

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ СОЦИОКУЛЬТУРНЫХ КОММУНИКАЦИЙ
Кафедра информационных технологий

РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ МАГАЗИНА ПЛАТЬЕВ
«АСОЛИЯ»

Курсовой проект

Кравчени Анны Владимировны
Студентки 4 курса,
специальности
«Прикладная информатика»
Научный руководитель:
Доцент кафедры
Информационных технологий
Царик Сергей Всеволодович

Допущен к защите

«___» _____ 2019 г.

Минск, 2019

РЕФЕРАТ

Количество страниц – 18, изображений – 7, листингов – 2, источников – 10, приложений – 1.

Ключевые слова: ИНТЕРНЕТ-ВИТРИНА, ВЕБ-СТРАНИЦА, ВЕБ-САЙТ, ЖЕНСКИЕ ПЛАТЬЯ, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC.

Объектом исследования курсовой работы является интернет-витрина женских платьев «Асолия». Методы исследования – архитектура клиент-серверного приложения, шаблон проектирования Model-Viewer-Controller.

В результате выполнения курсовой работы было разработана интернет-витрина женских платьев «Асолия». Веб-приложение реализовано на языке Java с использованием Spring Framework и Framework Bootstrap. Область применения – сектор Интернета, посвященный разработке веб-приложений.

РЭФЕРАТ

Колькасць старонак - 18, малюнкаў - 7, лістынгаў - 2, крыніц - 10, прыкладанняў - 1.

Ключавыя словы: ІНТЭРНЭТ-вітрын, ВЭБ-СТАРОНКА, ВЭБ-САЙТ, жаночыя сукенкі, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC.

Аб'ектам даследавання курсавой работы з'яўляецца інтэрнэт-вітрына жаночых сукенак «Асоль». Метады даследавання - архітэктара кліент-сервернага прыкладання, шаблон праектавання Model-Viewer-Controller.

У выніку выканання курсавой работы было распрацавана інтэрнэт-вітрына жаночых сукенак «Асоль». Вэб-дадатак рэалізавана на мове Java з выкарыстаннем Spring Framework і Framework Bootstrap. Вобласць ужывання - сектар Інтэрнэту, прысвечаны распрацоўцы вэб-прыкладанняў.

ESSAY

Number of pages - 18, images - 7, listings - 2, sources - 10, applications - 1.

Keywords: INTERNET SHOWROOM, WEB PAGE, WEBSITE, WOMEN'S DRESSES, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC.

The Object of study of the course work is an online showcase of women's dresses "Asolia". Research methods - architecture of the client-server application, design pattern Model-Viewer-Controller.

As a result of the course work, an Internet showcase of Asolia women's dresses was developed. The web application is implemented in Java using the Spring Framework and Framework Bootstrap. Scope - The Internet sector dedicated to the development of web applications.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. ОСНОВНЫЕ ИНСТРУМЕНТЫ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЯ	5
1.1 Основные инструменты прототипирования	5
1.2 Инструменты разработки интернет-витрины	7
ГЛАВА 2. РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ	12
2.1 Разработка прототипов страниц	12
2.2 Разработка интернет-витрины	14
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18
ПРИЛОЖЕНИЯ	19

ВВЕДЕНИЕ

Использование Интернета стало необходимостью во многих областях нашей жизни. Поиск и получение информации, заработок, обучение, развлечение, общение и покупки – все это является лишь частью предоставляемых интернетом возможностей. Разнообразие контента и большое количество сайтов в сети упрощают нашу жизнь и даже делают ее интересней.

Актуальность создания интернет-витрины заключается в том, что покупателям удобнее ознакомиться с ассортиментом магазина перед тем, как пойти туда. Магазинам же это является дополнительной рекламой, когда пользователям на поисковые запросы будет выдаваться сайт, где можно узнать о товарах магазина, их характеристиках, стоимости и отзывах на них.

Целью данного курсового проекта является разработка интернет-витрины магазина платьев «Асолия» с использованием языка Java, фреймворков Spring Framework и Bootstrap.

Чтобы соответствовать данным требованиям, сайт должен обладать следующим функционалом:

- 1) содержать каталог товаров;
- 2) содержать страницу каждого товара.

Для достижения данной цели были поставлены следующие задачи:

- 1) изучить возможные средства разработки интернет-витрины;
- 2) реализовать MVC модель веб-приложения интернет-витрины;
- 3) реализовать адаптивное отображение веб-приложения с помощью Bootstrap Framework.

ГЛАВА 1. ОСНОВНЫЕ ИНСТРУМЕНТЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ

Разработка дизайна сайтов и приложений играет важную роль в наши дни. От внешнего вида конечного продукта зависит все: от удобства использования и до привлечения новых клиентов. Поэтому главную часть любого проекта составляет дизайн интерфейса.

Рассмотрим несколько программ, которые лучше всего подходят для прототипирования.

1.1 Основные инструменты прототипирования

Adobe Photoshop. Adobe Photoshop – многофункциональный графический редактор, разработанный и распространяемый фирмой Adobe Systems. В основном работает с растровыми изображениями, однако имеет некоторые векторные инструменты. Продукт является лидером рынка в области коммерческих средств редактирования растровых изображений и наиболее известной программой фирмы Adobe. В настоящее время Photoshop доступен на платформах macOS, Windows, в мобильных системах iOS, Windows Phone и Android.

Adobe Photoshop является одним из самых популярных графических редакторов. Он используется фотографами, дизайнерами и т.д.

Программа Adobe PhotoShop позволяет как загружать готовые изображения из любых источников, редактировать их, так и создавать новые. Инструменты программы позволяют легко выполнить ретуширование, изменение яркости, контрастности, корректировку цвета и другие операции над изображением. Множество графических инструментов, разнообразные спецэффекты позволяют придать изображению самый разнообразный вид.

Рассмотрим преимущества и недостатки программы. Преимущества:

1) Поддерживает все форматы медиа-файлов и инструментарию для их редактирования.

2) Имеет полноценный функционал для коррекции и создания картинок.

3) Поддерживает полноценную работу графических планшетов.

4) Может расширяться за счет плагинов.

5) Существует бесплатная версия.

Недостатки:

1) Adobe Photoshop работает только с растровой графикой.

Figma. Figma – кросс-платформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков.

Figma позволяет разрабатывать интерфейсы в онлайн-приложении. У Figma две ключевые особенности: доступ к макету прямо из окна браузера и

возможность совместной работы над документами. Интерфейс программы представлен на рисунке 1.1.

Преимущества программы Figma:

- 1) Простота освоения.
- 2) Производительность.
- 3) Автоматическое сохранение и контроль версий.
- 4) Удобное выравнивание элементов.
- 5) Копирование стилей объектов.
- 6) Работа с графикой.
- 7) Интерактивные прототипы и анимированные переходы.
- 8) Командная работа.
- 9) Комментирование.

Недостатки работы в программе:

- 1) Слабые возможности работы с текстом.
- 2) Обязательное наличие интернета.

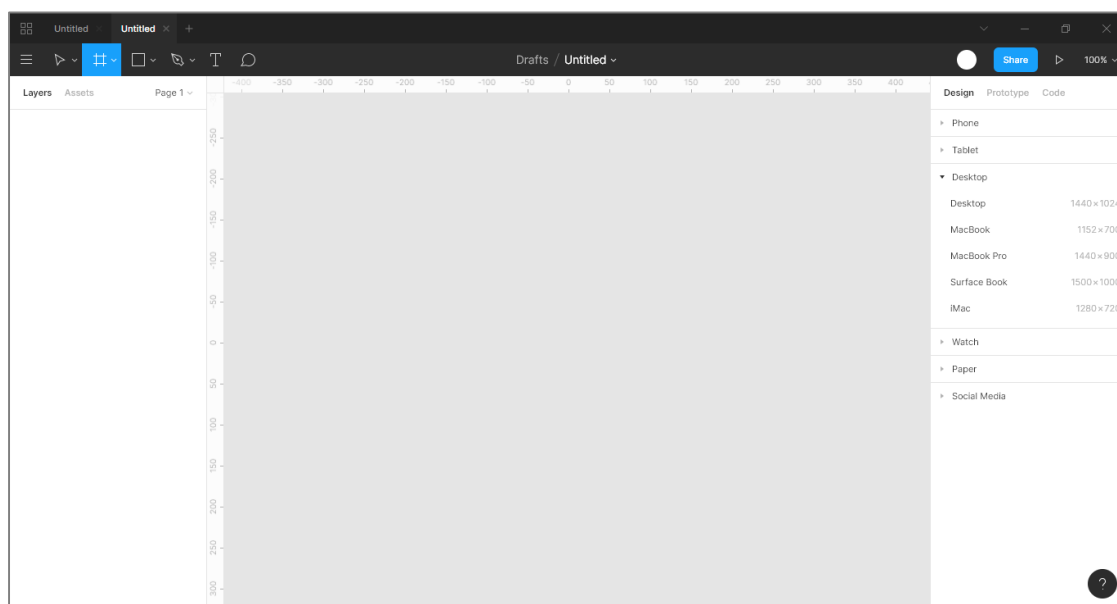


Рисунок 1.1 – Figma

Sketch. Sketch – платный редактор векторной графики, выпускаемый голландской компанией Bohemian Coding исключительно для устройств на базе операционной системы macOS. Sketch – фактически один из основных инструментов графического дизайна и безальтернативное для пользователей Apple по набору возможностей. Распространяется по модели подписки.

Основным конкурентом для Sketch являются продукты Adobe, Figma и Framer.

В отличие от Photoshop, Sketch не использует собственных средств для отображения создаваемого контента – эти функции возложены на встроенные инструменты macOS: QuickTime, Quartz Extreme, Core Image, Core Animation, OpenGL и ColorSync.

В своей работе и создании продуктов Sketch активно используют такие компании как Apple, Facebook, Google. Sketch имеет большие возможности интеграции с другим ПО и веб-сервисами Jira, Zeplin и Avocode.

Можно выделить следующие преимущества и недостатки работы в Sketch. Преимущества:

- 1) Узкопрофессиональный инструмент для дизайна интерфейсов.
- 2) Экспорт графики.
- 3) Простота интерфейса.
- 4) Удобная работа с текстом.
- 5) Регулярные обновления.

Недостатки:

- 1) Нет совместимости с продуктами Adobe.
- 2) Проблемы при работе с большими файлами.
- 3) Несовершенные векторные инструменты.

1.2 Инструменты разработки интернет-витрины

Веб-приложение – клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети.

Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует веб-страницу и отправляет ее клиенту по сети с использованием протокола HTTP.

Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

Фреймворк Spring. Spring Framework (или Spring) – универсальный фреймворк с открытым исходным кодом для Java-платформы. Также существует ответвление для платформы .NET Framework, названный Spring.NET. Фреймворк был впервые выпущен под лицензией Apache 2.0 license в июне 2003 года. Первая стабильная версия 1.0 была выпущена в марте 2004. Текущая версия – 5.0.1. Официальный сайт – spring.io.

Spring имеет собственную MVC-платформу веб-приложений. Класс DispatcherServlet является основным контроллером фреймворка и отвечает за делегирование управления различным интерфейсам, на всех этапах выполнения HTTP-запроса. Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса – быть простым и ясным, чтобы пользователям было

легко его заново реализовать, если это будет необходимо. MVC позволяют создавать более чистый front-end-код. Все интерфейсы тесно связаны с Servlet API. Наиболее важные интерфейсы, определенные Spring MVC, это:

1) **HandlerMapping**: выбор класса и его метода, которые должны обработать данный входящий запрос на основе любого внутреннего или внешнего для этого запроса атрибута или состояния;

2) **HandlerAdapter**: вызов и выполнение выбранного метода обработки входящего запроса;

3) **Controller**: включен между Моделью (Model) и Представлением (View) и переключает поток информации из модели в представление и обратно;

4) **View**: отвечает за возвращение ответа клиенту в виде текстов и изображений;

5) **ViewResolver**: выбор, какое именно View должно быть показано клиенту;

6) **LocaleResolver**: получение и, возможно, сохранение локальных настроек (язык, страна, часовой пояс) пользователя;

7) **MultipartResolver**: обеспечивает загрузку на сервер локальных файлов клиента;

8) **HandlerInterceptor**: перехват входящих запросов. Сопоставим, но не эквивалентен сервлет-фильтрам (использование не является обязательным и не контролируется DispatcherServlet-ом).

Spring MVC предоставляет разработчику следующие возможности:

1) понятное разделение между слоями в MVC и запросах;

2) стратегия интерфейсов – каждый интерфейс делает только свою часть работы;

3) интерфейс всегда может быть заменен альтернативной реализацией;

4) интерфейсы тесно связаны с Servlet API;

5) предоставляет доступ к базам данных и поддерживает все популярные СУБД;

6) высокий уровень абстракции для веб-приложений.

В веб-приложениях можно использовать различные части Spring, а не только Spring MVC.

Язык разметки HTML5 и фреймворк Bootstrap. Язык разметки HTML5 – стандартизированный язык разметки, позволяющий создавать разнообразные сайты и веб–приложения.

Одним из главнейших отличий HTML5 от предыдущих версий является семантичность. Данные, которые заключены между открывающим и закрывающим тегом должны соответствовать смыслу данного тега. HTML5 вводит несколько новых элементов и атрибутов, которые отражают типичное

использование разметки на современных веб-сайтах. Некоторые из них – семантические замены для использования универсальных блочных (<div>) и строчных () элементов, например, <nav> (блок навигации по сайту), <footer> (обычно относится к нижней части страницы или последней строке HTML кода) или <audio> и <video> вместо <object>. Основными тегами являются: <html>, <head> и <body>.

Тег <head> хранит элементы, которые помогают браузеру в работе с данными. Внутри этого тега находятся метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем, такой как описание сайта, ключевые слова и другие данные. Содержимое тега <head> не отображается на веб-странице, кроме тега <title>, который устанавливает заголовок окна веб-страницы.

В теге <body> находится информация, которая отображается в окне браузера. Теги, которые могут находиться внутри тега <body>, с указанием их назначения представлены в таблице 1.1. Согласно спецификации HTML5 теги <head> и <body> можно не прописывать. HTML – страница легко загрузится и при отсутствии данных тегов, однако желательно их прописывать, так как они несут в себе определенную семантику.

Чтобы расширить возможности отдельных тегов и более гибко управлять содержимым контейнеров и применяются атрибуты тегов. Основными атрибутами, которые есть практически у всех тегов, являются id и class. Атрибут id предоставляет возможность связать элемент с CSS. Он всегда уникальный и может существовать только в единственном экземпляре на странице.

Атрибут class устанавливает стилевой класс, который позволяет связать тег с CSS. Один и тот же класс можно задать для нескольких различных тегов.

Таблица 1.1 – Основные теги, размещаемые в теге <body>

Тег	Назначение тега
<header>	Включает в себя главную информацию о сайте: логотип со ссылкой на главную страницу, название сайта. Обычно выглядит одинаково на всех страницах сайта.
<nav>	Используется для обозначения панели навигации по сайту.
<footer>	Обычно относится к нижней части страницы. Сюда выносятся копирайт, контакты и ссылки на социальные сети, если они имеются.
<figure>	Содержит картинки, диаграммы, фотографии, схемы.
<section>	Обозначает раздел веб-страницы.

<h1> – <h6>	Теги заголовков. Каждый из тегов указывают на сравнительную важность контента, расположенного после каждого из заголовков.
<a>	Обозначает ссылки.
	Предназначен для добавления изображения на страницу.

Фреймворк Bootstrap – фреймворк для создания адаптивных веб-приложений и сайтов. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Основное отличие верстки с помощью Bootstrap от верстки нативными средствами заключается в том, что Bootstrap включает в себя использование компонентов. Компоненты представляют из себя часто используемые готовые HTML блоки с предопределенными стилями. Иногда компоненты используют JavaScript. Верстальщик может использовать как готовый компонент, так и определить свой внешний вид для него. Для этого часто нужно лишь поменять значение переменных в Bootstrap. Если нужны более гибкие изменения, верстальщик всегда может изменить HTML и CSS по своему усмотрению.

Bootstrap используется, если:

- 1) в проекте много страниц;
- 2) страницы собираются из простых базовых элементов (Базовый – это кнопка или таблица, а сложный – это пост в Facebook или комментарий к нему, они состоят из нескольких тегов);
- 3) нет мест, где изменение и эксперименты с дизайном могут сильно повышать/понижать метрики (например, корзина или страница товара в интернет-магазине);
- 4) никогда не будет глобального редизайна;
- 5) типизация страниц окупается скоростью их внедрения.

Bootstrap не используется для:

- 1) больших интернет-магазинов и новостных сайтов;
- 2) сайты со сложными повторно используемыми в разных местах компонентами;
- 3) промо-сайты и лендинги (одностраничные сайты, которые должны вызывать желание выполнить определенное целевое действие);
- 4) игры в браузерах.

Spring MVC. Фреймворк Spring MVC обеспечивает архитектуру паттерна Model – View – Controller (Модель – Отображение (Вид) – 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.

Вся логика работы Spring MVC построена вокруг DispatcherServlet, который принимает и обрабатывает все HTTP-запросы (из UI) и ответы на них. Процесс обработки запроса DispatcherServlet'ом показан на рисунке 1.1.

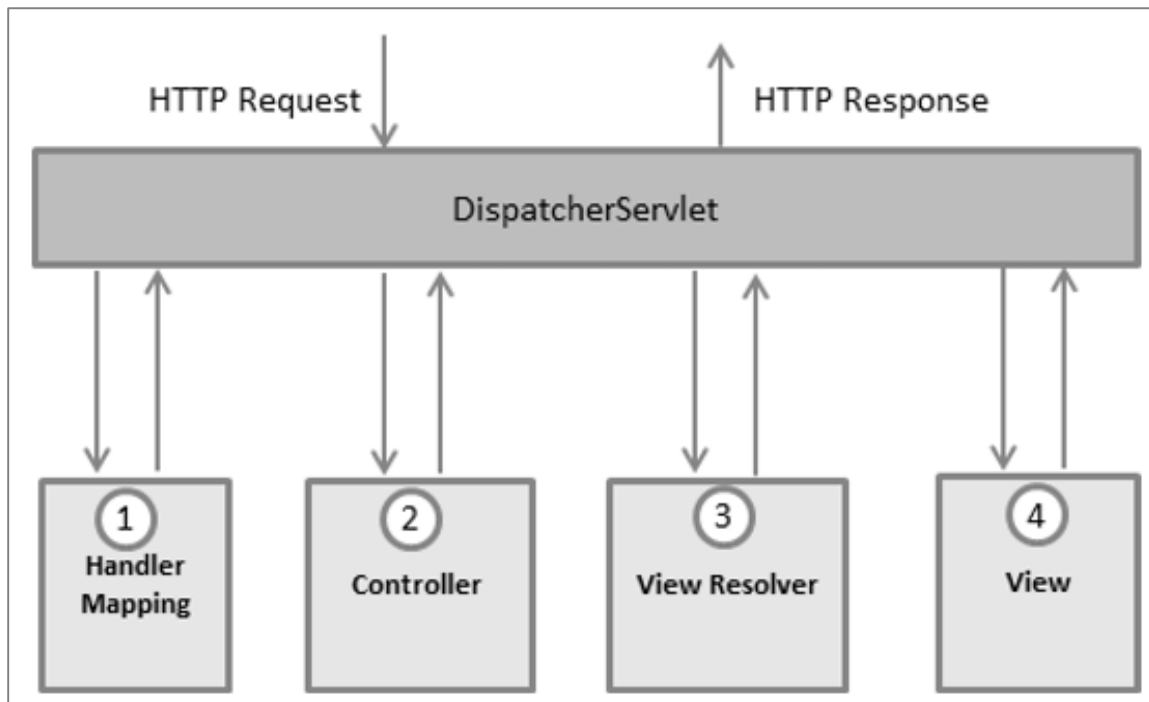


Рисунок 1.2 – Процесс обработки запроса DispatcherServlet'ом

Последовательность событий, соответствующая входящему HTTP-запросу:

- После получения HTTP-запроса DispatcherServlet обращается к интерфейсу HandlerMapping, который определяет, какой Контроллер должен быть вызван, после чего, отправляет запрос в нужный Контроллер.
- Контроллер принимает запрос и вызывает соответствующий служебный метод, основанный на GET или POST. Вызванный метод определяет данные Модели, основанные на определённой бизнес-логике и возвращает в DispatcherServlet имя Вида (View).
- При помощи интерфейса ViewResolver DispatcherServlet определяет, какой Вид нужно использовать на основании полученного имени.
- После того, как Вид (View) создан, DispatcherServlet отправляет данные Модели в виде атрибутов в Вид, который в конечном итоге отображается в браузере.

Компоненты HandlerMapping, Controller и ViewResolver, являются частями интерфейса WebApplicationContext extends plicationContext, с некоторыми дополнительными особенностями, необходимыми для создания web-приложений.

ГЛАВА 2. РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ

2.1 Разработка прототипов страниц

Инструментом прототипирования была выбрана программа Figma, т.к. она проста и удобна в использовании. Всего было разработано 4 страницы: каталог, страница товара, панель администратора и страница добавления нового товара.

На странице каталога товара расположена карусель с фотографиями и товар магазина с кратким описанием и кнопкой для перехода на страницу товара (рис. 2.1). На странице товара есть фотография товара с полным его описанием (рис. 2.2). На странице панели администратора расположена таблица с номером, названием товара и кнопки удаления и редактирования товара (рис. 2.3). Страница добавления товара содержит поля для ввода информации о новом товаре (рис. 2.4).

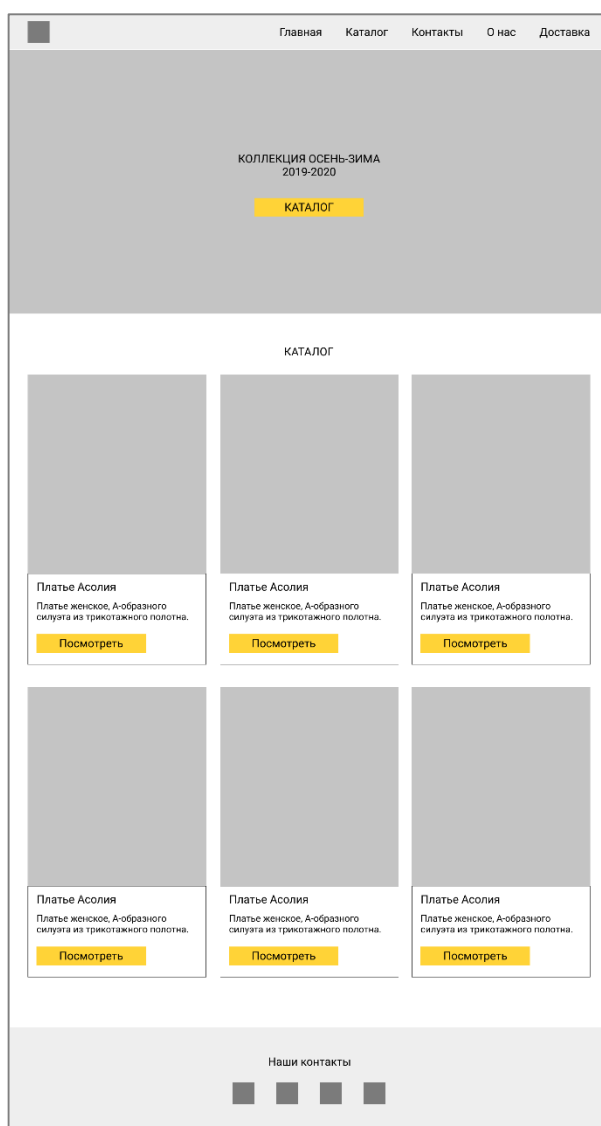


Рисунок 2.1 – Главная страница

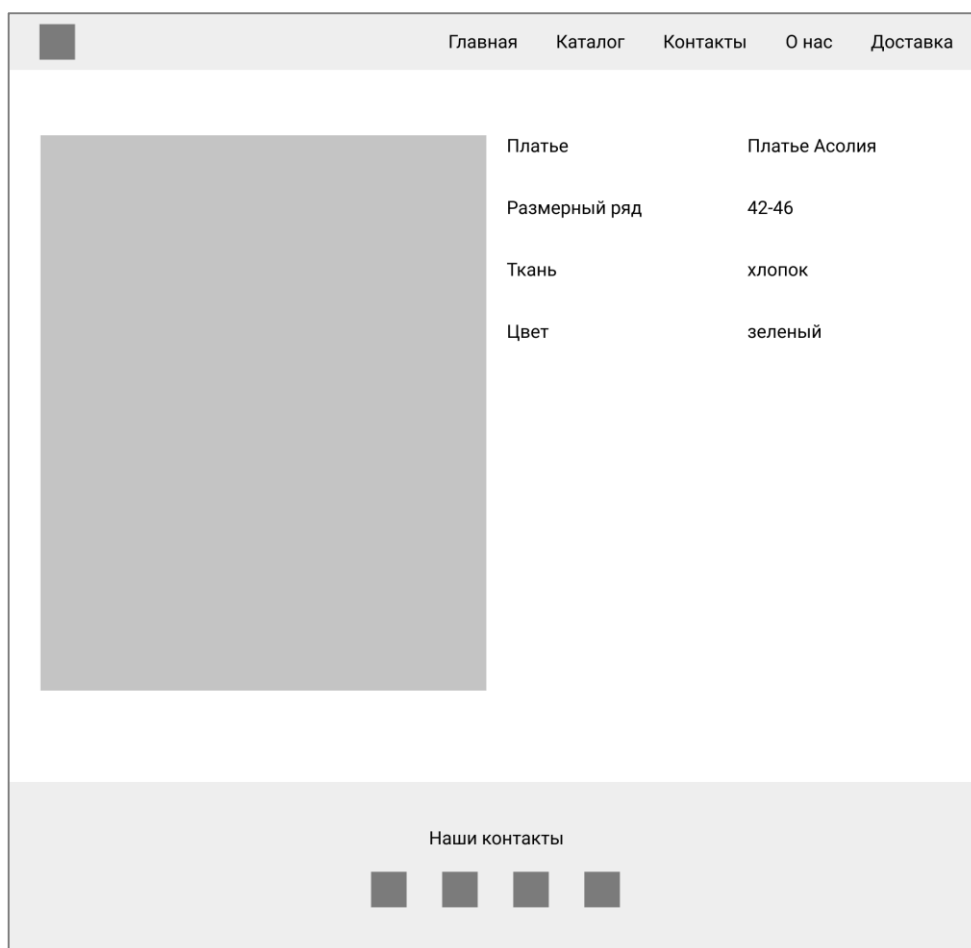


Рисунок 2.2 – Страница товара

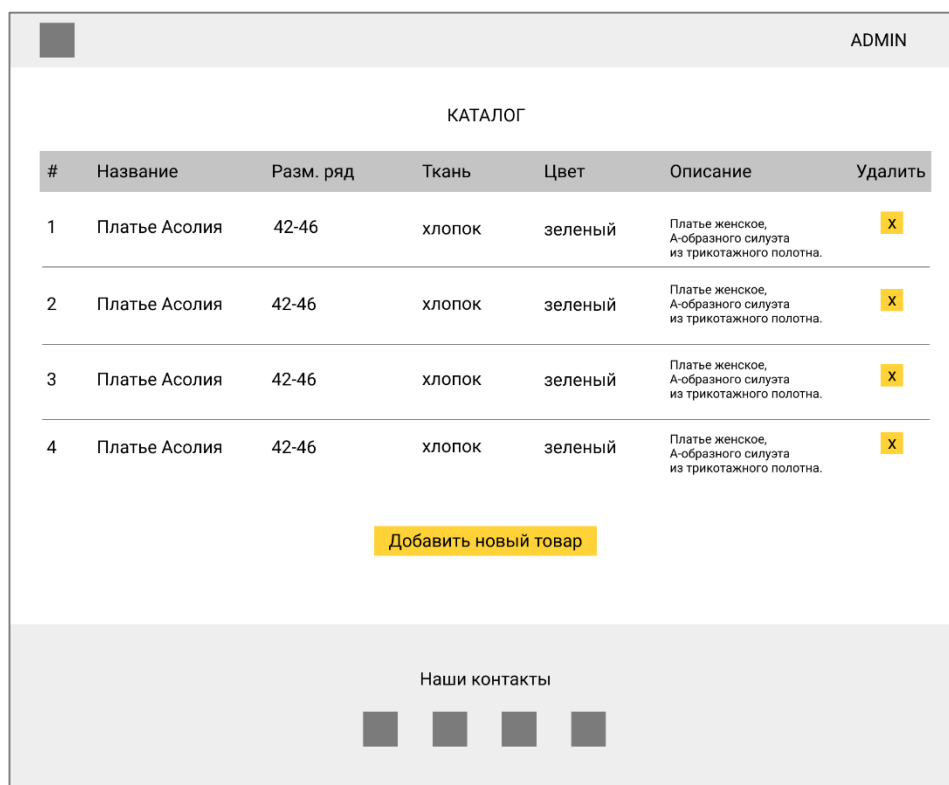


Рисунок 2.3 – Панель администратора

Рисунок 2.4 – Страница добавления товара

2.2 Разработка интернет-витрины

Целью данного курсового проекта является разработка интернет-витрины женских платьев «Асолия», с помощью которой покупатель может ознакомиться с ассортиментом магазина перед тем, как пойти туда. Исходя из поставленной задачи было выделено два вида страниц: «Каталог» и «Карточка товара». Кроме того, реализована страница с возможностью добавления новых товаров и изменения информации о существующих. Страница каталога представлена на рисунке 2.5.

За отображение страницы с каталогом, представленной на рисунке 2.1, отвечает файл `items-temp.jsp`. Все изображения хранятся в папке `/static/items/` и их названия совпадают с полем `id` товара. Чтобы картинки выводились на экран, сначала необходимо указать папку, в которой они находятся и далее прописывается расширение файла, в данном случае это `jpg`. Данный процесс показан в листинге 2.1.

Листинг 2.1 – Процесс отображения страницы каталога

```
<c:forEach var="item" items="${items.content}">
  <div class="col-md-4">
    <div class="card">
      
      <div class="card-body">
        <h4 class="card-title">${item.name}</h4>
```

```

        <p class="card-
text">${item.description}</p>
        <a href="/items/${item.id}" class="btn
btn-warning">Посмотреть</a>
    </div>
</div>
</div>
</c:forEach>

```

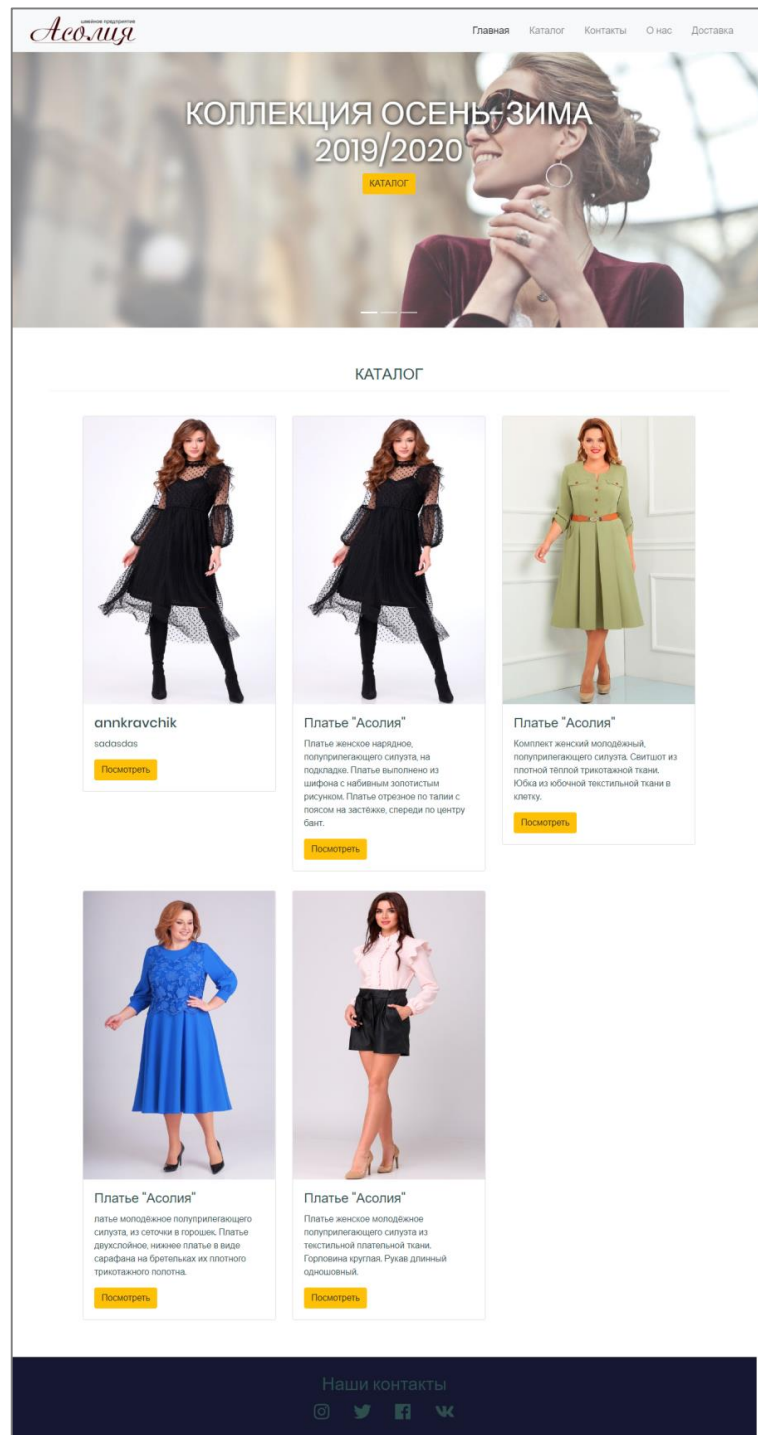


Рисунок 2.5 – Страница каталога

Основные страницы интернет-витрины представлены в приложении А.

DispatcherServlet отправляет запрос контроллерам для выполнения определённых функций. Аннотация @Controller указывает, что конкретный класс является контроллером. Аннотация @GetMapping используется для связывания с URL для всего класса или для конкретного метода обработчика. Фрагмент кода разработанного контроллера ItemController приведена в листинге 2.2.

Листинг 2.2 – Фрагмент кода контроллера ItemController

```
@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:/admin/items";
    }
}
```

Аннотация Controller определяет класс как Контроллер Spring MVC. Также, нужно отметить важные моменты в отношении контроллера:

- 1) Можно определить бизнес-логику внутри связанного таким образом служебного метода. Из него можно вызывать любые другие методы.
- 2) Основываясь на заданной бизнес-логике, в рамках этого метода создается Модель (Model). Можно добавлять атрибуты Модели, которые будут добавлены в Вид (View).

Для разрабатываемой интернет-витрины была создана база данных для хранения данных о названии, описании и коде товара.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были рассмотрены основные принципы создания веб-приложения и клиент-серверной архитектуры, такие как Spring Framework, Framework Bootstrap, Spring MVC, а также язык разметки HTML.

Кроме того, в результате выполнения курсовой работы была разработана интернет-витрина магазина женских платьев «Асолия». Интернет-витрина реализовано на языке Java с использованием Spring Framework и Framework Bootstrap. Разработка проводилась в IntelliJ Idea.

Разработанная интернет-витрина в полной мере реализует поставленную цель. Данный сайт позволяет посетителям ознакомиться со всей информацией о товаре. Сайт может быть использован для магазина платьев в качестве интернет-витрины для привлечения новых покупателей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Введение в HTML5 [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/HTML/HTML5Введение_в_HTML5 – Дата доступа: 28.10.2019.
2. Sketch [Электронный ресурс] – Режим доступа: <https://www.sketch.com/> – Дата доступа: 15.10.2019.
3. Figma [Электронный ресурс] – Режим доступа: <https://www.figma.com> – Дата доступа: 15.10.2019.
4. Photoshop [Электронный ресурс] – Режим доступа: <https://www.photoshop.com> – Дата доступа: 17.10.2019.
5. Дронов В. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов. / СПб, 2011 – 414 с.
6. Снайдер, Л. Photoshop CC 2014. Исчерпывающее руководство / Л. Снайдер – Москва: «Эксмо», 2015. – 1040 с.
7. Spring MVC – основные принципы [Электронный ресурс]. – Режим доступа: <https://habr.com/post/336816/> – Дата доступа: 03.11.2019.
8. Bootstrap в примерах [Электронный ресурс]. – Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-423-6.pdf> – Дата доступа: 05.11.2019.
9. Эффективная разработка веб-сайтов. Bootstrap. Учебное пособие [Электронный ресурс] – Режим доступа: <https://books.ifmo.ru/file/pdf/1828.pdf> – Дата доступа: 10.11.2019.
10. The Basics of User Experience Design [Электронный ресурс]. – Режим доступа: <https://tofasakademi.com/wp-content/uploads/2018/06/the-basics-of-ux-design.pdf/> – Дата доступа: 10.11.2019.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

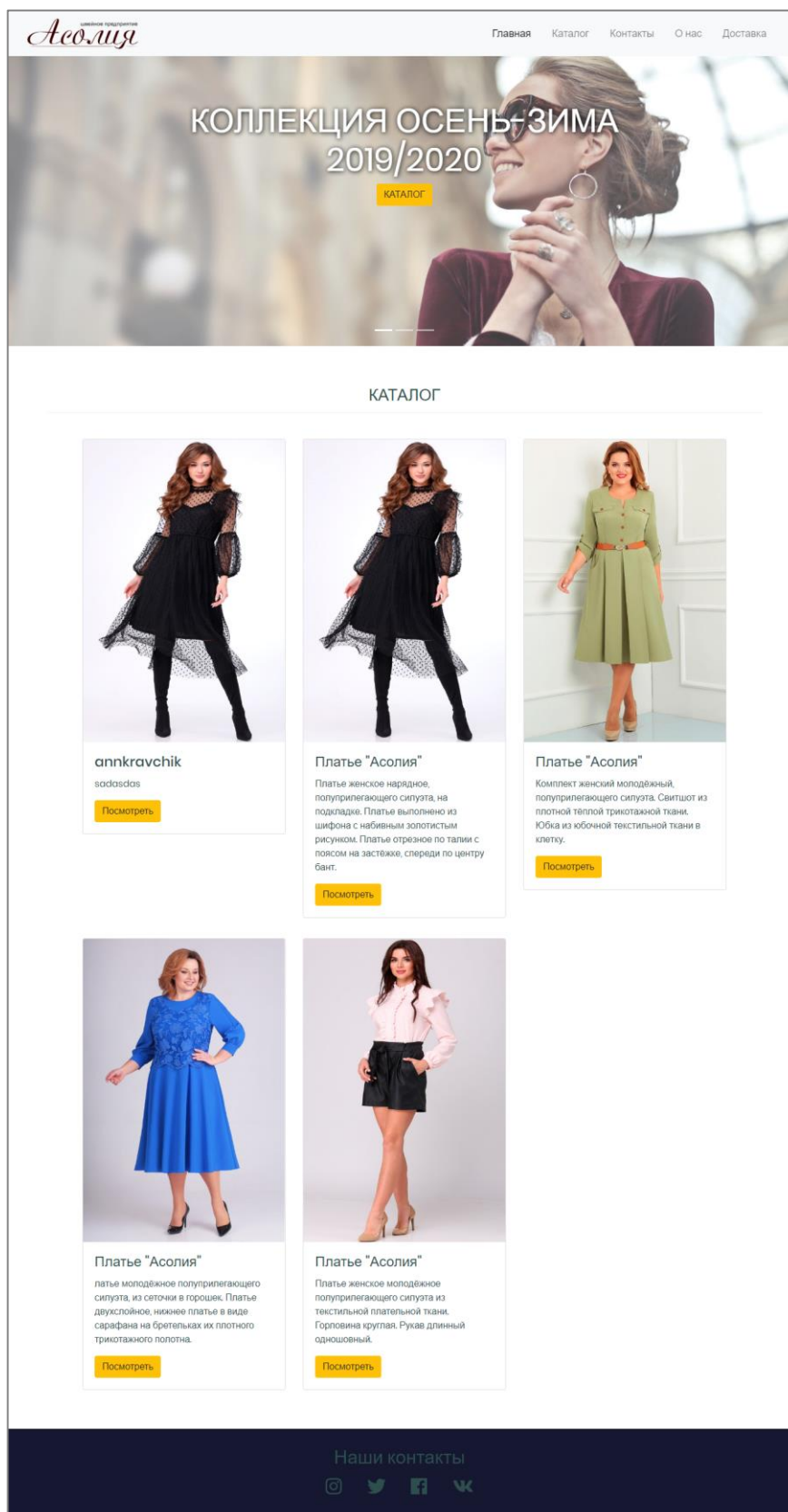


Рисунок А1 – Главная страница

ашейное предприятие

Асолия

Главная Каталог Контакты О нас Доставка

ВХОД

Логин

admin

Пароль

.....

Вход

Наши контакты

Instagram Twitter Facebook VK

Рисунок А2 – Вход

ашейное предприятие

Асолия

Главная Каталог Контакты О нас Доставка

ТАБЛИЦА ТОВАРОВ

#	Название	Описание	Удалить/ Редактировать	
6	Платье "Асолия"	Платье женское нарядное, полуприлегающего силуэта, на подкладке. Платье выполнено из шифона с набивным золотистым рисунком. Платье отрезное по талии с поясом на застёжке, спереди по центру бант.	e	x
7	Платье "Асолия"	Комплект женский молодёжный, полуприлегающего силуэта. Свитшот из плотной тёплой трикотажной ткани. Юбка из юбочной текстильной ткани в клетку.	e	x
8	Платье "Асолия"	Платье молодёжное полуприлегающего силуэта, из сеточки в горошек. Платье двухслойное, нижнее платье в виде сарафана на бретельках из плотного трикотажного полотна.	e	x
9	Платье "Асолия"	Платье женское молодёжное полуприлегающего силуэта из текстильной плательной ткани. Горловина круглая. Рукав длинный одношовный.	e	x
10	Платье "Асолия"	Платье женское полуприлегающего силуэта, расширенное к низу, из трикотажного полотна. Горловина V-образная.	e	x

Добавить новый товар

Наши контакты

Instagram Twitter Facebook VK

Рисунок А3 – Таблица товаров

The screenshot shows the 'ДОБАВИТЬ НОВЫЙ ТОВАР' (Add New Product) form. At the top left is the 'Асолья' logo with the tagline 'швейное предприятие'. The top navigation bar includes links for 'Главная', 'Каталог', 'Контакты', 'О нас', and 'Доставка'. The form itself has a title 'ДОБАВИТЬ НОВЫЙ ТОВАР' and three input fields: 'Название товара' (empty), 'Описание' (empty), and a file selection area with the text 'Выберите файл' and 'Файл не выбран'. A blue 'Добавить' button is at the bottom of the form. The footer is a dark blue bar with the text 'Наши контакты' and social media icons for Instagram, Twitter, Facebook, and VK.

Рисунок А4 – Добавление товара

The screenshot shows the 'РЕДАКТИРОВАТЬ ТОВАР' (Edit Product) form. It features the same header and navigation as the previous form. The title is 'РЕДАКТИРОВАТЬ ТОВАР'. The 'Название товара' field contains the text 'Платье'. The 'Описание' field contains the text 'Платье женское нарядное, полуприлегающего силуэта, на подкладке. Платье выполн...'. The file selection area shows 'Выберите файл' and 'Файл не выбран'. A blue 'Редактировать' button is at the bottom of the form. The footer is identical to the previous form, with 'Наши контакты' and social media icons.

Рисунок А5 – Редактирование товара