МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет информационных систем и технологий

Кафедра «Информационные системы»

Дисциплина «Интернет программирование»

**Отчёт по лабораторной работе №5**

**MVC**

Выполнил студент

гр. ПИбд-21:

Сергеев Н.И.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Научный руководитель:

Филиппов А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ульяновск

2022

**Задание**

1. Для каждого класса с бизнес-логикой из предыдущей работы создать свой MVC-контроллер.
2. Для каждого MVC-контроллера следует разработать страницы на языке HTML с использованием шаблонизатора Thymeleaf. Полученное MVC-приложение должно полностью соответствовать по внешнему виду и функциональности клиентскому приложению из предыдущей работы.

**Сервер**

1. **MVC-контроллеры**

package ip.labwork.shop.controller;

import ip.labwork.shop.service.ComponentService;

import jakarta.validation.Valid;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.\*;

@Controller

@RequestMapping("/component")

public class ComponentMvcController {

private final ComponentService componentService;

public ComponentMvcController(ComponentService componentService) {

this.componentService = componentService;

}

@GetMapping

public String getComponents(Model model) {

model.addAttribute("components", componentService.findAllComponent());

return "component";

}

@GetMapping(value = {"/edit", "/edit/{id}"})

public String editComponent(@PathVariable(required = false) Long id,

Model model) {

if (id == null || id <= 0) {

model.addAttribute("componentDto", new ComponentDTO());

} else {

model.addAttribute("componentId", id);

model.addAttribute("componentDto", new ComponentDTO(componentService.findComponent(id)));

}

return "component-edit";

}

@PostMapping(value = {"/", "/{id}"})

public String saveComponent(@PathVariable(required = false) Long id,

@ModelAttribute @Valid ComponentDTO componentDTO,

BindingResult bindingResult,

Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("errors", bindingResult.getAllErrors());

return "component-edit";

}

if (id == null || id <= 0) {

componentService.create(componentDTO);

} else {

componentService.updateComponent(id, componentDTO);

}

return "redirect:/component";

}

@PostMapping("/delete/{id}")

public String deleteComponent(@PathVariable Long id) {

componentService.deleteComponent(id);

return "redirect:/component";

}

}

package ip.labwork.shop.controller;

import ip.labwork.shop.service.ComponentService;

import ip.labwork.shop.service.ProductService;

import jakarta.validation.Valid;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;

import java.util.Base64;

@Controller

@RequestMapping("/product")

public class ProductMvcController {

private final ProductService productService;

private final ComponentService componentService;

public ProductMvcController(ProductService productService, ComponentService componentService) {

this.productService = productService;

this.componentService = componentService;

}

@GetMapping

public String getProducts(Model model) {

model.addAttribute("products", productService.findAllProduct());

return "product";

}

@GetMapping(value = {"/edit", "/edit/{id}"})

public String editProduct(@PathVariable(required = false) Long id,

Model model) {

if (id == null || id <= 0) {

model.addAttribute("productDto", new ProductDTO());

} else {

model.addAttribute("productId", id);

model.addAttribute("componentDto", new ComponentDTO());

model.addAttribute("productDto", new ProductDTO(productService.findProduct(id)));

}

model.addAttribute("components", componentService.findAllComponent());

return "product-edit";

}

@PostMapping(value = {"/", "/{id}"})

public String saveProduct(@PathVariable(required = false) Long id,

@RequestParam("file") MultipartFile file,

@ModelAttribute @Valid ProductDTO productDTO,

BindingResult bindingResult,

Model model) throws IOException {

if (bindingResult.hasErrors()) {

model.addAttribute("errors", bindingResult.getAllErrors());

return "product-edit";

}

productDTO.setImage("data:image/jpeg;base64," + Base64.getEncoder().encodeToString(file.getBytes()));

if (id == null || id <= 0) {

return "redirect:/product/edit/" + productService.create(productDTO).getId();

} else {

productService.updateFields(id, productDTO);

return "redirect:/product/edit/" + id;

}

}

@PostMapping("/delete/{id}")

public String deleteProduct(@PathVariable Long id) {

productService.deleteProduct(id);

return "redirect:/product";

}

@PostMapping("/delete/{id}/{componentid}")

public String deleteComponent(@PathVariable Long id,

@PathVariable Long componentid) {

productService.deleteComponent(id,componentid);

return "redirect:/product/edit/" + id;

}

@PostMapping("/addproduct/{id}")

public String addComponent(@PathVariable Long id,

@ModelAttribute @Valid ComponentDTO componentDTO,

BindingResult bindingResult,

Model model) {

if (bindingResult.hasErrors()) {

model.addAttribute("errors", bindingResult.getAllErrors());

return "product-edit";

}

productService.addComponent(id, componentDTO);

return "redirect:/product/edit/" + id;

}

}

package ip.labwork.shop.controller;

import ip.labwork.shop.model.OrderStatus;

import ip.labwork.shop.service.OrderService;

import ip.labwork.shop.service.ProductService;

import jakarta.servlet.http.Cookie;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import org.apache.commons.lang3.StringUtils;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.\*;

import java.util.ArrayList;

import java.util.List;

@Controller

@RequestMapping("/order")

public class OrderMvcController {

private final OrderService orderService;

private final ProductService productService;

public OrderMvcController(OrderService orderService, ProductService productService) {

this.orderService = orderService;

this.productService = productService;

}

@GetMapping

public String getOrders(HttpServletRequest request,

Model model) {

Cookie[] cookies = request.getCookies();

List<ProductDTO> productDTOS = new ArrayList<>();

int totalPrice = 0;

if (cookies != null){

for(Cookie cookie : cookies){

if (StringUtils.isNumeric(cookie.getName())){

ProductDTO productDTO = new ProductDTO(productService.findProduct(Long.parseLong(cookie.getName())),Integer.parseInt(cookie.getValue()));

productDTOS.add(productDTO);

totalPrice += productDTO.getPrice() \* productDTO.getCount();

}

}

}

model.addAttribute("productDTOS", productDTOS);

model.addAttribute("totalPrice", totalPrice);

return "order";

}

@PostMapping

public String createOrder(HttpServletRequest request,

HttpServletResponse response) {

Cookie[] cookies = request.getCookies();

OrderDTO orderDTO = new OrderDTO();

List<ProductDTO> productDTOS = new ArrayList<>();

int totalPrice = 0;

for(Cookie temp : cookies){

if (StringUtils.isNumeric(temp.getName())){

ProductDTO productDTO = new ProductDTO(productService.findProduct(Long.parseLong(temp.getName())),Integer.parseInt(temp.getValue()));

productDTOS.add(productDTO);

totalPrice += productDTO.getPrice() \* productDTO.getCount();

}

}

orderDTO.setPrice(totalPrice);

orderDTO.setProductDTOList(productDTOS);

orderDTO.setStatus(OrderStatus.Готов);

orderService.create(orderDTO);

response.addCookie(new Cookie("delete",""));

return "redirect:/order";

}

@GetMapping(value = {"/all"})

public String getOrders(Model model) {

model.addAttribute("orders", orderService.findAllOrder());

return "orders";

}

@GetMapping(value = {"/view/{id}"})

public String getOrder(@PathVariable(required = false) Long id,

Model model) {

model.addAttribute("orderDto", new OrderDTO(orderService.findOrder(id)));

return "order-view";

}

}

1. **Thymeleaf**

component-edit.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head> </head>

<body>

<main style="background-color: white" layout:fragment="content">

<div th:text="${errors}" class="margin-bottom alert-danger"></div>

<form

action="#"

th:action="@{/component/{id}(id=${id})}"

th:object="${componentDto}"

method="post"

>

<div class="mb-3">

<label for="componentName" class="form-label">Название</label>

<input

type="text"

class="form-control"

id="componentName"

th:field="${componentDto.componentName}"

required="true"

/>

</div>

<div class="mb-3">

<label for="price" class="form-label">Цена</label>

<input

type="text"

class="form-control"

id="price"

th:field="${componentDto.price}"

required="true"

/>

</div>

<div class="mb-3">

<button type="submit" class="btn btn-primary button-fixed">

<span th:if="${id == null}">Добавить</span>

<span th:if="${id != null}">Обновить</span>

</button>

<a class="btn btn-secondary button-fixed" th:href="@{/component}">

Назад

</a>

</div>

</form>

</main>

</body>

</html>

component.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head> </head>

<body>

<main style="background-color: white" layout:fragment="content">

<div>

<a class="btn btn-success button-fixed" th:href="@{/component/edit}">

<i class="fa-solid fa-plus"></i> Добавить

</a>

</div>

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Название</th>

<th scope="col">Цена</th>

<th scope="col"></th>

</tr>

</thead>

<tbody>

<tr th:each="component, iterator: ${components}">

<th scope="row" th:text="${iterator.index} + 1" />

<td th:text="${component.componentName}" />

<td th:text="${component.price}" style="width: 60%" />

<td style="width: 10%">

<div class="btn-group" role="group" aria-label="Basic example">

<a

class="btn btn-warning button-fixed button-sm"

th:href="@{/component/edit/{id}(id=${component.id})}"

>

<i class="fa fa-pencil" aria-hidden="true"></i> Изменить

</a>

<button

type="button"

class="btn btn-danger button-fixed button-sm"

th:attr="onclick=|confirm('Удалить запись?') && document.getElementById('remove-${component.id}').click()|"

>

<i class="fa fa-trash" aria-hidden="true"></i> Удалить

</button>

</div>

<form

th:action="@{/component/delete/{id}(id=${component.id})}"

method="post"

>

<button

th:id="'remove-' + ${component.id}"

type="submit"

style="display: none"

>

Удалить

</button>

</form>

</td>

</tr>

</tbody>

</table>

</div>

</main>

</body>

</html>

default.html

<!DOCTYPE html>

<html

lang="ru"

xmlns:th="http://www.thymeleaf.org"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

>

<head>

<meta charset="UTF-8" />

<title>Очень вкусно и запятая</title>

<meta name="viewport" content="width=device-width, initial-scale=1" />

<script

type="text/javascript"

src="/webjars/bootstrap/5.1.3/js/bootstrap.bundle.min.js"

></script>

<link

rel="stylesheet"

href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css"

/>

<link rel="stylesheet" href="/webjars/font-awesome/6.1.0/css/all.min.css" />

<link rel="stylesheet" href="/css/style.css" />

</head>

<body class="d-flex flex-column h-100">

<div id="app">

<nav class="navbar navbar-expand-lg">

<div class="container-fluid">

<a class="navbar-brand" href="/product">

<h1 class="text-black">Очень вкусно и запятая</h1>

</a>

<button

class="navbar-toggler"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<a class="nav-link fs-4 text-black" href="/component"

>Компоненты</a

>

<a class="nav-link fs-4 text-black" href="/product">Продукты</a>

<a class="nav-link fs-4 text-black" href="/order">Заказы</a>

<a class="nav-link fs-4 text-black" href="/order/all">История заказов</a>

</ul>

</div>

</div>

</nav>

<div class="container-fluid p-0">

<div

class="container container-padding"

layout:fragment="content"

></div>

</div>

<footer

class="footer mt-auto d-flex justify-content-center align-items-center"

>

ООО "Вкусно" © 2022

</footer>

</div>

</body>

<th:block layout:fragment="scripts"> </th:block>

</html>

error.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head> </head>

<body>

<div layout:fragment="content">

<div><span th:text="${error}"></span></div>

<a href="/">На главную</a>

</div>

</body>

</html>

index.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head> </head>

<body>

<div layout:fragment="content">

<div>It's works!</div>

<a href="123">ERROR</a>

</div>

</body>

</html>

order-view.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head>

<script

type="text/javascript"

src="/webjars/jquery/3.6.0/jquery.min.js"

></script>

</head>

<body>

<main style="background-color: white" layout:fragment="content">

<div th:text="${errors}" class="margin-bottom alert-danger"></div>

<div class="mb-3">

<label for="name" class="form-label">Дата</label>

<input

type="text"

class="form-control"

id="name"

th:field="${orderDto.date}"

disabled

required="true"

/>

</div>

<div class="mb-3">

<label for="price" class="form-label">Общая стоимость</label>

<input

type="text"

class="form-control"

id="price"

th:value="${orderDto.price}"

disabled

required="true"

/>

</div>

<div class="mb-3">

<label for="image" class="form-label">Статус</label>

<input type="text" class="form-control" disabled id="image" th:value="${orderDto.status}" />

</div>

<div class="mb-3">

<a class="btn btn-secondary button-fixed" th:href="@{/order/all}">

Назад

</a>

</div>

<div class="table-responsive" th:if="${id != null}">

<table class="table">

<thead>

<tr>

<th scope="col">Название</th>

<th scope="col">Количество</th>

</tr>

</thead>

<tbody>

<tr

th:each="product, iterator: ${orderDto.getProductDTOList()}"

>

<th

scope="row"

th:text="${product.getName()}"

style="width: 60%"

/>

<td th:text="${product.getCount()}" />

</tr>

</tbody>

</table>

</div>

</main>

</body>

</html>

order.html

<!DOCTYPE html>

<html

lang="en"

layout:decorate="~{default}"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

>

<head> </head>

<body>

<main

class="flex-shrink-0"

style="background-color: white"

layout:fragment="content"

>

<h1 class="my-5 ms-5 fs-1">

<b>Корзина</b>

</h1>

<h2 class="my-5 ms-5 fs-3"></h2>

<div class="ms-5 my-5">Список товаров</div>

<div>

<div style="max-width: 35%">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Позиция</th>

<th scope="col"></th>

<th scope="col">Стоимость</th>

<th scope="col"></th>

</tr>

</thead>

<tbody class="table-group-divider">

<tr th:each="product, iterator: ${productDTOS}">

<th scope="row" th:text="${iterator.index+1}"></th>

<td colspan="2" th:text="${product.name}"></td>

<td th:text="${product.count+'x'+product.price+' руб'}"></td>

<td>

<button th:attr="onclick=|remove(${product.id})|">

Удалить

</button>

</td>

</tr>

</tbody>

</table>

</div>

<h2 class="ms-5 my-5" th:text="${'Итого: ' + totalPrice + ' руб'}"></h2>

<button

class="btn btn-success ms-5 w-25"

type="button"

style="color: black"

th:attr="onclick=|document.getElementById('accept').click()|"

>

Купить

</button>

<form th:action="@{/order}" method="post">

<button th:id="'accept'" type="submit" style="display: none">

Купить

</button>

</form>

</div>

<p></p>

<p></p>

</main>

</body>

<th:block layout:fragment="scripts">

<script>

function deleteAll() {

let l = getCookie();

for (let i = 0; i < l.length; i++) {

if (l[i].id === "delete") {

let cookies = document.cookie.split(";");

for (let j = 0; j < cookies.length; j++) {

let cookie = cookies[j];

let eqPos = cookie.indexOf("=");

let name = eqPos > -1 ? cookie.substr(0, eqPos) : cookie;

document.cookie =

name + "=;expires=Thu, 01 Jan 1970 00:00:00 GMT;";

document.cookie =

name + "=; path=/; expires=Thu, 01 Jan 1970 00:00:01 GMT;";

}

window.location.href = "/order";

}

}

}

deleteAll();

function remove(name) {

let i = getCookie(name);

const d = new Date();

d.setTime(d.getTime() + 24 \* 60 \* 60 \* 1000);

let expires = "expires=" + d.toUTCString();

if (i == "1") {

document.cookie = name + "=;expires=Thu, 01 Jan 1970 00:00:00 GMT;";

document.cookie =

name + "=; path=/; expires=Thu, 01 Jan 1970 00:00:01 GMT;";

} else {

document.cookie =

name + "=" + (Number(i) - Number(1)) + ";" + expires + ";path=/";

}

window.location.href = "/order";

}

function getCookie(name) {

if (name) {

name = name + "=";

let decodedCookie = decodeURIComponent(document.cookie);

let ca = decodedCookie.split(";");

for (let i = 0; i < ca.length; i++) {

let c = ca[i];

while (c.charAt(0) == " ") {

c = c.substring(1);

}

if (c.indexOf(name) == 0) {

return c.substring(name.length, c.length);

}

}

return "";

} else {

let arrObjects = [];

let decodedCookie = decodeURIComponent(document.cookie);

let ca = decodedCookie.split("; ");

for (let i = 0; i < ca.length; i++) {

let c = ca[i];

let cs = c.split("=");

arrObjects[i] = {

id: cs[0],

count: cs[1],

};

}

return arrObjects;

}

}

</script>

</th:block>

</html>

orders.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head> </head>

<body>

<main style="background-color: white" layout:fragment="content">

<div class="table-responsive">

<table class="table">

<thead>

<tr>

<th scope="col">#</th>

<th scope="col">Дата оформления</th>

<th scope="col">Общая стоимость</th>

<th scope="col">Статус</th>

<th scope="col"></th>

</tr>

</thead>

<tbody>

<tr th:each="order, iterator: ${orders}">

<th scope="row" th:text="${iterator.index} + 1" />

<td th:text="${order.date}" style="width: 60%"/>

<td th:text="${order.price}" />

<td th:text="${order.status}" />

<td style="width: 10%">

<div class="btn-group" role="group" aria-label="Basic example">

<a

class="btn btn-warning button-fixed button-sm"

th:href="@{/order/view/{id}(id=${order.id})}"

>

<i class="fa fa-pencil" aria-hidden="true"></i> Подробнее

</a>

</div>

</td>

</tr>

</tbody>

</table>

</div>

</main>

</body>

</html>

product-edit.html

<!DOCTYPE html>

<html

lang="en"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

layout:decorate="~{default}"

>

<head>

<script

type="text/javascript"

src="/webjars/jquery/3.6.0/jquery.min.js"

></script>

</head>

<body>

<main style="background-color: white" layout:fragment="content">

<div th:text="${errors}" class="margin-bottom alert-danger"></div>

<form

action="#"

th:action="@{/product/{id}(id=${id})}"

th:object="${productDto}"

enctype="multipart/form-data"

method="post"

>

<div class="mb-3">

<label for="name" class="form-label">Название</label>

<input

type="text"

class="form-control"

id="name"

th:field="${productDto.name}"

required="true"

/>

</div>

<div class="mb-3">

<label for="price" class="form-label">Цена</label>

<input

type="text"

class="form-control"

id="price"

th:value="${productDto.price}"

disabled

required="true"

/>

</div>

<div class="mb-3">

<label for="image" class="form-label">Изображение</label>

<input type="file" class="form-control" id="image" th:name="file" />

</div>

<div class="mb-3">

<button type="submit" class="btn btn-primary button-fixed">

<span th:if="${id == null}">Добавить</span>

<span th:if="${id != null}">Обновить</span>

</button>

<a class="btn btn-secondary button-fixed" th:href="@{/product}">

Назад

</a>

</div>

</form>

<form

th:if="${id != null}"

action="#"

th:action="@{/product/addproduct/{id}(id=${id})}"

th:object="${componentDto}"

method="post"

>

<div class="mb-3">

<label for="lang" class="form-label">Компонент</label>

<select id="lang" class="form-select" th:field="${componentDto.id}">

<option

th:each="value: ${components}"

th:value="${value.getId()}"

th:selected="true"

th:text="${value.getComponentName()}"

></option>

</select>

</div>

<div class="mb-3">

<label for="count" class="form-label">Количество</label>

<input

type="text"

class="form-control"

id="count"

th:field="${componentDto.count}"

required="true"

/>

</div>

<div>

<div>

<button class="btn btn-success button-fixed" type="submit">

Добавить

</button>

</div>

</div>

</form>

<div class="table-responsive" th:if="${id != null}">

<table class="table">

<thead>

<tr>

<th scope="col">Название</th>

<th scope="col">Количество</th>

<th scope="col"></th>

</tr>

</thead>

<tbody>

<tr

th:each="component, iterator: ${productDto.getComponentDTOList()}"

>

<th

scope="row"

th:text="${component.getComponentName()}"

style="width: 60%"

/>

<td th:text="${component.getCount()}" />

<td style="width: 10%">

<div class="btn-group" role="group" aria-label="Basic example">

<button

type="button"

class="btn btn-danger button-fixed button-sm"

th:attr="onclick=|confirm('Удалить запись?') && document.getElementById('remove-${component.id}').click()|"

>

<i class="fa fa-trash" aria-hidden="true"></i> Удалить

</button>

</div>

<form

th:action="@{/product/delete/{productid}/{id} (id=${component.getId()}, productid=${id})}"

method="post"

>

<button

th:id="'remove-' + ${component.getId()}"

type="submit"

style="display: none"

>

Удалить

</button>

</form>

</td>

</tr>

</tbody>

</table>

</div>

</main>

</body>

</html>

product.html

<!DOCTYPE html>

<html

lang="en"

layout:decorate="~{default}"

xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"

>

<head> </head>

<body>

<main class="flex-shrink-0" layout:fragment="content">

<div class="btn-group mt-2" role="group">

<a

type="button"

class="btn btn-outline-dark text-center d-flex justify-content-md-center mx-2 mb-3"

th:href="@{/product/edit}"

>

Добавить

</a>

</div>

<div class="temp row row-cols-1 row-cols-md-3 g-4" id="tbl-items">

<div

class="col"

th:each="product, iterator: ${products}"

th:key="${product.getId()}"

>

<div class="card">

<div class="container w-100 h-350px">

<img

alt="Бугер"

class="img-fluid rounded mx-auto d-block"

style="width: 100%; height: 350px; objectfit: contain"

th:src="${product.image}"

/>

</div>

<div class="card-body">

<h5

class="card-title text-center fs-1"

th:text="${product.getPrice()}"

></h5>

<a

class="btn btn-outline-dark text-center d-flex justify-content-md-center mx-5"

th:attr="onclick=|confirm('Удалить запись?') && document.getElementById('remove-${product.id}').click()|"

>

Удалить

</a>

<a

class="btn btn-outline-dark text-center d-flex justify-content-md-center mx-5"

th:href="@{/product/edit/{id}(id=${product.id})}"

type="button"

>Изменить

</a>

<a

class="btn btn-outline-dark text-center d-flex justify-content-md-center mx-5"

type="button"

th:attr="onclick=|add(${product.id})|"

>

в корзину

</a>

<form

th:action="@{/product/delete/{id}(id=${product.id})}"

method="post"

>

<button

th:id="'remove-' + ${product.id}"

type="submit"

style="display: none"

>

Удалить

</button>

</form>

</div>

</div>

</div>

</div>

</main>

</body>

<th:block layout:fragment="scripts">

<script>

function add(name) {

let i = getCookie(name);

const d = new Date();

d.setTime(d.getTime() + 24 \* 60 \* 60 \* 1000);

let expires = "expires=" + d.toUTCString();

if (i == "") {

document.cookie = name + "=" + 1 + ";" + expires + ";path=/";

} else {

document.cookie =

name + "=" + (Number(i) + Number(1)) + ";" + expires + ";path=/";

}

}

function getCookie(name) {

name = name + "=";

let decodedCookie = decodeURIComponent(document.cookie);

let ca = decodedCookie.split(";");

for (let i = 0; i < ca.length; i++) {

let c = ca[i];

while (c.charAt(0) == " ") {

c = c.substring(1);

}

if (c.indexOf(name) == 0) {

return c.substring(name.length, c.length);

}

}

return "";

}

</script>

</th:block>

</html>

style.css

nav {

background-color: #c8afaf;

}

main {

background-color: #8c7b7b;

min-height: 90vh;

}

footer {

background-color: #c8afaf;

color: black;

}

.size {

overflow-x: hidden;

}

.card {

background-color: #8c7b7b;

border-color: #8c7b7b;

}

.card-body {

background-color: #8c7b7b;

}

h1 {

word-wrap: break-word;

}

#banner {

margin: 0px 15px 15px 15px;

padding-top: 15px;

display: flex;

align-items: center;

flex-direction: column;

}

#banner img {

border-radius: 5px;

}

#banner img.show {

width: 100%;

opacity: 1;

transition: opacity 1s, visibility 0s;

}

#banner img.hide {

height: 0;

width: 0;

opacity: 0;

visibility: hidden;

transition: opacity 1s, visibility 0s 1s;

}

@media (max-width: 767px) {

.btn {

padding: 1px 5px;

font-size: 12px;

line-height: 1.5;

border-radius: 3px;

}

}

@media (min-width: 768px) and (max-width: 991px) {

.btn {

padding: 5px 10px;

font-size: 12px;

line-height: 1.5;

border-radius: 3px;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.btn {

padding: 6px 12px;

font-size: 14px;

line-height: 1.42857143;

border-radius: 4px;

}

}

@media (min-width: 1200px) {

.btn {

padding: 10px 16px;

font-size: 18px;

line-height: 1.3333333;

border-radius: 6px;

}

}

@media (max-width: 600px) {

.tem {

padding-right: 50px;

}

}