/products/delete/{id}(id=${product.id})}"
   onclick="return confirm('Are you sure?')">Delete</a>
```

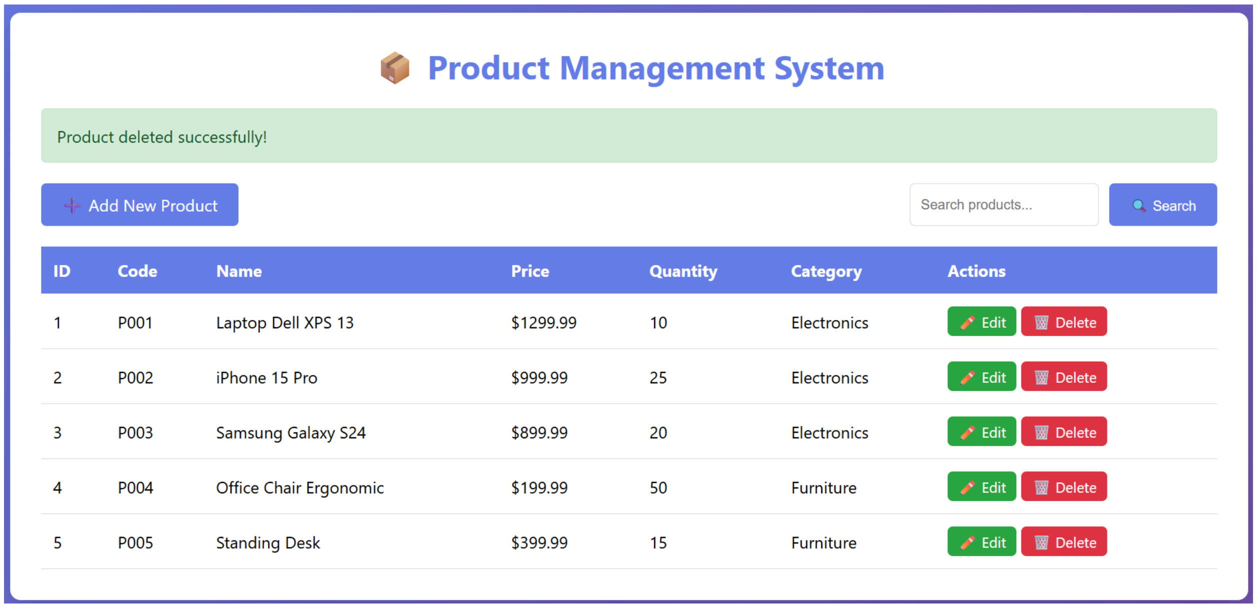


localhost:8089 says

Are you sure you want to delete this product?

Product updated successfully!

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	$1299.99	10	Electronics	Edit Delete
2	P002	iPhone 15 Pro	$999.99	25	Electronics	Edit Delete
3	P003	Samsung Galaxy S24	$899.99	20	Electronics	Edit Delete
4	P004	Office Chair Ergonomic	$199.99	50	Furniture	Edit Delete
5	P005	Standing Desk	$399.99	15	Furniture	Edit Delete
7	P006	Ghe Cong Thai Hoc	$6996.00	1	Furniture	Edit Delete



Product Management System

Product deleted successfully!

ID	Code	Name	Price	Quantity	Category	Actions
1	P001	Laptop Dell XPS 13	$1299.99	10	Electronics	Edit Delete
2	P002	iPhone 15 Pro	$999.99	25	Electronics	Edit Delete
3	P003	Samsung Galaxy S24	$899.99	20	Electronics	Edit Delete
4	P004	Office Chair Ergonomic	$199.99	50	Furniture	Edit Delete
5	P005	Standing Desk	$399.99	15	Furniture	Edit Delete