

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

**РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ СПОРТИВНОГО ПИТАНИЯ**  
**НИКА»**

Курсовой проект

Шостика Кирилл Александровича  
Студента 4 курса,  
специальности  
«Прикладная информатика»  
Научный руководитель:  
Доцент кафедры  
Информационных технологий  
Царик Сергей Всеволодович

Допущен к защите

«\_\_\_» \_\_\_\_\_ 2019 г.

Минск, 2019

## РЕФЕРАТ

Количество страниц – 19, изображений – 10, листингов – 5, источников – 5, приложений – 1.

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

Объектом исследования курсового проекта является интернет-витрина спортивного питания «НИКА». Методы исследования – архитектура клиент-серверного приложения, шаблон проектирования Model-Viewer-Controller.

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

## РЭФЕРАТ

Колькасць старонак - 19, малюнкаў - 10, лістынгаў - 5, крыніц - 5, прыкладанняў - 1.

Ключавыя словы: ІНТЭРНЭТ-вітрына, ВЭБ-СТАРОНКА, ВЭБ-САЙТ, SPRING FRAMEWORK, BOOTSTRAP, HTML, SPRING MVC.

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

У выніку выканання курсавой работы было распрацавана інтэрнэт-вітрына спартыўнага харчавання «НІКА». 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 - 16, images - 10, listings - 5, sources - 5, applications - 1.

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

The Object of study of the course work is an online showcase of sport food "NIKA". Research methods - architecture of the client-server application, design pattern Model-Viewer-Controller.

As a result of the course work, an Internet showcase of NIKA sport food 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

ОГЛАВЛЕНИЕ .....	3
ВВЕДЕНИЕ .....	4
ГЛАВА 1. ИНСТРУМЕНТЫ РАЗРАБОТКИ ВЕБ-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
ЗАКЛЮЧЕНИЕ .....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	20
ПРИЛОЖЕНИЯ.....	21

## ВВЕДЕНИЕ

В наши дни очень актуально использование интернет-витрин. Такие ресурсы позволяют получить информацию о бренде, компании, получить подробное описание ассортимента и услуг. Многие организации предпочитают создавать подобные сайты, вместо интернет-магазинов, так как это удобно по ряду причин.

Целью данного курсового проекта является разработка интернет-витрины спортивного питания «НИКА» 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.

Целью курсового проекта является закрепление опыта в разработке прототипов адаптивных страниц сайта, а также получить опыт разработки серверной части сайта.

Для достижения данной цели были поставлены следующие задачи:

- 1) проанализировать предоставленный дизайн страниц сайта;
- 2) выбрать инструментальные средства для создания адаптивных вариантов страниц сайта и разработать дизайн;
- 3) создать клиентскую часть приложения.
- 4) разработать серверную часть приложения.

# ГЛАВА 1. ИНСТРУМЕНТЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ

Разработка дизайна и прототипов играет большую роль в разработке приложения. Прототипы позволяют приложение сделать более удобным и привлекательным для пользователей.

Рассмотрим некоторые инструменты для разработки прототипов и приложений.

## 1.1 Основные инструменты прототипирования

**Adobe Photoshop.** Adobe Photoshop – многофункциональный редактор для работы с фото и видеофайлами (растровые изображения и несколько векторных инструментов). Программный продукт работает на ПК с операционными системами macOS, Windows и в мобильных версиях iOS, Windows Phone и Android. Используется графический редактор для коммерческих целей (телевидение, кинематограф, реклама, игры, ретушь и так далее) [1].

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.

Интерфейс графического редактора представлен на рисунке 1.1:

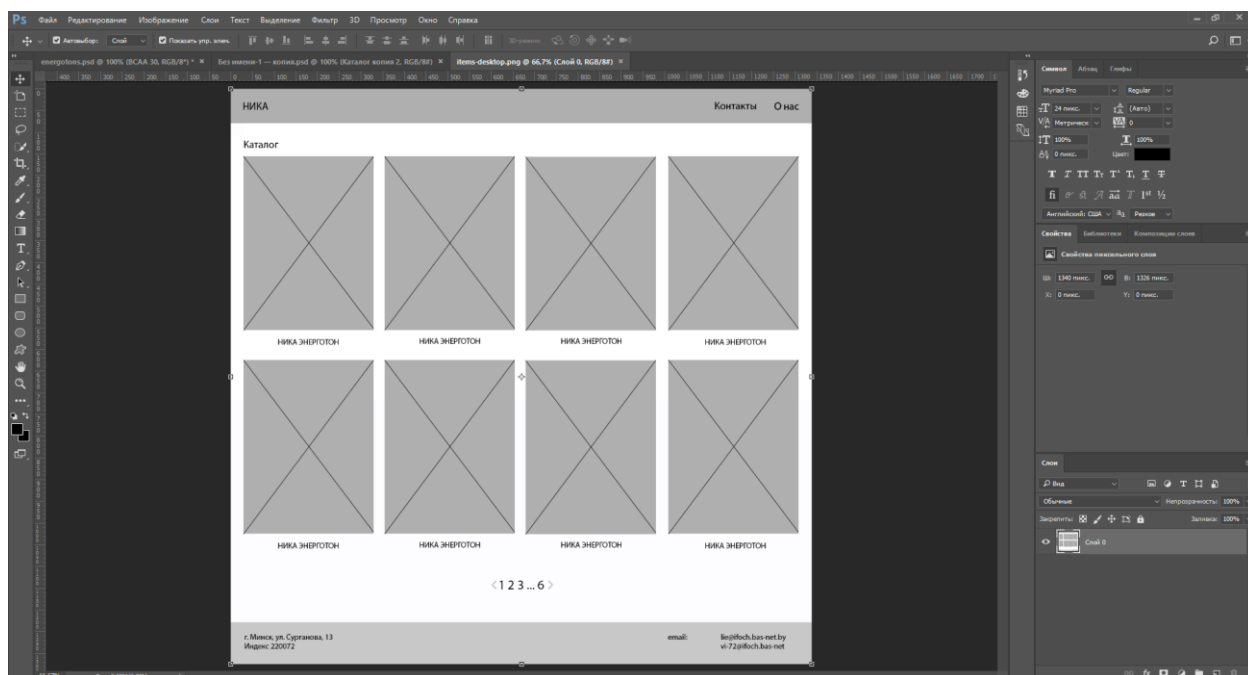


Рисунок 1.1 – Adobe Photoshop

Преимуществами Adobe Photoshop является:

- 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.

Недостатками является то, что Adobe Photoshop работает только с растровой графикой.

**Figma.** Figma – онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени. [2].

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 1.2.

Преимуществами Figma являются:

- 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.

Недостатками работы в данной программе:

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

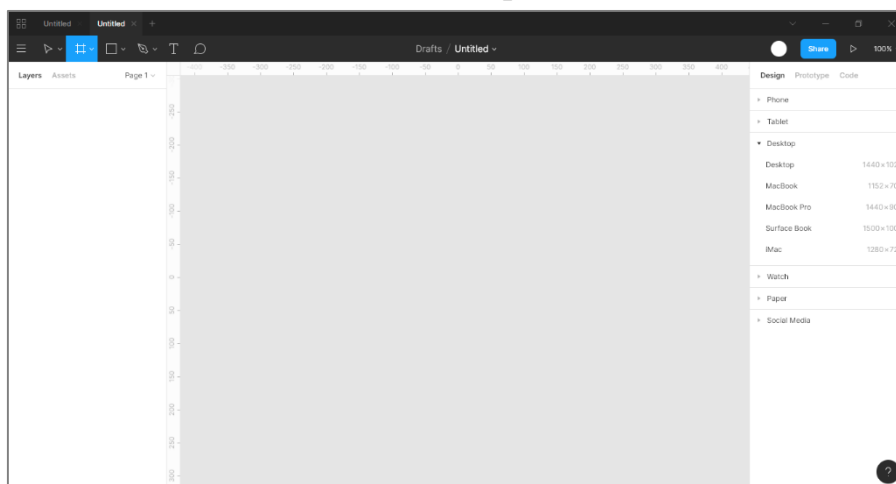


Рисунок 1.2 – Figma

## 1.2 Инструменты разработки интернет-витрины

Веб приложение – это решение, в основе которого лежит взаимодействие браузера и веб-сервера. Такие приложения являются кроссплатформенными сервисами, доступными с любого современного устройства, и не привязаны к архитектуре сети: можно получить доступ к ним с локального компьютера или со смартфона на другом конце света по удобному протоколу, например, наиболее быстрому или зашифрованному

Веб-приложение делится на две части: клиентскую и серверную. Клиентская часть отвечает за интерфейс и непосредственное взаимодействие с пользователем. Для выполнения более сложных операций клиент 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.

Такая архитектура позволяет разделить зоны ответственности между двумя подсистемами и сделать их более независимыми. Схема взаимодействия фиксируется в так называемом API, которое может быть использовано многократно не только при работе с браузером, но и, к примеру, с мобильными приложениями. Нет необходимости переписывать серверную часть под различные реализации клиента и, наоборот, при изменении внутренней логики сервера – все клиенты могут продолжать его использовать, пока соблюдается установленный API.

**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](http://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 [3].

Одним из главнейших отличий HTML5 от предыдущих версий является семанτικότητα. 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 (<span>) 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.

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.

Преимуществами HTML5 являются:

- 1) применения более простого вида кода, например, div заменены более совершенными элементами.
- 2) новая более совершенная семантика HTML5, которая дает возможность быстрее и проще различать футер, заголовки, панель навигации. Для этого используются разработанные по новому теги для упрощенного назначения основных элементов в разметки.
- 3) установку музыки, чартов, видео и рисунков можно производить без использования сторонних программ.
- 4) восприятие разметки страницы: исходный код стал простым и понятным.
- 5) используется Geolocation API, а полученные данные могут применяться в приложениях.

Фреймворк Bootstrap [4] – 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.

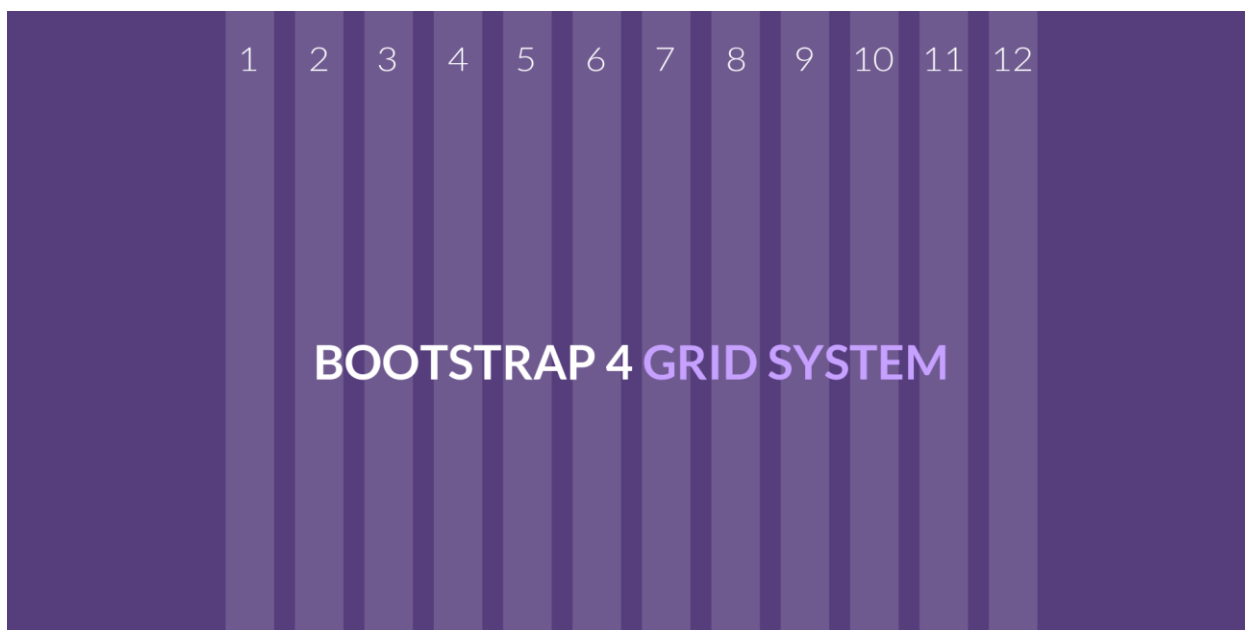
Основное отличие верстки с помощью Bootstrap от верстки нативными средствами заключается в том, что Bootstrap включает в себя использование компонентов. Компоненты представляют из себя часто используемые готовые HTML блоки с предопределенными стилями. Иногда компоненты используют JavaScript. Верстальщик 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.

Преимуществами Bootstrap являются:

- 1) возможность экономии времени в процессе разработки, так как доступно много уже готовых шаблонов;
- 2) открытый исходный код и бесплатная загрузка;
- 3) простота в использовании и гармоничный дизайн;
- 4) совместимость с Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer и Opera;
- 5) высокая скорость работы – динамичные макеты масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке.

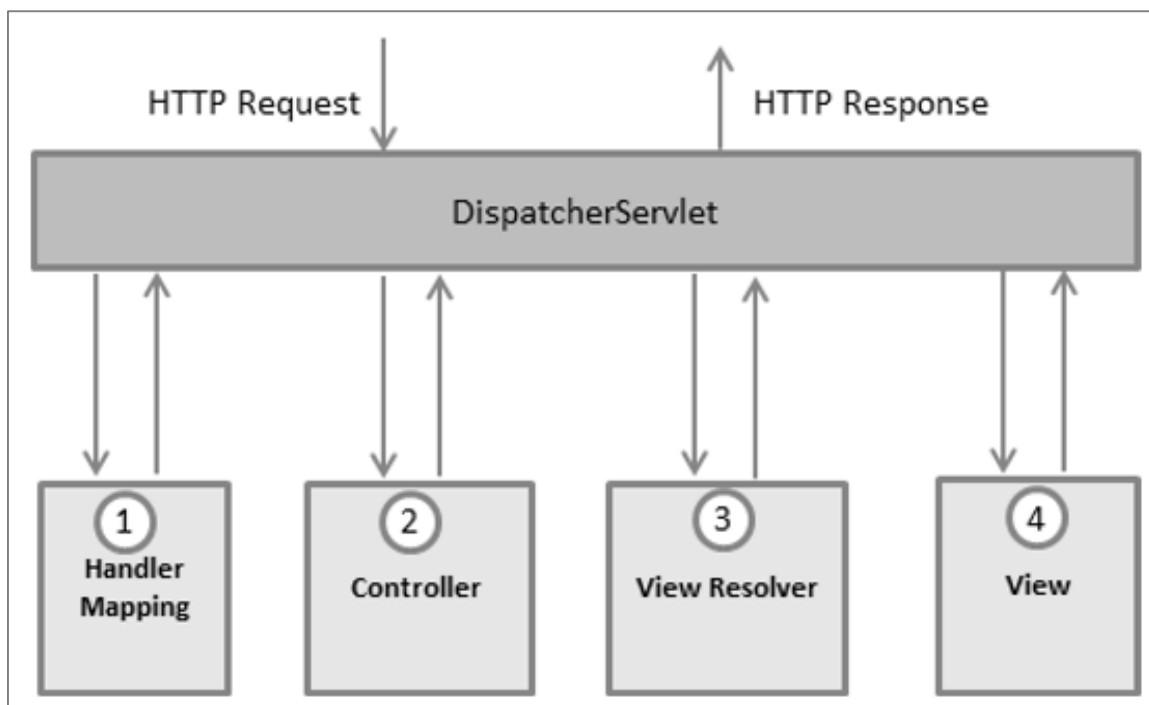
Система сеток Bootstrap 4 использует контейнеры, ряды и колонки, чтобы удобно располагать содержимое. Бутстрап реализован с помощью флексбокса и полностью адаптивен. Ниже приведен пример и глубокий взгляд на объединение сетки. Система сеток представлена на рисунке 1.3.



*Рисунок 1.3 – Система сеток Bootstrap*

**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 [5].

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 1.4 – Процесс обработки запроса DispatcherServlet'ом*

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 определённой 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.

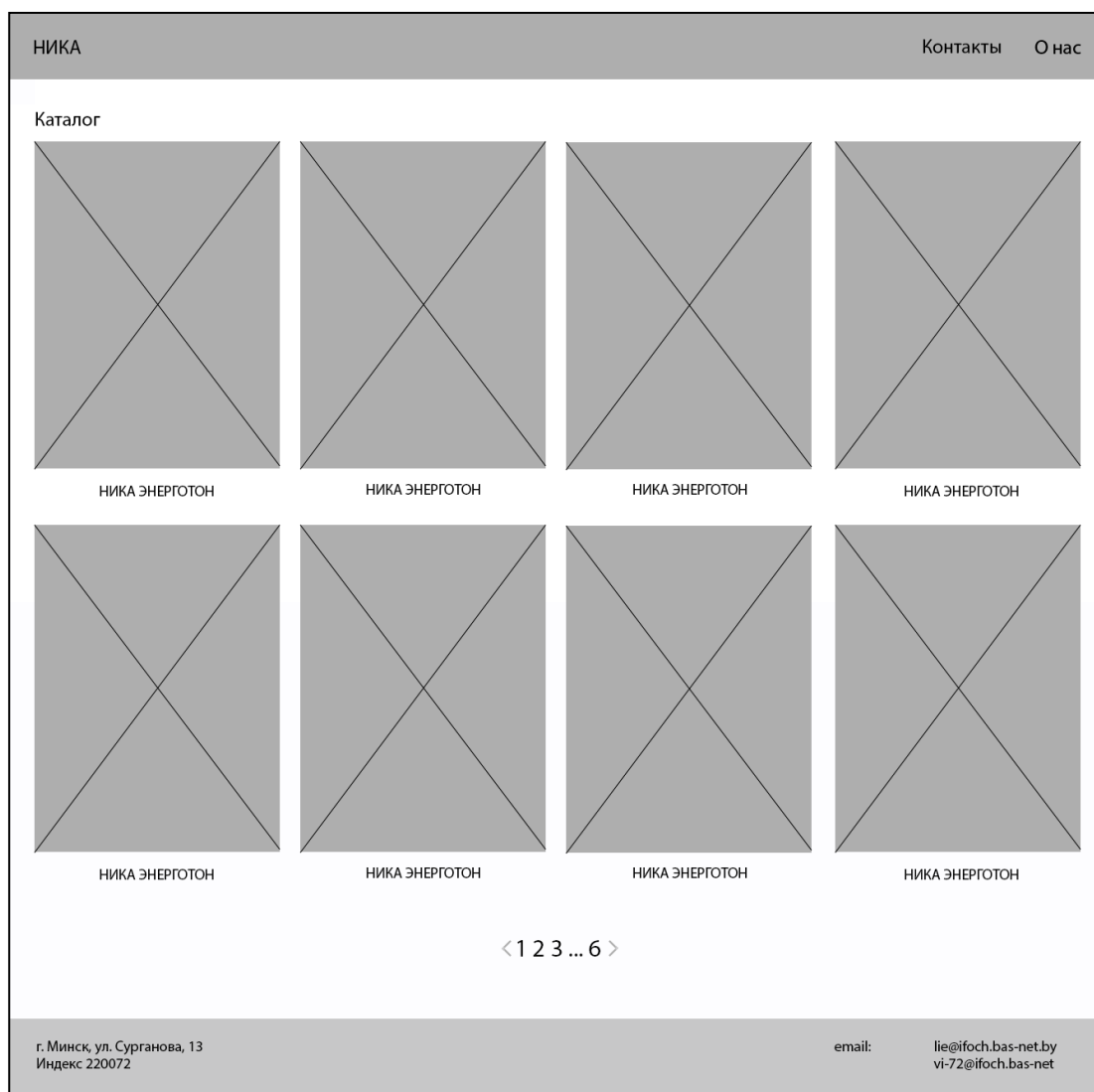
Компоненты HandlerMapping, Controller и ViewResolver, являются частями интерфейса WebApplicationContext extends plicationContext, с некоторыми дополнительными особенностями, необходимыми для создания web-приложений.

## ГЛАВА 2. РАЗРАБОТКА ИНТЕРНЕТ-ВИТРИНЫ

### 2.1 Разработка прототипов страниц

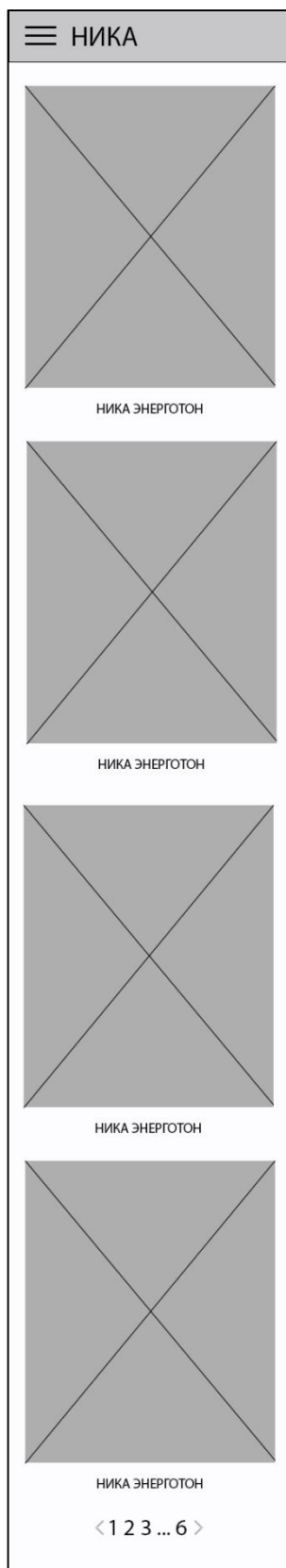
Приложение Invision было выбрано инструментом для прототипирования, т.к. оно просто и удобно в использовании. Были разработаны следующие прототипы: каталог с товарами, страничка товара, панель администратора с возможностью редактировать либо добавить товар

На странице каталога товаров расположены товары и их названия, а также шапка и подвал сайта. Прототип каталога представлен на рисунке 2.1:



*Рисунок 2.1 – Прототип каталога*

Мобильная версия страницы каталога отличается от десктопной версии тем, что шапка сайта преобразовывается в шапку-бургер, каждый товар располагается по одному в строке, а не по четыре. Мобильная версия каталога сайта представлена на рисунке 2.2:



*Рисунок 2.2 – Мобильная версия каталога*  
Все прототипы представлены в приложении А.

Для реализации проекта были определены контролеры, созданы виды, создана база данных с товарами. Проект имеет вид, представленный на рисунке 2.3

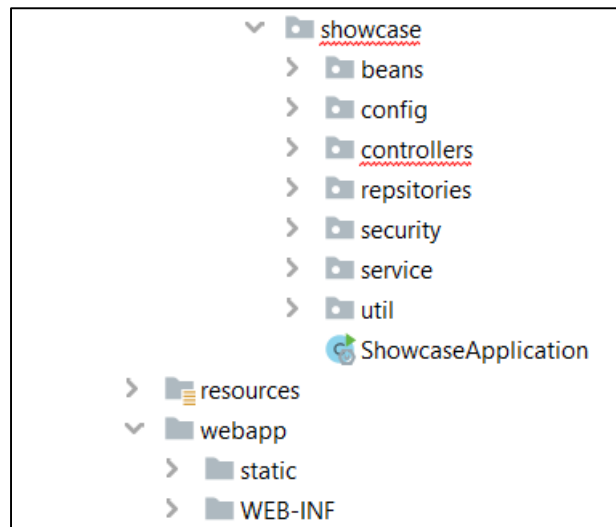


Рисунок 2.3 – Структура проекта

База данных содержит следующие поля: id (идентификатор товара), name (название товара), description (описание товара), contraindications (противопоказания), ingredients (состав), treatment (показания для применения), date (дата добавления или обновления товара). Класс создания таблицы товаров представлен в листинге 2.1

Листинг 2.1 – Класс создания таблицы

```
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 contraindications;  
  
    @Column(length = 3000)  
    private String ingredients;  
  
    @Column(length = 3000)
```

```

private String treatment;

@Column
private Date date;
}

```

Шапка и подвал сайта были реализованы как отдельные компоненты, так как они используются на нескольких страницах в одном и том же виде.

Добавление шапки и подвала сайта на страницу представлены в листинге 2.2.

*Листинг 2.2 – Добавление шапки и подвала*

```

<jsp:include page="inport_header.jsp"/>
<jsp:include page="footer.jsp"/>

```

За отображение страницы с каталогом, представленной на рисунке 2.4, 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 png. Данный процесс показан в листинге 2.3.

*Листинг 2.3 – Вывод товаров на страницу*

```

<div class="items">
    <div class="container">
        <div class="row">
            <c:forEach var="item" items="${items.content}">

                <div class="col-lg-3 col-md-6 col-sm-12">
                    <a href="/items/${item.id}">
                        <div class="card">
                            <div class="card-img">
                                
                            </div>
                            <div class="card-body">
                                <p>${item.name}</p>
                            </div>
                        </div>
                    </a>

                    <sec:authorize access="isAuthenticated()">
                        <div class="remove">

```

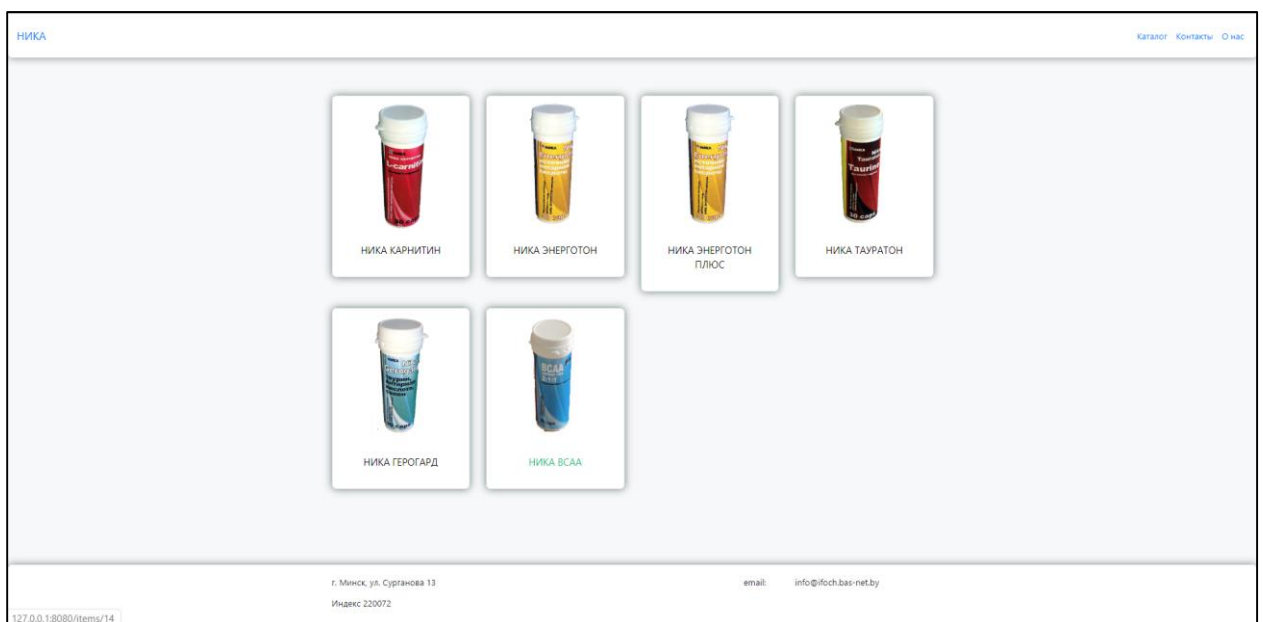
```

        <a
href="/admin/items/remove/${item.id}">remove</a>
    </div>
</sec:authorize>

</div>
</c:forEach>

</div>
</div>
</div>

```



*Рисунок 2.4 – Страница каталога товаров*

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 `LoginController` приведена в листинге 2.4.

#### *Листинг 2.4 – Контроллер LoginController*

```

@Controller
@RequestMapping("/login")
public class LoginController {

    @GetMapping
    public String toLoginPage(Authentication authentication,
Model model, HttpServletRequest request)

```



```

        if (authentication != null) {
            return "redirect:/";
        }

        if (request.getParameterMap().containsKey("error")) {
            model.addAttribute("error", true);
        }

        return "login_temp";
    }
}

```

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 обычного пользователя скрыты такие элементы управления, как добавить, редактировать и удалить товар. Это достигнуто с помощью проверки на аутентификацию администратора. Пример вывода кнопки для выхода из панели администратора представлен в листинге 2.5.

*Листинг 2.5 – Кнопка выхода из панели администратора*  
`<sec:authorize access="isAuthenticated()">`

```

        <a class="logout"
href="{pageContext.request.contextPath}/logout">
            Logout
        </a>

</sec:authorize>

```

Для доступа к панели администратора требуется авторизация. Данные о логине и пароле администратора хранятся в отдельной таблице admins. Страница авторизации представлена на рисунке 2.5.

НИКА

Каталог Контакты О нас

Логин  
admin

Пароль  
\*\*\*\*\*

Запомнить ☐

Войти

г. Минск, ул. Сурганова 13  
Индекс 220072

email: info@ifoch.bas-net.by

*Рисунок 2.5 – Страница авторизации*

После авторизации появляется возможность редактировать, добавлять и удалять товар. Страница добавления товара показана на рисунке 2.5.

НИКА

Каталог Контакты О нас Logout

Добавить

Выберите файл Файл не выбран

Название:

Описание:

Противопоказания:

Состав:

Показания к применению:

Сохранить

*Рисунок 2.6 – Страница добавления 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.

В результате выполнения курсового проекта была разработана интернет-витрина магазина женских платьев спортивного питания «НИКА». Интернет-витрина реализовано на языке Java с использованием Spring Framework и Framework Bootstrap. Разработка проводилась в IntelliJ Idea.

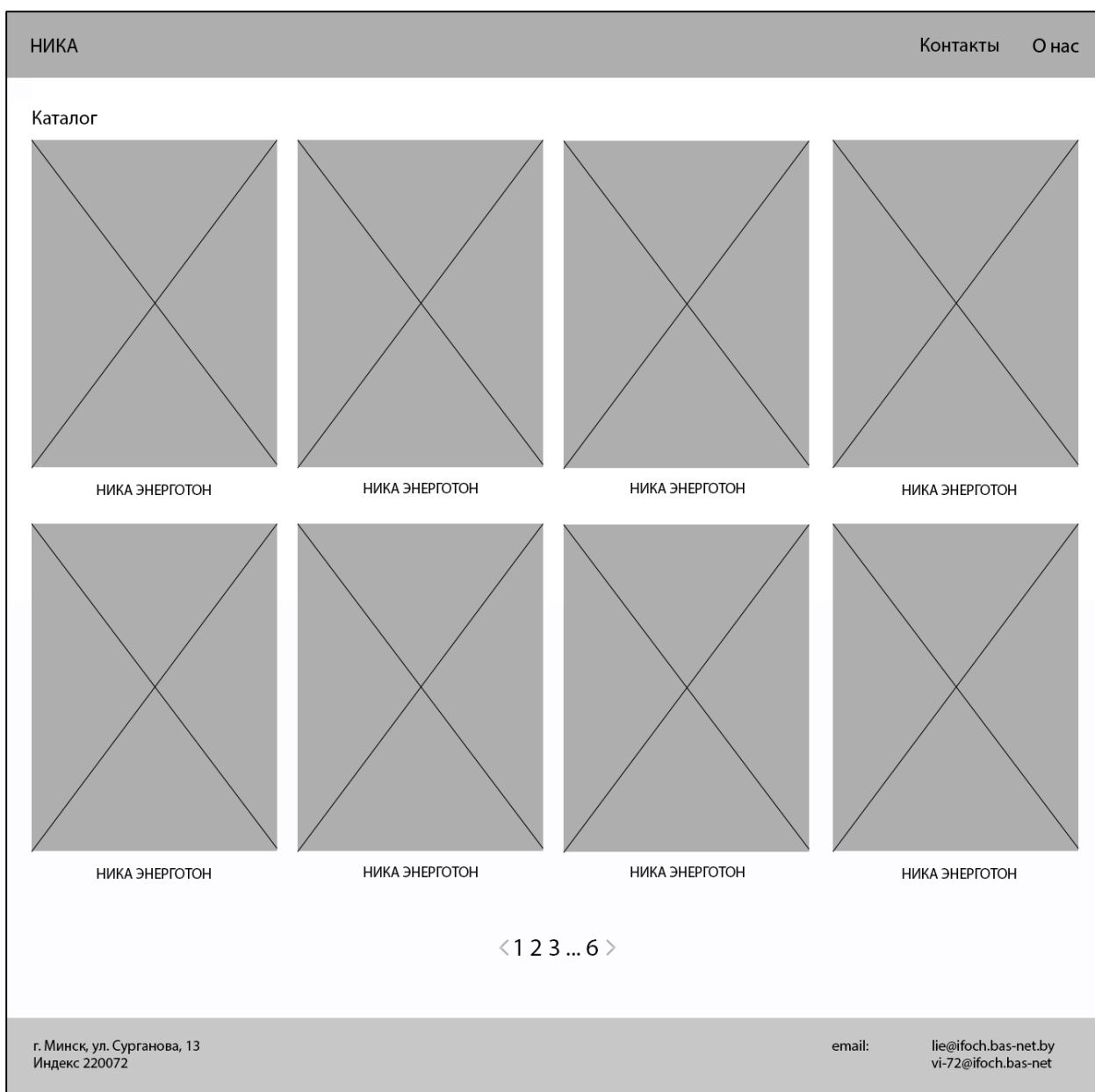
Разработанная интернет-витрина в полной мере реализует поставленную цель. Данный сайт позволяет посетителям ознакомиться со всей информацией о товаре. Сайт может быть использован для магазина спортивного питания в качестве интернет-витрины для привлечения новых покупателей.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Photoshop [Электронный ресурс] – Режим доступа: <https://www.photoshop.com>– Дата доступа: 18.12.2019.
2. Figma [Электронный ресурс] – Режим доступа: <https://www.figma.com> – Дата доступа: 18.12.2019.
3. Дронов В. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов. / СПб, 2011 – 414 с.
4. Bootstrap в примерах [Электронный ресурс]. – Режим доступа: <https://dmkpress.com/files/PDF/978-5-97060-423-6.pdf> – Дата доступа: 17.12.2019.
5. Spring MVC – основные принципы [Электронный ресурс]. – Режим доступа: <https://habr.com/post/336816/> – Дата доступа: 17.12.2019.

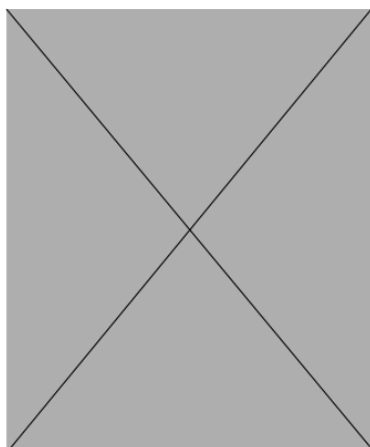
# ПРИЛОЖЕНИЯ

## ПРИЛОЖЕНИЕ А

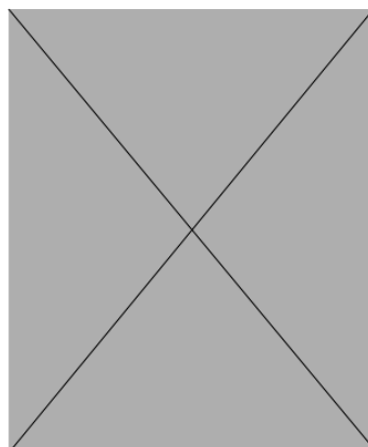


*Рисунок А1 – Прототип страницы товаров*

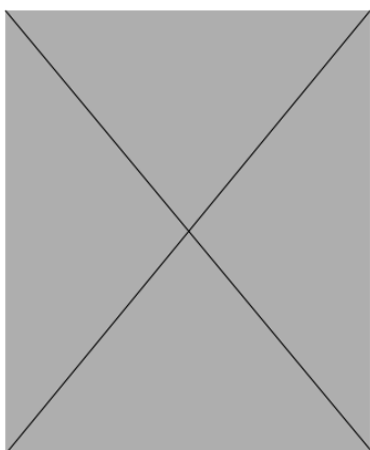
*Рисунок А1 – Прототип страницы товаров*



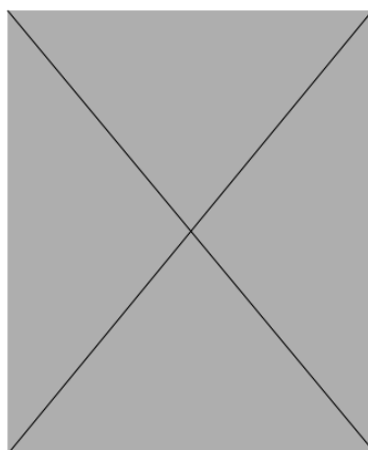
НИКА ЭНЕРГОТОН



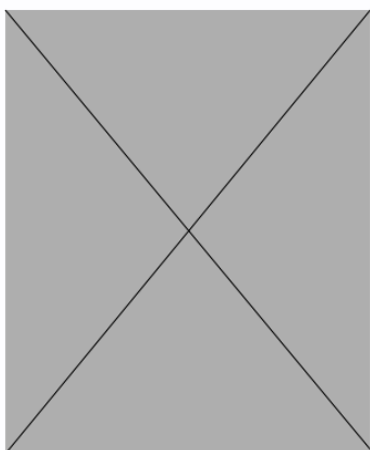
НИКА ЭНЕРГОТОН



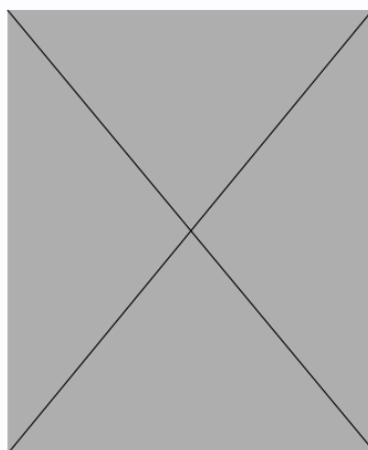
НИКА ЭНЕРГОТОН



НИКА ЭНЕРГОТОН



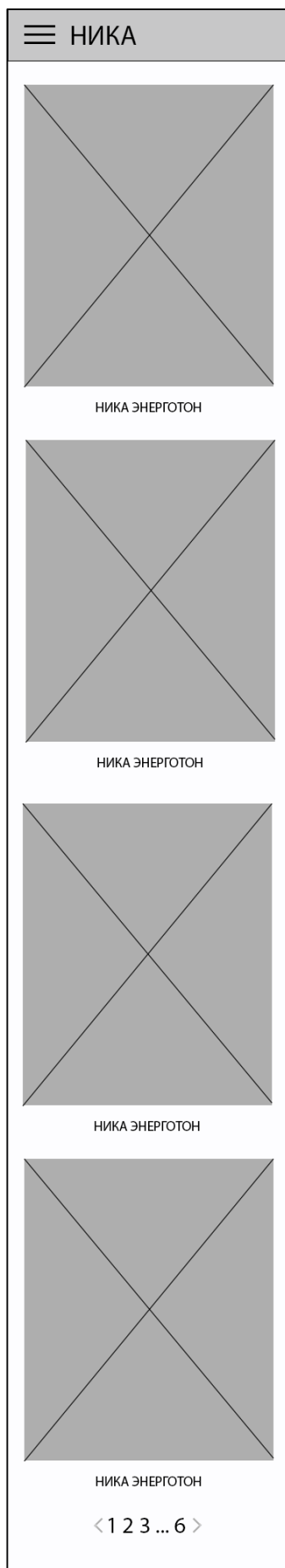
НИКА ЭНЕРГОТОН



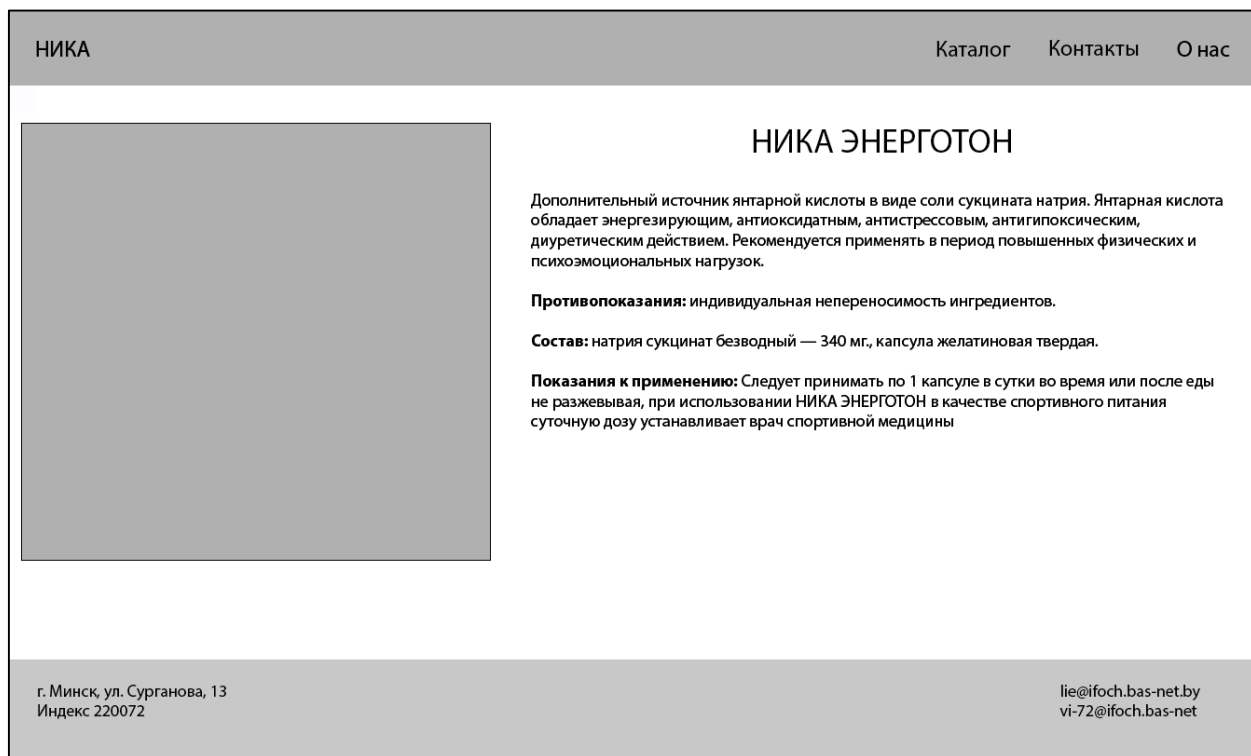
НИКА ЭНЕРГОТОН

< 1 2 3 ... 6 >

Рисунок А2 – Прототип страницы товаров для планшета



*Рисунок А3 – Прототип страницы товаров для мобильных устройств*



*Рисунок А4 – Прототип страницы товара*

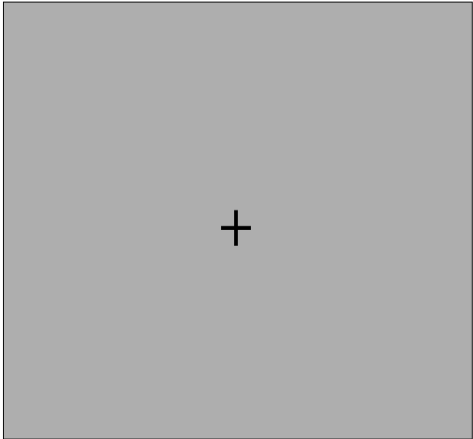


25

НИКА		Каталог	Контакты	О нас
------	--	---------	----------	-------

## Редактирование товара



Название:

НИКА ЭНЕРГОТОН

Описание:

Дополнительный источник янтарной кислоты в виде соли сукцината натрия. Янтарная кислота обладает энергизирующим, антиоксидатным, антистрессовым, антигипоксическим, диуретическим действием. Рекомендуется применять в период повышенных физических и психоэмоциональных нагрузок.

Противопоказания:

индивидуальная непереносимость ингредиентов.

Состав:

натрия сукцинат безводный — 340 мг., капсула желатиновая твердая.

Показания:

Следует принимать по 1 капсуле в сутки во время или после еды не разжевывая, при использовании НИКА ЭНЕРГОТОН в качестве спортивного питания суточную дозу устанавливает врач спортивной медицины

Сохранить

г. Минск, ул. Сурганова, 13  
Индекс 220072

lie@ifoch.bas-net.by  
vi-72@ifoch.bas-net

*Рисунок А1 – Прототип страницы редактирования товара*