

Product Management System

Spring Boot project

A. application.properties

```
spring.application.name=OrderMS-1
```

```
spring.datasource.username=root
```

```
spring.datasource.password=
```

```
spring.datasource.url=jdbc:mysql://localhost:3309/orderdb
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=true
```

B. Models

a) Customer.java

```
@Entity
public class Customer {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String name;
    @OneToMany(mappedBy = "customer")
    private List<Product> products;
```

b) Product.java

```
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String name;
    private double price;
    @ManyToOne
    private Customer customer;
```

C. Repositories

a) CustomerRepo Interface

```
@Repository  
public interface CustomerRepo extends JpaRepository<Customer, Integer>{  
  
}
```

b) ProductRepo Interface

```
@Repository  
public interface ProductRepo extends JpaRepository<Product, Integer>{  
  
}
```

D. Services

a) CustomerService.java

```
@Service
public class CustomerService {

    @Autowired
    private CustomerRepo repo;
    public List<Customer> getAllCustomers() {
        return repo.findAll();
    }
    public Customer getCustomer(Integer id) {
        return repo.findById(id).get();
    }
    public Customer createCustomer(Customer customer) {
        return repo.save(customer);
    }
    public void deleteCustomer(Integer id) {
        repo.deleteById(id);
    }
}
```

b) ProductService.java

```
@Service
public class ProductService {
    @Autowired
    private ProductRepo repo;
    public List<Product> getAllProducts() {
        return repo.findAll();
    }
    public Product getProduct(Integer id) {
        return repo.findById(id).get();
    }
    public Product createProduct(Product product) {
        return repo.save(product);
    }
    public void deleteProduct(Integer id) {
        repo.deleteById(id);
    }
}
```

E. Contollers

a) CustomerController.java

```
@Controller
@RequestMapping("/customer")
public class CustomerController {
    //Call the Customer Service Class
    @Autowired
    private CustomerService service;
    //Read All Customers
    @GetMapping("/")
    public String view(Model model) {
        model.addAttribute("customers", service.getAllCustomers());
        return "customer/view";
    }
    //New Customer form fetched
    @GetMapping("/create")
    public String createForm(Model model) {
        model.addAttribute("customer", new Customer());
        return "customer/new";
    }
    //New Customer Form Submitted
    @PostMapping("/create")
    public String create(@ModelAttribute Customer customer, Model model) {
```

```
service.createCustomer(customer);//save new customer
model.addAttribute("customers", service.getAllCustomers());
return "redirect:/customer/";
}
//Edit form is fetched
@GetMapping("/edit/{id}")
public String edit(@PathVariable Integer id, Model model) {
    Customer customer=service.getCustomer(id);//Find customer with given id
    model.addAttribute("customer", customer);
    return "customer/edit";
}
//Edit form is submitted
@PostMapping("/edit")
public String edit(@ModelAttribute Customer customer, Model model) {
    service.createCustomer(customer);//save edited customer
    model.addAttribute("customers", service.getAllCustomers());
    return "redirect:/customer/";
}
//Delete Customer with the given id
@GetMapping("/delete/{id}")
public String getMethodName(@PathVariable Integer id, Model model) {
    service.deleteCustomer(id);
    model.addAttribute("customers", service.getAllCustomers());
    return "redirect:/customer/";
}
```



```
}
```

b) ProductController.java

```
@Controller
@RequestMapping("/product")
public class ProductController {
    @Autowired
    private ProductService service;
    @Autowired
    private CustomerService customerService;
    //Read All
    @GetMapping("/")
    public String view(Model model) {
        model.addAttribute("products", service.getAllProducts());
        return "product/view";
    }
    @GetMapping("/create")
    public String createForm(Model model) {
        model.addAttribute("product", new Product());
        model.addAttribute("customers", customerService.getAllCustomers());
        System.out.println("product1= "+new Product());
        return "product/new";
    }
}
```

```
@PostMapping("/create")
public String create(@ModelAttribute Product product, Model model) {
    System.out.println("product="+product);
    System.out.println(product.getCustomer());
    service.createProduct(product);//save new customer
    model.addAttribute("products", service.getAllProducts());
    return "redirect:/product/";
}
//Edit form is fetched
@GetMapping("/edit/{id}")
public String edit(@PathVariable Integer id, Model model) {
    Product product=service.getProduct(id);//Find customer with given id
    model.addAttribute("product", product);
    model.addAttribute("customers", customerService.getAllCustomers());
    return "product/edit";
}
//Edit form is submitted
@PostMapping("/edit")
public String edit(@ModelAttribute Product product, Model model) {
    service.createProduct(product);//save edited product
    model.addAttribute("products", service.getAllProducts());
    return "redirect:/product/";
}
//Delete Customer with the given id
@GetMapping("/delete/{id}")
```

```
public String delete(@PathVariable Integer id, Model model) {  
    service.deleteProduct(id);  
    model.addAttribute("products", service.getAllProducts());  
    return "redirect:/product/";  
}  
  
}
```

F. Views Pages

a) Product

i. View.html

```
<!DOCTYPE html>
<html xmlns:th="www.thymeleaf.org">
<head>
</head>
<body>
<h1>View Products</h1>
<a th:href="@{/product/create}">Create New Product</a>
<table border="2">
<tr>
<th>ID</th>
<th>Name</th>
<th>Price</th>
<th>Customer Name</th>
</tr>
<tr th:each="p:${products}">
<td th:text="${p.id}">ID</td>
<td th:text="${p.name}">Name</td>
<td th:text="${p.price}"></td>
<td th:text="${p.customer.name}"></td>
<td>
```

```
<a th:href="@{'/product/delete/{id}' (id=${p.id})}">delete</a>  
<a th:href="@{'/product/edit/{id}' (id=${p.id})}">edit</a>  
</td>  
</tr>  
</table>  
</body>  
</html>
```

ii. Edit.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Edit Customers</title>
</head>
<body>
<h1>Edit Products</h1>
<form method="post" th:action="@{/product/edit}" th:object="${product}">
<div>
<input th:field="*{id}" type="hidden">
</div>
<div>
<input th:field="*{name}" type="text">
</div>
<div>
<input th:field="*{price}" type="number">
</div>
<div>
<label>Customer Name:</label>
<select th:field="*{customer.id}">
<option th:each="c:${customers}" th:value="${c.id}"
th:text="${c.name}"></option>
```

```
</select>  
</div>  
<button type="submit" >Submit</button>  
</form>  
</body>  
</html>
```

iii.New.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
</head>
<body>
<h1>Add New Product</h1>
<form method="post" th:action="@{/product/create}" th:object=${product}>
<div>
<label>Product Name:</label>
<input th:field="*{name}">
</div>
<div>
<label>Product Price:</label>
<input th:field="*{price}">
</div>
<div>
<label>Customer Name:</label>
<select th:field="*{customer.id}">
<option th:each="c:${customers}" th:value="${c.id}"
th:text="${c.name}"></option>
</select>
</div>
```



```
<button type="submit" >Submit</button>  
</form>  
</body>  
</html>
```

b) Customer

i. View.html

```
<!DOCTYPE html>  
<html xmlns:th="http://www.thymeleaf.org">  
<head>  
</head>  
<body>
```

```
<h1>View Customers</h1>
<a th:href="@{/customer/create}">Create New Customer</a>
<table border="2">
<tr>
<th>ID</th>
<th>Name</th>
</tr>
<tr th:each="c:${customers}">
<td th:text="${c.id}">ID</td>
<td th:text="${c.name}">Name</td>
<td>
<a th:href="@{'/customer/delete/{id}' (id=${c.id})}">delete</a>
<a th:href="@{'/customer/edit/{id}' (id=${c.id})}">edit</a>
</td>
</tr>
</table>
</body>
</html>
```

ii. Edit.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Edit Customers</title>
</head>
<body>
<h1>Edit Customers</h1>
<form method="post" th:action="@{/customer/edit}" th:object="${customer}">
<div>
<input th:field="*{id}" type="hidden">
</div>
<div>
<input th:field="*{name}" type="text">
</div>
<button type="submit">Submit</button>
</form>
</body>
</html>
```

iii.New.html

```
<!DOCTYPE html>
<html xmlns:th="www.thymeleaf.org">
<head>
</head>
<body>
<h1>Create New Customer</h1>
<form method="post" th:action="@{/customer/create}" th:object="${customer}">
<div>
<label>Name:</label>
<input th:field="*{name}">
</div>
<button type="submit" >Submit</button>
</form>
</body>
</html>
```

New tab ×The Spring-Boo! ×localhost / 127.0.0.1 ×localhost:8080/c ×google - Search ×Introducing Chs ×ChatGPT ×

localhost:8080/customer/

View Customers

[Create New Customer](#)

ID	Name	
1	Sam	delete edit
2	Usoh	delete edit
3	Chidimma	delete edit
4	Julius	delete edit
5	Tosin	delete edit

Windows Taskbar

Type here to search

26°C Partly cloudy

23:11 16/01/2025







