Оглавление

[ВВЕДЕНИЕ 3](#_Toc73020049)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ 6](#_Toc73020050)

[1.1. Описание предметной области 6](#_Toc73020051)

[1.2. Функциональные возможности будущих пользователей 7](#_Toc73020052)

[1.3. Формирование требований к веб-сайту 10](#_Toc73020053)

[1.4. Выбор и обоснование критериев качества продукта 12](#_Toc73020054)

[1.6. Анализ аналогов и прототипов 16](#_Toc73020055)

[ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 19](#_Toc73020056)

[2.1. Программные средства, используемые для решения задач 19](#_Toc73020057)

[2.2. Информационная модель данных 20](#_Toc73020058)

[2.3. Разработка пользовательского интерфейса 22](#_Toc73020059)

[2.3.1. Создание «Главной страницы». 23](#_Toc73020060)

[2.3.2. Страница «О нас» и «Оплата» 27](#_Toc73020061)

[2.3.3. Страница «Корзина» 28](#_Toc73020062)

[2.4. Разработка административного интерфейса 30](#_Toc73020063)

[2.5. Описание функционирования веб-сайта 32](#_Toc73020064)

[2.6. Тестирование программного продукта 34](#_Toc73020065)

[ЗАКЛЮЧЕНИЕ 37](#_Toc73020066)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 38](#_Toc73020067)

[ПРИЛОЖЕНИЕ 39](#_Toc73020068)

## **ВВЕДЕНИЕ**

В современном информационном обществе каждая стабильная компания должна иметь собственный представительский сайт в сети Интернет, который обеспечит информационную поддержку существующего бизнеса. С помощью web-сайта фирмы решают такие задачи, как представление компании в сети Интернет, расширение потенциальной аудитории потребителей, поддержка бренда, повышение узнаваемости, информирование общественности.

С началом самоизоляции во время пандемии коронавируса выросла доля онлайн-покупок. Большинство российских интернет-пользователей (67 процентов) за время самоизоляции совершали покупки онлайн, а каждый четвертый (26 процентов) заказывал доставку продуктов питания на дом, показало совместное исследование Роскачества и Аналитического центра НАФИ.

По некоторым оценкам, одними из самых популярных товаров, приобретенными именно через Интернет является детская одежда и обувь.

Продажа детской одежды - актуальный вид предпринимательской деятельности, ведь спрос на товары для детей всегда высокий и не зависит от времени года. Любая семья, вне зависимости от уровня дохода, социального статуса или жизненных устоев, нуждается в одежде, которая имеет свойство изнашиваться и устаревать. Современный человек подвержен тенденциям моды, поэтому зачастую решает купить ту или иную вещь, потому что она в тренде. Этот бизнес имеет достаточно быструю окупаемость, а его рентабельность может достигать 30%.

**Покупки онлайн – это не только один из вариантов сэкономить семейных бюджет, но и возможность сэкономить свое время и нервы на поиск одежды для малыша.** Для того, чтобы совершить покупку больше не нужно часами ездить по городу, и просить кого-то посидеть с ребенком. Покупка-онлайн поможет выбрать малышу одежду, не выходя из дома, при этом занимаясь своими делами. К тому же, на большинстве сайтов предусмотрен поиск, с помощью которого можно максимально упростить себе задачу, выбрав нужный размер, цвет, фасон, и даже страну производителя. Благодаря такой возможности, значительно сокращаются временные и физические затраты.

Проанализировав существующие тенденции на современном рынке было принято решение о разработке интернет-магазина компании, которая занимается продажей детской одежды «Сказка» в городе Нижневартовске.

Магазин детской одежды «Сказка» является торговой компанией. Основными видами деятельности предприятия является организация закупки и продажи детской одежды.

Целевой аудиторией являются родители детей возраста от 0 до 12 лет, а также бабушки и дедушки. Ценовой диапазон находится в пределах границ, соответствующих среднему доходу покупателя. Расположение магазина считается удачным, поскольку находится в центре города, вблизи многих торговых центров, жилых домов и магазинов.

***Основной проблемой* данного магазина является отсутствие большого количества покупателей, низкий спрос на товары. Одним из способов привлечения новых покупателей является создание интернет-магазина, что послужило основанием для разработки программного модуля.**

*Актуальностью данного дипломного проекта* является популярность интернет-магазинов на современном рынке, а также необходимость его создания для торгового предприятия «Сказка» в городе Нижневартовск, решая вопрос о привлечении большего числа покупателей.

*Объектом данного проекта* является деятельность предприятия «Сказка».

*Предметом проекта* является процесс разработки интернет-магазина для продажи товаров детской одежды и поиск дополнительной клиентской аудитории.

*Цель данного дипломного проекта*: разработать сайт интернета-магазина для предприятия, занимающегося продажей детской одежды, который будет служить решением основной проблемы предприятия – низкий спрос.

Для достижения поставленной цели в работе определены следующие *задачи*:

1. Анализ предметной области;
2. Функциональные задачи и возможности будущих пользователей;
3. Формирование требований к интернет-магазину;
4. Анализ методов и технологий веб-сайта;
5. Обзор существующих аналогов сайтов интернет-магазинов;
6. Выбор инструментария реализации;
7. Проектирование базы данных;
8. Разработка сайта интернет-магазина и административного модуля;
9. Тестирование и отладка на сервере.

## **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ**

### **Описание предметной области**

В настоящее время присутствует ряд подходов, которые раскрывают понятие Интернет-магазина. Для рассмотрения данной области используют как технологический, так и маркетингово-коммуникационный подход.

Изначально хотелось бы отметить, что Интернет-магазин реализуют функцию витрины, которая представляет имеющийся товар или услуга на просторах Интернета. На страницах сайта обычно оформляют подробный перечень имеющихся товаров с указанием их стоимости. Компания реализует сайт, наполняет его контентом о своих услугах и товарах, ценах и гарантиях для покупателей. Далее она привлекает как можно больше посетителей для того, чтобы показать им эту информацию.

Логическая цепочка, реализуемая в Интернет-магазине, строится на основании принципа реального магазина. В частности, к требованиям реализации Интернет-магазина относят обязательно продуманный структурированный подход к каталогам товара, должна быть реализована функция, обеспечивающая поиск по ключевым понятиям, должна быть представлена возможность добавлять товар в корзину, на сайте идеальным вариантом считается возможность оплаты различными платежными системами и предоставления услуг доставки до адресата. Интернет-магазин должен ясно и прямо информировать будущих покупателей о товарах, об уникальном торговом предложении, скидках, распродажах, способах покупки и оплаты товара.

Интернет с каждым годом занимает лидирующую позицию основного источника информации для достаточно большого количества потенциальных покупателей. С каждым годом, по мере развития технологий большинство людей просматривает информацию касаемо товара, вещей, все то, что касается производителя и продавца в сети интернет. Изучив альтернативные предложения разных продавцов, потребители делают вполне осознанный выбор, который заканчивается сделкой (совершением покупки).

«Продающий» сайт не должен отвлекать посетителя от процесса выбора товара различными красивыми, полезными, но не связанными с продажами функциями – форумом, гостевой книгой, витиеватым обращением генерального директора или каким-либо другим развлекательным контентом. Для того, чтобы продать информацию людям уже не обязательно траться на бумагу, тиражи, точки продаж.

По результатам изученного материала было принято решение о разработке интернет-магазина предприятия «Сказка», занимающейся продажей детской одежды. Реализация данного проекта позволит предприятию более эффективно работать с клиентами и позиционировать свои услуги на более обширную аудиторию.

В результате разработки и внедрения интернет-магазина в нашей компании решится вопрос с большим штатом менеджеров, так как множество ответов на вопросы клиенты будет получать через ресурсы сайта компании.

### **Функциональные возможности будущих пользователей**

Для наглядного представления функциональных возможности будущих пользователей воспользуемся диаграммой вариантов использования **(Use Case**). Основные элементы диаграммы - участник (**actor**) и прецедент (**вариант**).

**Участник** - это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности. Графически участник изображается «человечком».

**Прецедент (Use Case)** - описание множества последовательных событий (включая варианты), выполняемых системой, которые приводят к наблюдаемому участником результату. Прецедент представляет поведение сущности, описывая взаимодействие между участниками и системой. Прецедент не показывает, «как» достигается некоторый результат, а только «что» именно выполняется. Прецеденты обозначаются очень простым образом - в виде эллипса, внутри которого указано его название.

Функциональные возможности неавторизованного пользователя:

* регистрация;
* авторизация;
* забронировать товара на 3 часа;
* изменения содержания корзины;
* просмотр корзины;
* просмотреть информацию о способах оплаты;
* включить версию сайта для слабовидящих;
* получить обратную связь;
* просмотр товаров на сайте;
* просмотреть информацию о физической точке магазина;
* отобрать товары по цене, цвету и размеру.

Use Case диаграмма для неавторизованного пользователя представлена на рисунке 1.

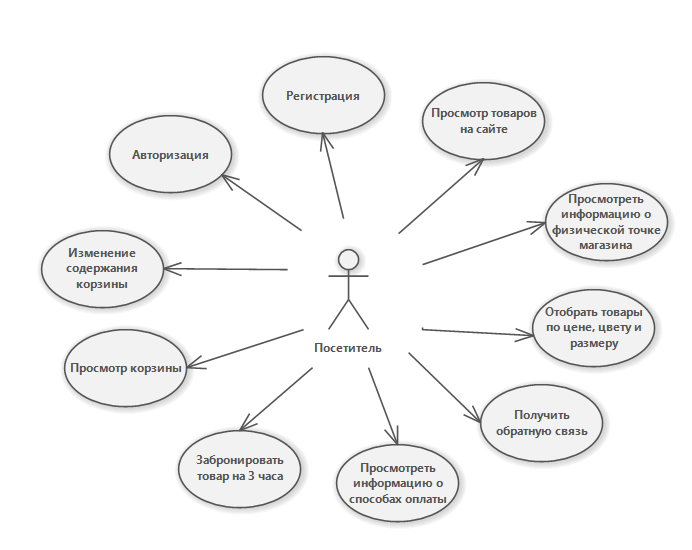


Рисунок 1. Use Case диаграмма для неавторизованного пользователя.

Функциональные возможности авторизованного пользователя:

* просмотреть история покупок;
* узнать статус заказа;
* изменить персональные данные;
* выбор способа оплаты;
* оформление заказа;
* просмотреть информацию о физической точке магазина;
* просмотр товаров на сайте;
* получить обратную связь;
* отобрать товары по цене, цвету и размеру;
* просмотреть информацию о способах оплаты;
* включить версию сайта для слабовидящих;
* авторизация;
* изменение содержания корзины;
* просмотр корзины;
* забронировать товар на 3 часа.

Use Case диаграмма для авторизованного пользователя представлена на рисунке 2.

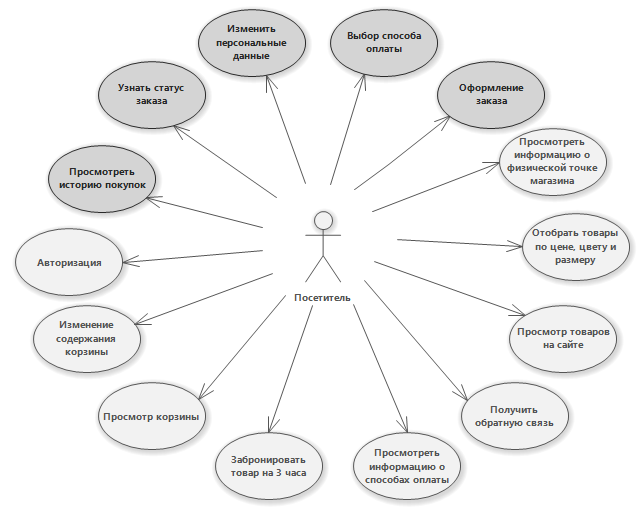


Рисунок 2. Use Case диаграмма для авторизованного пользователя

Функциональные возможности администратора интернет-магазина:

* выдать заказ;
* изменить статус заказа;
* удалить товар;
* изменить цену товара и его количество;
* добавить товар.

Use Case диаграмма для администратора интернет-магазина представлена на рисунке 3.

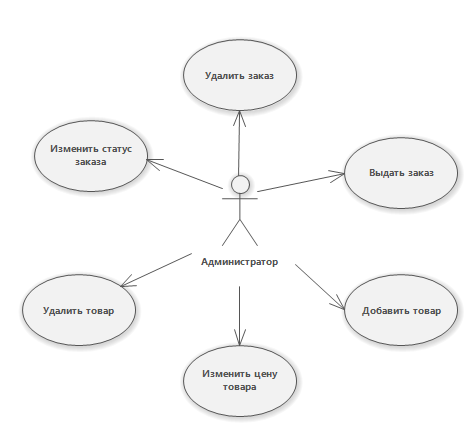


Рисунок 3. Use Case диаграмма для администратора

### **Формирование требований к веб-сайту**

На протяжении лет требования к интернет ресурсам менялись и шли к совершенству. Хоть эти требования продолжают усовершенствоваться, сейчас уже выработались определенные требования к созданию интернет-магазинов, придерживаясь которых, можно создавать наиболее эффективные Интернет-ресурсы.

Интернет-магазин должен предоставлять следующие возможности:

* регистрация/авторизация пользователя;
* онлайн-просмотр товаров магазина;
* добавление/удаление товаров в корзине;
* фильтр товаров по отдельным категориям;
* оформление заказа, выбор способа оплаты;
* администрирование сайта;
* обеспечение онлайн-чата с консультантами интернет-магазина.

1. В Интернет-магазине информация про товар должна включать:

* артикул;
* название;
* цена;
* краткое описание — оно нужно для показа небольшой информации про товар в списке категорий;
* детальное описание — если клиент заинтересовался и захотел узнать больше на товар — он нажимает на него и может прочитать больше информации;
* привязка товара к нужной категории;
* комбинации или свойства — это нужно, если вы продаете обувь или футболки, которые имеют свой размер и цвет и соответствующую цену;
* фотография товара.

1. Категории товаров используются, чтобы сгруппировать продукты по типу или по функционалу. Эта возможность позволяет клиенты быстро выбрать товар из списка.

Категории должны содержать:

* код категории;
* название.

1. О каждом пользователе должна храниться следующая информация:

* адрес электронной почты (для регистрации);
* фИО;
* номер телефона;
* история покупок.

1. Раздел заказов будет контролировать и управлять статусом сделанного заказа. В этом случае, администратор магазина, можете менять статус заказа (в ожидании, оплачен, в подготовке, готов к выдаче). Данный раздел будет доступен авторизованным пользователям.
2. Пользователи должны иметь возможность оплачивать выбранный товар с помощью различных способов оплаты. Это может быть: оплата при доставке курьером, банковский перевод на счет, оплата виртуальными деньгами.
3. Для обеспечения взаимосвязи клиентов с менеджерами или организаторами ресурсами необходимо предусмотреть форму обратной связи.
4. Так же необходимо обеспечить способность сайта менять дизайн страницы в зависимости от поведения пользователя, платформы, размера экрана и ориентации девайса. Это является неотъемлемой частью современной веб-разработки.

### **Выбор и обоснование критериев качества продукта**

Классическое определение качества заключается в том, что разработанный программный продукт подтверждает свою спецификацию – при этом спецификация должна быть ориентирована на характеристики, которые желает получить клиент.

В соответствии с таким определением общепринятые критерии качества ПО определить весьма затруднительно. Однако современные стандарты уточняют понятие качества, вводя совокупность черт и характеристик, которые влияют на его способность удовлетворять заданные потребности пользователей. Рассмотрим некоторые из них.

**Функциональная пригодност**ь - это характеристика программного средства (ПС), соответствующая набору атрибутов, определяющих назначение, номенклатуру, необходимые и достаточные функции ПС, заданные техническим заданием (ТЗ) заказчика или потенциального пользователя.

**Сопровождаемость** – это характеристики программного продукта, позволяющие минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей. Удобство (понятность, эффективность освоения, эргономичность).

**Корректность** - способность ПС обеспечивать правильные или приемлемые результаты и внешние эффекты для пользователей.

**Защищенность** – способность программного средства соответствовать критериям и требованиям защиты от предумышленных угроз безопасности ПС.

**Эффективность** (по времени и по ресурсам). Эффективность – это отношение уровня услуг, предоставляемых программным продуктом пользователю при заданных условиях, к объему используемых ресурсов.

При выборе критериев качества программного продукта я руководствовалась требованиями ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование.

* 1. **Анализ методов и технологий создания веб-сайта**

На сегодняшний день у разработчиков веб-сайтов есть богатый выбор относительно того, какой язык использовать для создания сайта. Много различных вариантов, из которых PHP, JSP, ASP и так далее.

Мною использованы:

* языки – JavaScript (совместно с библиотекой jQuery), PHP;
* язык разметки – HTML;
* язык стилей – CSS;
* технология – Ajax;
* фреймворк – Materialize CSS.

**JavaScript** – это мультипарадигменный язык программирования. Cам по себе довольно компактный, но очень гибкий, и разработчиками написано много инструментов поверх основного JavaScript языка, которые разблокируют огромное количество дополнительных функций с очень небольшим усилием. К ним относятся:

* Программные Интерфейсы приложения (API) встроенные в браузеры, обеспечивающие различные функциональные возможности, такие как динамическое создание HTML и установку CSS стилей, захват и манипуляция видеопотоком, работа с веб-камерой пользователя или генерация 3D графики и аудио сэмплов.
* Также вы можете применить к вашему HTML сторонние фреймворки и библиотеки, что позволит вам ускорить создание сайтов и приложений.

**jQuery** - это быстрая, небольшая и многофункциональная библиотека JavaScript. Он значительно упрощает такие вещи, как обход и манипуляции с HTML-документами, обработка событий, анимация и Ajax с помощью простого в использовании API, который работает во множестве браузеров. Благодаря сочетанию универсальности и расширяемости jQuery изменил способ написания JavaScript миллионами людей.

**PHP** – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Он сконструирован специально для ведения веб разработок и его код может внедряться непосредственно в HTML. PHP отличается JavaScript тем, что PHP- скрипты выполняются на сервере и генерируют HTML, который посылается клиенту. PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов.

**HTML** - это стандартный язык разметки документов во Всемирной паутине. Все веб- страницы создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузером и отображается в виде документа, удобном для человека.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без искажений воспроизводиться на оборудовании с различной технической оснащённостью.

Однако современное применение HTML очень далеко от его изначальной задачи. С течением времени, основная идея платформ независимости языка HTML была отдана в своеобразную жертву современным потребностям в мультимедийном и графическом оформлении.

В целом язык HTML предлагает следующие возможности:

• публиковать в web-документы, содержащие заголовки, текст, таблицы, списки, фотографии и т.д.;

• получать дополнительную информацию, используя гипертекстовые ссылки;

• создавать формы для передачи данных удалённым серверам;

• включать непосредственно в документ видео- и аудиоклипы, анимационные ролики и другие объекты.

Технология **Ajax** (аббревиатура от «Asynchronous Javascript And Xml») – технология обращения к серверу без перезагрузки страницы. За счёт этого уменьшается время отклика и веб-приложение по интерактивности больше напоминает десктоп. Несмотря на то, что в названии технологии присутствует буква X (от слова XML), использовать XML вовсе не обязательно. Под AJAX подразумевают любое общение с сервером без перезагрузки страницы, организованное при помощи JavaScript.

**CSS (Cascading Style Sheets)** - это язык стилей, определяющий отображение HTML-документов. Например, CSS работает с шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами. HTML может использоваться для оформления web-сайтов. Но CSS предоставляет большие возможности и более точен, и проработан. CSS, на сегодняшний день, поддерживается всеми браузерами (программами просмотра).

**Materialize CSS** – это новый фреймворк, который использует большое количество стилей, элементов, компонентов, анимации и, конечно, принципы материального дизайна от Google. Помогает создавать привлекательные, согласованные и функциональные веб-страницы и веб-приложения, придерживаясь современных принципов веб-дизайна, таких как переносимость браузера, независимость от устройства и постепенная деградация.

### **Анализ аналогов и прототипов**

На данный момент в интернете имеется множество различных персональных веб-сайтов фотографов. Все сайты имеют относительно схожий интерфейс и набор вкладок.

Перед началом разработки веб-сайта я провела анализ многих аналогичных продуктов. Но для описания анализа в дипломном проекте я взяла несколько примеров сетевых интернет-магазинов.

* Интернет-магазин «СмоллСити» (Рисунок 4).

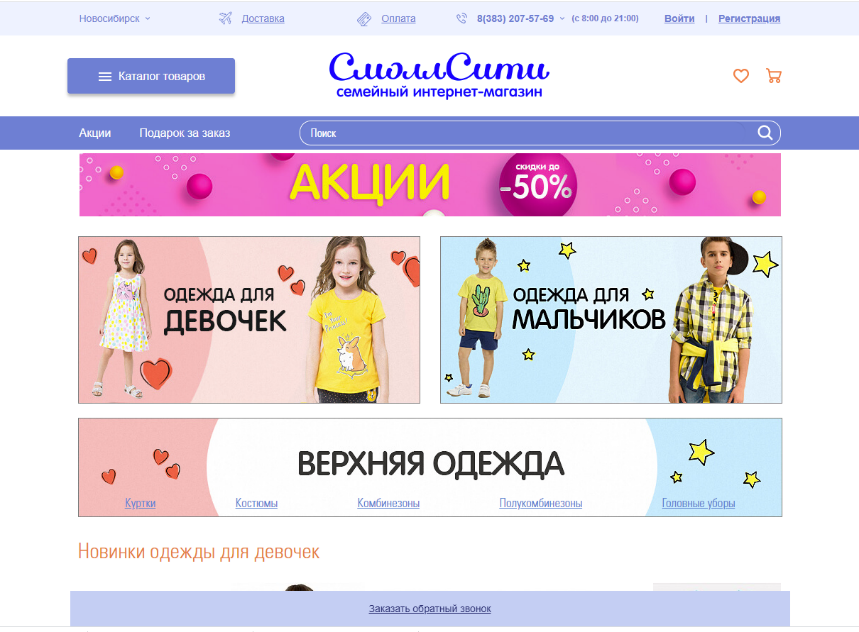
****

Рисунок 4. Интернет-магазин «СмоллСити».

Достоинства: Есть обратная связь в виде небольшой анкеты для сообщения, которая отправляется на почту компании. Имеется строка поиска, что позволяет пользователю быстро и удобно подобрать интересующий контент. Расположение информации об акциях магазина на главной странице сайта повышает заинтересованность потенциального покупателя.

Недостатки: Неудачно подобранная цветовая гамма элементов сайта, неудобное меню: тяжелое для зрительного восприятия (Рисунок 5).

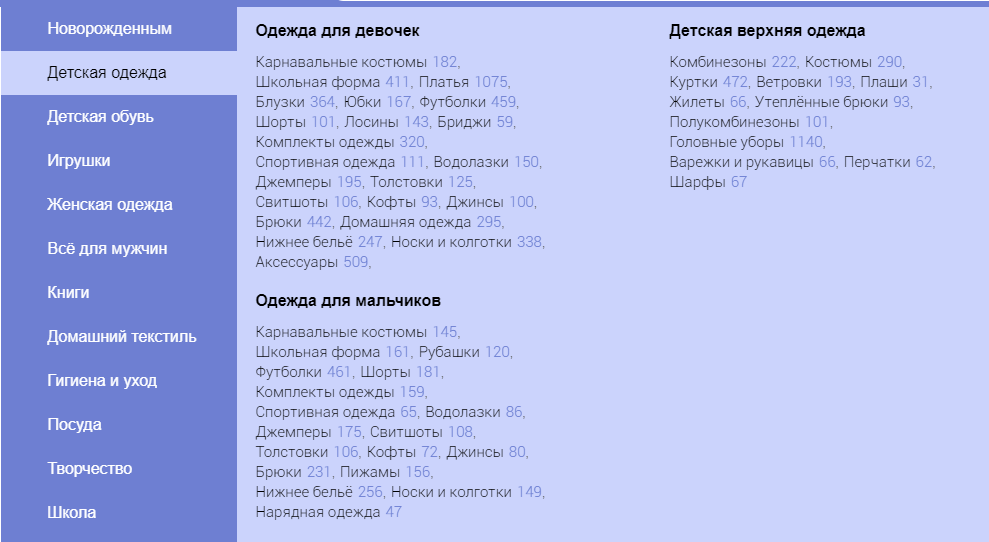


Рисунок 5. Меню сайта «СмоллСити».

* Интернет-магазин «Acoola» (Рисунок 6).

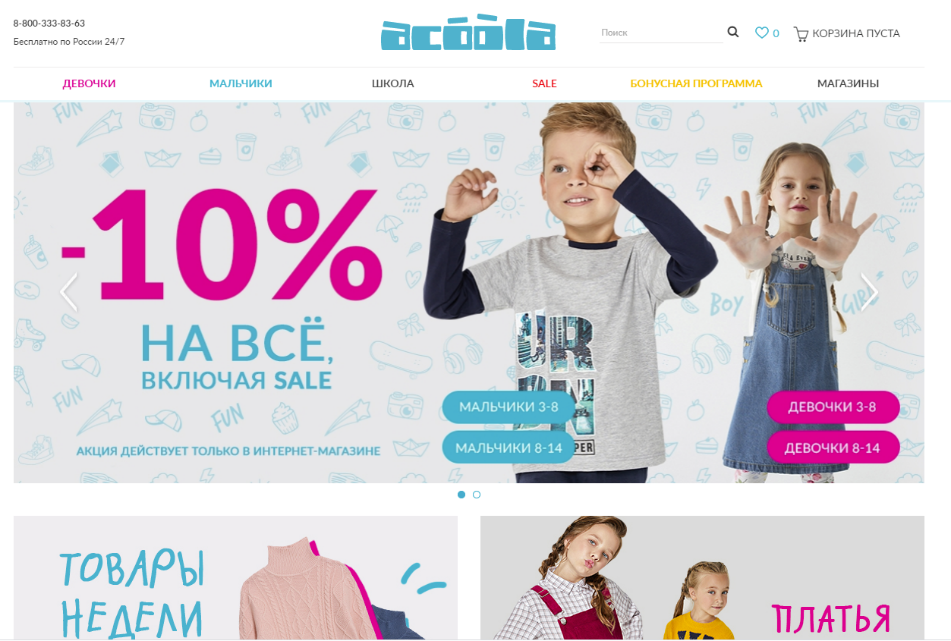


Рисунок 6. Интернет-магазин «Аcoola».

Достоинства: Дизайн сайта весьма прост, приятен для восприятия. В современных тенденциях разработки сайта приветствуется простота и понятность, отсутствие лишний элементов, отвлекающих от просмотра контента. Также имеется строка поиска, которая помогает пользователю быстрее ориентироваться на сайте.

Недостатки: На сайте имеется справочная информация, которую тяжело читать, неправильно определен размер текста (Рисунок 7).



Рисунок 7. Справочная информация на сайте.

Данные интернет-магазины представляют деятельность сетевых компании. Изучив информацию в сети Интернет, мною был сделан вывод, что интернет-магазинов не сетевых предприятий, только начавших заниматься продажей детской одежды в городе Нижневартовск, не существует, что можно также можно рассматривать как актуальность, новизну разрабатываемого проекта.

## **ГЛАВА 2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**

### **2.1. Программные средства, используемые для решения задач**

Для разработки проекта было использовано следующее программное обеспечение:

* среда разработки: Visual Studio Code;
* веб-интерфейс для администрирования СУБД: PhpMyAdmin;
* СУБД: MySQL;
* графический редактор: Figma.

**Visual Studio Code** — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для  кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

**Figma -** это графический редактор для веб-дизайна. В Figma можно создавать:

* интерактивные прототипы сайтов и мобильных приложений;
* элементы интерфейса — иконки, кнопки, меню, окна, формы обратной связи;
* векторные иллюстрации.

В Figma все документы хранятся в облаке. Благодаря этому в редакторе можно коллективно работать над макетами и открывать их по ссылке, без скачивания.

Можно заходить в Figma через браузер или же скачать программу на компьютер. Она подходит как для Windows, так и для Mac. В десктопной версии можно работать офлайн, а когда появляется доступ к интернету, изменения синхронизируются

**PhpMyAdmin** - это приложение, написанное на PHP и обеспечивающее полноценную, в том числе удаленную, работу с базами данных MySQL через браузер. Так как phpMyAdmin позволяет во многих случаях обойтись без непосредственного ввода команд SQL, то работа с базами данных становится вполне посильной задачей даже для человека весьма поверхностно знакомого с MySQL.

На сегодняшний день PhpMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения СУБД MySQL. Подавляющее большинство российских провайдеров используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

**MySQL** – это одна из самых популярных и самых распространенных СУБД в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет-сайтов, как небольших, так и достаточно крупных. MySQL отличается хорошей скоростью работы, надежность, гибкостью. Работа с ней, как правило, не вызывает больших трудностей.

Активное использование MySQL в веб-программировании обусловило его актуальность, а интуитивно понятный интерфейс в совокупности с широкой функциональностью и поддержкой более 60 языков (в т.ч. и русского) обеспечило ему популярность среди веб-разработчиков.

### **2.2. Информационная модель данных**

На сайте имеется разграничение прав доступа. Пользователь ограничивается просмотром страниц сайта. Администратор, в свою очередь, полностью управляет контентом сайта с помощью административной панели.

Для четкого представления структуры создаваемого сайта была спроектирована структура разделов сайта (Рисунок 8).

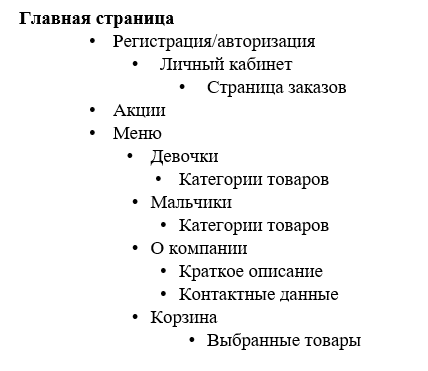


Рисунок 8. Структура разделов сайта.

Регистрация/авторизация. Для регистрации/авторизации пользователя необходимо указать e-mail, телефон, создать/ввести пароль. На указанную почту придет код подтверждения. Покупатель может отплатить покупку только при наличии регистрации на сайте.

Личный кабинет. В личном кабинете храниться информация о:

* ФИО;
* e-mail;
* телефон;
* текущих заказах и их статусах.

Акции. На главной странице в виде информационного блока должны быть продемонстрированы актуальные акции магазина, информация о скидках.

Меню. В компьютерной версии меню должно быть горизонтальным, в мобильной – вертикальным. При наведении на раздел «Девочки» или «Мальчики» должно появляться меню с популярными категориями товаров.

При нажатии на выбранную категорию товара должен открываться каталог с товарами - список товаров, представленный в линейку по 4 шт. Предусмотреть фильтр товара по цвету, размеру, категории товара.

Формат:

* картинка;
* название;
* описание имеющихся размеров;
* цена;
* кнопка «Просмотр».

По клику на название или картинку переход на страницу с подробным описанием товара. На странице приводится полное описание товара:

* название;
* картинка или несколько (не более 3);
* описание имеющихся размеров;
* кнопка «В корзину»;

О компании. Страница с описанием истории магазина, условиях работы, оплаты, контактные данные предприятия: номер телефона, электронный адрес, адрес магазина.

Корзина. Страница с описанием выбранных пользователем товаров, подсчитанной суммой заказа, а также блоком оформления заказа. На данной странице также будет предусмотрена возможность очистки корзины. В Интернет-магазине будет реализована система бронирования, поскольку важно для небольших предприятий важно отслеживать каждую единицу товара. Продукт будет находиться в пользовательской корзине 3 часа, после чего корзина очистится и товар вернется обратно.

### **2.3. Разработка пользовательского интерфейса**

Все страницы сайта выполнены в едином дизайне и с использованием адаптивной верстки для использования на разных устройствах, каждая страница сайта содержит все основные элементы меню: «Войти/Регистрация» (при регистрации или авторизации пользователя появляется раздел «Личный кабинет», логотип с переадресацией на главную страницу интернет-магазина, «Девочки» и «Мальчики» с раскрывающимися блоками популярных категорий товаров, «О нас», «Оплата», «Корзина» с подсчетом выбранных товаров.

### **2.3.1. Создание «Главной страницы».**

Так как сайт имеет многостраничную структуру, то при активации кнопки элемента меню, сайт будет автоматически открывать соответствующую страницу по выбранному раздела. При открытии сайта пользователь видит приветствующую видеовставку с лозунгом компании и кнопкой «Оформить заказ» (Рисунок 9). Такое решение придает сайту динамичность, мотивирует пользователя сделать заказ.

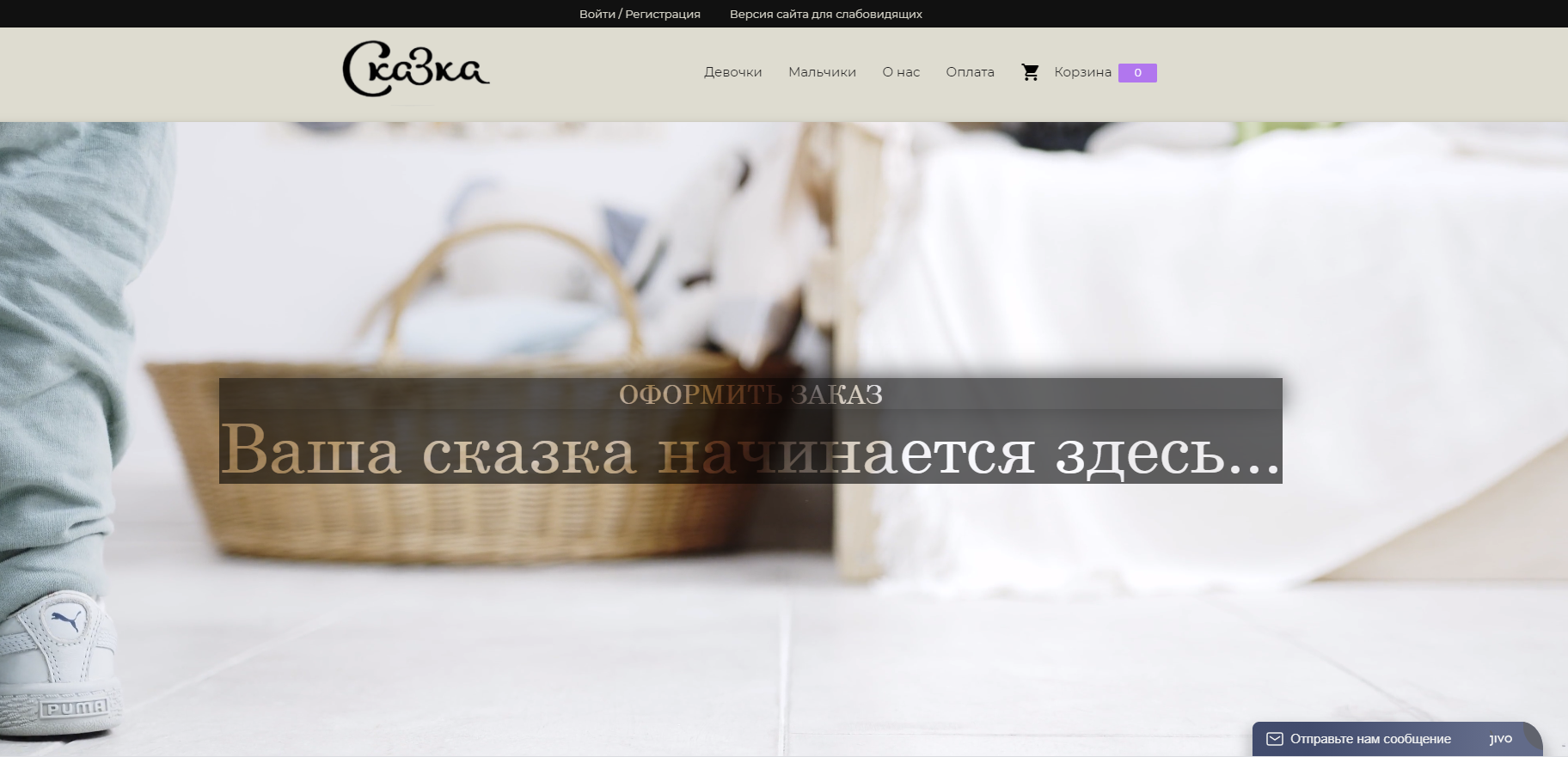


Рисунок 9. Главная страница сайта.

В верхней части главной страницы расположено меню, а также ссылки на формы авторизации и регистрации («Войти / Регистрация»), а также ссылка, по которой можно перейти на версию для слабовидящих. В данной версии сайта предусмотрено изменение размера шрифта текста, цвета сайта, работа с изображениями, а также озвучка происходящих действий. Так, например, на рисунке 10 продемонстрированы изменения главной страницы после перехода на выбранную версию сайта.

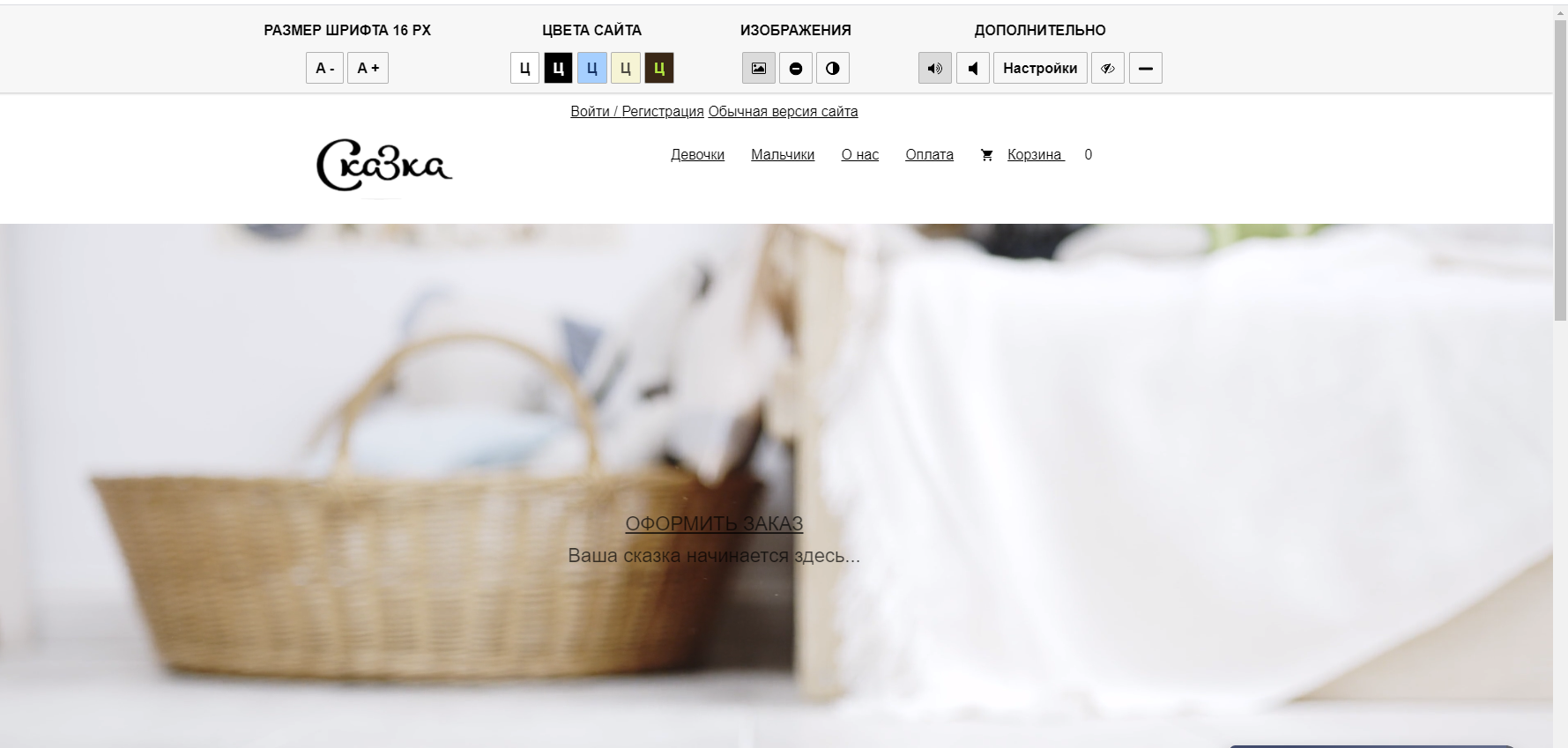


Рисунок 10. Изменения главной страницы.

В нижней части главной странице расположен информационный блок с кратким описание компании, который поможет сформировать представление о деятельности интернет-магазина, а также слайдер с повествованием о новых коллекциях детской одежды. (Рисунок 11).

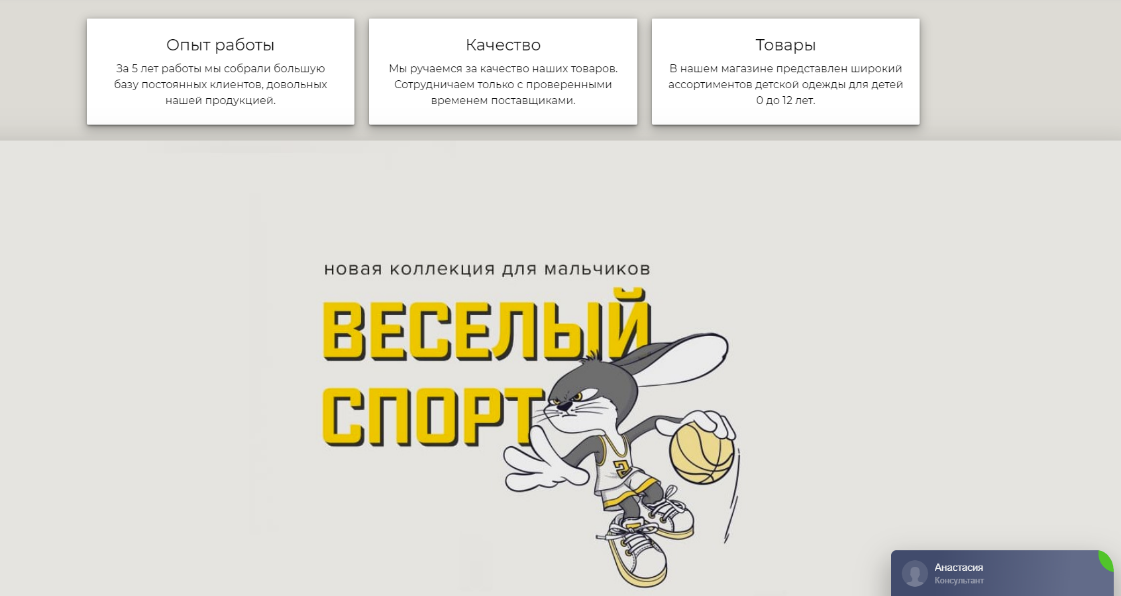


Рисунок 11. Информационный блок главной страницы.

В правом нижнем углу распложена форма взаимодействия с онлайн-консультантом интернет-магазина в случае, если у пользователя будут какие-либо вопросы (Рисунок 12). Данная форма подключена, используется популярный бизнес-менеджер Jivo.

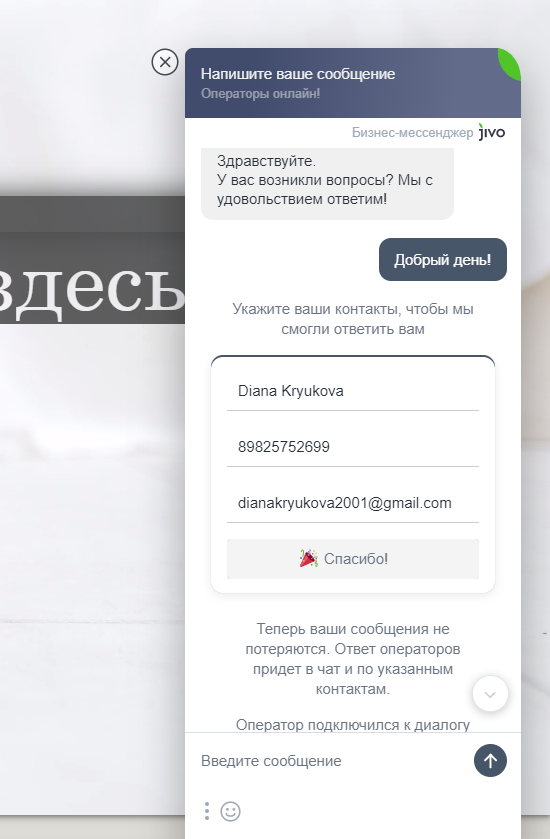


Рисунок 12. Форма взаимодействия с онлайн-консультантом.

**2.3.2. Создание страницы товаров**

Страница товар представляет собой таблицу с описанием ассортимента товаров интернет-магазина. Каждая информационная ячейка содержит изображение товара, его цену, имеющиеся в наличии размеры, а также кнопку «Просмотр» для перехода к странице с описанием товара. Слева от товаров расположен отбор по категории, цвету и размеру, что поможет пользователю сократить время в поисках необходимого продукта. В левом верхнем углу распложена система навигации, позволяющая просто переместиться к необходимому блоку. (Рисунок 13).

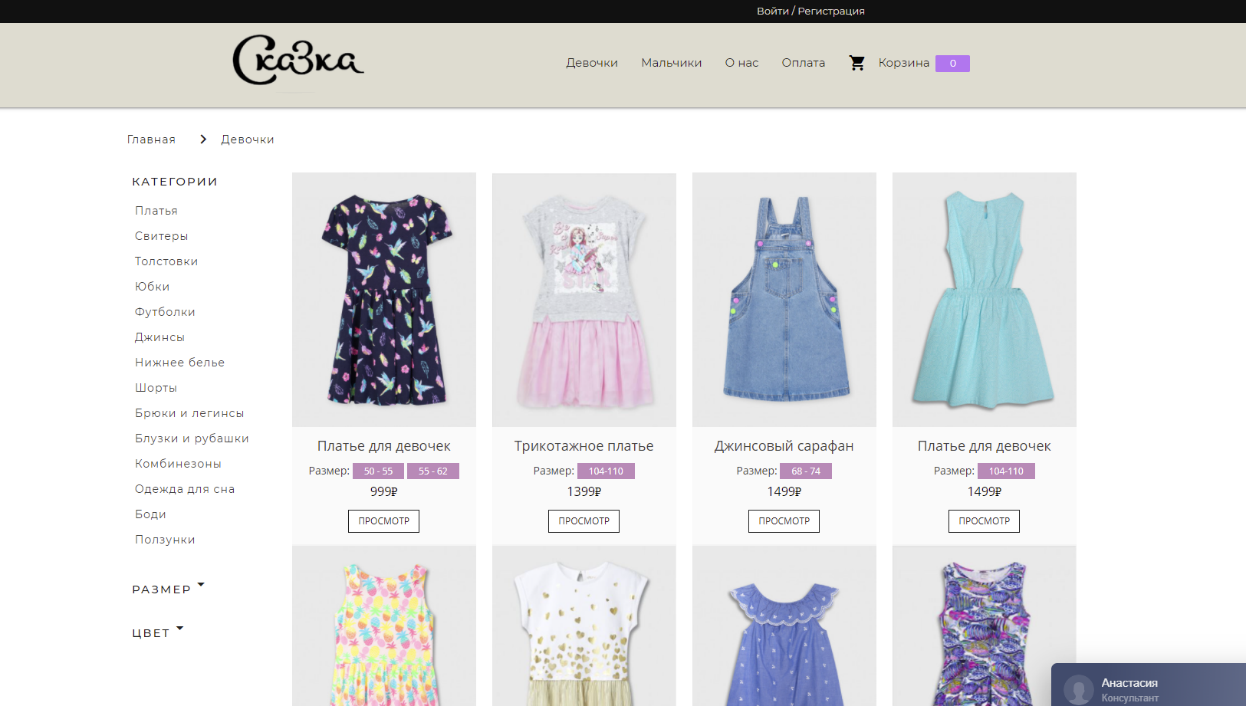


Рисунок 13. Страница товаров.

При нажатии на выбранный товар, открывается страница с его описанием, где расположена кнопка «В корзину» (Рисунок 14). После выбора необходимого размера, пользователь активирует кнопку, тем самым добавляя товар в свою корзину. Важно отметить, что магазин «Сказка» является молодым, развивающимся предприятием, поэтому количество каждого товара может не превышать даже и 10-ти единиц. В следствии чего было принято решение отслеживать движение каждой единицы продукции, путем введения ограничения на 3-х часов бронирование товара.

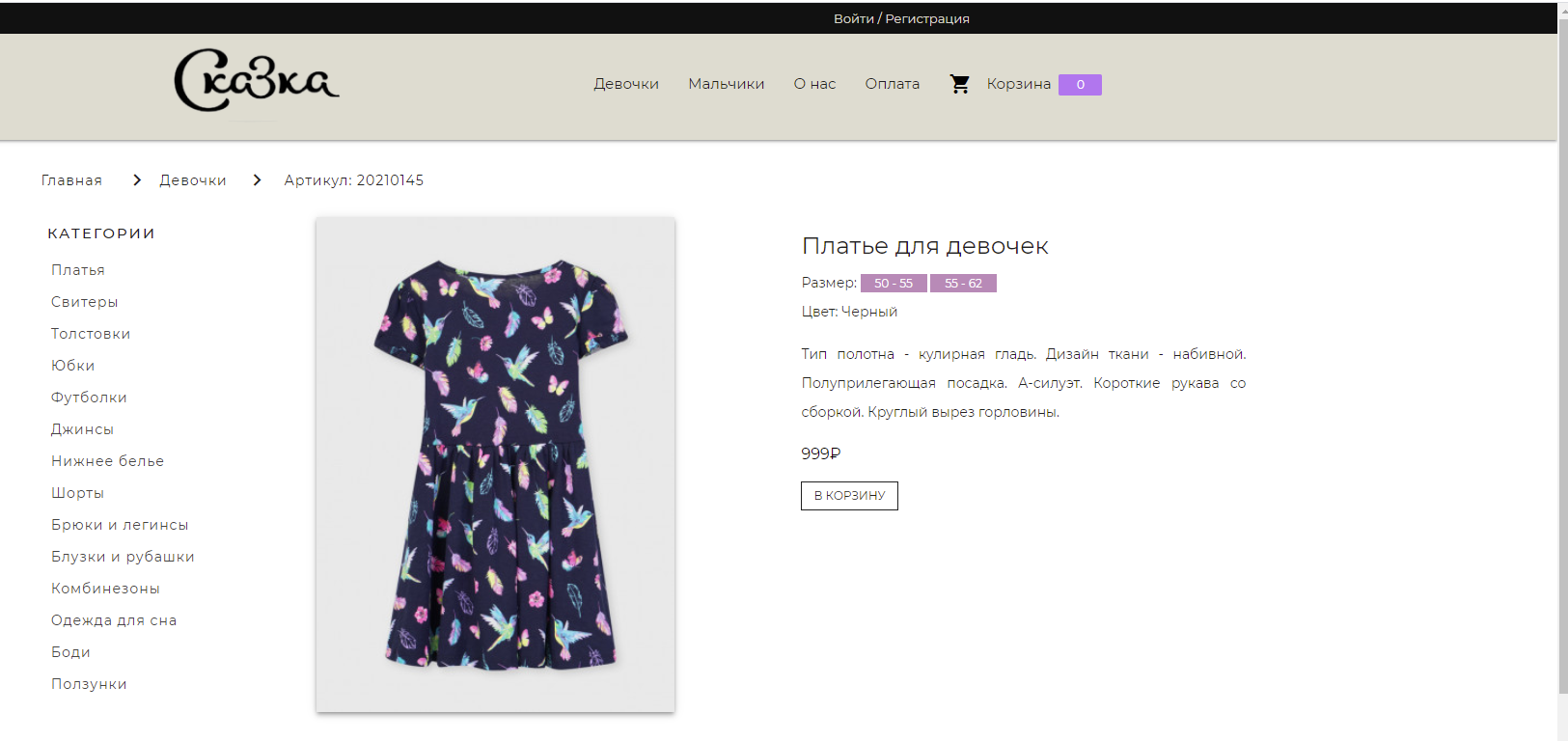


Рисунок 14. Страница с описанием товара.

### **2.3.2. Страница «О нас» и «Оплата»**

Раздел «О нас» представляет собой страницу с описанием деятельности предприятия, его контактными данными, месторасположением физической точки магазина. Такая страница информирует пользователя о способах связи с представителями интернет-магазина (Рисунок 15). Страница «Оплата» описывает способы оплаты заказа в данном интернет-магазине (Рисунок 16).

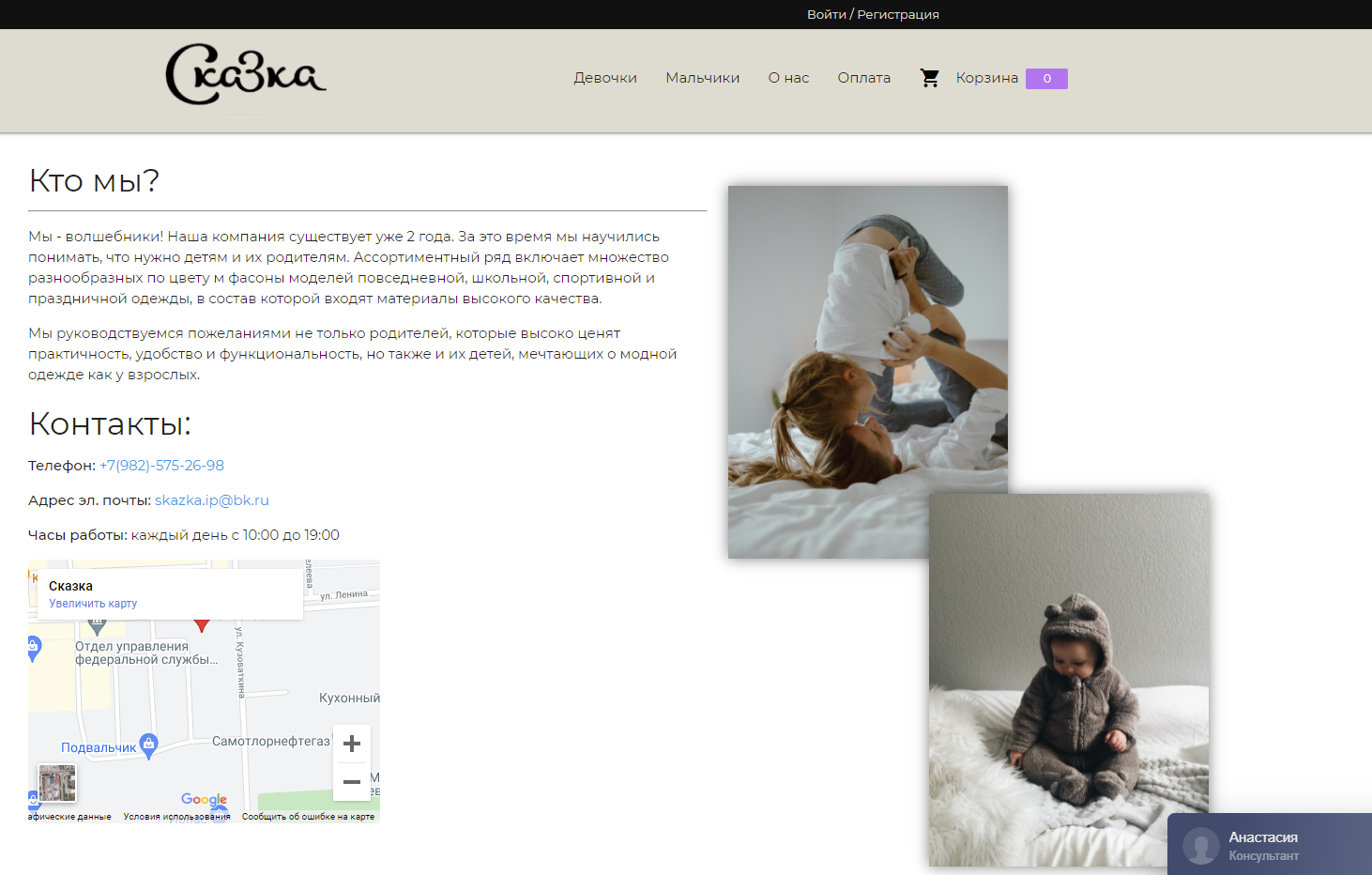


Рисунок 15. Страница с контактными данными организации.

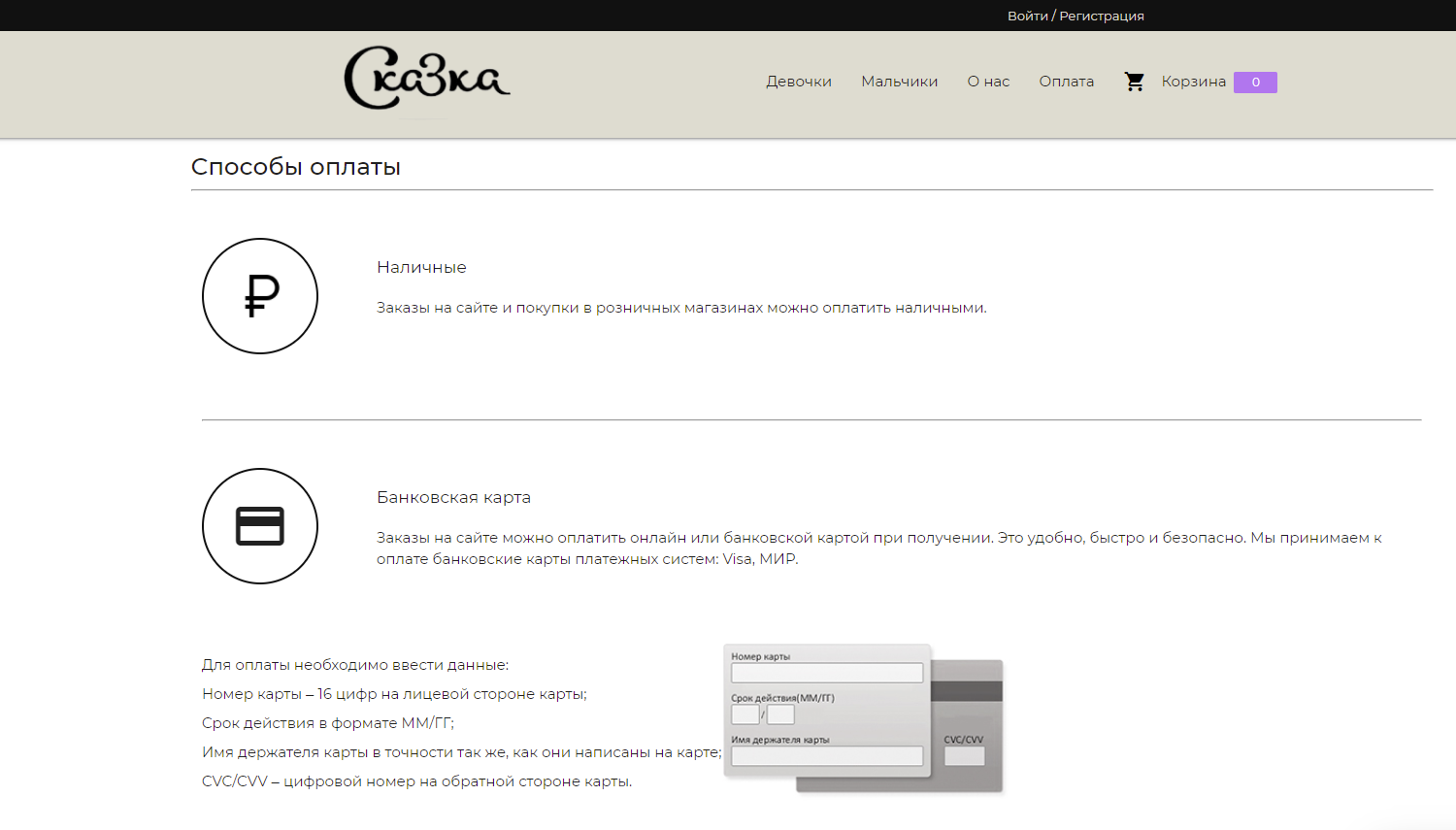


Рисунок 16. Страница «Способы оплаты».

### **2.3.3. Страница «Корзина»**

Страница «Корзина» от части является главным элементом интернет-магазина, поскольку именно там происходит основное действие – оформление заказа и его оплата. Содержание страницы авторизованного пользователя формируется из последней сформированной им корзины. Такая корзина может меняться, после выхода из системы товары сохранятся, но после истечения 3 часов будут также удалены.

После перехода в раздел «Корзина» пользователь увидит список выбранных товаров с подсчетом итоговой суммы заказа, а также напоминание о сроках истечения бронирования товаров. (Рисунок 17) Если пользователь авторизирован, ему будет доступно оформления заказа и его оплата, иначе система не даст этого сделать. Оформление заказа состоит из этапов: проверки контактных данных, выбора способа оплаты и непосредственно самой оплаты (Рисунок 18).

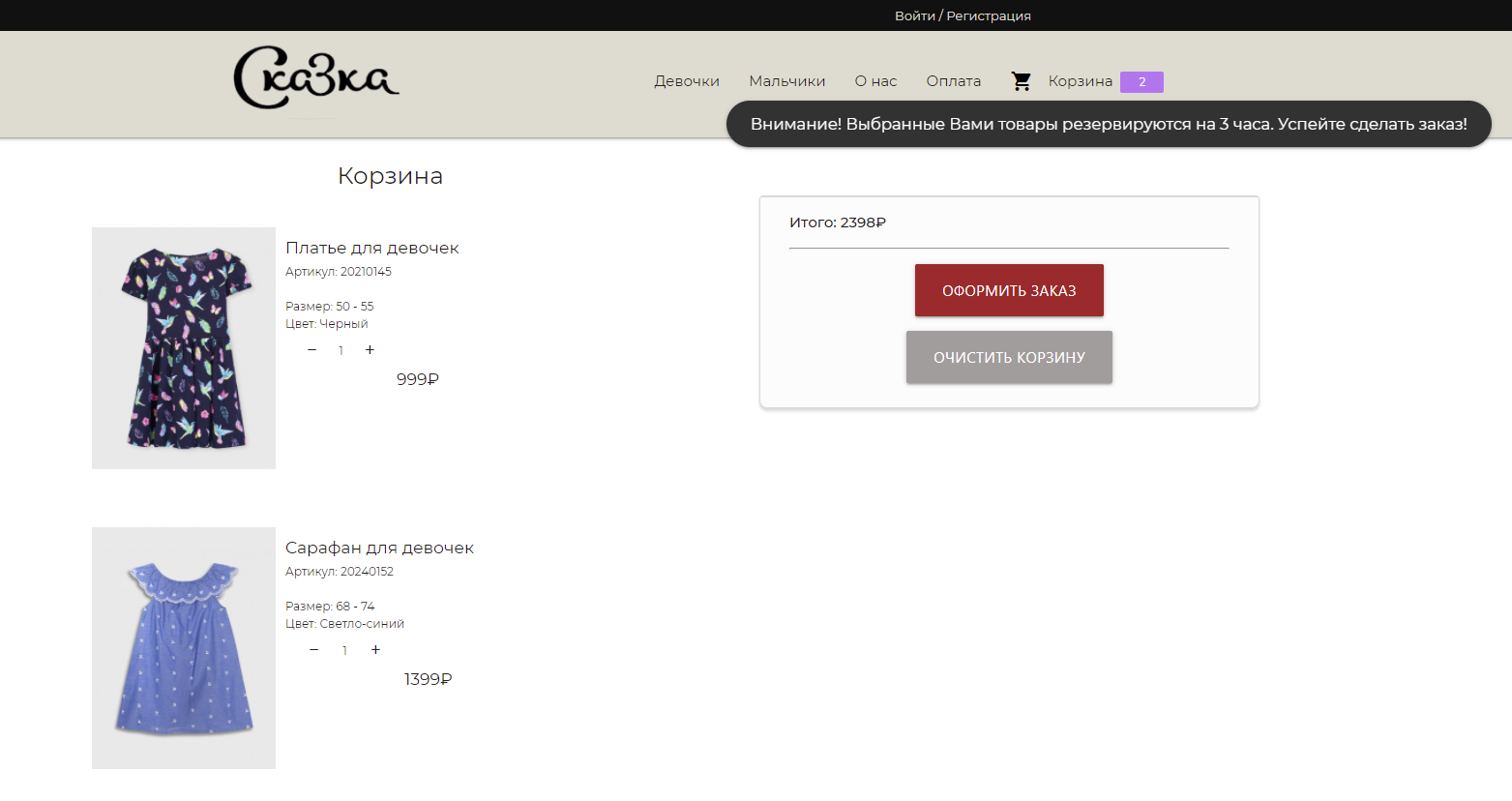


Рисунок 17. Содержание корзины.

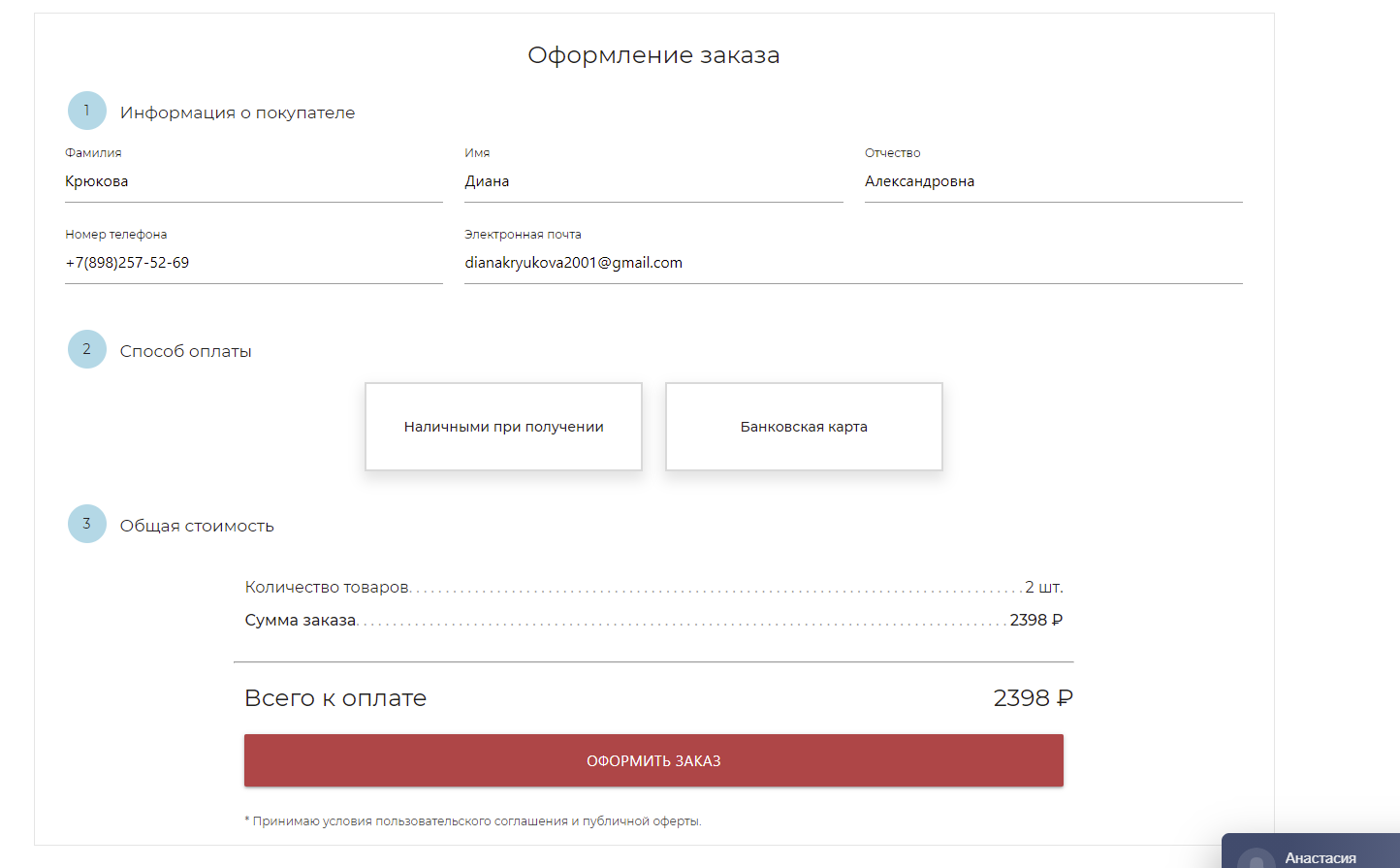


Рисунок 18. Блок оформления заказа.

После успешного оформления заказа в личном кабинете пользователя будет отображен номер текущего заказа, а также его статус, который будет меняться администратором интернет-магазина (Рисунок 19). Также в личном кабинете пользователь может редактировать персональные данные, просмотреть историю покупок (предыдущие заказы).

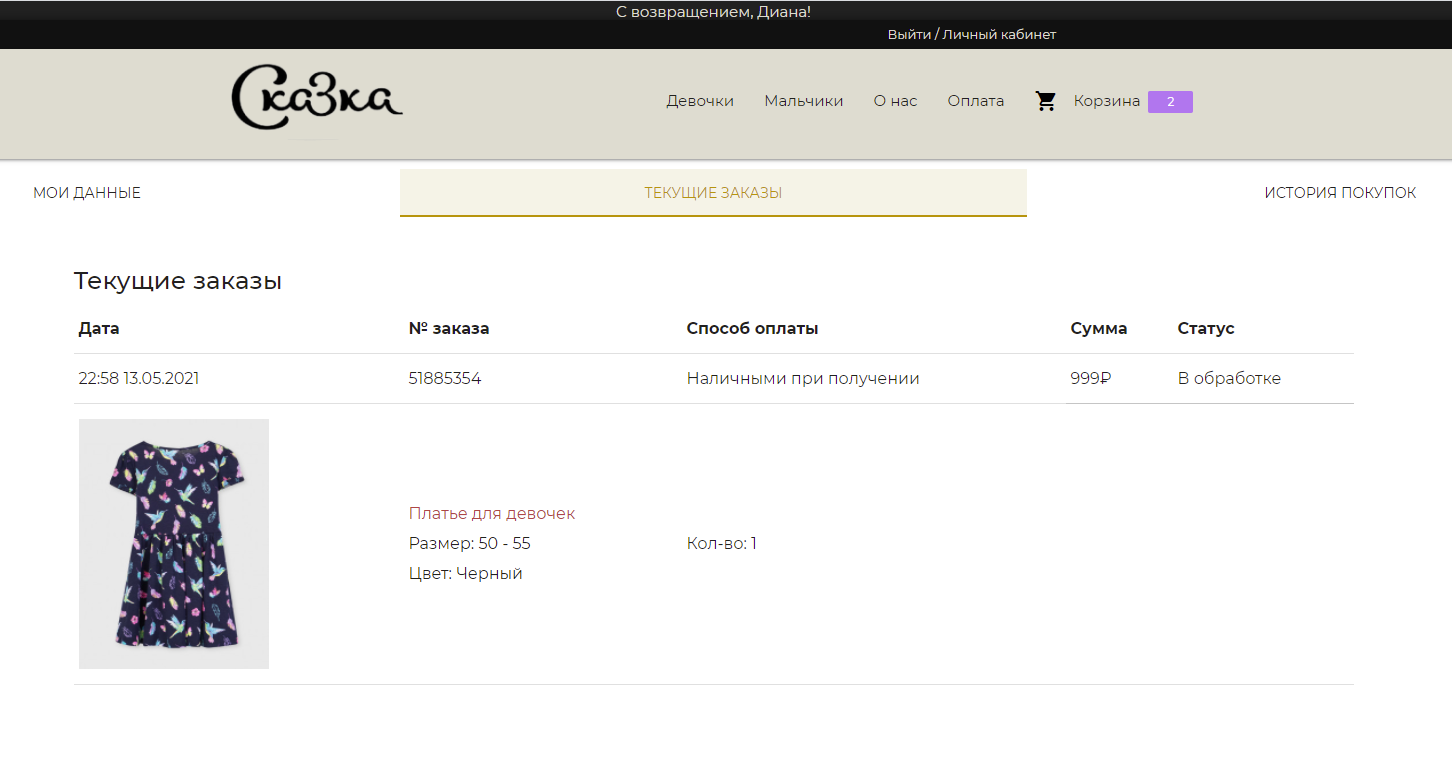


Рисунок 19. Информация о текущих заказах пользователя.

### **2.4. Разработка административного интерфейса**

Админ-панель интернет – магазина – также не мало важная часть проекта. Из нее администратор производит все необходимые действия: добавляет, изменяет, удаляет товары, отслеживает статусы заказов, изменяет их.

Админ-панель есть у всех сайтов с CMS — системой управления контентом. Такая система позволяет с легкостью создавать сайт и наполнять его материалами любого формата без навыков программирования. Если сайт сделан не с помощью CMS, а разработан программистами с нуля, то его редактирование потребует изучения HTML, CSS, PHP и других языков программирования. В некоторых случаях разработчики также создают адаптированную под нужды проекта CMS для облегчения работы с сайтом, тогда вход в админ-панель будет осуществляться определенным создателями ресурса способом.

В разрабатываемом мной интернет-магазине у администратора будет возможность добавлять, изменять, удалять товары на странице, а также уточнять содержимое заказа, при его выдаче и удалять заказ, если он был выдан или был оформлен возврат. Панель администрирования состоит их двух блоков: «Товары» и «Заказы» (Рисунок 20, 21).

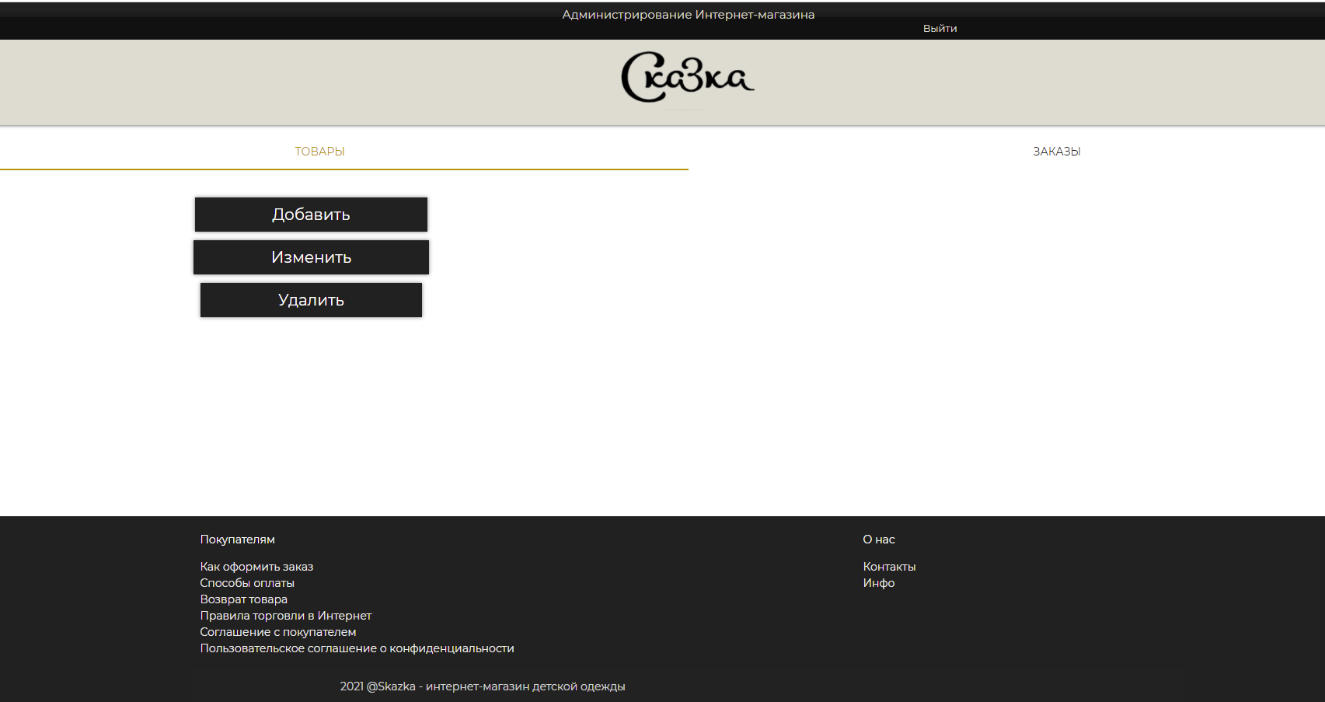


Рисунок 20. Блок «Товары».

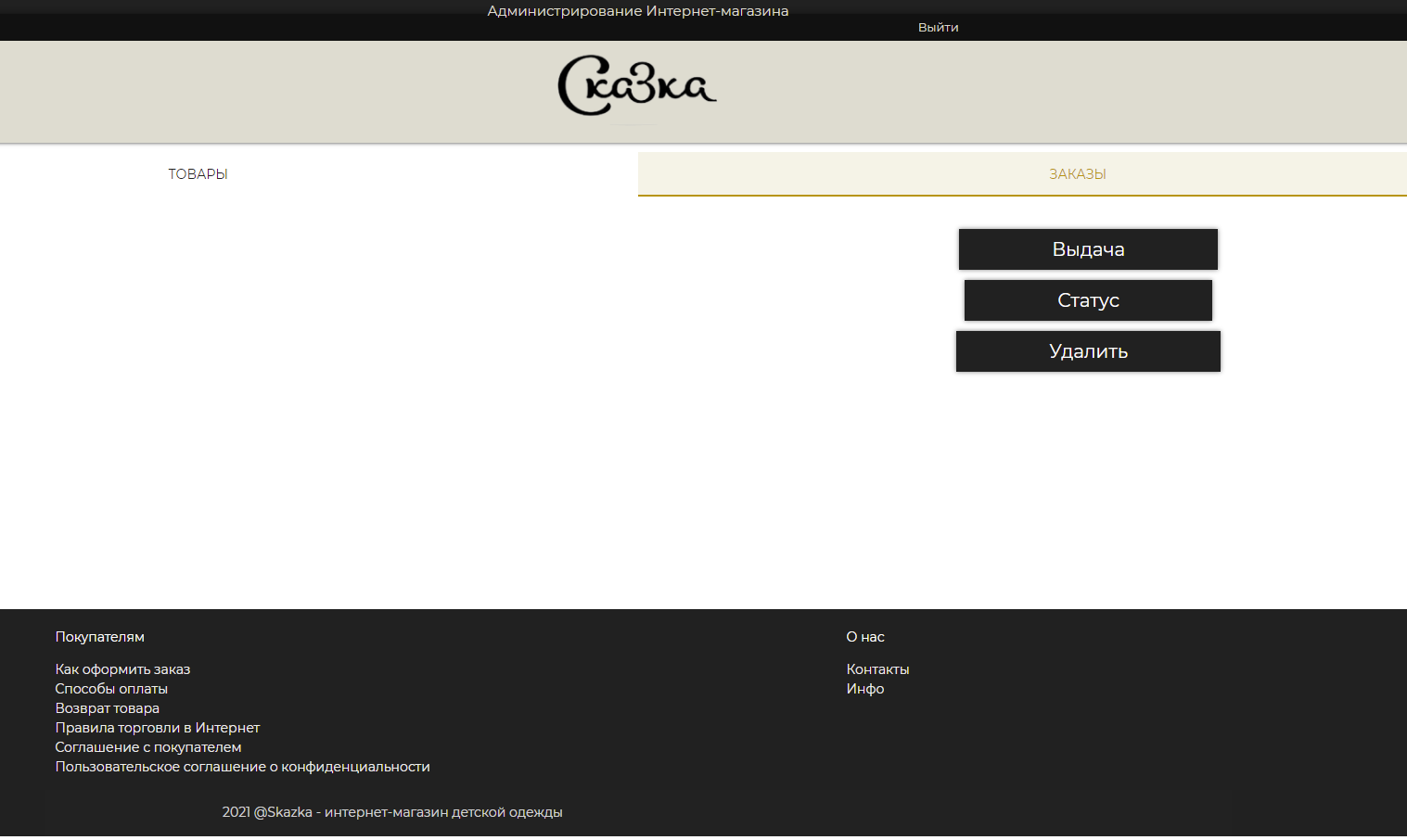


Рисунок 21. Блок «Заказы».

Так, например, чтобы найти определенный товар и изменить его цену, достаточно выбрать «Товары – Изменить». Указав артикул товара, можно найти все товары, имеющие его (Рисунок 22).

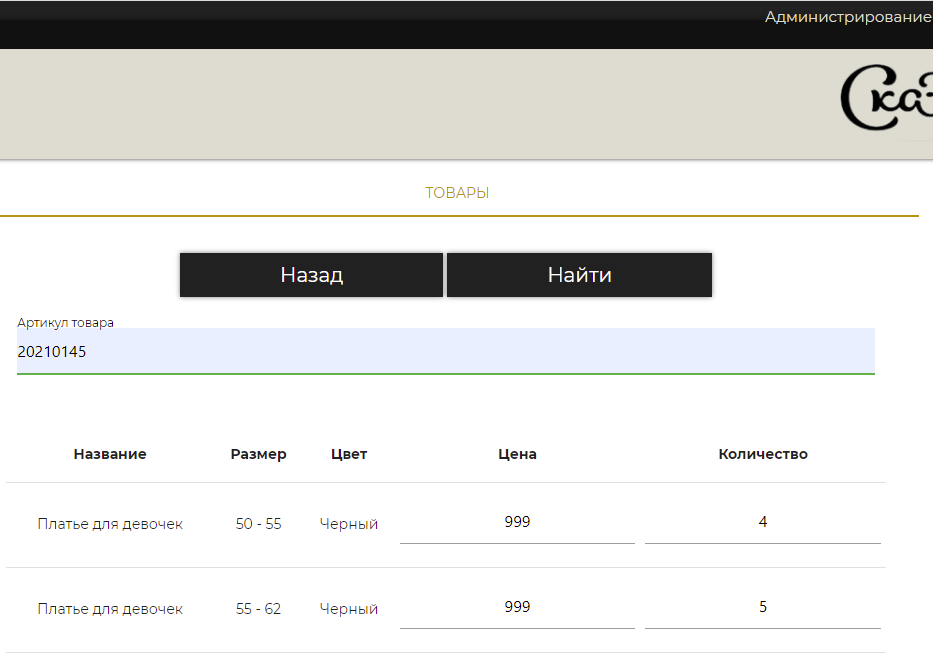


Рисунок 22. Отбор по артикулу товара.

В приложении представлен листинг основных функций страниц сайта.

### **2.5. Описание функционирования веб-сайта**

При разработке сайта были использованы:

* СУБД MySQL для создания базы данных;
* веб-интерфейс для администрирования СУБД – phpMyAdmin.

Совокупность этих программных решений позволило разработать удобный и функционирующий сайт, со структурированными разделами, удобным и интуитивным интерфейсом как для простых пользователей, так и для администратора.

База данных интернет-магазина содержит 11 сущностей (рисунок 23, таблица 1).

Таблица 1. Сущности БД

|  |  |
| --- | --- |
| Сущность **Users** хранит информацию о зарегистрированных пользователях интернет-магазина | |
| ID\_user | int(11) ключевой |
| patron | text |
| name | text |
| surname | text |
| email | text |
| phone | text |
| login | text |
| password | text |
| Сущность **Images** хранит информацию об изображениях товаров на сайте. | |
| ID\_images | int(11) ключевой |
| path | text |
| Сущность **Categories** хранит категории товаров | |
| ID\_category | int(11) ключевой |
| name\_category | text |
| Сущность **Sizes** хранит информацию о размерах товаров | |
| ID\_size | int(11) ключевой |
| name\_size | text |
| Сущность **Carts** хранит информацию о пользователях, их корзинах | |
| ID\_cart | int(11) ключевой |
| ID\_user | int(11) индекс |
| ID\_product | int(11) индекс |
| count | int(11) |
| reservation | datetime |
| Сущность **Colors** хранит информацию о цветах товаров | |
| ID\_color | int(11) ключевой |
| name\_color | text |

Продолжение таблицы 1.

|  |  |
| --- | --- |
| Сущность **Pays** хранит информацию о способах оплаты товаров | |
| ID\_pay | int(11) ключевой |
| name\_pay | text |
| Сущность **Genders** хранит две записи: «Девочки» и «Мальчики» | |
| ID\_gender | int(11) ключевой |
| name\_gender | text |
| Сущность **Statuses** хранит информацию о возможных статусах товаров для дальнейшего присвоения | |
| ID\_status | int(11) ключевой |
| name\_status | text |
| Сущность **Products** хранит подробную информацию о товаре | |
| ID\_product | int(11) ключевой |
| name | text |
| vendor | int(11) |
| ID\_images | int(11) индекс |
| ID\_category | int(11) индекс |
| ID\_color | int(11) индекс |
| ID\_size | int(11) индекс |
| price | int(5) |
| description | text |
| count | int(11) |
| ID\_gender | int(11) индекс |
| Сущность **Products** хранит информацию о заказах пользователей | |
| ID\_order | int(11) ключевой |
| number\_order | int(11) |
| ID\_user | int(11) индекс |
| ID\_product | text |
| count | int(11) |
| ID\_pay | int(11) индекс |
| summa | int(11) |
| date | datatime |
| ID\_status | int(11) |

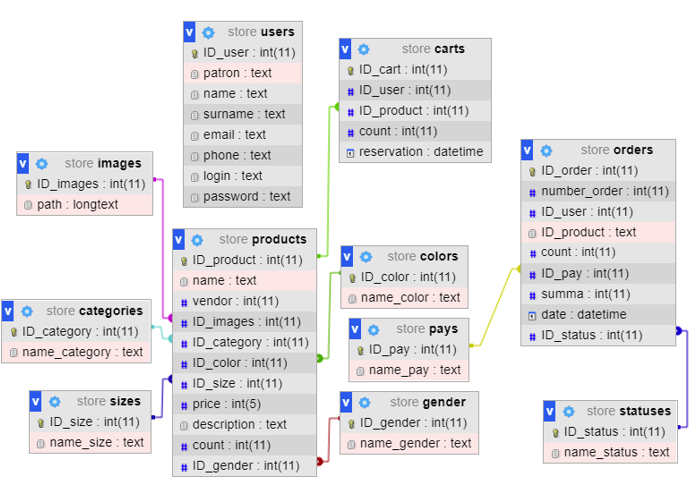


Рисунок 23. Структура базы данных.

### **2.6. Тестирование программного продукта**

Тестирование сайта – это проверка сайта различными способами на правильную работу.

Для тестирования сайтов были выбраны следующие процедуры:

• просмотр сайта на мониторах, имеющих различную разрешающую способность;

• проверки работы всех обязательных функций сайта;

• проверка правильности отображения шрифтов различными браузерами и их версиями;

• проверки гиперссылок, поиск нерабочих ссылок;

• проверка графических материалов: фотографий, картинок;

• проверка соответствия наполнения страниц исходному контенту.

В связи с этим был разработан сценарий тестирования (таблица 2).

Таблица 2. Сценарий тестирования программного продукта

|  |  |  |
| --- | --- | --- |
| **№** | **Действие** | **Результат** |
| **1.** | **Сценарий «Регистрация в системе»** | |
|  | Нажать на кнопку «Регистрация». | Откроется форма регистрации с полями ввода данных. |
|  | Заполнить поля личных данных, указать почту. Нажать «Регистрация». | Проверка введенных данных, в случае успеха – новая запись в БД в таблице «Users». В случае неправильного ввода данных – сообщение с указанием рекомендаций по исправлению ошибок. |
| **2** | **Сценарий «Авторизация в системе»** | |
|  | Нажать на кнопку «Войти». | Откроется форма с полями ввода данных. |
|  | Указать данные несуществующего в системе пользователя. | Отказ входа. |
|  | Указать данные существующего в системе пользователя. | Разрешить доступ. |
|  | Указать служебные данные администратора. | Откроется панель администрирования сайта. |
| **3** | **Сценарий «Проверка отображения товаров»** | |
|  | В меню выбрать категорию «Девочки». | Откроется страница с детской одеждой для девочек. |
|  | Нажать на любой товар. | Откроется страница с подробным описанием товара. |
|  | Нажать на кнопку «Перейти на следующую страницу». | Страница обновится и появятся следующие 20 товаров данной категории. |
|  | Нажать на кнопку «Выбрать размер». Выбрать любой размер. | На странице отобразятся товары с указанной фильтрацией. |
| **4** | **Сценарий «Работа корзины»** | |
|  | Выбрать товар из списка товаров. Нажать на кнопку «Добавить в корзину». | В базе данных добавилась запись в статусе корзины, отображение количества выбранных товаров в корзине. |
|  | Нажать на кнопку «Корзина». | Откроется страница с ранее выбранным товаров с указанием суммы заказа. |
|  | Нажать на кнопку «Удалить из корзины». | В базе данных удалилась запись в статусе корзины, отображение количества выбранных товаров в корзине. |
|  | **Сценарий «Оформление заказа»** |  |
| **5** | Выбрать любой товар. | Товар добавится в корзину. |
|  | Нажать на кнопку «Корзина». | Откроется страница с ранее выбранным товаров с указанием суммы заказа. |
|  | Нажать на кнопку «Оформить заказ». | Откроется форма с полями ввода данных. |
|  | Указать способ оплаты. | Выбран способ оплаты. |
|  | Выполнить онлайн-оплату. | Оплата прошла успешно. Изменен статус заказа в БД. |
|  | Нажать на кнопку «Выйти». | Пользователь выходит из учетной записи. |
|  | Нажать на кнопку «Оформить заказ». | Откроется форма с полями ввода данных. |
|  | Указать способ оплаты. | Появится уведомление с просьбой регистрации/авторизации. |

Самой часто - встречающейся ошибкой при разработке была ошибка в синтаксисе языка программирования, реже встречались несоответствия названия сопоставляемых переменных, и отсутствие запрашиваемых полей в базе данных.

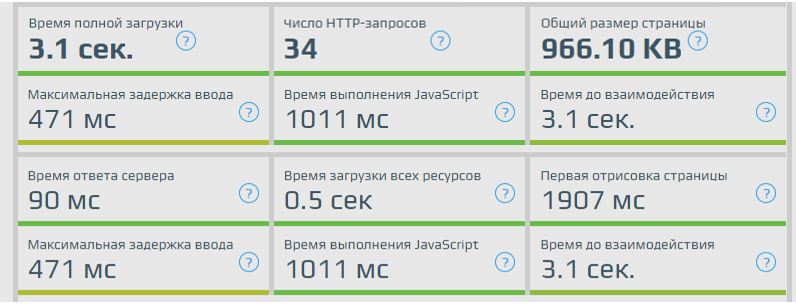
Тестирование проводилось после загрузки и установки программного продукта на хостинг. Оценки скорости работы сайта продемонстрированы на рисунке 24. В целом, результаты положительны. 

Рисунок 24. Результаты тестирования сайта.

## **ЗАКЛЮЧЕНИЕ**

В данной выпускной квалификационной работе был спроектирован и разработан сайт интернет-магазина для торгового предприятия «Сказка» города Нижневартовска. Сайт соответствует всем техническим требованиям, разработанным в процессе анализа предметной области.

Основными результатами, полученной в данной работе, являются:

* проанализирована предметная область;
* изучена структура веб-сайта;
* выявлены и разработаны функциональные задачи и возможности будущих пользователей;
* произведен обзор существующих аналогов;
* сформированы требования к веб-сайту;
* изучены программные продукты, использованные для выполнения работы.
* разработана функциональная схема;
* разработан интерфейс веб-сайта и административного модуля;
* спроектирована база данных;
* разработан веб-сайт, полностью готовый к размещению на хостинге в сети Интернет;
* произведено тестирование и отладка на сервере.

Данный сайт позволит решить основную проблему компании, обеспечит продвижение. Пользователи могут посмотреть информацию о товарах предприятия и сделать заказ. За время работы над выпускным квалификационным проектом опыт и навыки в веб-разработке, повысились в разы, было прочитано и изучено множество различной литературы и интернет ресурсов. В перспективе планируется доработка панели администрирования, улучшения качества функциональной работы интернет-магазина.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

* + 1. Рассел Джесси. Вёрстка веб-страниц; Книга по Требованию - Москва, 2012.
    2. Макнейл Патрик. Веб-дизайн. Книга идей веб-разработчика; Питер - Москва, 2014.
    3. Киселев С. В., Алексахин С. В., Остроух А.А В. Веб-дизайн; Академия - Москва, 2012.
    4. Маркин, А.В. Основы Web-программирования на PHP / А.В. Маркин. - М.: Диалог-МИФИ, 2012.
    5. Будилов В. PHP Экспресс-курс - Спб. БХВ-Петербург, 2005.
    6. Гаевский А.Ю. Создание Web-страниц и Web-сайтов. / Гаевский А.Ю., Романовский В.А, 2008.
    7. Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2017.
    8. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: БХВПетербург, 2017.
    9. Дари К., Баланеску Э. PHP и MySQL. Создание интернет-магазина. – М.: Издательский дом «Вильямс».
    10. К. Акила. Интернет-магазин с нуля. Полное пошаговое руководство.
    11. Пауэрс Дэвид. PHP. Создание динамические страниц, Изд-во: Рид Групп. 2012.
    12. Робинс Дженнифер.HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Дженнифер Роббинс; 4-е издание. - Изд-во М.: Эксмо, 2014.
    13. ХенникБех. HTML и CSS Путь к совершенству. - Изд-во - СПб.: Питер, 2011.
    14. Янк Кевин. PHP и MySQL. От новичка к профессионалу, Изд-во: Эксмо, 2013.

## **ПРИЛОЖЕНИЕ**

«Авторизация пользователя».

<?php

require 'config.php';

session\_start();

//Проверка на совпадение введенного пароля и логина

$selectSQL =

'SELECT \* FROM users

    WHERE login =  "'.$\_POST["login"].'" AND password = "'.md5(md5($\_POST['password'])).'"';  
     $result = mysqli\_query($link, $selectSQL);

    $info = mysqli\_fetch\_assoc($result);

//Создание сессии с информацией о пользователе

    $\_SESSION['user'] = [

        "id" => $info['ID\_user'],

        "login" => $info['login'],

        "username" => $info['name'],

        "usersurname" => $info['surname'],

        "patron" => $info['patron'],

        "email" => $info['email'],

        "phone" => $info['phone'],

    ];  
    $\_SESSION['message'] = [

      "text" => "С возвращением, " . $\_SESSION['user']['username'] . "!",

      "type\_message" => "text"

    ];  
//Если по результату выполнения запроса число строк больше > 0, потверждаем успешный вход

    if($result->num\_rows>0){

        if($\_POST["login"] == "admin"){

            $\_SESSION['message']['type\_message'] = 'admin\_ok';

            $\_SESSION['message']['text'] = "Администрирование Интернет-магазина";

            header("Location: admin.php");

        }

        else {$\_SESSION['message']['type\_message'] = 'log\_ok';

        header("Location: cart\_tovar.php");}

    }

//Иначе выводим ошибку входа

    else {

        $\_SESSION['message']['text'] = "Ошибка входа";

        $\_SESSION['message']['type\_message'] = 'log\_error';

        header("Location: cart\_tovar.php");

       }

mysqli\_close($link);

?>

«Регистрация пользователя».

<?php

    require 'config.php';  
    //Запуск сессий;

    session\_start();

    $text = $\_SESSION['message']['activate'];  
    //Выбор всех пользователей с данным логином

    $selectSQL =

    'SELECT \* FROM users

        WHERE login =  "'.$\_POST["login"].'"';

    $insertSQL =

                'INSERT INTO users

                    (surname, name, patron, email, phone, login, password)

                    VALUES

                    (

                    "'.$\_POST["last\_name"].'",

                    "'.$\_POST["first\_name"].'",

                    "'.$\_POST["patron"].'",

                    "'.$\_POST["email"].'",

                    "'.$\_POST["phone"].'",

                    "'.$\_POST["login"].'",

                    "'.md5(md5(trim($\_POST['password']))).'"

                );';

                //Регистрация пользователя, если такой записи в БД нет

                $result = mysqli\_query($link, $selectSQL);

                if($result->num\_rows>0){

                    $\_SESSION['message']['text'] = "Ошибка";

                    $\_SESSION['message']['type\_message'] = 'reg\_error';

                    // header("Location: index.php");

                }

                else {

                    if ($\_POST["activate"] == $text) {

                     $query = mysqli\_query($link, $insertSQL);

                     $repeat\_select = mysqli\_query($link, $selectSQL);  
                     $info = mysqli\_fetch\_assoc($repeat\_select);

                     $\_SESSION['user'] = [

                        "id" => $info['ID\_user'],

                        "login" => $info['login'],

                        "username" => $info['name'],

                        "usersurname" => $info['surname'],

                        "patron" => $info['patron'],

                        "email" => $info['email'],

                        "phone" => $info['phone'],

                    ];

                    //Вывод сообщения об успешной регистрации

                     $\_SESSION['message']['text'] = "Добро пожаловать, " .$\_SESSION['user']['username']. "!";

                     $\_SESSION['message']['type\_message'] = 'reg\_ok';

                     $\_SESSION['message']['activate'] = "good";

                        header("Location: cart\_tovar.php");

                    } else {

                        $\_SESSION['message']['activate'] = "error";

                        header("Location: cart\_tovar.php");

                    }

                }

    mysqli\_close($link);

?>

«Добавление товара в корзину».

<?php

        include ("header.php");

        session\_start();

        $id =  $\_POST['ID'];

            //Получаем ID пользователя

            if($\_SESSION['user']['id'] == ""){

                $\_SESSION['user']['id'] = $\_POST['userID'];

            }  
            if ($\_POST['ID'] != ""){ //ID выбранного товара

                $check\_count = 'SELECT count FROM products WHERE ID\_product = '.$id.'';

                $update\_count = 'UPDATE products SET count = count-1 WHERE ID\_product = '.$id.'';

                //Если корзина с выбранным ID товара не создана, создаем и   
указываем его ID, количество

                // и максимальное количество, чтобы избежать ситуации, когда   
пользователь выбирает два товара, а на складе всего 1

                if(empty($\_SESSION['cart\_items'][$id])){

                        $\_SESSION['cart\_items'][$id]['ID\_product'] = $\_POST['ID'];

                        $\_SESSION['cart\_items'][$id]['quantity'] = 1;

                        $query\_update\_count=mysqli\_query($link, $update\_count);  // Обновляем количество товара на складе

                        $query\_count=mysqli\_query($link, $check\_count);   
                //Создаем в БД корзину для пользователя, чтобы хранить его корзину, в случае, если сайт будет закрыт

                //до истечения срока бронирования товаров

                            $select =

                            'SELECT \* FROM carts

                            WHERE ID\_user =  "'.$\_SESSION['user']['id'].'" and ID\_product = "'.$\_POST['ID'].'"';

                            $query\_selectSQL=mysqli\_query($link, $select);  
                            if($query\_selectSQL->num\_rows == 0){

                                $insert =

                                'INSERT INTO carts

                                (ID\_product, ID\_user, count, reservation)

                                VALUES

                                (

                                "'.$\_POST['ID'].'",

                                "'.$\_SESSION['user']['id'].'",

                                "'.$\_SESSION['cart\_items'][$id]['quantity'].'",

                                now()

                                );';

                            $query\_insertSQL=mysqli\_query($link, $insert);

                        }

                        echo "<p id = 'message-box'>Товар успешно добавлен в корзину!</p>";

                            foreach($query\_count as $product) {

                                $\_SESSION['cart\_items'][$id]['max'] = $product['count'];

                            }

                }

                //Если информация о таком товаре в корзине уже есть, добавляем к количеству 1, обновляем количество товара на складе

                else {

                        $\_SESSION['cart\_items'][$id]['ID\_product'] = $\_POST['ID'];

                        $query\_count=mysqli\_query($link, $check\_count); //Обновляем максимальное количество для выбора товара

                        foreach($query\_count as $product) {

                            $\_SESSION['cart\_items'][$id]['max'] = $product['count'];

                        }

                        if($\_SESSION['cart\_items'][$id]['max'] == 0){

                            echo "<p id = 'message-box'>Отсутствует такое количество товара</p>";

                        } else {

                                $\_SESSION['cart\_items'][$id]['quantity']++;

                                $query\_update\_count=mysqli\_query($link, $update\_count);                                         $count = $\_SESSION['cart\_items'][$id]['quantity'];

                                $update\_cart = 'UPDATE carts SET count = '.$count.' WHERE ID\_user =  '.$\_SESSION['user']['id'].' and ID\_product = '.$id.'';

                                $query\_update\_cart=mysqli\_query($link, $update\_cart);     
                            echo "<p id = 'message-box'>Товар успешно добавлен в корзину!</p>";

                            }  
            }

            }

«Удаление товаров из корзины».

  //Удаление товаров из корзины

        function check\_count($idQ,  $link, $action\_BD){

            //Отнимаем одну единицу товара из общего числа на складе, если пользователь захотел увеличить число товара

            if($action\_BD == "delete1"){

  $check\_count = 'SELECT count FROM products WHERE ID\_product = '.$idQ.'';

 $update\_count = 'UPDATE products SET count = count-1 WHERE ID\_product = '.$idQ.'';

                $query\_update\_count=mysqli\_query($link, $update\_count);

                $query\_count=mysqli\_query($link, $check\_count);

                foreach($query\_count as $product) {

                    $\_SESSION['cart\_items'][$idQ]['max'] = $product['count'];

                }

            }

            //Возвращаем все, если пользователь нажал "Очистить корзину"

            if($action\_BD == "returnAll"){

 $check\_count = 'SELECT count FROM products WHERE ID\_product = '.$idQ.'';

                    foreach($\_SESSION['cart\_items'] as $id => $value) {

$update\_count = 'UPDATE products SET count = count+ ' . $value["quantity"] . '

WHERE ID\_product = '.$value["ID\_product"] .'';

                        $query\_update\_count=mysqli\_query($link, $update\_count);

                    }

            }

            //Возвращаем одну единицу товара, если пользователь захотел уменьшить число товара

            if($action\_BD == "return1"){

 $check\_count = 'SELECT count FROM products WHERE ID\_product = '.$idQ.'';

                $query\_count=mysqli\_query($link, $check\_count);

                    foreach($query\_count as $product) {

                        $\_SESSION['cart\_items'][$idQ]['max'] = $product['count'];

                    }

  $update\_count = 'UPDATE products SET count = count+1  WHERE ID\_product = '.$idQ.'';

                        $query\_update\_count=mysqli\_query($link, $update\_count);

            }

        }

«Очищение корзины – AJAX».

    //Очищение корзины при нажатии "Очистить корзину"

                    $("#clear").click(function(event) {

                        event.preventDefault();

    //Отправляем запрос на очищение корзины без обновления страницы

                            $.ajax({

                                    type: "POST",

                                    url: "cart\_tovar.php",

                                    data: {action: "clear"},

                                    success: function()

                                            {

          //Динамично изменяем содержимое корзины

           var cart = document.getElementById('cart-change');

            cart.innerHTML = "Ваша корзина пуста";

            $('.cart-product').css('display','none');

            $('.div-oform').css('display','none');

            var cart = document.getElementById('itogo');

             cart.innerHTML = "0";  
                       event.target.classList.add("disabled");

             document.getElementById('oform').classList.add("disabled");  
                  var carts = document.querySelectorAll('.badge');

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

                         var val = parseInt(carts[i].innerHTML) + 1;

                          carts[i].innerHTML = 0;

                             }

                       }

                 });

         });

«Оформление заказа».

//Создание нового заказа

        //Доступно только при авторизации пользователя

        function add\_new\_order($formdata, $link, $about\_user){

            //Функция для генерации строки

            $permitted\_chars = '0123456789';

            function generate\_string($input, $strength = 8) {

                $input\_length = strlen($input);

                $random\_string = '';

                for($i = 0; $i < $strength; $i++) {

                    $random\_character = $input[mt\_rand(0, $input\_length - 1)];

                    $random\_string .= $random\_character;

                }

                return $random\_string;

            }

                $number\_order = generate\_string($permitted\_chars, 8); //Генерируем номер заказа

                $users = $\_SESSION['user']['id'];

                //Получаем содержимое корзины

                foreach($\_SESSION['cart\_items'] as $id => $value) {

                    $ids.=$id.",";

                $count\_prodicts.=$\_SESSION['cart\_items'][$id]['quantity'].",";

                }

                $ids=substr($ids, 0, -1); //Формируем строку с ID товаров

                $count\_prodicts=substr($count\_prodicts, 0, -1);

                $date = date("Y-m-d H:i:s"); //Формируем дату заказа

                $summa=$formdata["summa"];  //Получаем сумму заказа

                if ($formdata["pay"] == "nal"){ //Получаем способ оплаты

                    $pay = 1;

                } else {

                    $pay = 2;

                }

                $status = 1; //Указываем статус 1, т.к. в БД ему присвоен "1" означает - "В обработке"

                //Добавляем в БД новую запись

            $new\_order\_SQL =

                'INSERT INTO orders

   (number\_order, ID\_user, ID\_product, count, ID\_pay, summa, date, ID\_status)

                VALUES

                (

                "'.$number\_order.'",

                "'.$users.'",

                "'.$ids.'",

                "'.$count\_prodicts.'",

                "'.$pay.'",

                "'.$summa.'",

                "'.$date.'",

                "'.$status.'"

                );';

                    $query\_new\_order=mysqli\_query($link, $new\_order\_SQL);

                    echo "<p id = 'message-box'>Ваш заказ успешно оформлен! Отслеживайте статус заказа в личном кабинете. Спасибо!</p>";

                    unset($\_SESSION['cart\_items']); //Очищаем корзину

                    $control0 = 'DELETE FROM carts WHERE ID\_user =  '.$\_SESSION['user']['id'].''; // Удаляем корзину пользователя

                    $query\_updateSQL=mysqli\_query($link, $control0);

        }

«Отправка кода подтверждения на электронную почту».

<?php  
//send.php

// Файлы phpmailer

session\_start();

require 'PHPMailer/PHPMailer.php';

require 'PHPMailer/SMTP.php';

require 'PHPMailer/Exception.php';  
$permitted\_chars = '0123456789';

function generate\_string($input, $strength = 8) {

    $input\_length = strlen($input);

    $random\_string = '';

    for($i = 0; $i < $strength; $i++) {

        $random\_character = $input[mt\_rand(0, $input\_length - 1)];

        $random\_string .= $random\_character;

    }

    return $random\_string;

}  
    $email = $\_POST['email'];

    $name = $\_POST['first\_name'];

    $surname = $\_POST['last\_name'];

    $patron = $\_POST['patron'];

    // Формирование самого письма

    $title = "Подтверждение Email";  
    $text = generate\_string($permitted\_chars, 8); //Формируем код потверждения

    // Сообщение для Email

    $body = '

    <html>

    <body>

    <h3>Здравствуйте,  '. $surname .'  ' . $name .' ' . $patron .'! </h3>

    <p>Чтобы подтвердить Email, вернитесь на сайт и в строке "Код подтверждения" введите код активации: </p>' . $text . ' . Всего доброго!

    </body>

    </html>

    ';  
    // Настройки PHPMailer

    $mail = new PHPMailer\PHPMailer\PHPMailer();  
    $mail->isSMTP();

    $mail->CharSet = "UTF-8";

    $mail->SMTPAuth   = true;  
    // Настройки  почты

    $mail->Host       = 'smtp.mail.ru'; // SMTP сервера вашей почты

    $mail->Username   = 'skazka.ip@bk.ru'; // Логин на почте

    $mail->Password   = '\*\*\*\*\*\*\*\*\*'; // Пароль на почте

    $mail->SMTPSecure = 'ssl';

    $mail->Port       = 465;

    $mail->setFrom('skazka.ip@bk.ru', '"Интернет-магазин "Сказка""'); // Адрес самой почты и имя отправителя  
    // Получатель письма

    $mail->addAddress($\_POST['email']);

    // Отправка сообщения

    $mail->isHTML(true);

    $mail->Subject = $title;

    $mail->Body = $body;      
    if ($mail->send()) {

         $\_SESSION['message']['send'] = 'send\_ok';

         $\_SESSION['message']['activate'] = $text;

         echo  $\_SESSION['message']['send'];

    }

    else{

        $\_SESSION['message']['send'] = 'send\_error';

        echo  $\_SESSION['message']['send'];

    }

?>

«Реализация вывода корзины неавторизованного пользователя».

$about\_user = 0;

 $select\_foreach = 'SELECT \* FROM carts WHERE carts.ID\_user = "'.$userID.'"';

                $query\_select\_foreach=mysqli\_query($link, $select\_foreach);

                        if($query\_select\_foreach != ""){

                            $arr\_ids = [];

                            $arr\_counts = [];  
                    unset($\_SESSION['cart\_items']);

                    foreach($query\_select\_foreach as $products) {

                         array\_push($arr\_ids,  $products['ID\_product']);

                          array\_push($arr\_counts,  $products['count']);

                            $comb=array\_combine($arr\_ids, $arr\_counts);

                            foreach ($comb as $word\_ids => $word\_count) {

                            $word\_count = trim($word\_count);

                            $word\_ids = trim($word\_ids);

                                if(empty($\_SESSION['cart\_items'][$word\_ids])){

                    $\_SESSION['cart\_items'][$word\_ids]['ID\_product'] = $word\_ids;

                    $\_SESSION['cart\_items'][$word\_ids]['quantity'] = $word\_count;                                                      }      }         }    }

«Бронирование товара».

   //Бронирование товара

        $now = time() + 7200;

        $time\_limit\_cart = 2000; //Указываем срок бронирования в мс   
        $select\_about\_user = 'SELECT \* FROM carts WHERE ID\_user =  "'.$\_SESSION['user']['id'].'"';

        $query\_select\_user=mysqli\_query($link, $select\_about\_user);   
        foreach ($query\_select\_user as $carts){

        if ($now > strtotime($carts['reservation']) + $time\_limit\_cart){ //Если текущее время больше, чем дата резервирования товара + срок бронирования -  удаляем корзину

            $control\_time = $carts['reservation'];

            if(!empty($\_SESSION['cart\_items'])) {

                        foreach($\_SESSION['cart\_items'] as $id => $value) {                      $update\_count = 'UPDATE products SET count = count+ ' . $value["quantity"] . '   WHERE ID\_product = '.$value["ID\_product"] .'';

 $control\_delete = 'DELETE FROM carts WHERE ID\_user =  '.$\_SESSION['user']['id'].' and ID\_product = '.$value["ID\_product"] .' AND reservation = "'.$control\_time.'"';

            $query\_update\_count=mysqli\_query($link, $update\_count);

            $query\_update\_cart=mysqli\_query($link, $control\_delete);

                             }

}

 }

  }

«Обновление цены и количества товара – админ-панель». Часть 1.

function update\_about\_vendor($dataForm, $link){

        $name = $dataForm['name'];

        $value = $dataForm['value'];

        $id = $dataForm['id'];

            if ($name == "new\_price"){

            $update\_product = 'UPDATE products SET products.pricе = '.$value.' WHERE ID\_product = '.$id.'';

                    $query=mysqli\_query($link, $update\_product);

            } else if ($name == "new\_count"){

             $update\_product = 'UPDATE products SET products.count = '.$value.' WHERE ID\_product = '.$id.'';

                    $query=mysqli\_query($link, $update\_product);

            }

    }

«Обновление цены и количества товара – админ-панель». Часть 2.

$(document).on('click', '#action-get\_vendor\_\_post', function(event) {

                        event.preventDefault();

                        if ($("form[id='get-tovars\_\_form'] input").hasClass( "invalid")){

                            M.toast({html: "Введите артикул товара.", classes: 'rounded'});

                        } else {

                                $.ajax({

                                                url: 'admin.php',

                                                type: "POST",

                                                data: $("#get-tovars\_\_form").serialize() +"&action=get",

                                                success: function (data) {

                                                var array =  $(data).get();

                var arrs = array.find(x => x.id === 'return\_info\_vendor');

          if (arrs.innerHTML != "Не существует"){

                 var product = document.getElementById('get\_product');

                 product.replaceWith(arrs);

                 $("#update-tovars\_\_form").show();

                var product\_now = document.getElementById('return\_info\_vendor');

                  product\_now.setAttribute("id",  "get\_product");

                } else { M.toast({html: "Товаров с введенным артикулом не найдено.", classes: 'rounded'});

                            }

                      }

            });           } });

«Таблица с товарами».

<div class="tovars">

                            <?php

                            //Формирование страницы товаров

                                    if($resultP>num\_rows == 0){ echo "<h3 class = 'widgettitle'> К сожалению, по Вашему запросу не найдено товаров.</h3>";  }

                                foreach ($resultP as $row) {

                                    $slova = $row["name\_size"];

                                    $arr = explode(',', $slova);

                                    $img = $row["images"];

                                    $arr\_img = explode(',', $img);

                            ?>

   <div class="product-item" id = "<?php echo $row['vendor']; ?>">

   <?php echo "<a href = ''><img class = 'productimg' src=" . $arr\_img[0] . " width = '300px' height = '332px' id = ". $row['vendor'] . "></a>"; ?>  
                                        <div class="product-list">

                                 <?php

                                     echo "<h3>" . $row["name"] . "</h3>"

                                      echo "<h8> Размер: </h8> ";   
                                        foreach ($arr as $word) {

                      $word = trim($word);

                     echo " <a href = '' class = 'sizes'> " . $word . "</a>";  }

                     echo "<span class='price'> " . $row["pricе"] . "₽</span>"?>

                    <a href="about\_tovar.php" class="button prosmt">Просмотр</a>

                                        </div>

                                    </div>

                                <?php

                                }

                                ?>

                        </div>

«Отбор товара по категории». Часть 1.

var categories = document.querySelectorAll('.category');

                    Array.from(categories).forEach(function(categories){

                    categories.addEventListener('click', function(e) {

                            event.preventDefault();

                            var category = e.target.innerHTML;

                            var cat = document.getElementById('category\_input');

                            cat.setAttribute("value",category);

                            $.ajax({

                                    type: "POST",

                                    url: "list\_tovar.php",

                                    data: {category: category},

                                    success: function(data)

                                            {

                                            var content = $(data).find('.tovars');

                                             $(".tovars").replaceWith( content );

                                           var breadcrumb = $(data).find('.nav-breadcrumb');

                                 $(".nav-breadcrumb").replaceWith( breadcrumb );

                                            }

                                });

                        });

                    });

«Отбор товара по категории». Часть 2.

if ($category == 'Девочки' || $category == "Мальчики"){

            $selectSQL\_product =

            'SELECT products.ID\_product, products.vendor, gender.name\_gender, products.name, products.count,  categories.name\_category,  group\_concat(distinct images.path SEPARATOR ", ") as images, colors.name\_color,  group\_concat(distinct sizes.name\_size SEPARATOR ", ") as name\_size, products.pricе, products.description

                FROM products

                INNER JOIN sizes

                    ON products.ID\_size = sizes.ID\_size

                INNER JOIN images

                    ON products.ID\_images = images.ID\_images

                INNER JOIN colors

                    ON products.ID\_color = colors.ID\_color

                INNER JOIN gender

                    ON products.ID\_gender = gender.ID\_gender

                INNER JOIN categories

            ON products.ID\_category = categories.ID\_category WHERE (gender.name\_gender = "'. $category .'"  AND products.count !=0) group by products.vendor';

        }

        else {

            $selectSQL\_product =

            'SELECT products.ID\_product, products.vendor, gender.name\_gender, products.name, products.count,  categories.name\_category,  group\_concat(distinct images.path SEPARATOR ", ") as images, colors.name\_color,  group\_concat(distinct sizes.name\_size SEPARATOR ", ") as name\_size, products.pricе, products.description

                FROM products

                INNER JOIN sizes

                    ON products.ID\_size = sizes.ID\_size

                INNER JOIN images

                    ON products.ID\_images = images.ID\_images

                INNER JOIN colors

                    ON products.ID\_color = colors.ID\_color

                INNER JOIN gender

                    ON products.ID\_gender = gender.ID\_gender

                INNER JOIN categories

            ON products.ID\_category = categories.ID\_category WHERE (categories.name\_category =  "'. $category .'"  AND gender.name\_gender = "'. $gender .'"  AND products.count !=0) group by products.vendor';

        }

Присваивание неавторизованному пользователю уникального кода.

    $(".incart").click(function(e) {

                    event.preventDefault();

                    var about\_user = '<? echo $about\_user; ?>';

                            if(about\_user == 0){

                                var User = get\_cookie("username");

                                if (User == ""){

            var UserCode = randomString(6);

            let date = new Date(Date.now() + 1,08e7); // время жизни куки 3 часа

                 date = date.toUTCString();

                document.cookie = "username=" + UserCode + "expires=" + date;

                                }

                                 else {

                                    var UserCode = get\_cookie("username");

                                 }}