**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ СОЦИОКУЛЬТУРНЫХ КОММУНИКАЦИЙ**

**Кафедра информационных технологий**

**РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ ДЛЯ ТАКЕЛАЖА**

Курсовой проект

Кривулько Дианы Романовны

студентки 4 курса,

специальности

«прикладная информатика»

Научный руководитель:

доцент кафедры

Царик Сергей Всеволодович

Допущена к защите

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г

**РЕФЕРАТ**

Количество страниц **–** 24,изображений **–** 16, листингов – 4, источников **–**4, приложений **–** 1.

**Ключевые слова:** ИНТЕРНЕТ-ВИТРИНА, ВЕБ-СТРАНИЦА, ВЕБ-САЙТ, ТАКЕЛАЖ, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC, ССSS.

**Объектом исследования** курсовой работы является интернет-витрина такелажа.

**Методы исследования –** архитектура клиент-серверного приложения, шаблон проектирования Model-Viewer-Controller.

В результате выполнения курсовой работы было разработана интернет-витрина фенов для волос. Веб-приложение реализовано на языке Java с использованием Spring Framework и Famework Bootstrap. Разработка проводилась в IDE Intellij IDEA и MySQL.

**Область применения –** сектор Интернета, посвященный разработке веб-приложений.

**ESSAY**

Number of pages - 24, images - 16, listings - 4, sources - 4, applications - 1.

**Keywords:** INTERNET SHOWROOM, WEB PAGE, WEBSITE, RIGGING, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC, CCSS.

**The Object of study** of the course work is an online showcase rigging.

**Research methods** - architecture of the client-server application, design pattern Model-Viewer-Controller.

As a result of the course work, an Internet showcase of hair dryers was developed. The web application is implemented in Java using the Spring Framework and Famework Bootstrap. The development was carried out in the IDE Intellij IDEA and MySQL.

**Application Area** - Web Application Development Sector.

**РЭФЕРАТ**

Колькасць старонак - 24, малюнкаў - 16, лістынгаў - 4, крыніц - 4, прыкладанняў - 1.

**Ключавыя словы**: ІНТЭРНЭТ-вітрын, ВЭБ-СТАРОНКА, ВЭБ-САЙТ, такелаж, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC, ССSS.

**Аб'ектам даследавання** курсавой работы з'яўляецца інтэрнэт-вітрына такелажа.

**Метады даследавання** - архітэктура кліент-сервернага прыкладання, шаблон праектавання Model-Viewer-Controller.

У выніку выканання курсавой работы было распрацавана інтэрнэт-вітрына фенаў для валасоў. Вэб-дадатак рэалізавана на мове Java з выкарыстаннем Spring Framework і Famework Bootstrap. Распрацоўка праводзілася ў IDE Intellij IDEA і MySQL.

**Вобласць ужывання** - сектар Інтэрнэту, прысвечаны распрацоўцы вэб-прыкладанняў.

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ 4**](#_Toc27579514)

[**ГЛАВА 1. ОСНОВНЫЕ ИНСТРУМЕНТЫ РАЗБРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ 5**](#_Toc27579515)

[**1. Применение системы прототипирования Invision 5**](#_Toc27579516)

[**2. Работа HTML и CSS 7**](#_Toc27579517)

[**2. Фреймворк для веб-сайтов Bootstrap 4 9**](#_Toc27579518)

[**3. Spring MVC 11**](#_Toc27579519)

[**ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ИНТЕРНЕТ-ВИТРИНЫ ТАКЕЛАЖА» 13**](#_Toc27579520)

[**2.1 Разработка прототипов страниц интернет-витрины 13**](#_Toc27579521)

[**2.2 Работа со Spring 14**](#_Toc27579522)

[**ЗАКЛЮЧЕНИЕ 20**](#_Toc27579523)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21**](#_Toc27579524)

[**ПРИЛОЖЕНИЕ 22**](#_Toc27579525)

# ВВЕДЕНИЕ

Интернет-витрина является одним из популярных и востребованных типов сайтов в Интернете. Такие веб-ресурсы как правило включают в себя информацию о бренде, компании, организации, и главное описание их широкого ассортимента услуг или продукции с доступностью online. Посетители, которые заходят на сайт-витрину могут легко ознакомиться с детальным описание продукции или услуги, прочитать все характеристики и по фото наглядно понять, что есть что. Не все фирмы и компании предпочитают заказать интернет-магазин – многим по ряду причин целесообразнее сделать свой выбор в пользу online каталога.

Целью данного курсового проекта является создание интернет-витрины для такелажа. Интернет-витрина должна содержать полную информацию о товаре, возможность регистрации пользователей для заказа товара, а также административную панель для добавления, удаления и редактирования продуктов.

Для достижения цели были поставлены следующие задачи:

1) изучить возможные средства разработки веб-приложений;

2) разработать базу данных системы;

3) разработать веб-приложение «интернет-витрина фенов для волос».

## ГЛАВА 1. ОСНОВНЫЕ ИНСТРУМЕНТЫ РАЗБРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ

## Применение системы прототипирования Invision

InVision – платформа для разработки макетов интерфейсов. Данная платформа предоставляет возможность разработки программного продукта, в котором можно создавать дизайнерские макеты и просматривать работу макета в программе.

Основными характеристиками InVision является:

* Возможность работы нескольких разработчиков над одним проектом.
* Все данные хранятся в облаке, что позволяет иметь доступ к проекту из любого места.

Система содержит в себе пакет инструментов быстрой обработки, что позволяет загружать в учетную запись пользователя графические материалы или новые прототипы и работать с ними прямо в браузере.

Инструменты и интерфейсы InVision являются очень простыми в работе и интуитивно понятными, что позволяет разработчикам с первых действий понять функции платформы и не тратить время на освоение интерфейса.

При первоначальной работе с Invision очень удобным является предоставления выбора размеров будущего прототипа. Заранее прописанные размеры под соответствующие устройства значительно упрощают работу на первоначальном этапе (рисунок 1.1)

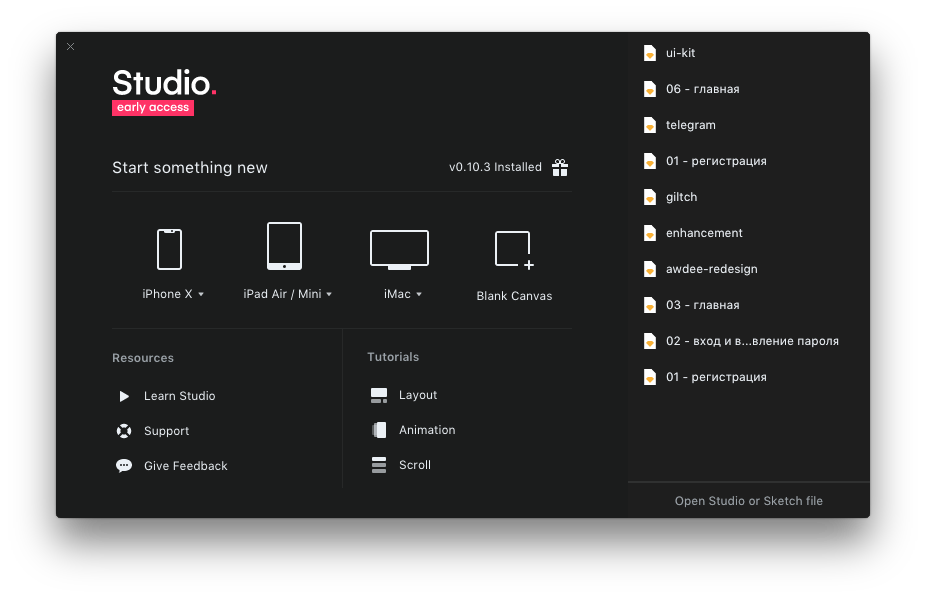


Рисунок 1.1 – Первоначальная панель

То, что InVision Studio кроссплатформенный — огромное стратегическое преимущество другими системами прототипирования. Это значительно расширяет аудиторию пользователей.

Слои в Invision представлены, как и во многих программах. Имеется возможность создания вложенных слоев, групп и помещения на страницу не одного объекта, а сразу группы объектов (рисунок 1.2).

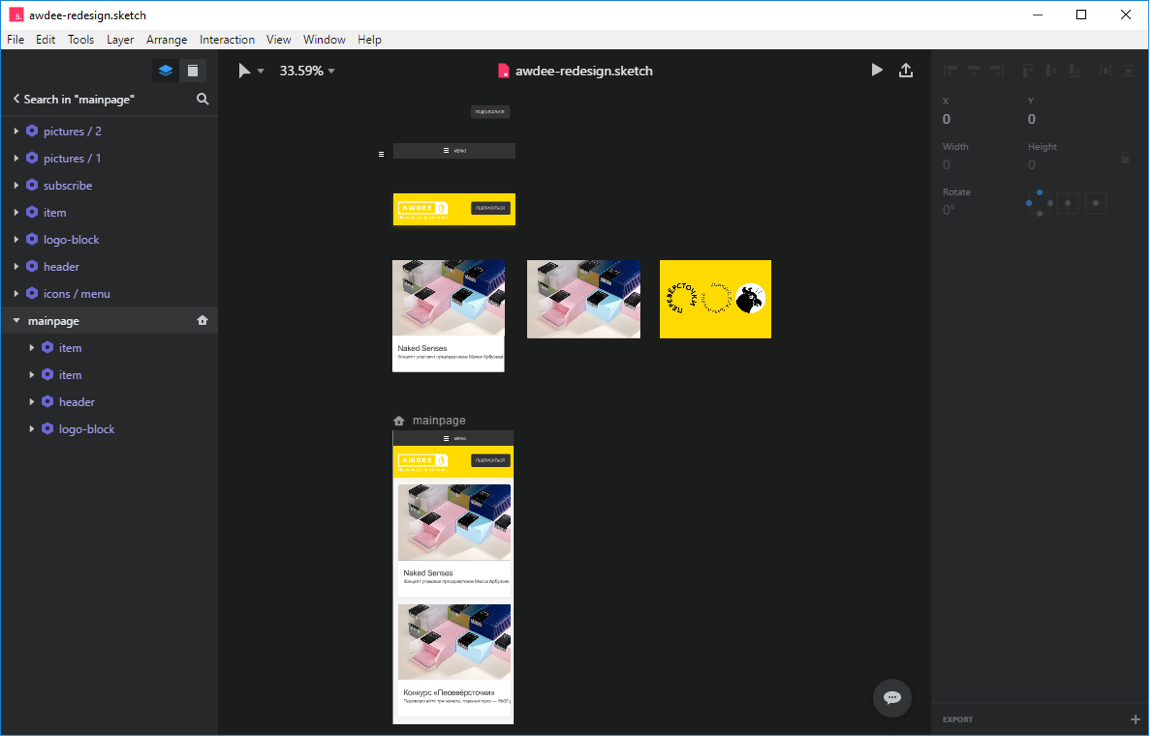


Рисунок 1.2 – Внешний вид панели слоев

Инспектор инструментов также представлен панелью, интуитивно понятной для разработки и изменения объектов (рисунок 1.3).

В случае работы всего над одним проектом, разработчикам предоставляется возможность бесплатного пользования. В остальных случаях, если вы собираетесь работать в команде или создавать несколько проектов, придется оплатить использование сервиса.

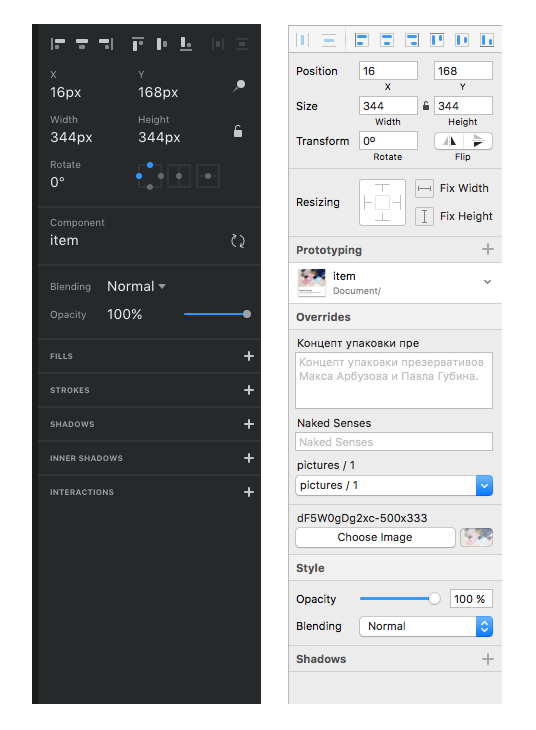
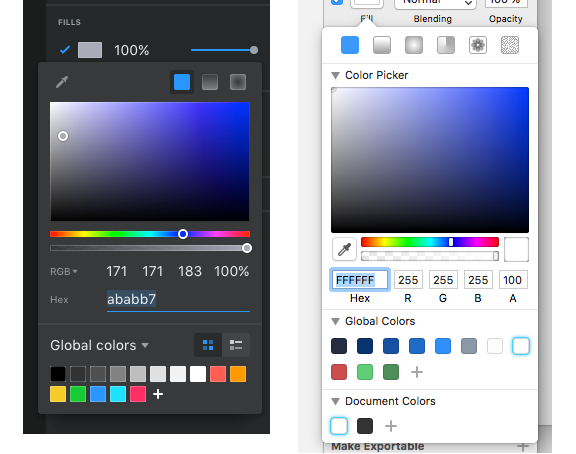
Рассмотрев основные возможности данной системы, можно сказать, что она является очень современной и динамичной системой для работы над созданием веб-проектов. Возможность работы с любой операционной системой и простой доступ к инструментам позволяет использовать эту платформу как сильный онлайн-инструмент для UX-разработки и тестирования.

Рисунок 1.3 – Внешний вид инспектора инструментов

## 2. Работа HTML и CSS

Каждая страница в Интернете, которую вы посещаете, строится на выполнении отдельных инструкций шаг за шагом. Ваш браузер (Chrome, Firefox, Safari и т.д) играет колоссальную роль в отображении кода и тем, что мы можем видеть на наших экранах и даже взаимодействовать. Когда вы открываете веб-страницу, браузер отображает HTML и другие языки программирования в максимально понятном для вас формате.

HTML и CSS всего лишь структура страницы и информация о стиле. Прежде чем перейти к JavaScript и другим языкам программирования, необходимо знать основы HTML и CSS, поскольку они находятся на передней части каждой веб-страницы и приложения.

HTML отображает язык разметки гипертекста. «Язык разметки» означает, что HTML использует теги для идентификации различных типов контента и целей, которые каждый преследует на веб-странице.

CSS — это каскадные таблицы стилей. Этот язык разметки определяет, как HTML-элементы веб-сайта должны отображаться на интерфейсе страницы.

В то время как HTML является основной структурой вашего сайта, CSS — это то, что дает всему вашему сайту стиль. Цвета, интересные шрифты и фоновые изображения – все это заслуга CSS. Этот язык влияет на все «настроение» веб-страницы, что делает его невероятно мощным инструментом и важным навыком для веб-разработчиков. Он также позволяет веб-сайтам адаптироваться к различным размерам экрана и типам устройств.

Пример HTML без CSS, приводящий к неформатированному сообщению блога (рисунок 1.4).

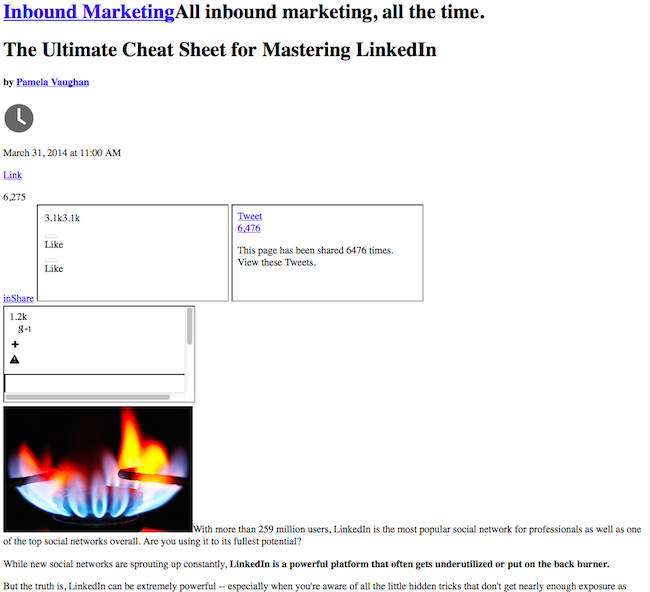
Обратите внимание, что все содержимое не изменилось, но визуально все выглядит иначе. Это то, что вы увидите, если таблица стилей по какой-либо причине не прогрузится на сайт. Теперь, вот как выглядит одна и та же веб-страница с добавлением CSS.

Рисунок 1.4 – Пример HTML-страницы без применения CSS

Пример HTML + CSS, используемый для форматирования сообщения блога (рисунок 1.5).

Любой браузер имеет таблицу стилей по умолчанию, поэтому на каждой веб-странице отражается хотя бы одна таблица стилей, в зависимости от того, какой браузер посетитель использует, независимо от дизайна страницы. К примеру, допустим стиль шрифта нашего браузера по умолчанию - Times New Roman, размер 12, если разработчик не применил собственную таблицу стилей, то посетитель увидит веб-страницу в Times New Roman размером 12.

Разумеется, в большинстве случаев используется пользовательский CSS. Последняя определенная таблица стилей сообщает моему браузеру, какие инструкции имеют приоритет.

Рисунок 1.5 – Пример HTML-страницы с применением CSS

## Фреймворк для веб-сайтов Bootstrap 4

Практически любая веб-страница содержит множество похожих компонентов, которые встречаются и на других сайтах. Это меню, навигация, элементы форм, заголовки и др. Не говоря уже про многоколоночную верстку, без которой вообще сложно превратить картинку макета в готовую веб-страницу. Чтобы меньше писать кода и по максимуму задействовать уже готовые решения применяются специализированные библиотеки, их часто называют фреймворками. Одним из таких фреймворков для верстки является Bootstrap.

Во-первых,Bootstrap является самым популярным фреймворком, у его ближайшего конкурента в 3-5 раз меньше сообщество.

Во-вторых, это не только css, но и js-фреймворк. То есть в Bootstrap написаны готовые стили и скрипты, для применения которых вам достаточно всего лишь прописать необходимые стилевые классы и атрибуты html-элементам.

На сегодняшний день Bootstrap давно перерос рамки одной компании, это открытый продукт, применяемый веб-разработчиками для верстки сайтов во всем мире.

Какие плюсы несет в себе использование библиотеки.

1) Высокая скорость разработки

Фактически Bootstrap представляет собой конструктор, фрагменты которого вы включаете в свой проект при необходимости. Это уменьшает время разработки, потому что не требуется придумывать и писать их самостоятельно.

2) Адаптивный дизайн

Bootstrap направлен на создание макета под разные устройства — ноутбуки, планшеты, смартфоны. При этом код пишется один, а масштабирование в зависимости от ширины устройства берет на себя фреймворк.

3) Открытая программа

Вы можете свободно изучать исходные коды библиотеки, изменять и расширять ее под собственные нужды.

4) Совместимость с браузерами

Компоненты библиотеки написаны и протестированы с учетом работы разных браузеров. Это гарантирует, что макет будет выглядеть одинаково независимо от выбранного браузера.

4) Низкий порог вхождения

Чтобы использовать библиотеку в своей работе, требуется обладать минимальными знаниями по HTML, CSS и JavaScript. Это позволяет создавать эффектные сайты даже начинающим разработчикам.

5) Единая работа компонент

Bootstrap не просто вставляет какие-то элементы на страницу, но сразу же устанавливает их оформление и взаимодействие с пользователем через JavaScript. Вы получаете полностью работающий набор компонент, который достаточно добавить и настроить под себя.

Недостатки, конечно же, тоже имеются и они вытекают в основном из универсальности системы.

1) Файлы библиотеки, даже сжатые, занимают довольно много места и увеличивают нагрузку на сервер. С повышением опыта нужно загружать и устанавливать только требуемые компоненты Bootstrap, сокращая тем самым объем файлов.

2) Предлагаемые в библиотеке стили могут не подходить под дизайн разрабатываемого сайта и придется много переделывать. В таких случаях, как известно, проще написать все самому с нуля.

3) Так же минусом является, что кода обычно в библиотеке написано больше, чем если бы вы написали при разработке с нуля. Потому что, когда вы делаете самостоятельно, вы реализуете только необходимый функционал и все. В Bootstrap же есть все на все случаи жизни. Даже то, что вам может не пригодиться. Но опять же, эта проблема очень легко решается тем, что вы можете сами выбирать, какие компоненты фреймворка загрузить в css-файл. Например, вы вообще можете скачать только сетку, а все остальное делать самостоятельно.

4) Еще один минус – шаблонный дизайн. Действительно, заходя на различные сайты можно увидеть одинаковые кнопки и другие элементы фреймворка. Сразу становиться понятно, что они сделаны в Bootstrap. Но и эта проблема легко решается, потому что она будет существовать только в том случае, если вы будете использовать только готовые компоненты фреймворка и ничего никогда не изменять под себя.

5) Если вы, например, подключите только сетку Bootstrap, то сможете воссоздать любой дизайн, при этом пользуясь очень удобной гибкой сеткой фреймворка.

Bootstrap является всеобъемлющим фреймворком. Это означает, что в него заложено много компонентов. Например, выпадающее меню, кнопки, индикаторы состояния, списки, заголовки и т.д. Знания этих простых элементов и свойств поможет в создании однотипного сайта. В Bootstrap достаточно изучить немного сам фреймворк и все эти вещи вы сможете использовать очень быстро.

## Spring MVC

Spring – это свободно распространяемый фреймворк, созданный Родом Джонсоном (Rod Johnson) и описанный в его книге «ExpertOne-on-One: J2EE Design and Development». Он был создан с целью устранить сложности разработки корпоративных приложений и сделать возможным использование простых компонентов JavaBean для достижения всего того, что ранее было возможным только с использованием EJB. Однако область применения Spring не ограничивается разработкой программных компонентов, выполняющихся на стороне сервера. Любое Java-приложение может использовать преимущества фреймворка в плане простоты, тестируемости и слабой связанности [1]

Фреймворк Spring MVC обеспечивает архитектуру паттерна Model – View – Controller (Модель – Отображение – Контроллер) при помощи слабо связанных готовых компонентов. Паттерн MVC разделяет аспекты приложения (логику ввода, бизнес-логику и логику UI), обеспечивая при этом свободную связь между ними [2]

1) Model (Модель) инкапсулирует (объединяет) данные приложения, в целом они будут состоять из POJO (Java-объектов).

2) View (Отображение, Вид) отвечает за отображение данных Модели, — как правило, генерируя HTML, которые мы видим в своем браузере.

3) Controller (Контроллер) обрабатывает запрос пользователя, создает соответствующую Модель и передает ее для отображения в Вид.

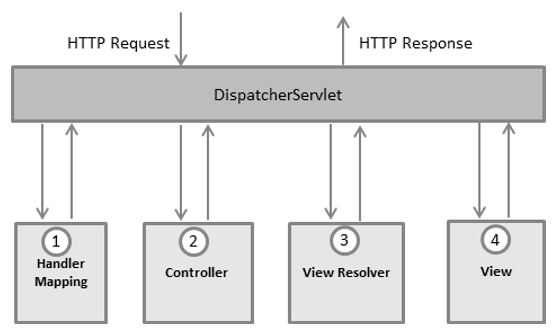
Вся логика работы Spring MVC построена вокруг DispatcherServlet, который принимает и обрабатывает все HTTP-запросы (из UI) и ответы на них. Рабочий процесс обработки запроса DispatcherServlet'ом проиллюстрирован на рисунке 1.6.

Рисунок 1.6 – Процесс обработки запроса

Ниже приведена последовательность событий, соответствующая входящему HTTP-запросу:

1) После получения HTTP-запроса DispatcherServlet обращается к интерфейсу HandlerMapping, который определяет, какой Контроллер должен быть вызван, после чего, отправляет запрос в нужный Контроллер.

2) Контроллер принимает запрос и вызывает соответствующий служебный метод, основанный на GET или POST. Вызванный метод определяет данные Модели, основанные на определенной бизнес-логике, и возвращает в DispatcherServlet имя Вида (View).

3) При помощи интерфейса ViewResolver DispatcherServlet определяет, какой Вид нужно использовать на основании полученного имени.

4) После того, как Вид (View) создан, DispatcherServlet отправляет данные Модели в виде атрибутов в Вид, который в конечном итоге отображается в браузере.

Все вышеупомянутые компоненты, а именно, HandlerMapping, Controller и ViewResolver, являются частями интерфейса WebApplicationContext extends ApplicationContext, с некоторыми дополнительными особенностями, необходимыми для создания web-приложений.

# ГЛАВА 2. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ИНТЕРНЕТ-ВИТРИНЫ ТАКЕЛАЖА»

## 2.1 Разработка прототипов страниц интернет-витрины

Перед началом разработки сайта стоит иметь представление о внешнем виде страниц.

Каждая страница содержит шапку с несколькими пунктами меню, которые содержат в себе ссылки для перехода страниц (рисунок 2.1).

Рисунок 2.1 – Шапка сайта

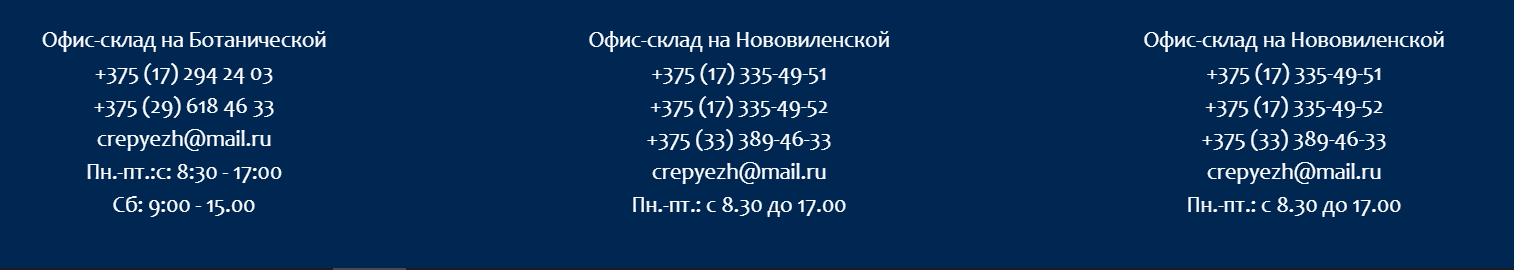
По мимо шапки на странице имеется подвал, в котором прописана контактная информация о магазине (рисунок 2.2).

Рисунок 2.2 – Подвал сайт

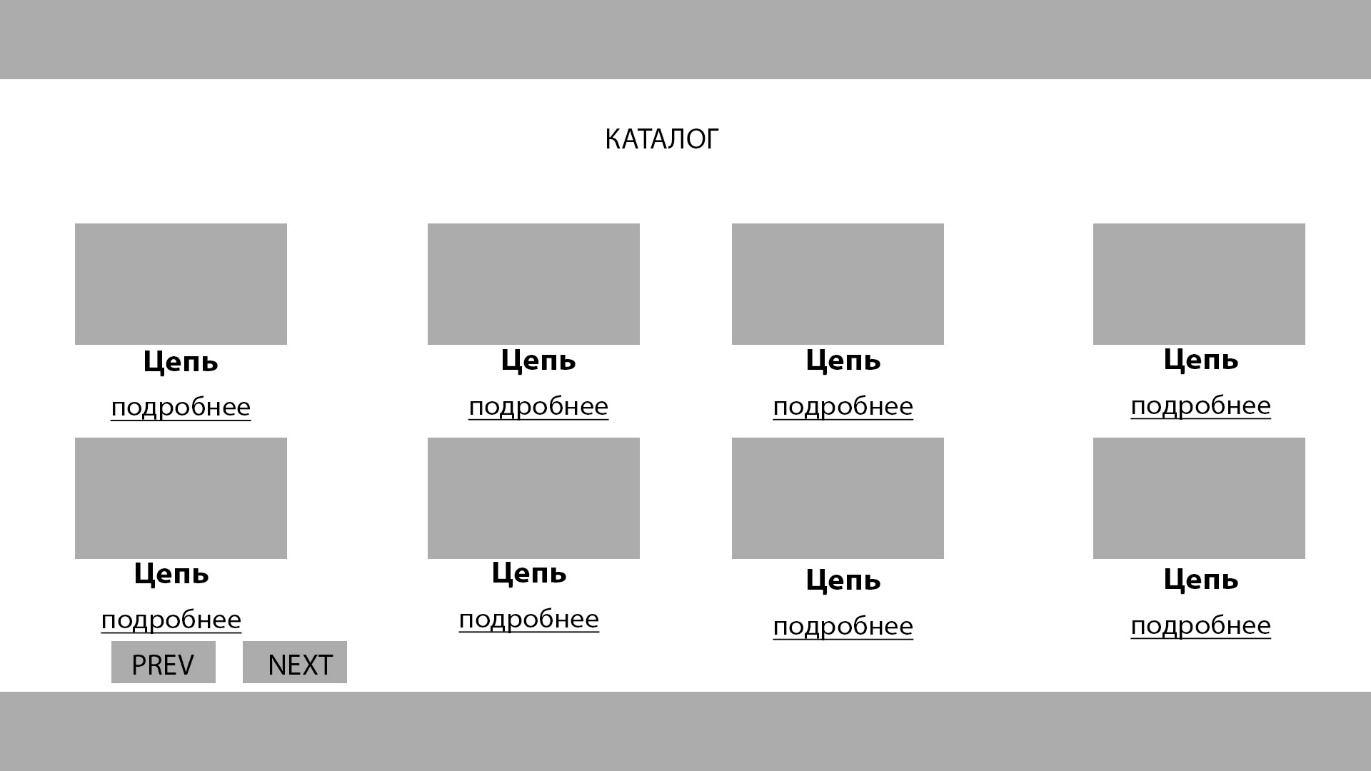
На рисунках 2.3 и 2.4 представлен прототип страниц каталога и отдельной страницы товара.

Рисунок 2.3 – Страница каталога сайта

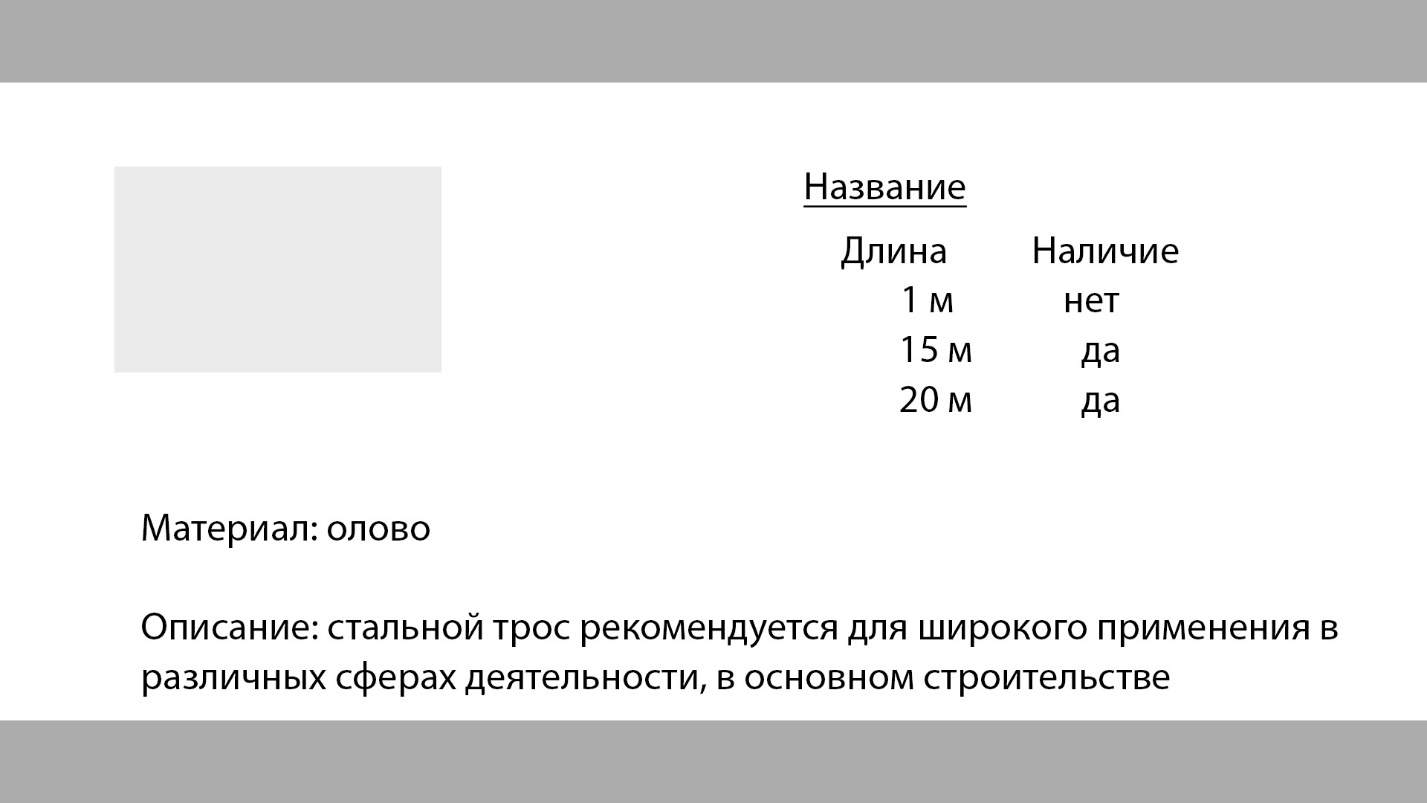


Рисунок 2.4 – Отдельная страница товара

## 2.2 Работа со Spring

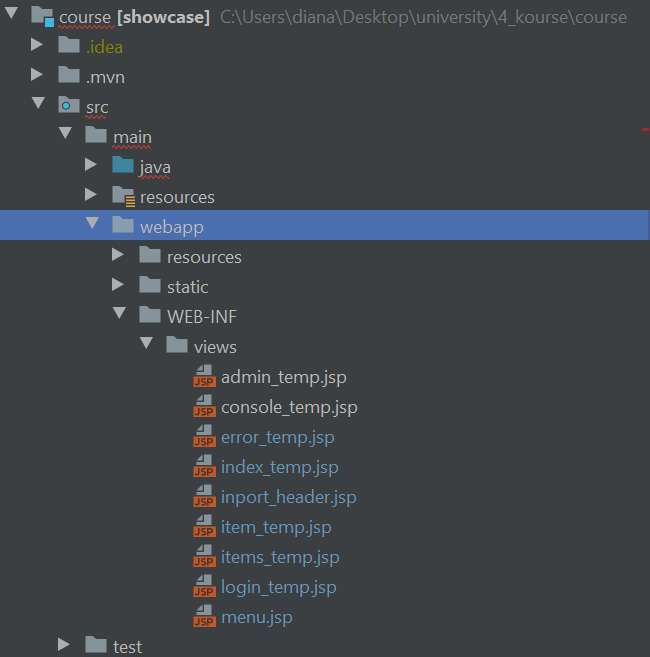
Структура файлов нашего проекта содержит следующие элементы (рисунок 2.5).

Рисунок 2.5 – Файловая структура проекта

Первым шагом при настройке Spring является изменения в файле pom.xml. Данный файл содержит в себе список зависимостей *spring-boot-starter-web*, *spring-boot-maven-plugin, jar*, а также настройка версии java и maven.

Файл *application.properties* содержит в себе порт локального сервера, по которому можно просмотреть написанные нами страницы. По мимо этого хранит в себе информации о имени пользователя базы данных и пароль от нее, максимальный размер загружаемых объектов на сайт, а также задается фиксированное количество товаров на странице каталога (листинг 2.1).

*Листинг 2.1 - Структура файла application.properties*

spring.datasource.url=jdbc:mysql://localhost:3306/showcase?serverTimezone=UTC  
spring.datasource.username=root  
spring.datasource.password=18512  
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect  
spring.jpa.generate-ddl=true spring.jpa.hibernate.ddl-auto=update  
server.port=8081  
  
spring.mvc.view.prefix=/WEB-INF/views/  
spring.mvc.view.suffix=.jsp  
  
  
spring.servlet.multipart.max-file-size=100MB  
spring.servlet.multipart.max-request-size=100MB  
  
  
count.per\_page=20

Наша интернет-витрина представляет из себя каталог, с возможность просмотра товаров и внесения изменения в них от лица админа. Для реализации этого мы создаем классы-контроллеры, в которые обращаются к серверу для дальнейшего извлечения данных в виде объектов из базы данных. Пример одного из классов-контроллеров находится на листинге 2.2.

Также существует класс, поля которого являются именами столбцов в базе данных. В дальнейшем, при обращении к полям этого класса мы получаем отдельные элементы из базы данных по определенному параметру (листинг 2.3).

*Листинг 2.2 – Пример части класса-контроллера*

Package by.fsc.showcase.controllers;  
  
import by.fsc.showcase.beans.Item;  
import by.fsc.showcase.service.ItemService;  
import by.fsc.showcase.util.FileUtil;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.\*;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.Part;  
import java.io.IOException;  
import java.util.Date;  
  
@Controller  
public class ItemController{  
 @Autowired

private ItemService itemService;  
 @RequestMapping(path="/items", method=RequestMethod.GET)  
 public String getItems(ModelMap model, @RequestParam(defaultValue = "0") int page) {  
 model.addAttribute("items", itemService.getAllItems(page));  
 return "items\_temp";  
 }  
  
  
 @GetMapping(path="/admin/items/remove/{id}")  
 public String removeItem(@PathVariable int id) throws IOException {  
 itemService.remove(id);  
 FileUtil.removeFile(id);  
 return "redirect:/items";  
 }

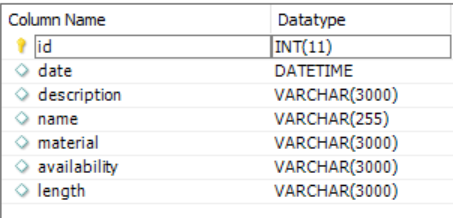
Сама база данных выглядит следующим образов и содержит такие поля как id элемента, описание, материал из которого изготовлен продукт, поле, отвечающее за наличие продукта и количество (рисунок 2.6).

Рисунок 2.6 – Структура базы данных

Основными файлами, содержащими в себе вид страниц, являются файлы с расширением jsp Интернет-витрина содержит в себе 3 основные страницы: страница каталога, страница товара, страница входа в панель админа. Дополнительными в данном проекте являются файлы содержащие меню, которое присутствует на каждой странице интернет-витрины и изменяется, если пользователь авторизирован, страница ошибки, на случай сбоя в работе частей проекта.

*Листинг 2.3 – Класс с именами столбцов базы данных*

Package by.fsc.showcase.beans;  
  
  
import lombok.Data;  
  
import javax.persistence.\*;  
import java.util.Date;  
  
@Data

@Entity(name="items")  
public class Item {  
 @Id  
 @GeneratedValue(strategy=GenerationType.AUTO)  
 private Integer id;  
  
 @Column private String name;  
  
 @Column(length =3000)  
 private String description;  
  
 @Column(length=3000)  
 private String material;  
  
 @Column(length=3000)  
 private String length;  
  
 @Column(length=3000)  
 private String availability;  
  
 @Column  
 private Date date;  
}

Страница каталога представляет из себя перечень товаров, представленных на витрине. Максимальное количество товаров на странице – 20, если оно превышает данное число, то появляются кнопки, которые отвечают за навигацию между последующими страницами (листинг 2.4).

*Листинг 2.4 – Кнопки переключения страниц*

<div class="navigation">  
 <c:if test="${items.number> 0 }">  
 <button class="btn-secondary"><a href="/items?page=${items.number-1}">prev</a>  
 </button>  
 </c:if>  
  
 <c:if test="${items.number < items.totalPages – 1 }">  
 <button class="btn-secondary">  
 <a href="/items?page=${items.number+1}"> next</a>  
 </button>  
 </c:if>  
</div>

На странице отдельного продукта расположено его изображение, с описанием данного товара: материал, длина и наличие. Все данные считываются из базы данных и могут быть изменены админом.

При необходимости зайти в панель админа вы попадаете на страницу с формой для входа, которая требует от вас логин и пароль (рисунок 2.7). В случае неправильного ввода появляется сообщение, которое информирует вас об этом. При правильном введении данных вы получаете доступ к странице каталога с дополнительным функционалом, доступным только админу.

Рисунок 2.6 – Форма входа в админ-панель

На рисунке 2.8 Вы видите, что при входе в админ панель изменилась шапка страниц, появилась дополнительная кнопка с функцией выхода из панели. По мимо этого, на странице каталога появилась форма для добавления нового продукта. Вы выбираете изображение товара, а в остальные пункты формы вписываете соответствующее описание (рисунок 2.9).

Рисунок 2.8 – Шапка сайта после входа в админ-панель

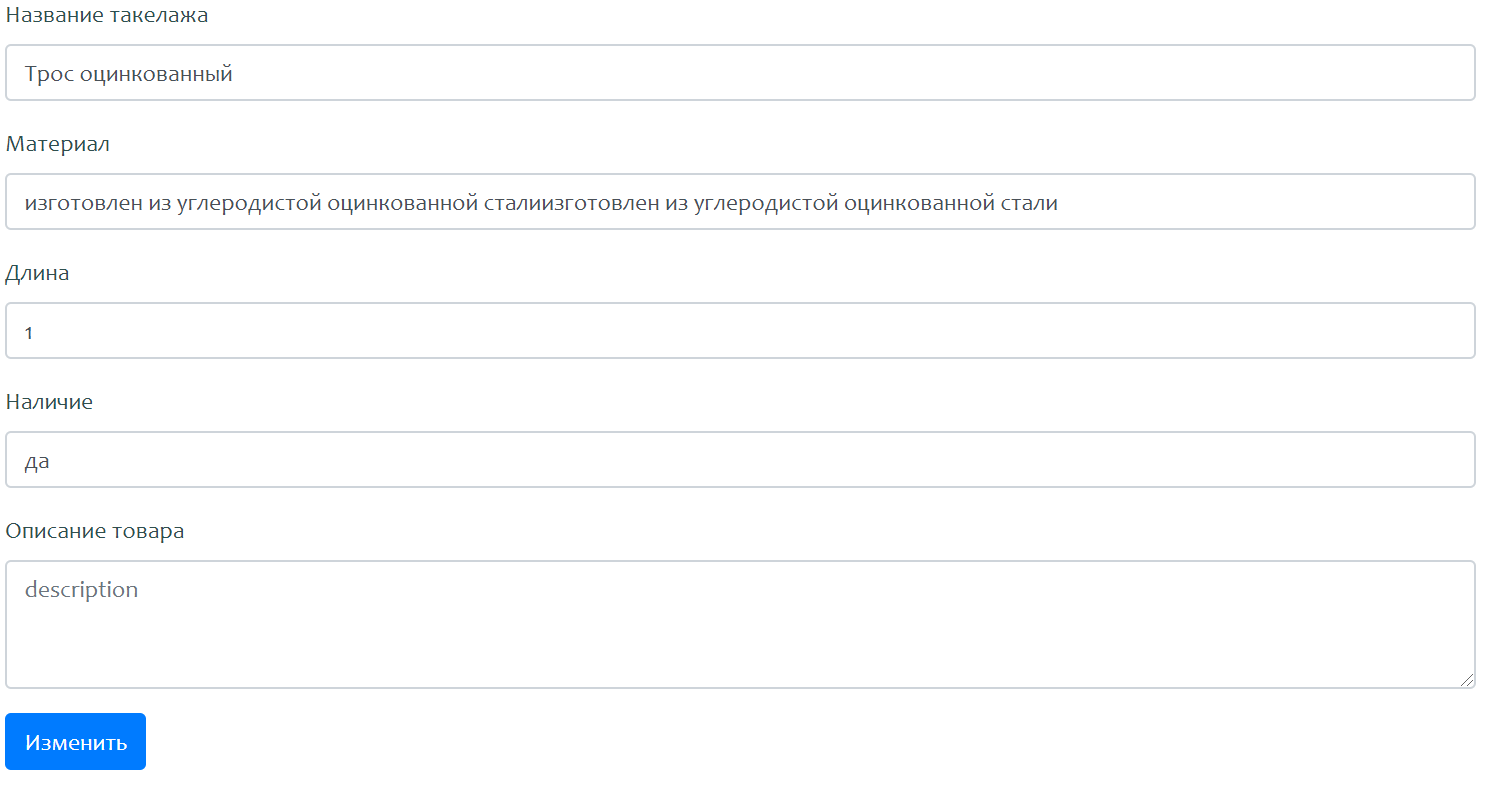
На странице товара появляется аналогичная форма, которая отвечает за изменение данных выбранного админом товаров. Можно загрузить новое изображение и изменить описание, свойства (рисунок 2.10).

Рисунок 2.10 – Форма изменения имеющегося товара на сайте

Рисунок 2.9 – Форма добавления нового товара

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были рассмотрены основные приципы создания веб-приложения и клиент-серверной архитектуры, такие как Spring Framework, Framework Bootstrap.

Разработанное веб-приложение в полной мере реализует поставленную цель и реализовано посредством последовательного выполнения задач проекта.

Система позволяет пользователю просмотреть каталог товаров на сайте, просмотреть полную информацию о товаре на персональной странице, а также при входе в админ-панель есть возможность добавления нового товара и редактирование при необходимости существующего.

Веб-приложения реализовано на языке Java с использованием Spring Framework, разработка проводилась в IDE Intellij IDEA, в качестве системы управления базами данных использовалась MySQL. Разработаны такие страницы как:

1) Каталог

2) Внутренняя каталожная

3) Страница входа в админ-панель

4) Каталог с доступом добавления товаров

5) Внутренняя каталожная с возможностью изменения данных о имеющемся товаре

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Уолс, К. Spring в действии. – М.: ДМК Пресс, 2013. – 752 с

2) Deinum, M. Pro Spring MVC: with web flow / M. Deinum, K. Serneels. – Apress, 2012. – 565 p.

3) Bootstrap. Документация на русском [Электронный ресурс]. – Режим доступа: <http://bootstrap-4.ru/>.

4) Введение в HTML5 [Электронный ресурс]/ Режим доступа: https://developer.mozilla.org/ru/docs/HTML/HTML5Введение\_в\_HTML5. — Дата доступа: 01.12.2019.

5) IntelliJ IDEA the Java IDE [Электронный ресурс]. – Режим доступа: https://www.jetbrains.com/idea/. – Дата доступа: 23.11.2019.

# ПРИЛОЖЕНИЕ

Рисунок A1 – Главная страница интернет-витрины

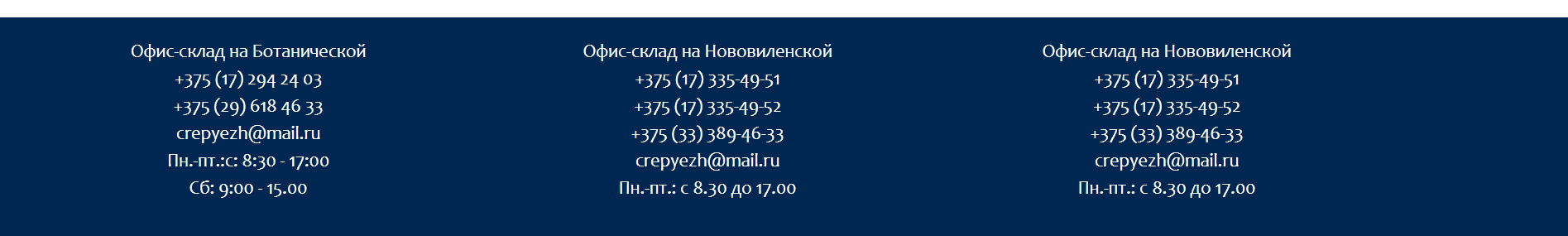


Рисунок A2 – Страница каталога

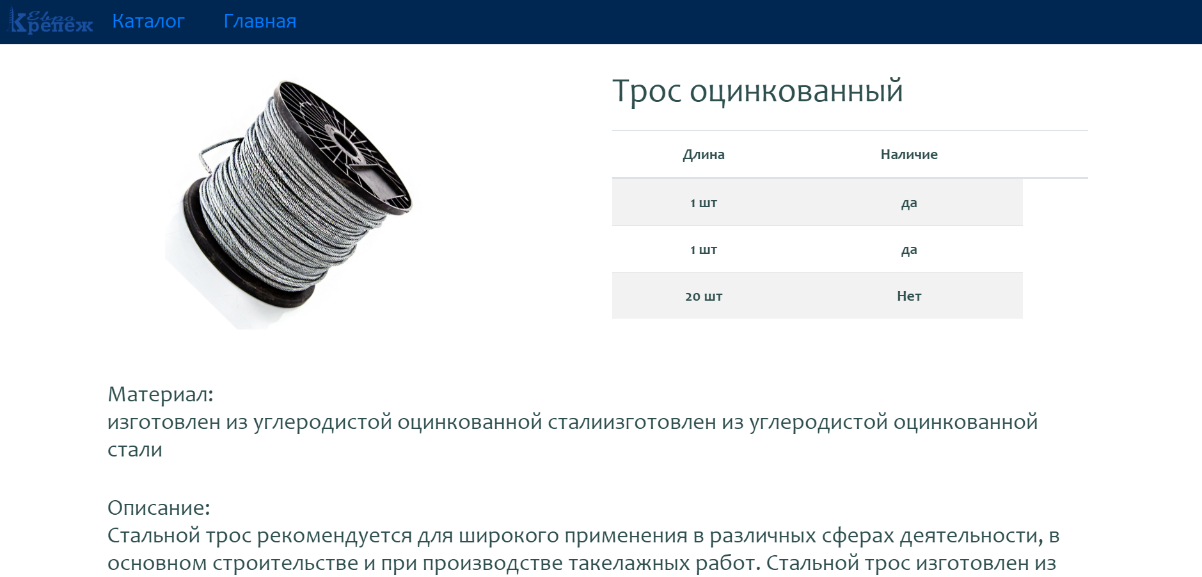
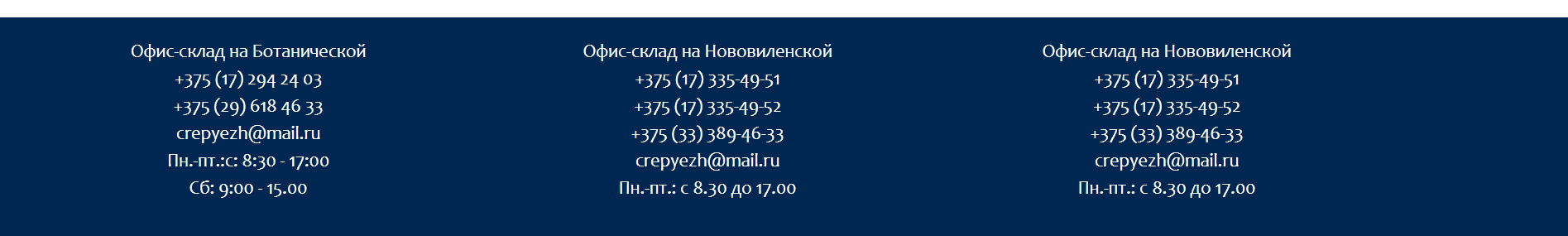
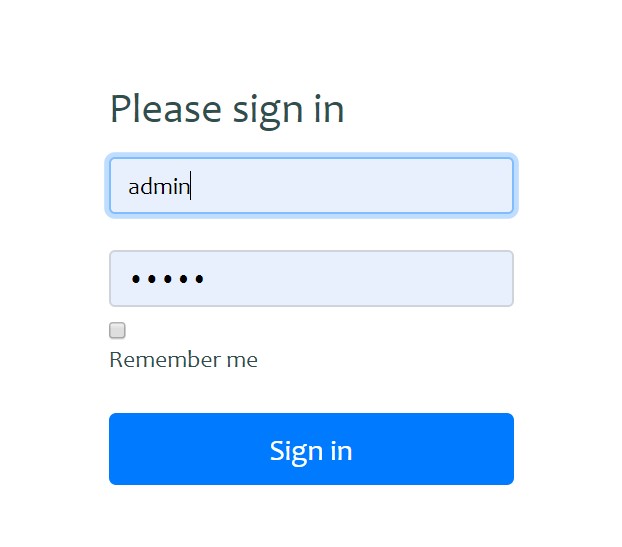


Рисунок A4 – Страница входа в админ панель

Рисунок A3 – Отдельная страница товара



Рисунок A5 – Форма добавления нового товара