

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования**

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,  
профессор

\_\_\_\_\_ Л.Б. Соколинский

«\_\_\_»\_\_\_\_\_ 2024 г.

## **Разработка интернет-магазина товаров для художников**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.03.04.2024.308-380.ВКР

Научный руководитель,  
доцент кафедры СП, к.т.н.  
\_\_\_\_\_ М.В. Сухов

Автор работы,  
студент группы КЭ-403  
\_\_\_\_\_ А.В. Бабенко

Ученый секретарь  
(нормоконтролер)  
\_\_\_\_\_ И.Д. Володченко  
«\_\_\_»\_\_\_\_\_ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Южно-Уральский государственный университет  
(национальный исследовательский университет)»**  
Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

УТВЕРЖДАЮ  
Зав. кафедрой СП  
\_\_\_\_\_ Л.Б. Соколинский  
29.01.2024 г.

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы бакалавра**  
студентке группы КЭ-403  
Бабенко Александре Валерьевне,  
обучающейся по направлению  
09.03.04 «Программная инженерия»

- 1. Тема работы** (утверждена приказом ректора от 22.04.2024 г. №764-13/12)  
Разработка интернет-магазина товаров для художников.
- 2. Срок сдачи студентом законченной работы:** 03.06.2024 г.
- 3. Исходные данные к работе**
  - 3.1. Браун И. Веб-разработка с применением Node и Express. – Санкт Петербург: Питер, 2017. – 336 с.
  - 3.2. Хэррон Д. Node.js Разработка сервисных веб-приложений в JavaScript: Пер. с англ. Слинкина А.А. – М.: ДМК Пресс, 2012. – 144 с.
  - 3.3. Основы HTML. [Электронный ресурс] URL:  
<https://htmlacademy.ru/courses/297> (дата обращения: 05.02.2024 г.).
  - 3.4. CSS справочник. [Электронный ресурс] URL: <http://htmlbook.ru/css> (дата обращения: 05.02.2024 г.).
- 4. Перечень подлежащих разработке вопросов**
  - 4.1. Провести анализ предметной области и выполнить обзор аналогов.
  - 4.2. Спроектировать интернет-магазин.

4.3. Реализовать интернет-магазин.

4.4. Провести тестирование интернет-магазина.

**5. Дата выдачи задания: 29.01.2024 г.**

**Научный руководитель,**  
доцент кафедры СП, к.т.н.

М.В. Сухов

**Задание принял к исполнению**

А.В. Бабенко

## **ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	7
2. ПРОЕКТИРОВАНИЕ .....	14
2.1. Функциональные и нефункциональные требования.....	14
2.2. Диаграмма вариантов использования.....	15
2.3. Диаграмма деятельности.....	16
2.4. Архитектура системы.....	17
2.5. Схема базы данных.....	18
2.6. Средства реализации .....	20
3. РЕАЛИЗАЦИЯ СИСТЕМЫ.....	23
3.1. Реализация работы с базой данных.....	23
3.2. Реализация каталога товаров .....	23
3.3. Пагинация, сортировка и поиск товаров .....	26
3.4. Реализация корзины.....	28
3.5. Реализация страницы товара.....	29
3.6. Реализация регистрации и авторизации .....	30
3.7. Реализация страницы оформления заказа .....	32
3.8. Реализация страницы отслеживания заказа .....	33
3.9. Панель администратора.....	34
4. ТЕСТИРОВАНИЕ .....	37
ЗАКЛЮЧЕНИЕ .....	40
ЛИТЕРАТУРА.....	41
ПРИЛОЖЕНИЯ.....	43
Приложение А. Спецификация вариантов использования.....	43
Приложение Б. Адаптивная верстка .....	48

## **ВВЕДЕНИЕ**

### **Актуальность**

Интернет является простой и удобной вещью, которая стала неотъемлемой частью жизни каждого человека. Все более популярной становится интернет-торговля и ключевым элементом является разработка таких платформ.

Интернет-магазин позволяет людям, не выходя из дома, просмотреть весь объем товаров и не торопясь выбрать для себя подходящий продукт. Практически всегда на таких сайтах есть удобная сортировка по категориям, что облегчает поиск и экономит время пользователя. Также люди могут увидеть внешний вид продукта и его параметры, что значительно помогает при выборе товара. Пользователи таких интернет-ресурсов могут добавить понравившийся товар в личную корзину и отложить, если это необходимо, что позволяет сохранять начатую покупку. Большим преимуществом является то, что выбор товара можно произвести из любой точки мира, и с помощью доставки, которая включена на такие сайты, получить свои покупки прямо из дома. Более того, одним из главных преимуществ интернет-магазинов является их способность предоставлять покупателям персонализированные рекомендации и предложения, основанные на их предпочтениях и истории покупок. Это помогает повысить лояльность клиентов и стимулировать повторные покупки.

Кроме того, данный способ ведения бизнеса позволяет владельцам оставлять магазин открытым круглые сутки, а также позволяет вести бизнес без аренды помещения и найма сотрудников, что значительно уменьшает затраты на его ведение и является выгодным решением. К тому же, не у всех есть возможность посещать физические магазины, а интернет-магазин позволит владельцу расширить потенциальную аудиторию.

Интерес к творчеству у людей не угасает и, благодаря многочисленным преимуществам, создание онлайн-магазинов для художников становится перспективным.

## **Постановка задачи**

Целью выпускной квалификационной работы является разработка интернет-магазина товаров для художников. Для достижения поставленной цели необходимо решить следующие задачи:

- 1) провести анализ предметной области и выполнить обзор существующих аналогов;
- 2) спроектировать интернет-магазин;
- 3) реализовать интернет-магазин;
- 4) провести тестирование интернет-магазина.

## **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения, списка литературы и приложений. Объем работы составляет 50 страниц, объем списка литературы – 16 источников.

В первой главе описывается анализ предметной области, обзор существующих похожих решений и выбор инструментов разработки.

Вторая глава посвящена проектированию системы, которое включает в себя формирование функциональных и нефункциональных требований к сайту, создание диаграмм компонентов и вариантов использования.

В третьей главе содержится реализация разрабатываемой системы.

Четвертая глава содержит результаты функционального тестирования системы со стороны пользователя.

В приложении А содержатся таблицы, описывающие основные варианты использования (ВИ) разрабатываемого приложения.

В приложении Б содержатся скриншоты веб-сайта на различных размерах экрана.

## 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Интернет-магазин представляет собой простой и доступный способ проведения покупок. Такой сервис представляет собой уникальный сегмент рынка, ориентированный на профессиональных и любительских художников, а также на всех, кто увлекается творчеством и искусством. В данной нише представлен широкий ассортимент товаров, необходимых для занятий искусством, создания произведений и развития творческого потенциала. Именно интернет-магазины позволяют людям совершать покупки быстро, удобно и приятно.

### **Сравнительный анализ аналогов**

Для более полного понимания, какой функционал должен иметь такой веб-сайт, необходимо провести сравнительный анализ аналогов. Это позволит выявить достоинства и недостатки подобных решений и учесть их при создании собственного сервиса, что позволит сделать приложение более удобным и функциональным.

#### **«Etudesite» [1]**

«Этюд» – интернет-магазин товаров для художников, дизайнеров, архитекторов и других творческих личностей, который занимается продажей художественных товаров разных брендов.

На главной странице сайта имеется меню, в котором пользователь может выбрать нужный ему бренд товара, прочитать отзывы, узнать новости магазина, акции и уцененные товары. Чуть ниже представлены популярные категории, включающие в себя чаще приобретаемые товары, скидки и акции с карточками самих товаров, и новинки. При нажатии на выбранный товар открывается страница с более подробным описанием товара, а также приводятся похожие товары от других брендов. На сайте есть возможность войти в личный кабинет, но заказ можно оформить и без авторизации, с помощью отдельной страницы, на которой пользователь вводит в поля все необходимые данные. Также имеется сортировка товаров по популярности и цене и фильтрация по категориям.

Достоинства:

- 1) широкий выбор брендов и материалов;
- 2) удобный поиск товара;
- 3) отображение акций и скидок на главной странице сайта;
- 4) есть чат с консультантом;
- 5) наличие статей и мастер-классов от известных художников.

Недостатком является неудобная навигация, поскольку сайт содержит много информации.

На рисунке 1 представлен скриншот главной страницы сайта «Etudesite».

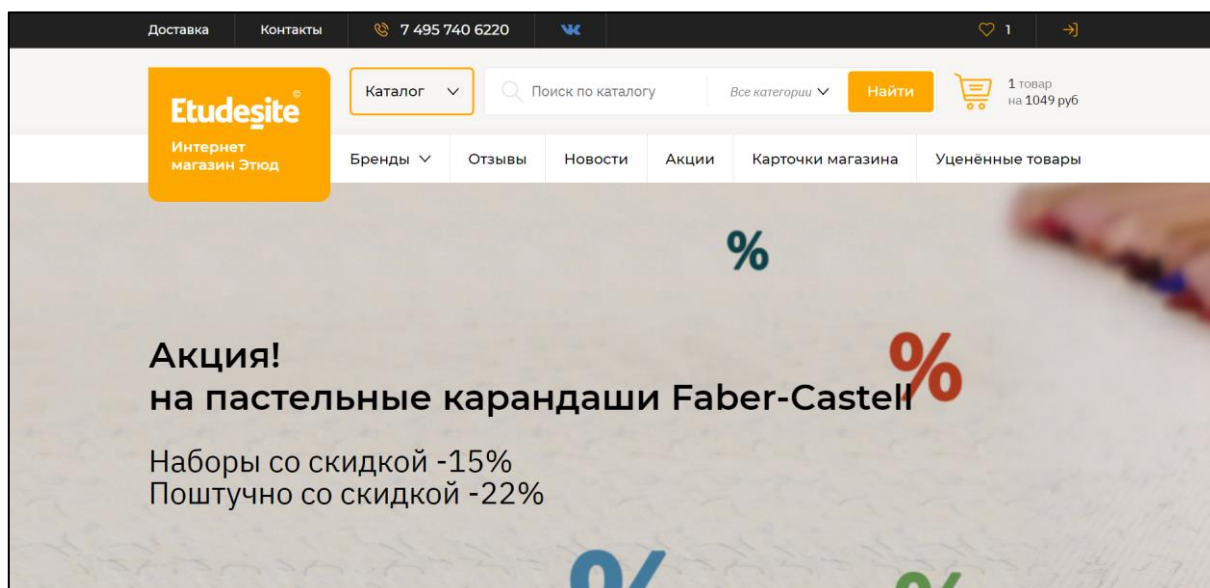


Рисунок 1 – Главная страница сайта «Etudesite»

### «Почерк» [2]

«Почерк» – небольшой магазин товаров для художников, специализирующийся на продаже художественных материалов для различных направлений.

На главной странице сайта представлены категории товаров, новости о скидках и акциях, кнопки для входа в личный кабинет и просмотра своих бонусных баллов. Есть возможность сортировки товаров по цене и популярности. Чуть ниже расположены новинки, распродажи и подборки акту-



альных материалов для создания подарков. При переходе на страницу товара, можно увидеть краткую информацию о нем, добавить его в корзину и перейти к покупке.

Достоинства:

- 1) есть возможность выбора доставки или оформление самовывоза;
- 2) наличие бонусной системы;
- 3) возможность создать личный кабинет.

Недостатки:

- 1) устаревший дизайн сайта;
- 2) незаметное расположение поля для поиска;
- 3) небольшой выбор для сортировки товаров.

На рисунке 2 представлен скриншот главной страницы сайта интернет-магазина «Почерк».

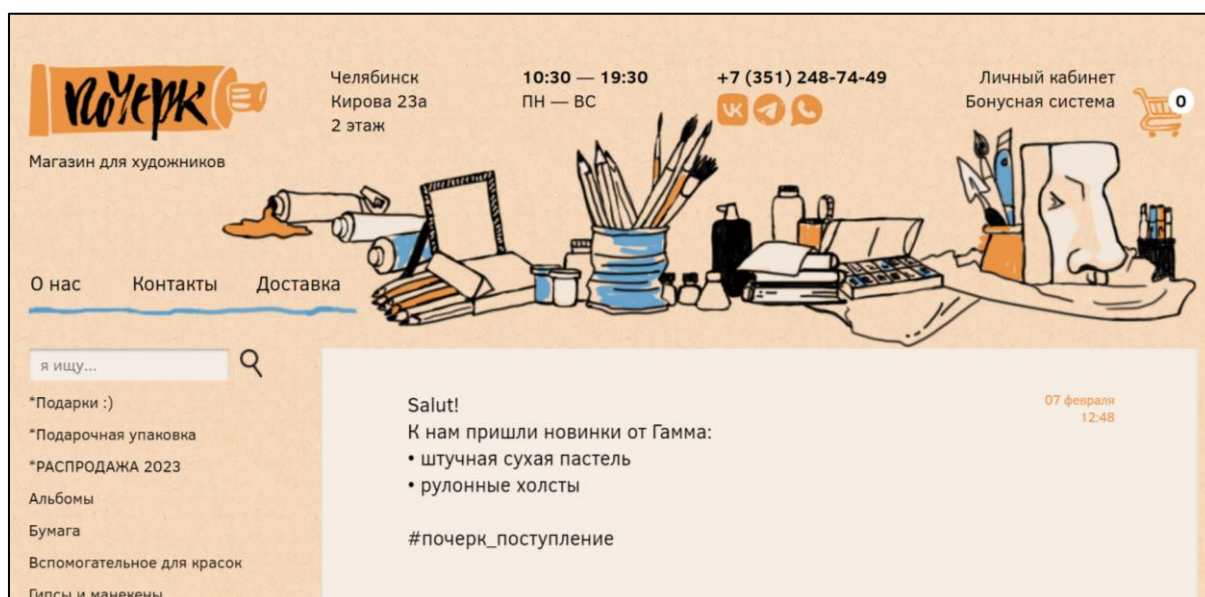


Рисунок 2 – Главная страница сайта интернет-магазина «Почерк»

### «Малевичъ» [3]

«Малевичъ» – это производственная оптовая компания, предлагающая товары для художников под собственной торговой маркой. Компания разрабатывает собственный модельный ряд с учетом потребностей художников и их цель наладить постоянное и взаимовыгодное сотрудничество с

магазинами, сетями, художественными студиями и другими потенциальными партнерами. Ассортимент постоянно расширяется, предлагая новые востребованные на рынки позиции.

На главной странице сайта есть шапка с категориями товара, кнопка входа в личный кабинет и возможность смены языка. Также присутствует возможность сортировки товаров по цене и популярности. Чуть ниже также находятся самые популярные для покупки категории, лучшие предложения по продуктам и новости компании. При выборе товара и переходе на его страницу, пользователь может сравнить цены различных вариаций одного товара, а также просмотреть рекомендации. Имеется возможность добавления товара в избранное или в корзину с дальнейшим оформлением заказа.

Достоинства:

- 1) лаконичный дизайн;
- 2) удобная навигация по сайту;
- 3) возможность добавить товар в избранное;
- 4) наличие личного кабинета;
- 5) возможность смены языка;
- 6) возможность поделиться корзиной с другим человеком;
- 7) наличие чата для консультации.

Недостатки:

- 1) необходимо много действий, чтобы добавить товар в корзину;
- 2) очень высокая минимальная сумма заказа;
- 3) неудобный поиск товара, сайт предлагает не всегда нужные варианты.

На рисунке 3 представлен скриншот главной страницы сайта интернет-магазина «Малевичь».

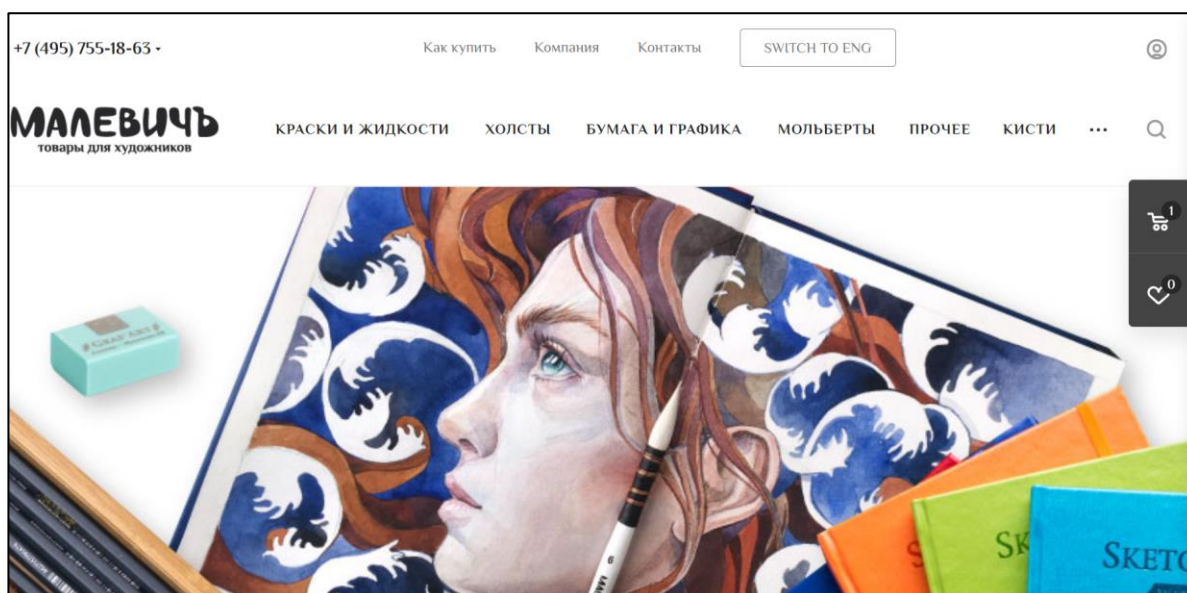


Рисунок 3 – Главная страница сайта интернет-магазина «Малевич»

#### «Леонардо» [4]

«Леонардо» – крупнейшая сеть хобби-гипермаркетов в России, имеющая одноименный интернет-магазин товаров для творчества и рукоделия, в котором каждый найдет для себя необходимое.

На главной странице сайта имеется шапка с выпадающим меню, в котором содержатся категории товаров, также поле для поиска и кнопки для перехода на страницы профиля, избранных товаров и корзины. Чуть ниже расположены такие иконки, как новинки, мастерские и прочие. При поиске товара очень удобно оформлены параметры, от бренда до мелких особенностей. Сама страница товара содержит описание, характеристики, похожие товары и возможность добавления в корзину. Около кнопки добавления в корзину описаны виды и сроки каждой из доставок.

Достоинства:

- 1) наличие личного кабинета;
- 2) удобный поиск товара с большим количеством параметров;
- 3) чат с консультантом;
- 4) добавление товара в избранное;
- 5) отображение сроков доставки до перехода в корзину.

Недостатками является то, что не всегда есть возможность добавить товар в корзину без перехода на страницу самого товара и то, что нет якоря на начале страницы.

На рисунке 4 представлен скриншот главной страницы сайта интернет-магазина «Леонардо».

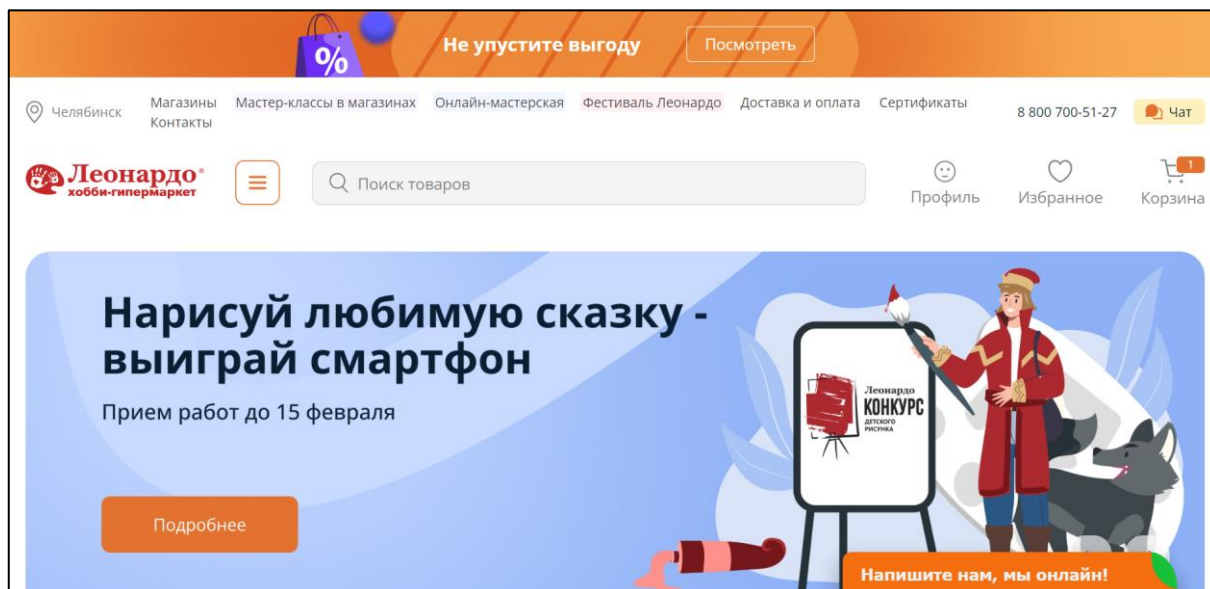


Рисунок 4 – Главная страница сайта интернет-магазина «Леонардо»

### «Красный карандаш» [5]

«Красный Карандаш» – это сеть творческих магазинов для профессиональных художников и творческих людей.

На главной странице сайта представлены хиты продаж, акции, новости и новинки товаров. В шапке сайта есть большой список категорий при нажатии на кнопку «Каталог», поле для поиска товара, а также возможность перейти на страницу избранного, корзины и личного кабинета.

При поиске товара есть много параметров для удобства, можно добавить товар в избранное или добавить его к сравнению, не переходя на страницу самого товара.

Достоинства:

- 1) возможность сравнить товары;
- 2) наличие личного кабинета;

- 3) удобный поиск и навигация;
- 4) фиксированная шапка с полем поиска и всеми необходимыми ссылками.

На рисунке 5 представлен скриншот главной страницы сайта интернет-магазина «Красный карандаш».

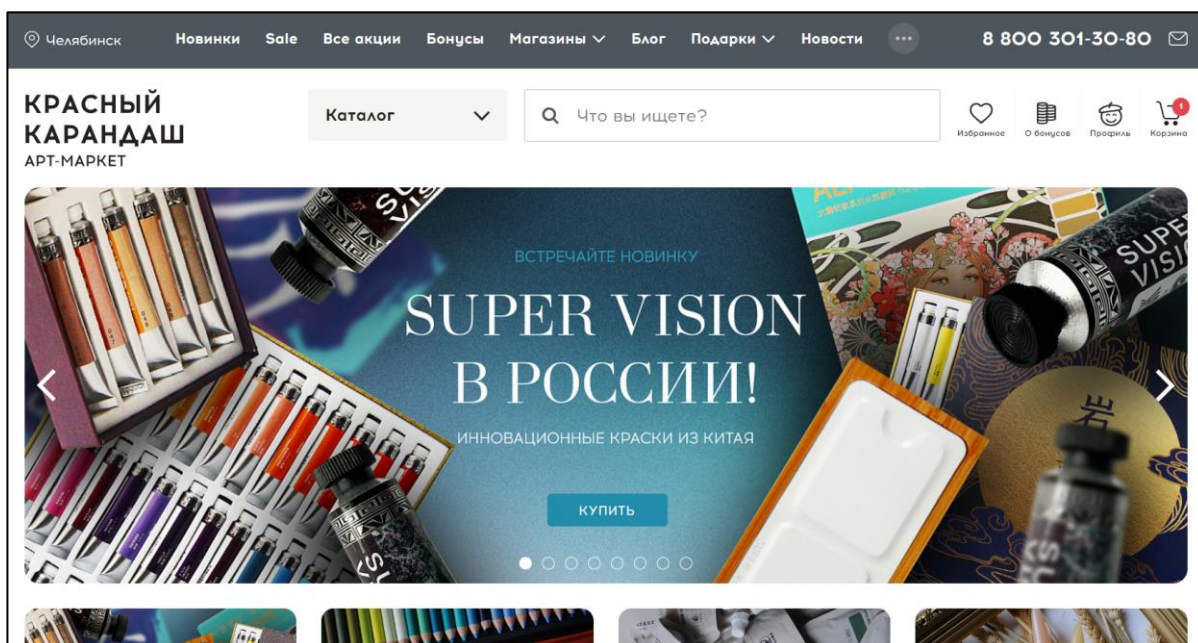


Рисунок 5 – Главная страница сайта магазина «Красный Карандаш»

При рассмотрении аналогов сайтов интернет-магазинов товаров для художников, было выяснено, что большинство систем имеют стандартный функционал, а именно поиск по категориям товаров с параметрами поиска, возможность просмотреть описание товара с его характеристиками, возможность добавления товара в избранное и в корзину, а также наличие личного кабинета пользователя.

### **Вывод по первой главе**

В данной главе был проведен анализ существующих решений для интернет-магазина товаров для художников, было произведено их сравнение и выявлены общие черты.

## **2. ПРОЕКТИРОВАНИЕ**

### **2.1. Функциональные и нефункциональные требования**

После рассмотрения предметной области и обзора существующих аналогов, были выявлены функциональные и нефункциональные требования к системе.

Сайт интернет-магазина товаров для художника должен отвечать следующим функциональным требованиям.

1. Пользователь должен иметь возможность сортировать товары по цене.
2. Пользователь должен иметь возможность добавлять товары в корзину и удалять их оттуда.
3. Пользователь должен иметь возможность поиска товара по названию.
4. Пользователь должен иметь возможность оформления заказа.
5. Пользователь должен иметь возможность входа в личный кабинет.
6. Администратор должен иметь возможность добавления, редактирования и удаления товаров в каталоге.
7. Администратор должен иметь возможность просматривать оформленные заказы и обрабатывать заявки.

Нефункциональные требования к системе.

1. Веб-сайт должен быть разработан на стеке MERN (MongoDB, Express, React, Node.js).
2. Сайт должен корректно отображаться при запуске в популярных браузерах.
3. Веб-сайт должен иметь адаптивный дизайн.
4. Панель управления администратора должна поддерживать работу в разных браузерах.

## 2.2. Диаграмма вариантов использования

Для изображения функциональных и нефункциональных требований системы была разработана диаграмма вариантов использования с помощью унифицированного языка моделирования UML [6]. При ее разработке были выделены следующие актеры: неавторизованный пользователь, авторизованный пользователь и администратор.

Неавторизованный пользователь – это человек, который не зарегистрирован на сайте и не имеет доступа к личному кабинету.

Авторизованный пользователь – это человек, который прошел процедуру авторизации на сайте и ему доступен основной функционал интернет-магазина.

Администратор – это человек, который имеет доступ к панели администратора и имеет права для внесения правок в содержимое сайта.

Диаграмма вариантов использования представлена на рисунке 6.

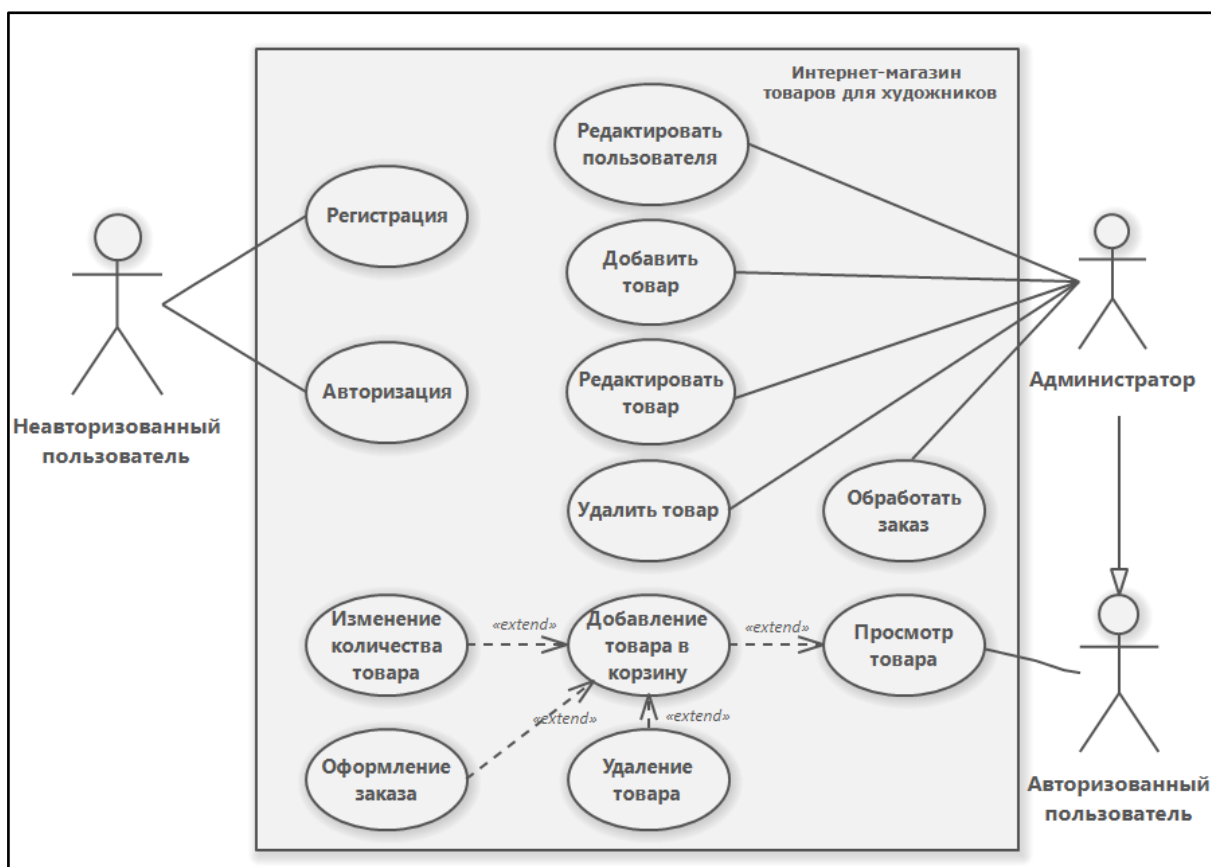


Рисунок 6 – Диаграмма вариантов использования

Краткое описание вариантов использования представлено ниже.

1. Неавторизованный пользователь может зарегистрироваться в системе, указав свою электронную почту, имя и пароль.
2. Неавторизованный пользователь может авторизоваться, используя электронную почту и пароль, введенные при регистрации.
3. Авторизованный пользователь может просмотреть товар и узнать о нем всю необходимую информацию.
4. Авторизованный пользователь может добавить товар в корзину для дальнейшей покупки.
5. Авторизованный пользователь может изменить количество товара в корзине при необходимости.
6. Авторизованный пользователь может удалить товар из корзины.
7. Авторизованный пользователь может оформить заказ.
8. Администратор может добавить новый товар.
9. Администратор может редактировать уже существующий товар.
10. Администратор может удалить ненужный товар с сайта.
11. Администратор может редактировать пользователя.
12. Администратор может обработать заказ пользователя.

Спецификация вариантов использования системы приведена в таблицах 1–12 приложения А.

### **2.3. Диаграмма деятельности**

Одной из основных задач интернет-магазина является возможность оформления заказа для авторизованного пользователя. Была разработана диаграмма деятельности, которая более подробно описывает вариант использования «Оформить заказ».

Предусловием для данного прецедента является то, что пользователь авторизован и находится на главной странице сайта. Диаграмма деятельности представлена на рисунке 7.



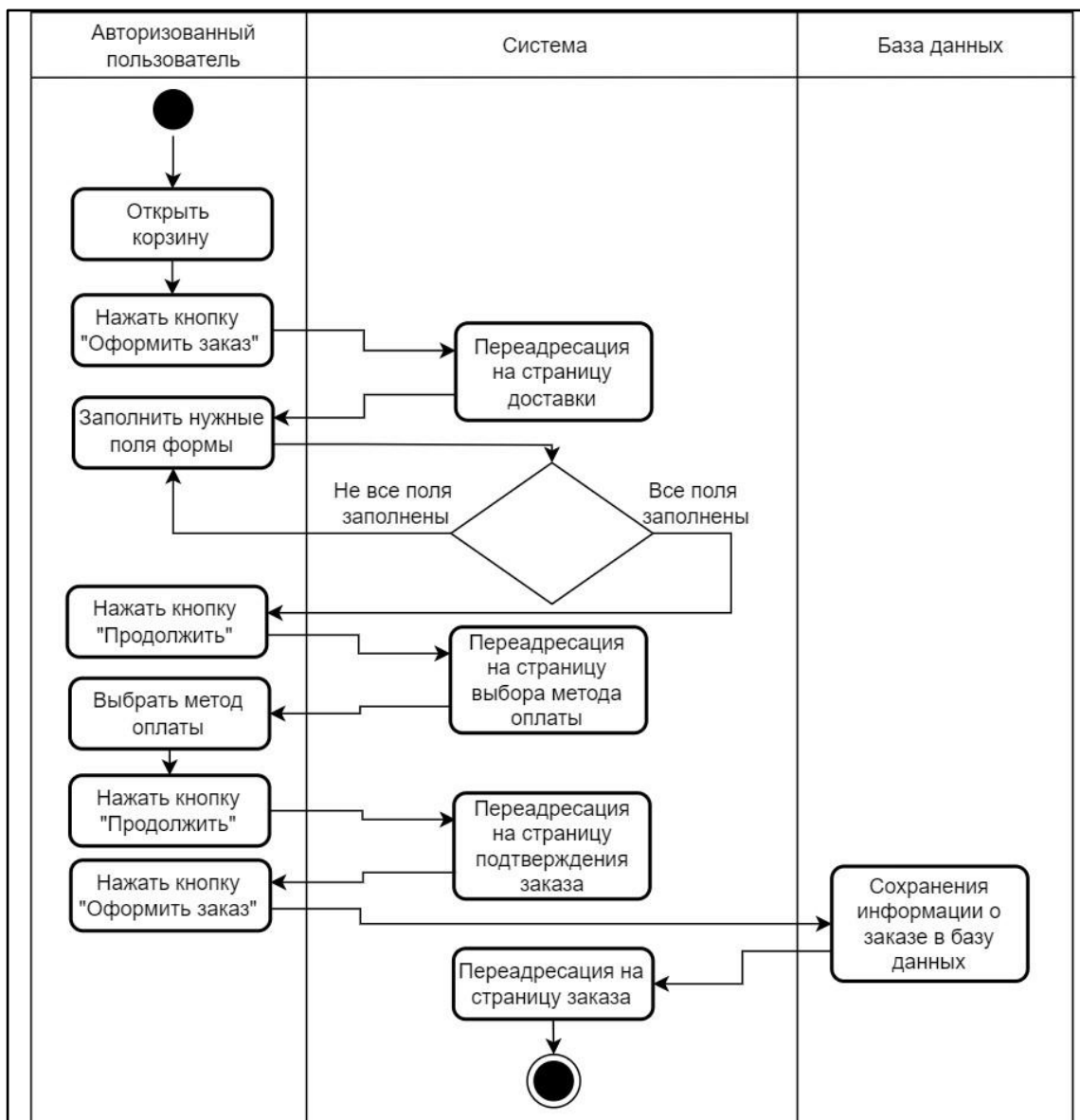


Рисунок 7 – Диаграмма деятельности

## 2.4. Архитектура системы

Для решения поставленной задачи по разработке интернет-магазина была спроектирована трехслойная архитектура. При работе веб-приложения интернет-магазина, пользователь взаимодействует с клиентским интерфейсом в своем браузере, просматривая страницы сайта. Когда пользователь отправляет запрос, сервер приложения обрабатывает данные, которые хранятся в базе данных. После обработки запроса сервер отправляет ответ веб-браузеру через сообщения с HTTP-ответом. Таким образом, диаграмма развертывания приложения учитывает распределение функций

между клиентским интерфейсом, сервером приложения и сервером базы данных для эффективной работы системы.

Исходя из этого была спроектирована диаграмма развертывания, которая представлена на рисунке 8.

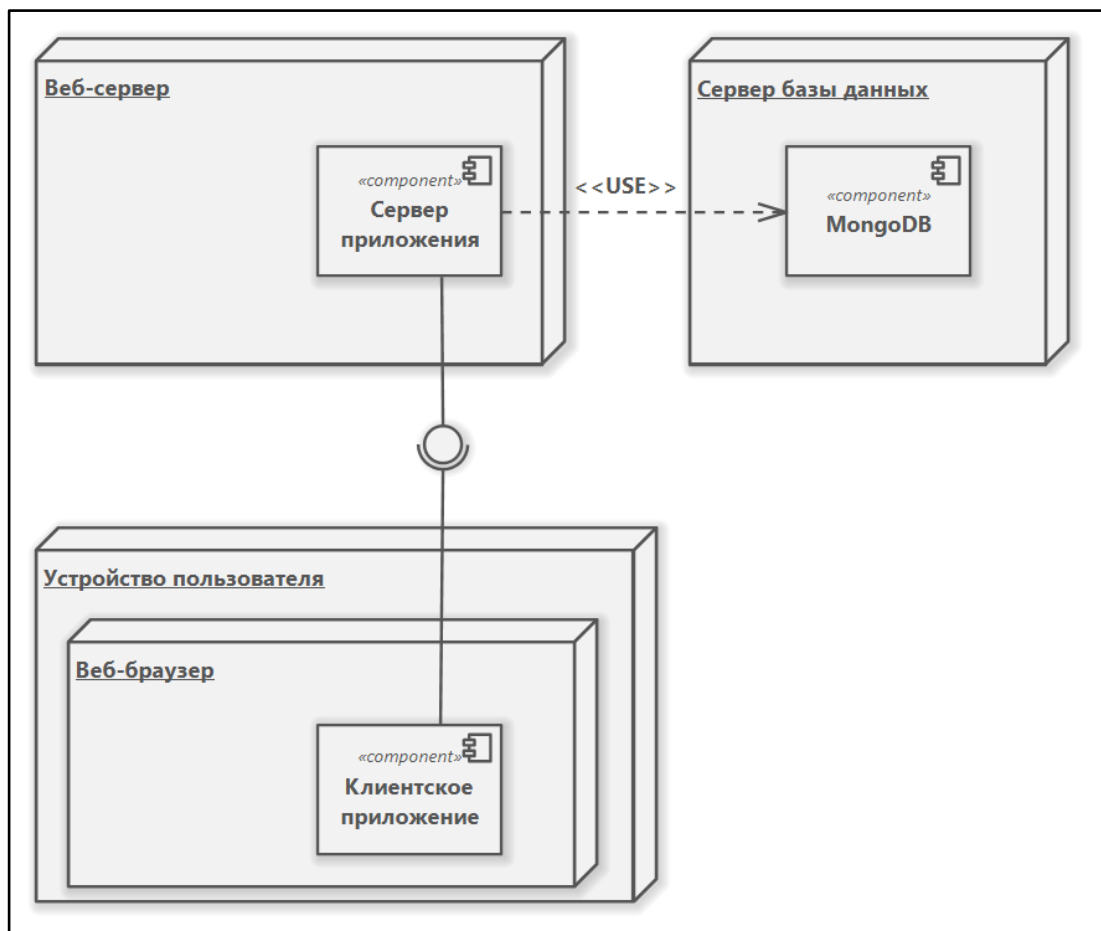


Рисунок 8 – Диаграмма развертывания

## 2.5. Схема базы данных

В качестве хранилища данных выбрана СУБД MongoDB, так как она является базой данных NoSQL, что означает, что она не требует строгой предварительной настройки. Это позволяет гибко изменять структуру данных в процессе разработки, что упрощает и ускоряет работу.

База данных должна хранить информацию обо всех пользователях, продуктах, заказах и отзывах о товарах.

Каждый продукт должен включать в себя уникальный идентификатор товара, название, изображение, бренд и категорию продукта. Модель

продукта хранит в себе id пользователя, создавшего этот продукт, а также описание, отзывы, рейтинг, количество отзывов, цену и количество товара в наличии.

Каждый заказ должен иметь уникальный идентификатор, пользователя, который совершил заказ, список товаров, адрес доставки, метод оплаты, результат оплаты, цену товара, цену доставки, общую стоимость товара. А также хранить информацию о том, оплачен ли заказ, доставлен ли заказ, и время оплаты или доставки соответственно.

Модель пользователя должна хранить уникальный идентификатор, имя, почту, пароль и поле проверки на администратора.

Модель отзыва должна хранить уникальный идентификатор, пользователя, который оставил отзыв, его имя, выставленную товару оценку и комментарий пользователя. Схема базы данных представлена на рисунке 9.

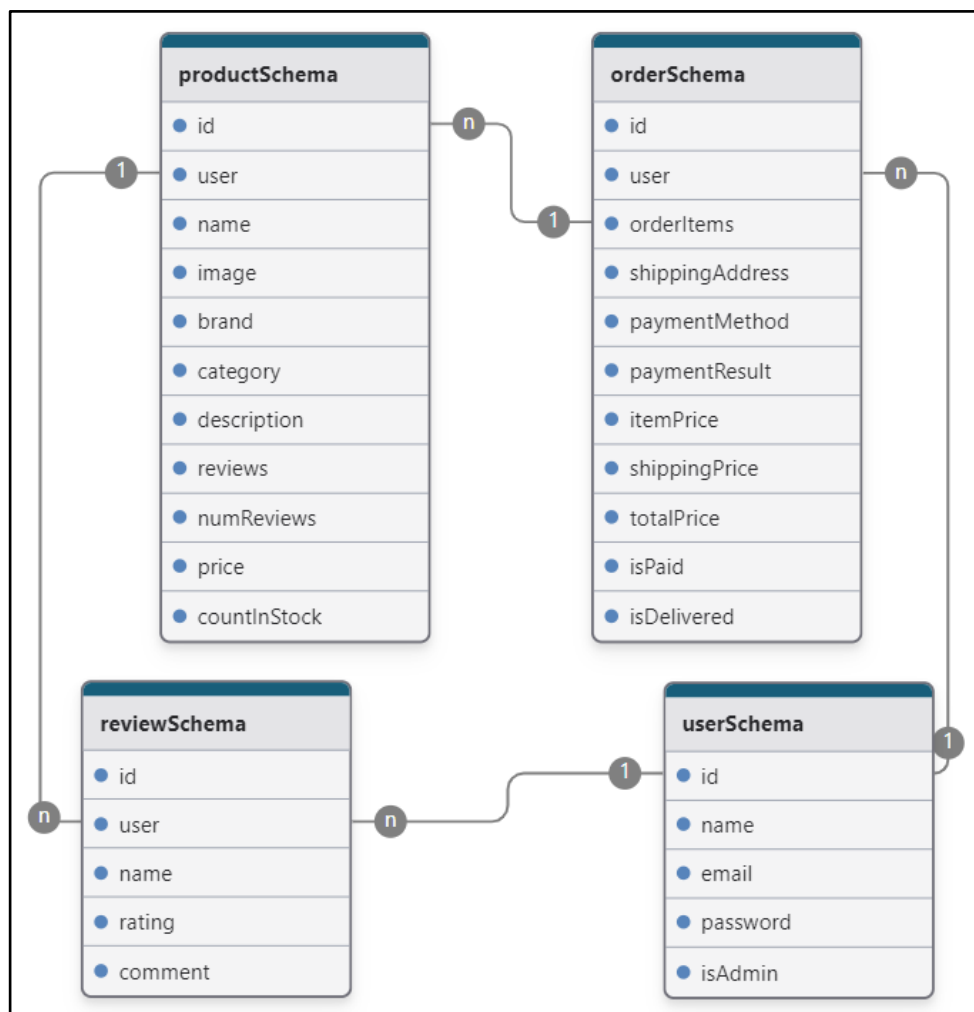


Рисунок 9 – Схема базы данных

## 2.6. Средства реализации

Для разработки веб-приложения интернет-магазина товаров для художника был выбран стек MERN (MongoDB, Express, React, Node.js). Необходимо реализовать две составляющие: клиентскую часть – это то, с чем взаимодействует пользователь напрямую в браузере, и серверную – это то, что будет обрабатывать информацию, полученную от пользователя на клиентской части.

В качестве редактора кода выбран Visual Studio Code [7] благодаря его функциональности, такой как выделение синтаксиса, автозаполнение кода, возможность установки различных расширений для повышения эффективности разработки и улучшения структуры кода.

Реализация клиентской части приложения будет реализована с использованием следующих технологий.

HTML (HyperText Markup Language) – это язык разметки, который используется для создания структуры веб-страниц. С помощью HTML определяются элементы на странице, такие как заголовки, параграфы, изображения, ссылки и другие. HTML используется для описания содержания веб-страницы [8].

CSS (Cascading Style Sheets) – это язык стилей, который используется для оформления веб-страниц. С помощью CSS можно задавать различные стили элементов HTML, такие как цвет текста, размер шрифта, отступы, рамки и другие аспекты внешнего вида страницы. CSS позволяет разделять содержимое и оформление страницы [9].

JavaScript – основной язык программирования интернет-ресурсов, который позволяет придавать веб-страницам дополнительную интерактивность. А также поддерживает объектно-ориентированные, императивные и функциональные стили [10].

React – это библиотека JavaScript для создания пользовательских интерфейсов. Он позволяет разрабатывать динамические и интерактивные

веб-приложения с использованием компонентного подхода. React также обеспечивает эффективное управление состоянием приложения [11].

Redux – это библиотека для управления состоянием приложения в JavaScript. Она предоставляет единое хранилище состояния, которое позволяет централизованно управлять данными приложения и обновлять интерфейс на основе этих данных [12].

Реализация серверной части приложения будет реализована с использованием следующих технологий.

MongoDB – это документоориентированная NoSQL база данных, которая использует формат JSON-подобных документов для хранения данных. Эта база данных отличается гибкостью и масштабируемостью, что делает ее популярным выбором для разработки веб-приложений. Она позволяет хранить данные без строгой схемы, что упрощает разработку и изменение структуры данных [13].

Express – это минималистичный и гибкий веб-фреймворк для Node.js, который упрощает создание веб-приложений и API. Express предоставляет удобные инструменты для работы с маршрутами, обработкой запросов и управлением сессиями.

Node.js – это среда выполнения JavaScript на стороне сервера, которая обеспечивает возможность использовать JavaScript как язык программирования как на стороне клиента, так и на стороне сервера [14].

### **Проектирование дизайна**

Для того, чтобы пользователю было приятно находиться на страницах интернет-магазина, сайт должен иметь качественный UX/UI дизайн. Для этого должны быть выбраны такие шрифты и цвета, чтобы прочтение текста не вызывало у пользователя трудностей. Компоненты сайта же должны быть расположены в интуитивно понятных местах.

С помощью сервиса Figma был разработан макет сайта, цветовое решение и логотип [15]. Для магазина товаров для художников стоит использовать яркие и вдохновляющие цвета, которые ассоциируются с творче-

ством и креативностью. Именно поэтому было принято решение сделать красочный градиент в шапке сайта. Логотип же должен отражать в себе как направленность интернет-магазина, так и ассоциироваться с определенной компанией. Макет главной страницы представлен на рисунке 10.

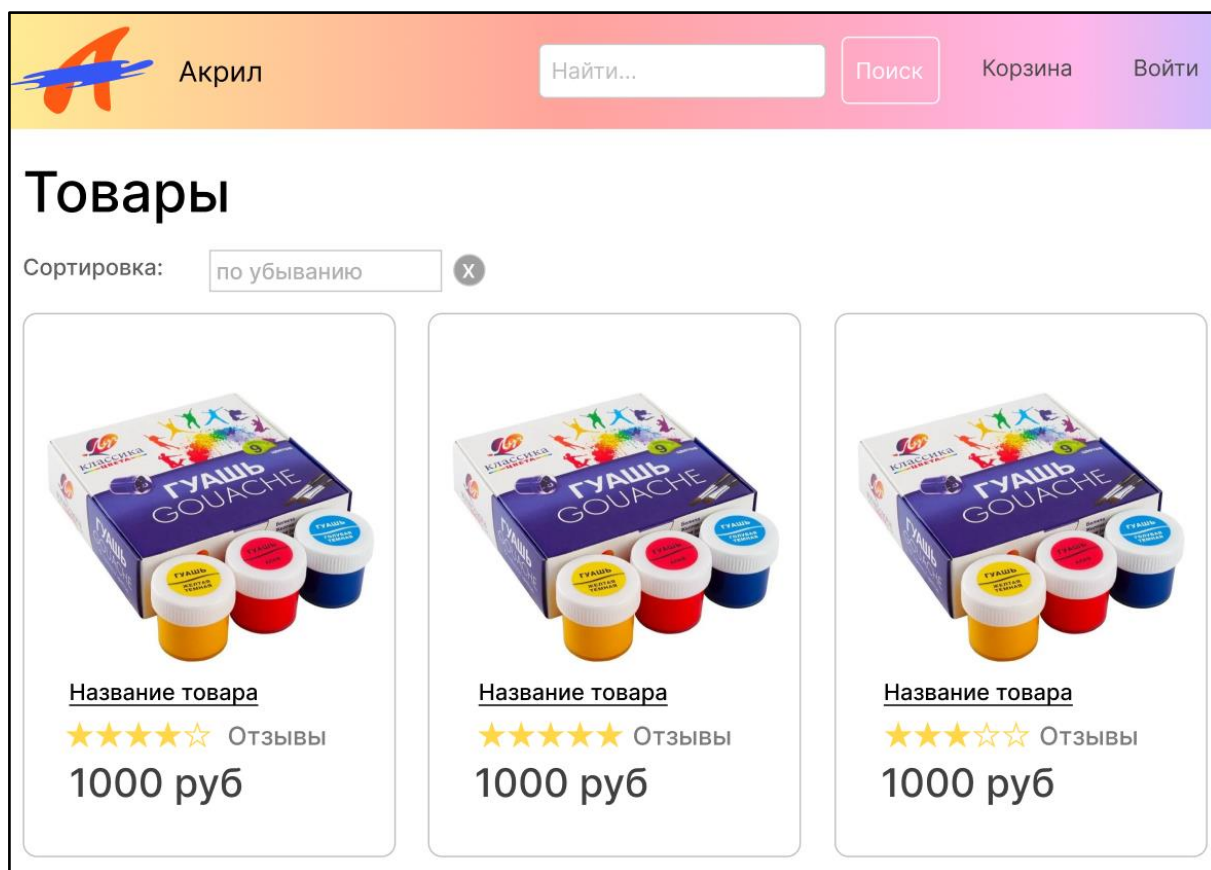


Рисунок 10 – Макет главной страницы

### Вывод по второй главе

В данной главе были проведены различные этапы проектирования системы, такие как определение функциональных и нефункциональных требований, разработка диаграмм вариантов использования, развертывания системы и схемы базы данных, а также были выбраны и описаны средства реализации клиентской и серверной частей веб-приложения и разработан макет.

## 3. РЕАЛИЗАЦИЯ СИСТЕМЫ

### 3.1. Реализация работы с базой данных

Для подключения к базе данных в проекте подключается библиотека `mongoose`, определяется асинхронная функция с именем `connectDB`, которая устанавливает соединение с базой данных `MongoDB`. Код для подключения к базе данных представлен в листинге 1.

#### Листинг 1 – Подключение к базе данных

```
import mongoose from 'mongoose';

const connectDB = async () => {
  try {
    const conn = await mongoose.connect(process.env.MONGO_URI)
    console.log(`Mongo DB connected: ${conn.connection.host}`);
  } catch (error) {
    console.log(`Error: ${error.message}`);
    process.exit(1);
  }
};

export default connectDB;
```

При успешном подключении базы данных выводится соответствующее сообщение. Если же во время процесса соединения возникает ошибка, ее обрабатывает блок `catch`, который записывает сообщение об ошибке и завершает работу приложения с кодом `1`, что означает ошибку во время запуска.

### 3.2. Реализация каталога товаров

#### Создание хранилища

В проекте используется библиотека `Redux`, которая позволит централизованно управлять состоянием приложения. Для начала создается глобальное хранилище в файле `store.js`. В нем будет храниться состояние. С помощью функции `configureStore()` комбинируем редьюсеры, которые отвечают за конкретную часть приложения.

Редьюсеры – это чистые функции в `Redux`, которые определяют, как состояние приложения изменяется в ответ на действия. Создание хранилища представлено в листинге 2.

## Листинг 2 – Создание хранилища

```
import { configureStore } from "@reduxjs/toolkit";
import { apiSlice } from "../slices/apiSlice";
import cartSliceReducer from '../slices/cartSlice';
import authSliceReducer from "../slices/authSlice";
const store = configureStore({
  reducer: {
    [apiSlice.reducerPath]: apiSlice.reducer,
    cart: cartSliceReducer,
    auth: authSliceReducer
  },
  middleware: (getDefaultMiddleware) => getDefaultMiddle-
ware().concat(apiSlice.middleware),
  devTools: true,
});
```

### Создание API Slice

Для реализации загрузки данных используются функции из библиотеки `@reduxjs/toolkit/query`, которые позволяют создавать API слайс. API слайс – это набор точек доступа для взаимодействия с внешним API. Он упрощает создание и управление запросами к серверу тем, что автоматически создает вебхуки для использования точек доступа в React-компонентах.

Для начала создаем пустой API Slice, от которого будут создаваться остальные для формирования запросов получения данных о заказах, продуктах и пользователях. Реализация API Slice представлена в листинге 3.

### Листинг 3 – Реализация Api Slice

```
import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";
import { BASE_URL } from '../constants'
const baseQuery = fetchBaseQuery({ baseUrl: BASE_URL })
export const apiSlice = createApi({
  baseQuery,
  tagTypes: ['Product', 'Order', 'User'],
  endpoints: (builder) => ({}))
```

При разработке каталога с продуктами был создан компонент `Product`, который отвечает за отрисовку карточки товара. В листинге 4 представлен компонент карточки товара.

### Листинг 4 – Компонент карточки товара

```
const Product = ({ product, addToCart }) => {
  return (
    <Card className="my-3 p-3 rounded">
      <Link to={`\product/${product._id}`}>
```



```

        <Card.Img src={product.image} variant="top" />
    </Link>
    <Card.Body>
        <Link to={` /product/${product._id}`}>
            <Card.Title as="div" className='product-title'>
                <strong>{product.name}</strong>
            </Card.Title>
        </Link>
        <Card.Text as='div'>
            <Rating
                value={ product.rating }
                text={`Отзывов: ${product.numReviews}`} />
        </Card.Text>
        <Card.Text as='h3'>
            {product.price} руб
        </Card.Text>
    </Card.Body>
</Card>
)
}

```

`FetchBaseQuery` создает базовую функцию для выполнения запросов к API. Она автоматически добавляет базовый URL к каждому запросу. Далее создаем API slice, в котором указываются базовый URL, массив строк, указывающий типы тегов, которые используются для автоматического управления кешем и функцию, которая принимает объект `builder` и возвращает объект с определенными эндпоинтами для выполнения запросов к API.

В данном случае объект пустой, так как этот API Slice будет использоваться в качестве основы для других слайсов и при помощи функции `injectEndpoints` будут добавляться эндпоинты для продуктов, заказов и пользователей. Для загрузки продуктов из базы данных используется созданный вебхук `useGetProductsQuery`.

После того, как данные были получены, они записываются в переменную `data`. Для того, чтобы трансформировать массив полученных значений в необходимый вид используется метод `map`, с его помощью каждый объект массива преобразуется в `Product`, созданный ранее.

Вид главной страницы с карточками продуктов представлен на рисунке 11.

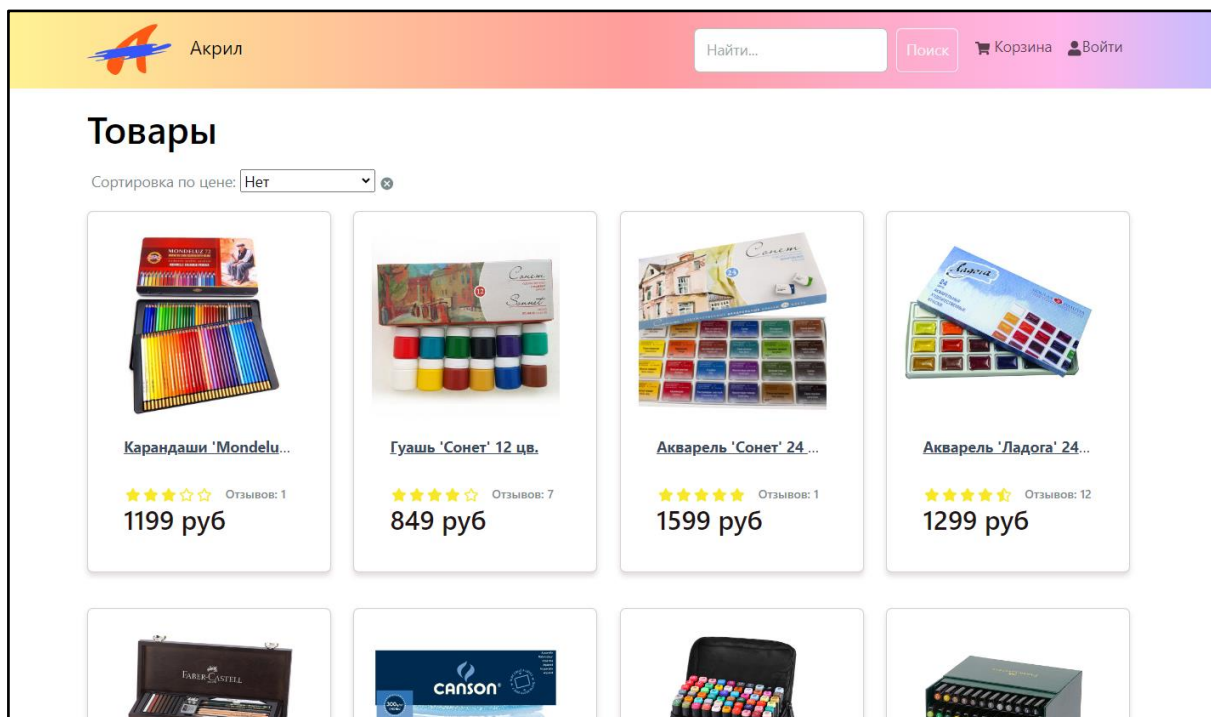


Рисунок 11 – Главная страница с карточками продуктов

### 3.3. Пагинация, сортировка и поиск товаров

Создание пагинации (постраничного вывода товаров), сортировки и поиска происходило в бэкенде в контроллере продуктов, который является асинхронной функцией для получения списка продуктов из базы данных. Код функции представлен в листинге 5.

#### Листинг 5 – Реализация пагинации в бэкенде

```
const getProducts = asyncHandler(async (req, res) => {
  const pageSize = 8;
  const page = Number(req.query.pageNumber) || 1;
  const keyword = req.query.keyword
    ? {name: { $regex: req.query.keyword, $options: 'i'}} : {}
  const count = await Product.countDocuments({...keyword});
  let products = req.query.sortType === 'none' ?
    await Product.find({...keyword}).limit(pageSize).skip(pageSize *
    (page - 1)) :
    await Product.find({...keyword}).limit(pageSize).skip(pageSize *
    (page - 1)).sort({price: req.query.sortType});
  res.json({products, page, pages: Math.ceil(count / pageSize)});
});
```

Во фронтенде пагинация и поиск представлены в компоненте `Paginate`. В зависимости от номера страницы и фразы для поиска проис-

ходит запрос к базе данных и перенаправление на URL с необходимыми параметрами. Код компонента представлен в листинге 6.

### Листинг 6 – Реализация пагинации и поиска во фронтенде

```
import { Pagination } from 'react-bootstrap'
import { LinkContainer } from 'react-router-bootstrap'
const Paginate = ({ pages, page, isAdmin = false, keyword = '' }) => {
  return (
    pages > 1 && (
      <Pagination>
        {[...Array(pages).keys()].map((x) => (
          <LinkContainer
            key={x + 1}
            to={!isAdmin ? keyword ? `/search/${keyword}/page/${x + 1}` : `/page/${x + 1}` : `/admin/productlist/${x + 1}`}>
            <Pagination.Item active={x + 1 === page}>{x + 1}</Pagination.Item>
          </LinkContainer>))}
        </Pagination>
      )
    )
  )
}
```

### Пагинация

При переходе на страницу с продуктами и отправлению запроса к базе данных в параметрах запроса указывается номер страницы. Для реализации пагинации используется функция `limit` и `skip`, которые ограничивают выборку данных, в зависимости от текущей страницы. Количество товаров на одной странице ограничено восемью штуками.

### Сортировка

На сайте есть возможность сортировки товаров по возрастанию и убыванию цены. Переключение типа сортировки по клику реализовано с помощью выпадающего списка с выбором сортировки. Все возможные типы сортировки хранятся в этом списке. После выбора типа сортировки пользователем происходит запрос и сервер возвращает данные в необходимом порядке.

### Поиск

Для поиска по товарам создано специальное поле ввода. После ввода фразы для поиска и клика по кнопке делается запрос на сервер и проверяется на соответствие названию товара. Результат поиска и сортировки показан на рисунке 12.

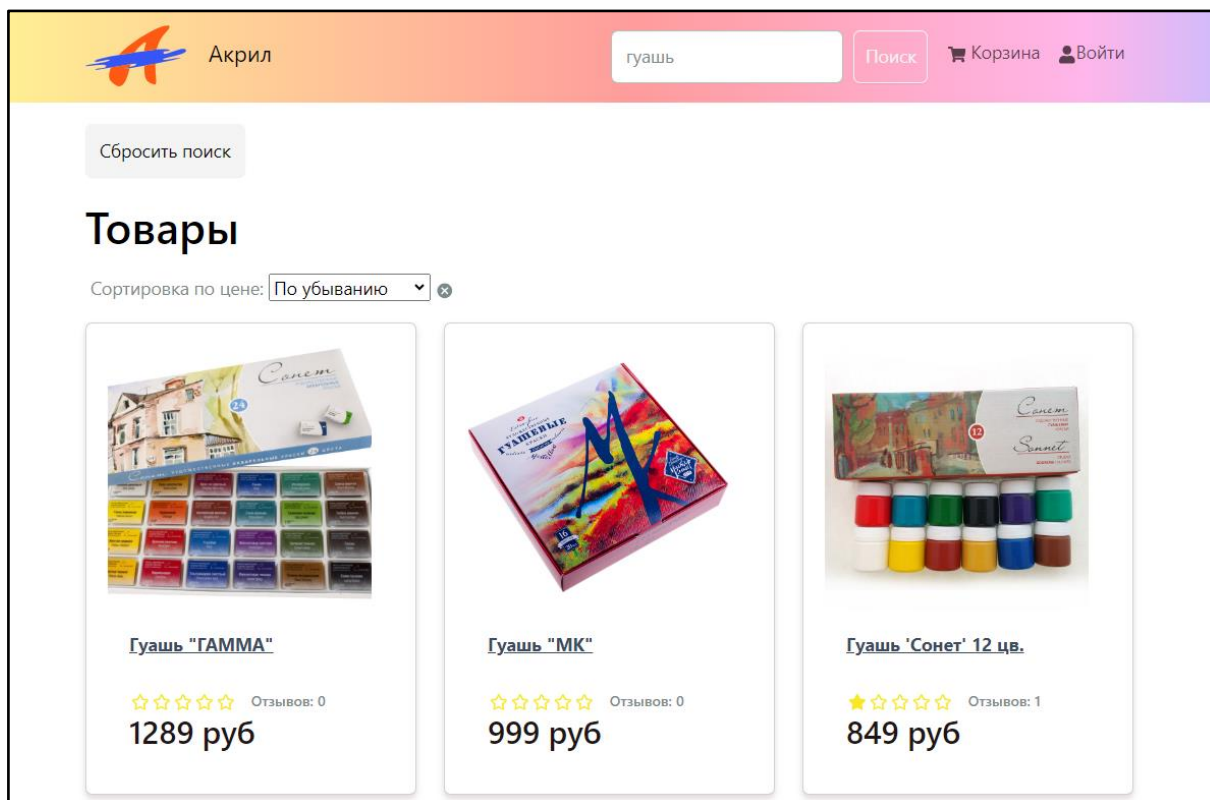


Рисунок 12 – Результат поиска и сортировки

### 3.4. Реализация корзины

Для разработки корзины был также создан отдельный слайс для хранения и управления состоянием и навигацией корзины. Компонент содержит несколько обработчиков событий для управления действиями пользователя. `AddToCartHandler` обновляет количество товара в корзине, `removeFromCartHandler` удаляет товар из корзины, а `checkoutHandler` перенаправляет пользователя на страницу входа для продолжения оформления заказа.

Если корзина пуста, отображается сообщение с предложением вернуться к покупкам. Если корзина не пуста, отображается список товаров с возможностью изменить количество или удалить товар.

Каждый товар в корзине отображается с его изображением, названием, ценой и количеством, которое можно изменить с помощью выпадающего списка. Также предусмотрена кнопка для удаления товара из корзины. В правой части экрана отображается информация о корзине: общее ко-

личество товаров и их общая стоимость. Кнопка "Оформить заказ" становится активной, если в корзине есть товары, и перенаправляет пользователя либо на страницу входа для продолжения процесса оформления заказа, если он не авторизован в системе, либо на страницу оформления. Скриншот корзины представлен на рисунке 13.

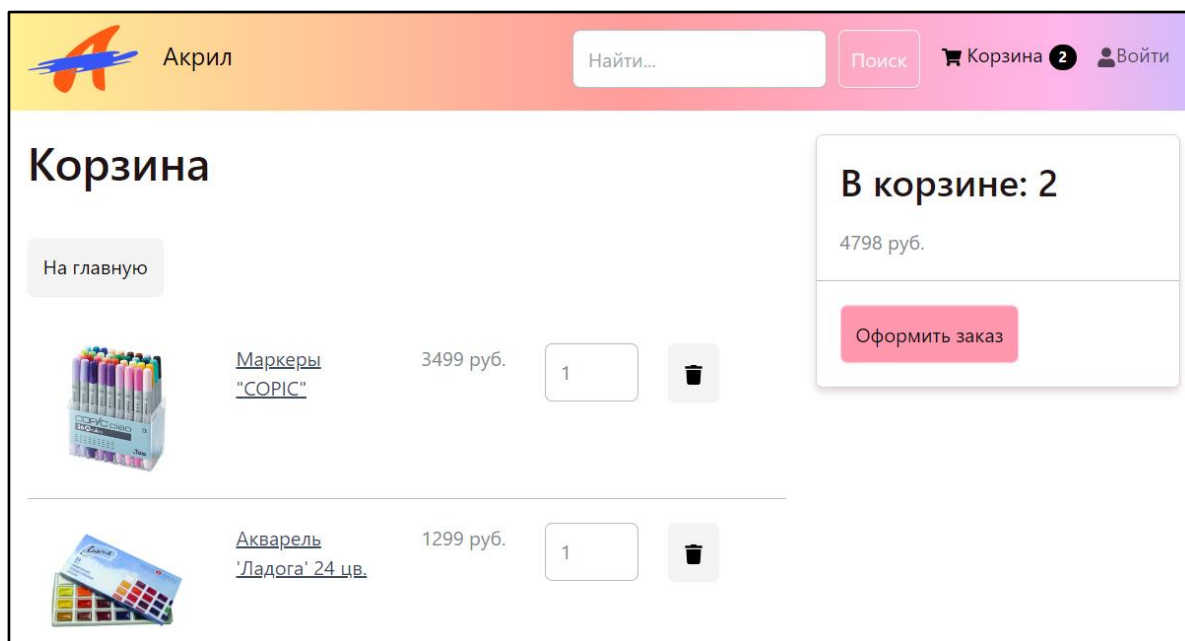


Рисунок 13 – Скриншот корзины

Чтобы содержимое корзины не пропадало после обновления страницы, было создано локальное хранилище (листинг 7), которое проверяет наличие сохраненных данных. Если пользователь добавляет товар в корзину, данные сохраняются в локальное хранилище, а при повторном посещении страницы корзина автоматически восстанавливается из него.

#### Листинг 7 – Сохранение данных в локальное хранилище

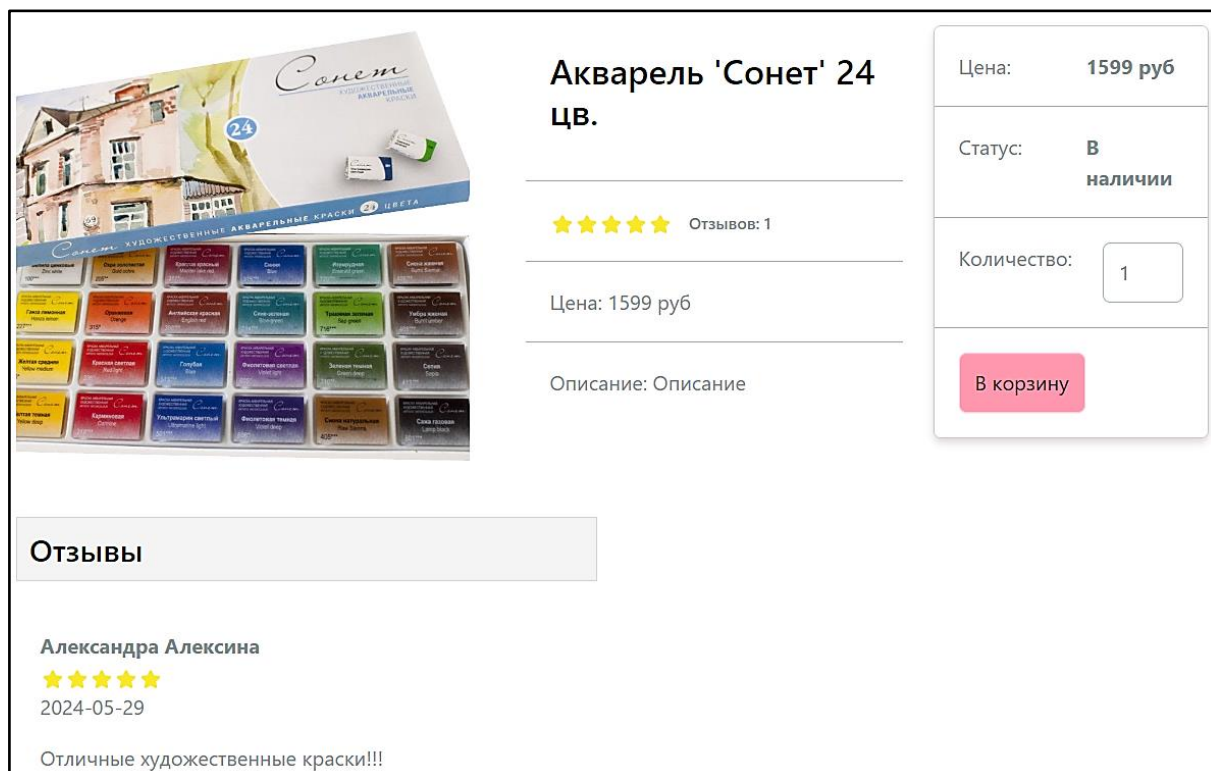
```
const initialState = localStorage.getItem("cart")
  ? JSON.parse(localStorage.getItem("cart"))
  : {cartItems: [], shippingAddress:{}, PaymentMethod: 'Card'};
```

### 3.5. Реализация страницы товара

Когда пользователь определился с выбором, кликнув на него, он может перейти на страницу товара, где хранится более подробная информация о нем. Чтобы получить необходимую информацию нужно сделать за-

прос на сервер по конкретному id. Реализация выполнена аналогично разработке каталога товаров.

Если данные о продукте успешно загружены, отображается информация о продукте, включая его изображение, название, рейтинг, цену и описание. Также показываются доступные отзывы и форма для написания нового отзыва, если пользователь аутентифицирован. Страница товара представлена на рисунке 14.



The screenshot shows a product page for 'Акварель 'Сонет' 24 ЦВ.'. On the left is a large image of the product box and a grid of 24 color swatches. To the right of the image, the product name 'Акварель 'Сонет' 24 ЦВ.' is displayed. Below the name is a 5-star rating and 'Отзывов: 1'. The price is listed as 'Цена: 1599 руб'. Below the price is a link for 'Описание: Описание'. On the far right, there is a summary box containing: 'Цена: 1599 руб', 'Статус: В наличии', 'Количество: 1' (with a text input field), and a pink 'В корзину' button. Below the product image is a section for 'Отзывы' (Reviews) with one review from 'Александра Алексина' dated '2024-05-29' with a 5-star rating and the text 'Отличные художественные краски!!!'.

Рисунок 14 – Страница товара

### 3.6. Реализация регистрации и авторизации

Для того, чтобы перейти к оформлению заказа, пользователь должен быть авторизован в системе. При обработке запроса на регистрацию пользователя отправляется POST-запрос к API-серверу, передавая в запросе данные пользователя, полученные из формы. Когда пользователь отправляет запрос на регистрацию, функция сначала извлекает имя, email и пароль из тела запроса. Затем она проверяет, существует ли уже пользователь с таким email в базе данных.

Метод `User.create` создает новую запись в базе данных, хэшируя пароль перед сохранением для обеспечения безопасности. Если создание пользователя прошло успешно, функция генерирует JWT-токен с помощью `generateToken`, который затем отправляется клиенту. Этот токен будет использоваться для последующей аутентификации пользователя в приложении. Реализация обработки запроса на регистрацию представлена в листинге 8.

### Листинг 8 – Реализация обработки запроса на регистрацию

```
const registerUser = asyncHandler(async (req, res) => {
  const { name, email, password } = req.body;
  const userExists = await User.findOne({ email });

  if (userExists) {
    res.status(400);
    throw new Error('Пользователь уже существует');}
  const user = await User.create({
    name,
    email,
    password});
  if(user) {
    generateToken(res, user._id);
    res.status(201).json({
      _id: user._id,
      name: user.name,
      email: user.email,
      isAdmin: user.isAdmin
    });
  } else {
    res.status(400);
    throw new Error('Неверные данные пользователя')
  }
});
```

Для реализации авторизации используется функция `authUser`, которая принимает запрос, извлекает из него учетные данные, такие как `email` и пароль, и проверяет их на корректность. Когда пользователь отправляет запрос на аутентификацию, функция сначала извлекает `email` и пароль из тела запроса, затем она пытается найти пользователя в базе данных по предоставленному `email`.

Если пользователь найден, следующая операция проверяет, соответствует ли предоставленный пароль сохраненному в базе данных паролю этого пользователя.

Для этого используется метод `matchPassword`, который сравнивает введенный пароль с хэшированным паролем, хранящимся в базе данных.

Реализация авторизации представлена в листинге 9.

### Листинг 9 – Реализация авторизации

```
const authUser = asyncHandler(async (req, res) => {
  const { email, password } = req.body;

  const user = await User.findOne({ email });

  if(user && (await user.matchPassword(password))) {
    generateToken(res, user._id);

    res.status(200).json({
      _id: user._id,
      name: user.name,
      email: user.email,
      isAdmin: user.isAdmin,
    });
  } else {
    res.status(401);
    throw new Error('Неверная почта или пароль');
  }
});
```

## 3.7. Реализация страницы оформления заказа

Для более удобной навигации по пунктам оформления заказа была реализована функция `CheckoutSteps`, которая позволяет пользователю отслеживать на какой стадии оформления он находится, и при необходимости, дает возможность вернуться на один из предыдущих шагов. Код данной функции с одним из шагов представлен в листинге 10.

### Листинг 10 – Функция `CheckoutSteps` с одним из шагов

```
const CheckoutSteps = ({ step1, step2, step3, step4 }) => {
  return (
    <div>
      <Nav className='justify-content-center mb-4'>
        <Nav.Item>
          { step1 ? (
            <LinkContainer to='/login'>
              <Nav.Link>Вход</Nav.Link>
            </LinkContainer>
          ) : (
            <Nav.Link disabled>Вход</Nav.Link>
          )}
        </Nav.Item>
      </Nav>
    </div>
  )
}
```



После того, как пользователь заполнил все необходимые поля, на финальном шаге оформления заказа выводится вся информация для проверки, общая стоимость заказа, все выбранные товары и кнопка оформления заказа. Страница оформления заказа представлена на рисунке 15.

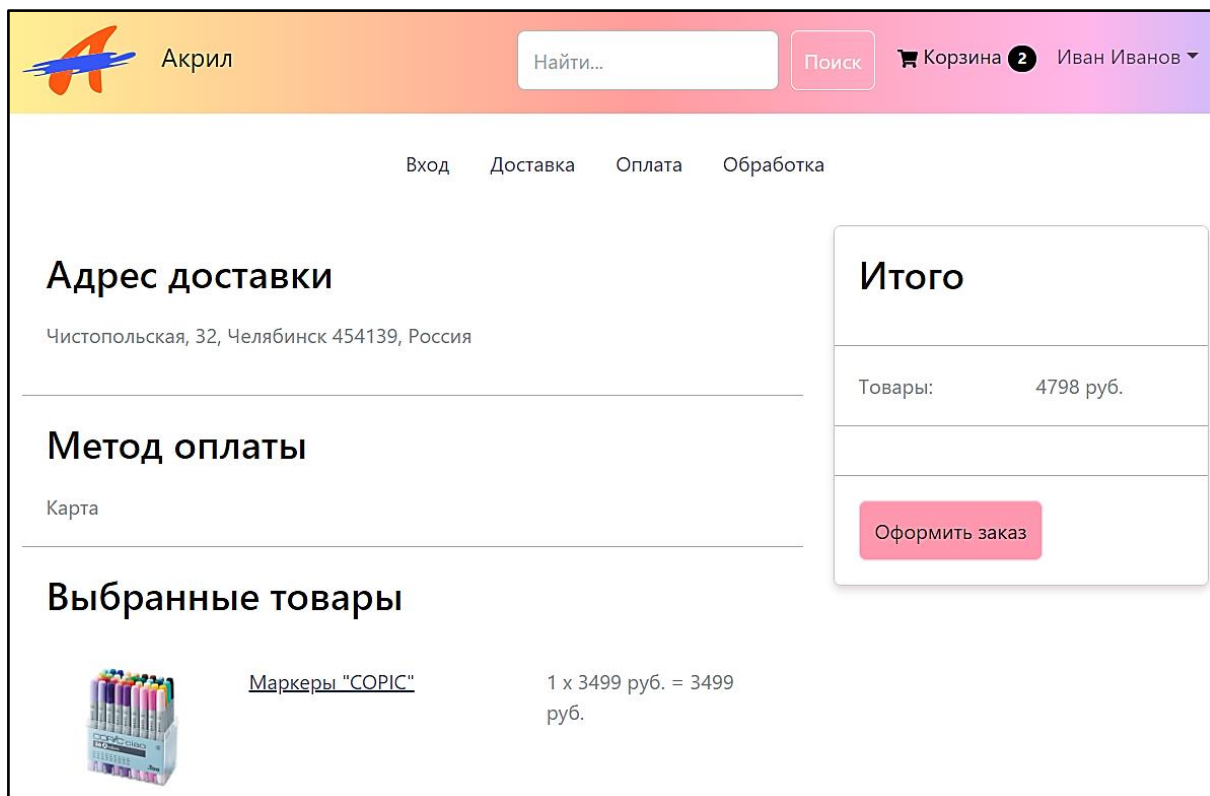


Рисунок 15 – Страница оформления заказа

### 3.8. Реализация страницы отслеживания заказа

Компонент страницы заказа начинается с извлечения идентификатора заказа, затем выполняет запрос деталей заказа с использованием `useGetOrderDetailsQuery`. Если заказ успешно получен, сохраняются данные заказа, а также состояния загрузки и ошибки.

В данный момент сайт не работает с реальной компанией, поэтому при нажатии на кнопку Оплаты срабатывает тестовая операция с сохранением информации о заказе в базу данных, без подключенного банка, и обновляет статус оплаты. При необходимости можно подключить вывод информации на сторонний ресурс. Скриншот страницы заказа представлен на рисунке 16.

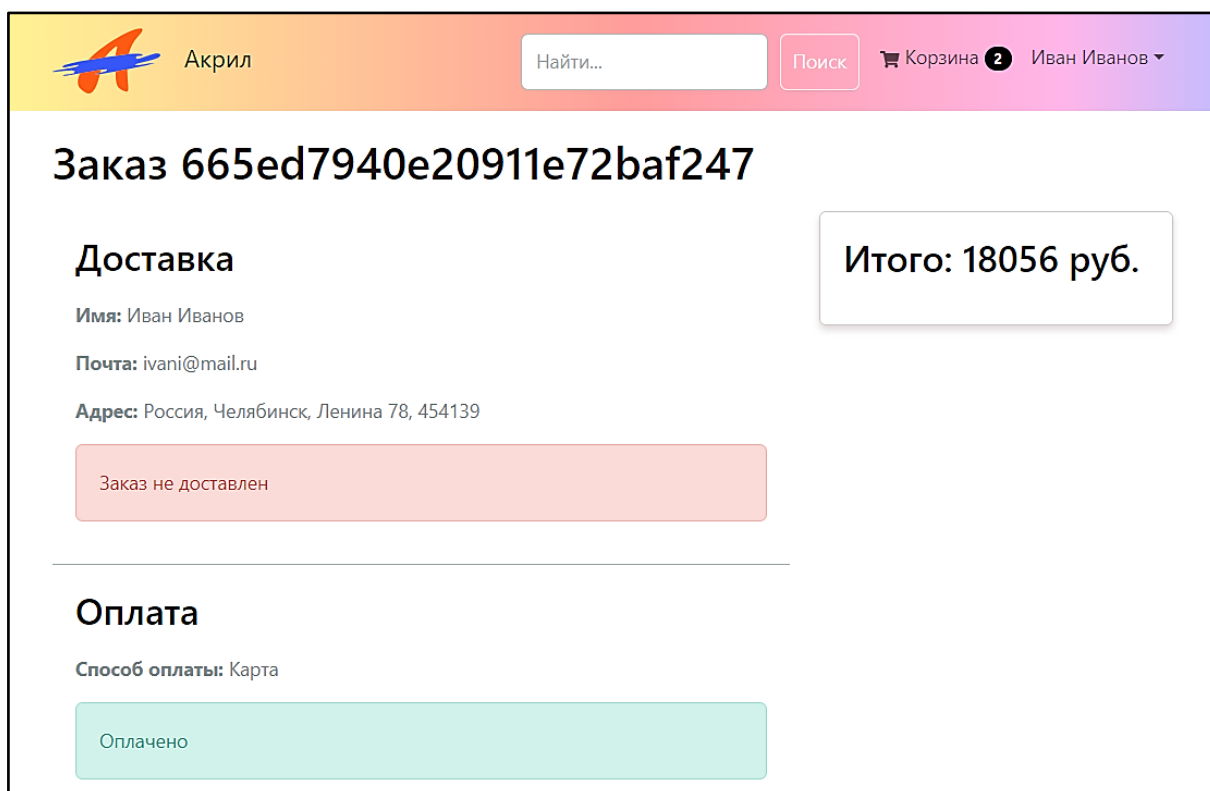


Рисунок 16 – Страница отслеживания заказа

### 3.9. Панель администратора

Для управления содержимым сайта была реализована панель администратора. Авторизовавшись с помощью логина и пароля, администратор получает доступ к администраторскому меню, где сосредоточены все функции панели.

#### Управление товарами и пользователями

С помощью соответствующего пункта меню администратор может просмотреть список всех товаров, включая такие данные как id товара, его название, стоимость, категория и бренд, или же добавить новый товар. При добавлении нового товара, продукт просто добавляется в список с базовыми данными.

Напротив каждого товара также расположены кнопки добавления и изменения товара. При нажатии на иконку редактирования, администратор переходит в меню редактирования товара, где имеет возможность исправить необходимые поля, изменить фотографию и сохранить изменения с

помощью кнопки «Сохранить». Страница полей для изменения товара администратором представлена на рисунке 17.

**Изменить товар**

Название  
Карандаши 'Mondeluz' 72 цв.

Стоимость  
1199

Изображение  
/uploads/image-1716998919081.jpg  
Выберите файл | Файл не выбран

Бренд  
Mondeluz

Наличие товара  
11

Категория  
Карандаши

Описание  
Описание

Сохранить

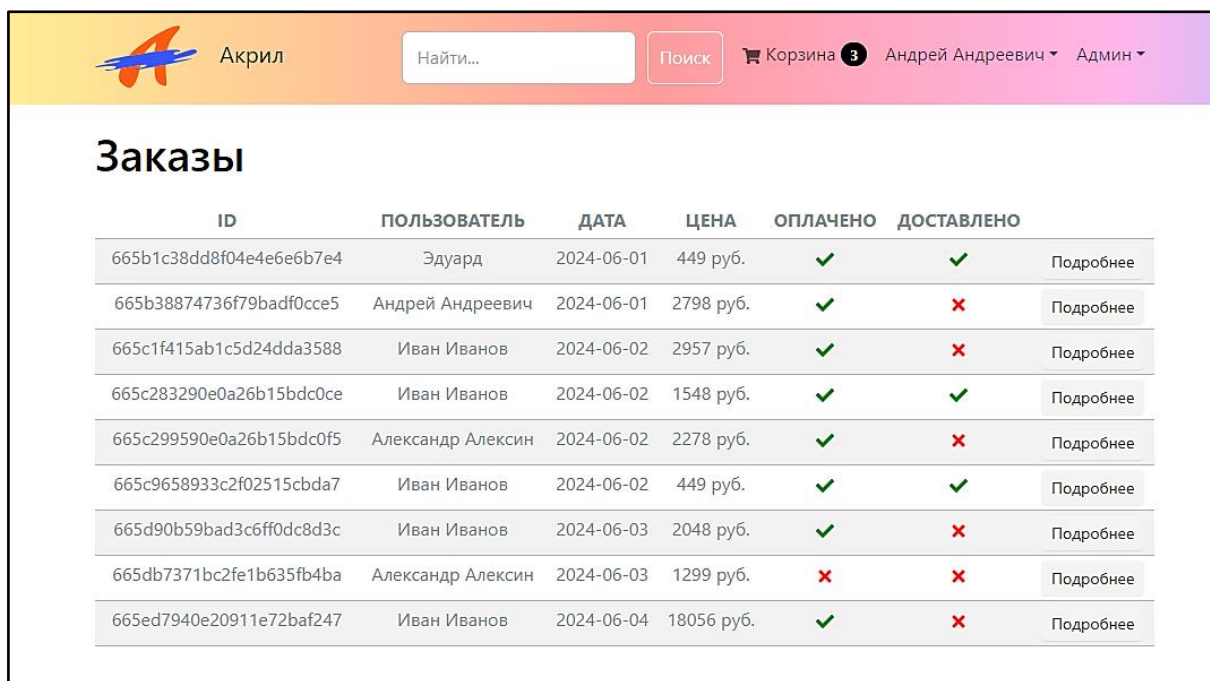
Рисунок 17 – Поля изменения товара администратором

Аналогично товарам, администратор также может просмотреть список существующих пользователей, удалить их при необходимости или же внести изменения в их профили. При изменении информации о пользователе администратор может назначать и снимать права администратора.

### **Управление заказами**

С помощью меню администратора можно просмотреть список заказов и отследить, какие из них оплачены и доставлены. Также есть возможность перейти на страницу самого заказа, где открывается возможность

провести оплату при получении товара или же пометить товар как доставленный. Страница со списком заказов представлена на рисунке 18.



ID	ПОЛЬЗОВАТЕЛЬ	ДАТА	ЦЕНА	ОПЛАЧЕНО	ДОСТАВЛЕНО	
665b1c38dd8f04e4e6e6b7e4	Эдуард	2024-06-01	449 руб.	✓	✓	<a href="#">Подробнее</a>
665b38874736f79badf0cce5	Андрей Андреевич	2024-06-01	2798 руб.	✓	✗	<a href="#">Подробнее</a>
665c1f415ab1c5d24dda3588	Иван Иванов	2024-06-02	2957 руб.	✓	✗	<a href="#">Подробнее</a>
665c283290e0a26b15bdc0ce	Иван Иванов	2024-06-02	1548 руб.	✓	✓	<a href="#">Подробнее</a>
665c299590e0a26b15bdc0f5	Александр Алексин	2024-06-02	2278 руб.	✓	✗	<a href="#">Подробнее</a>
665c9658933c2f02515cbda7	Иван Иванов	2024-06-02	449 руб.	✓	✓	<a href="#">Подробнее</a>
665d90b59bad3c6ff0dc8d3c	Иван Иванов	2024-06-03	2048 руб.	✓	✗	<a href="#">Подробнее</a>
665db7371bc2fe1b635fb4ba	Александр Алексин	2024-06-03	1299 руб.	✗	✗	<a href="#">Подробнее</a>
665ed7940e20911e72baf247	Иван Иванов	2024-06-04	18056 руб.	✓	✗	<a href="#">Подробнее</a>

Рисунок 18 – Страница со списком заказов

### Адаптивная верстка

Адаптивность на сайте реализована при помощи библиотеки Bootstrap и CSS-свойств [16]. При использовании медиа-запросов контент на сайте перестраивается, когда ширина экрана достигает определенной точки. Один из медиа-запросов представлен в листинге 11. Версии сайта для различных типов и размеров экранов представлены в приложении Б.

Листинг 11 – Медиа-запрос для экранов не более 768 пикселей

```
@media (min-width: 768px) {  
  .container-md,  
  .container-sm,  
  .container {  
    max-width: 720px;  
  }  
}
```

### Вывод по третьей главе

В данной главе были описаны ключевые этапы создания системы. Были реализованы и описаны необходимые компоненты, создание базы данных, а также панель администратора.

## 4. ТЕСТИРОВАНИЕ

### Функциональное тестирование

Функциональное тестирование является одним из ключевых этапов в процессе разработки программного обеспечения. Оно направлено на проверку работоспособности системы в соответствии с ее функциональными требованиями. Функциональные тесты системы представлены в таблице 1.

Таблица 1 – Функциональное тестирование системы

№	Название теста	Действие	Ожидаемый результат	Тест пройден?
1	Проверка пагинации	Находясь в каталоге товаров нажать на номер нужной страницы	Отображается группа товаров в соответствии с номером выбранной страницы	Да
2	Проверка сортировки товаров	Нажать на текущий тип сортировки и в выпадающем списке выбрать нужный вид сортировки	Товары отображаются в соответствии с выбранным типом сортировки	Да
3	Проверка поиска товаров	Ввести в поисковую строку ключевое слово, по которому будет производиться поиск	Отображаются все товары, соответствующие введенному слову	Да
4	Проверка перехода на страницу товара	Находясь в каталоге нажать на нужный товар	Отображается страница товара с информацией о нем	Да
5	Проверка добавления товара в корзину	Находясь на странице товара нажать на кнопку «Добавить в корзину»	Товар добавляется в корзину и открывается соответствующая страница	Да
6	Проверка изменения количества товара в корзине	Находясь в корзине нажать на поле с количеством товара и выбрать из списка подходящее количество	Количество товара меняется, также как меняется и общая сумма заказа	Да
7	Проверка удаления товара из корзины	Находясь в корзине нажать на иконку корзины у необходимого для удаления товара	Товар удаляется из корзины, и его цена вычитается из общей суммы заказа	Да
8	Переход к оформлению заказа	Находясь в корзине нажать на кнопку «Оформить заказ»	Открывается страница ввода данных пользователя	Да

Продолжение таблицы 1

№	Название теста	Действие	Ожидаемый результат	Тест пройден?
9	Проверка валидации формы	Заполнить поля при оформлении заказа	Если все данные заполнены корректно, то кнопка для продолжения оформления доступна. Если поля заполнены некорректно, появляется сообщение и кнопка не доступна	Да
10	Добавление отзыва на товар	Находясь на странице товара, поставить желаемое количество звезд в качестве оценки, написать комментарий и нажать кнопку «Отправить»	Отзыв добавляется к товару и изменяется среднее количество звезд товара	Да
11	Проверка редактирования профиля	Находясь на странице профиля, ввести данные для изменения в соответствующие поля и нажать кнопку «Сохранить»	Данные пользователя обновляются и выводится сообщение об успешном изменении	Да
12	Проверка отслеживания заказа	Находясь на вкладке профиля нажать кнопку «Подробнее» около нужного заказа	Открывается окно заказа и информация о его состоянии	Да
13	Проверка авторизации администратора	При авторизации введены данные пользователя, у которого есть доступ к функциям администратора	Появляется дополнительная кнопка в хэдере	Да
14	Добавление товара администратором	Находясь на вкладке «Товары» в меню администратора, нажать кнопку «Добавить товар»	Появится сообщение требующее подтверждение для создания нового товара. При согласии будет создан товар с базовыми полями.	Да
15	Редактирование товара	Находясь на вкладке товаров в меню администратора нажать на иконку изменения, ввести данные и нажать «Сохранить»	Произойдет замена данных товара на введенные и сохранится в базу данных	Да

№	Название теста	Действие	Ожидаемый результат	Тест пройден?
16	Добавление картинки в карточку товара администратором	Находясь в окне изменения товара администратором, нажать кнопку «Выберите файл» и выбрать файл из хранилища	Фотография добавилась в проект и при сохранении изменений изменилась в карточке товара	Да
17	Удаление товара администратором	Находясь в окне товаров администратора нажать на иконку корзины соответствующего товара	Появилось сообщение об изменении. При согласии товар будет удален	Да
18	Изменение статуса товара администратором	Находясь в окне заказов администратора нажать кнопку «Подробнее», а далее на кнопку «Пометить доставленным»	Состояние доставки изменится на «Доставлено»	Да
19	Изменение данных пользователя	Находясь на вкладке пользователей с аккаунта администратора нажать на иконку изменения соответствующего пользователя, изменить данные и нажать кнопку «Сохранить»	Данные пользователя заменятся на указанные	Да

### Тестирование верстки

При тестировании верстки сайт был открыт с разных браузеров в разных оконных размерах. Для тестирования использовались такие браузеры, как Google Chrome (версия: 125.0.6422.141), Opera (версия: 110.0.5130.49) и Яндекс Браузер (версия: 24.4.4.1168). Интерфейс пользователя во всех перечисленных браузерах остается идентичным, весь контент остается на своих местах. При увеличении и уменьшении размера экрана контент адаптируется под устройство. Тесты успешно пройдены.

### Вывод по четвертой главе

В результате проведения тестирования установлено, что разработанный веб-сайт соответствует исходным функциональным требованиям.

## **ЗАКЛЮЧЕНИЕ**

Целью данной выпускной квалификационной работы бакалавра являлась разработка интернет-магазина товаров для художников.

В ходе выполнения работы были выполнены следующие задачи.

1. Произведен анализ предметной области и выполнен обзор существующих аналогов.
2. Спроектирован интернет-магазин.
3. Реализован интернет-магазин.
4. Проведено тестирование интернет-магазина.

В рамках данной выпускной квалификационной работы был реализован интернет-магазин товаров для художников с использованием стека MERN, а также библиотеки Bootstrap для адаптивной верстки.

Все поставленные задачи были успешно решены, таким образом, цель данной работы выполнена.

В дальнейшем планируется расширение функционала данной системы, добавление новых разделов, добавление и расширение функционала личного кабинета, главной страницы и страницы товара, а также при необходимости добавление подключения выбранного банка.



## ЛИТЕРАТУРА

1. Etudesite. [Электронный ресурс] URL: [http:// www.etudesite.ru](http://www.etudesite.ru) (дата обращения: 10.02.2024 г.).
2. Почерк. [Электронный ресурс] URL: <https://pocherkartshop.ru> (дата обращения: 10.02.2024 г.).
3. Малевичъ. [Электронный ресурс] URL: <https://art-malevich.ru> (дата обращения: 10.02.2024 г.).
4. Леонардо. [Электронный ресурс] URL: <https://leonardo.ru> (дата обращения: 10.02.2024 г.).
5. Красный карандаш. [Электронный ресурс] URL: <https://krasniykarandash.ru> (дата обращения: 10.02.2024 г.).
6. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. – 2-е изд. ДМК Пресс, 2006. – С. 47–49.
7. Редактор кода VSCode. [Электронные ресурс] URL: <https://code.visualstudio.com> (дата обращения: 14.02.2024 г.).
8. Фримен Э., Фримен Э. Изучаем HTML, XHTML и CSS. 2-е изд. – СПб.: Питер, 2014. – С. 36–40.
9. Диков А.В. Клиентские технологии веб-дизайна. HTML5 и CSS3: учебное пособие. // Санкт-Петербург: Лань, 2022. – 188 с.
10. Дакетт Д. JavaScript и jQuery. Интерактивная веб-разработка. – Москва: Издательство «Э», 2018. – С. 1–23.
11. React documentation. [Электронный ресурс] URL: <https://www.postgresql.org/> (дата обращения: 15.02.2023 г.).
12. Бэнкс А., Порселло Е. React и Redux: функциональная веб-разработка. – Санкт-Петербург: Издательство «Питер», 2018. – С. 190-200.
13. MongoDB [Электронный ресурс] URL: [mongodb.com](https://www.mongodb.com) (дата обращения: 15.02.2024 г.).
14. Node.js [Электронный ресурс] URL: [nodejs.org](https://nodejs.org) (дата обращения: 18.02.2024 г.).

15. Figma. [Электронный ресурс] URL: <https://www.figma.com> (дата обращения: 02.04.2024 г.).

16. Морето С. Bootstrap в примерах. – Москва: ДМК Пресс, 2017. – С. 1–314.

## ПРИЛОЖЕНИЯ

### Приложение А. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) системы приведена в таблицах 1–12.

Таблица 1 – Спецификация ВИ «Регистрация»

Прецедент: Регистрация
ID: 1
Краткое описание: Пользователь регистрирует свой аккаунт в базе данных системы.
Главные актеры: Неавторизованный пользователь.
Предусловия: Пользователь заходит на главную страницу сайта не имея аккаунта.
Основной поток: 1. Пользователь нажимает на кнопку «Войти» в шапке сайта. 2. Пользователь нажимает кнопку «Зарегистрироваться» и переходит на страницу регистрации. 3. Пользователь вводит свои данные и отправляет их с помощью кнопки «Зарегистрироваться», соглашаясь с обработкой персональных данных.
Постусловия: Аккаунт пользователя сохраняется в базе данных.
Альтернативные потоки: Нет.

Таблица 2 – Спецификация ВИ «Авторизация»

Прецедент: Авторизация
ID: 2
Краткое описание: Пользователь авторизуется в системе.
Главные актеры: Неавторизованный пользователь.
Предусловия: Пользователь заходит на главную страницу.
Основной поток: 1. Пользователь нажимает на кнопку «Войти» в шапке сайта. 2. Пользователь вводит свои данные для входа в систему. 3. Пользователь нажимает на кнопку «Войти»
Постусловия: Система проверяет данные, введенные пользователем, если данные совпали с данными в базе данных, то авторизация завершается успешно и пользователь входит в свой аккаунт либо как обычный пользователь, либо как администратор.
Альтернативные потоки: Нет.

Таблица 3 – Спецификация ВИ «Просмотр товара»

Прецедент: Просмотр товара
ID: 3
Краткое описание: Пользователь просматривает страницу товара.
Главные актеры: Авторизованный пользователь.
Предусловия: Пользователь заходит на главную страницу сайта.
Основной поток: Пользователь выбирает товар и нажимает на его название.
Постусловия: Пользователь переходит на страницу товара.
Альтернативные потоки: Нет.

Таблица 4 – Спецификация ВИ «Добавление товара в корзину»

Прецедент: Добавление товара в корзину
ID: 4
Краткое описание: Пользователь добавляет товар в корзину.
Главные актеры: Авторизованный пользователь.
Предусловия: Пользователь заходит на страницу товара.
Основной поток: Пользователь нажимает на кнопку «Добавить в корзину»
Постусловия: Товар добавляется в корзину, и пользователь переходит на страницу корзины.
Альтернативные потоки: Нет.

Таблица 5 – Спецификация ВИ «Изменение количества товара»

Прецедент: Изменение количества товара
ID: 5
Краткое описание: Пользователь изменяет количество товара в корзине.
Главные актеры: Авторизованный пользователь.
Предусловия: Пользователь находится на странице корзины.
Основной поток: Пользователь нажимает на поле с количеством товара и выбирает нужное ему число.
Постусловия: Количество товара в корзине изменяется на число выбранное пользователем.
Альтернативные потоки: Нет.

Таблица 6 – Спецификация ВИ «Удаление товара»

Прецедент: Удаление товара
ID: 6
Краткое описание: Пользователь удаляет товар из корзины.
Главные актеры: Авторизованный пользователь.
Предусловия: Пользователь заходит на страницу корзины.
Основной поток: Пользователь нажимает на иконку корзины у выбранного товара.
Постусловия: Товар удаляется из корзины.
Альтернативные потоки: Нет.

Таблица 7 – Спецификация ВИ «Оформление заказа»

Прецедент: Оформление заказа
ID: 7
Краткое описание: Пользователь оформляет заказ.
Главные актеры: Авторизованный пользователь.
Предусловия: Пользователь заходит на страницу корзины.
Основной поток: 1. Пользователь нажимает кнопку «Оформить заказ». 2. Пользователь вводит все необходимые данные для оформления заказа. 3. Пользователь заканчивает все шаги для оформления заказа и нажимает кнопку «Оформить заказ»
Постусловия: Информация о заказе сохраняется в базу данных, и пользователь попадает на страницу заказа.
Альтернативные потоки: Нет.

Таблица 8 – Спецификация ВИ «Редактировать товар»

Прецедент: Редактировать товар
ID: 8
Краткое описание: Администратор редактирует существующий на сайте товар.
Главные актеры: Администратор.
Предусловия: Администратор заходит на страницу со списком товаров.

Основной поток: 1. Администратор нажимает на иконку изменения товара. 2. Администратор вводит необходимые изменения в соответствующие поля 3. Администратор нажимает кнопку «Сохранить»
Постусловия: Товар изменяется.
Альтернативные потоки: Нет.

Таблица 9 – Спецификация ВИ «Добавить товар»

Прецедент: Добавить товар
ID: 9
Краткое описание: Администратор добавляет товар на сайт.
Главные актеры: Администратор.
Предусловия: Администратор заходит на страницу со списком товаров.
Основной поток: Администратор нажимает на кнопку «Добавить товар».
Постусловия: Товар добавляется на сайт.
Альтернативные потоки: Нет.

Таблица 10 – Спецификация ВИ «Удалить товар»

Прецедент: Удалить товар
ID: 10
Краткое описание: Администратор удаляет товар с сайта.
Главные актеры: Администратор.
Предусловия: Администратор заходит на страницу со списком товаров.
Основной поток: Администратор нажимает на иконку удаления товара.
Постусловия: Товар удаляется с сайта.
Альтернативные потоки: Нет.

Таблица 11 – Спецификация ВИ «Редактировать пользователя»

Прецедент: Редактировать пользователя
ID: 11
Краткое описание: Администратор изменяет данные существующего пользователя.
Главные актеры: Администратор.
Предусловия: Администратор заходит на страницу со списком пользователей.
Основной поток: 1. Администратор нажимает на иконку изменения рядом с нужным пользователем. 2. Администратор вводит необходимые данные для изменения. 3. Администратор нажимает кнопку «Сохранить».
Постусловия: Пользователь изменен.
Альтернативные потоки: Нет.

Таблица 12 – Спецификация ВИ «Обработка заказа»

Прецедент: Обработка заказа
ID: 12
Краткое описание: Администратор обрабатывает заказ.
Главные актеры: Администратор.
Предусловия: Администратор заходит на страницу со списком заказов и нажимает кнопку «Подробнее» рядом с нужным ему заказом.
Основной поток: 1. Администратор нажимает на кнопку «Оплата» в случае оплаты при получении заказа. 2. Администратор нажимает на кнопку «Пометить доставленным» в случае доставки заказа.
Постусловия: Статус заказа меняется в соответствии с действиями администратора.
Альтернативные потоки: Нет.

## Приложение Б. Адаптивная верстка

На рисунках 1–8 приведены скриншоты интернет-магазина на различных размерах экрана.



Рисунок 1 – Вид главной страницы на мобильной версии сайта

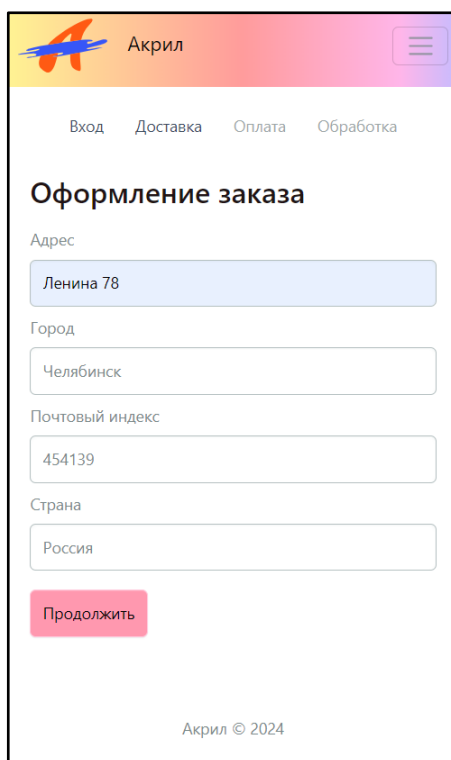


Рисунок 2 – Вид страницы оформления заказа на мобильной версии сайта



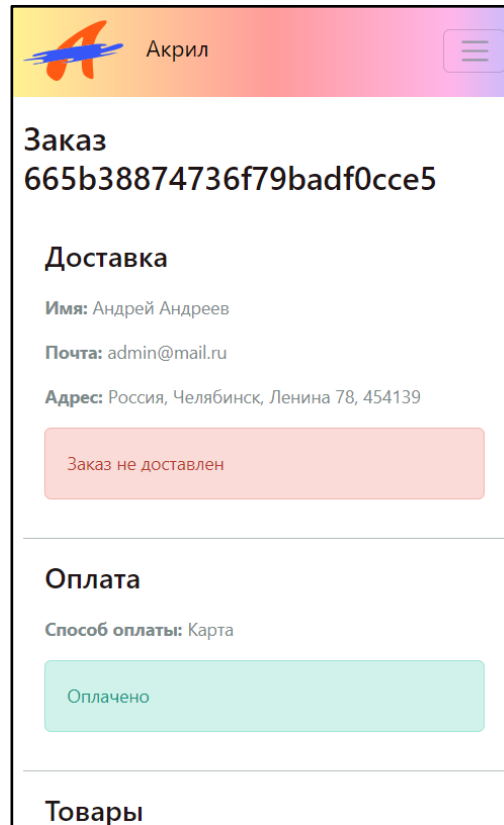


Рисунок 3 – Вид страницы заказа на мобильной версии сайта

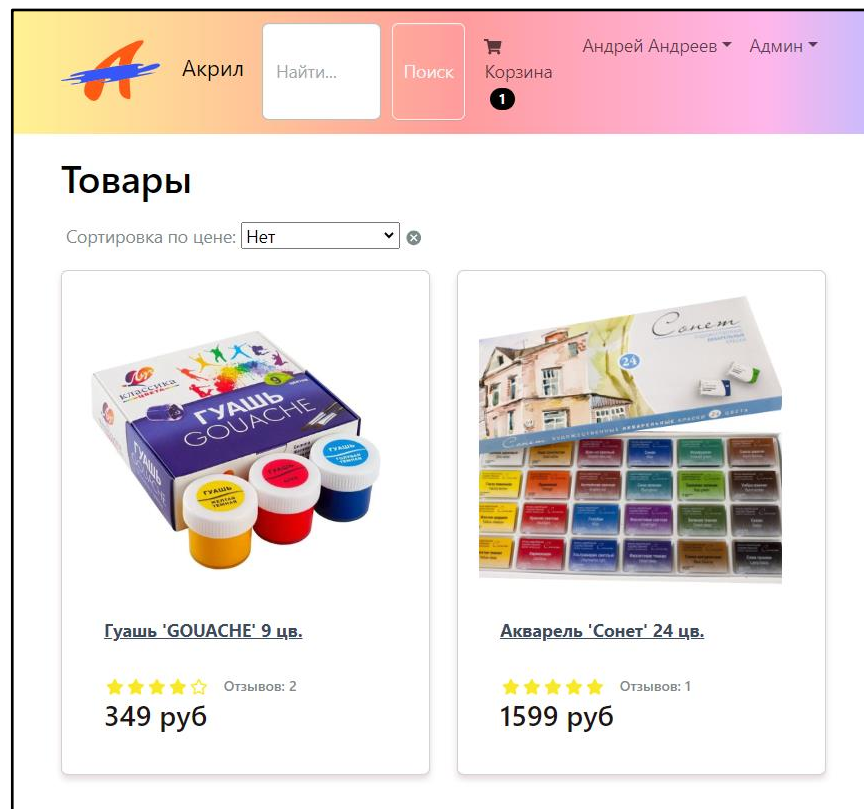


Рисунок 4 – Вид главной страницы на планшетной версии сайта

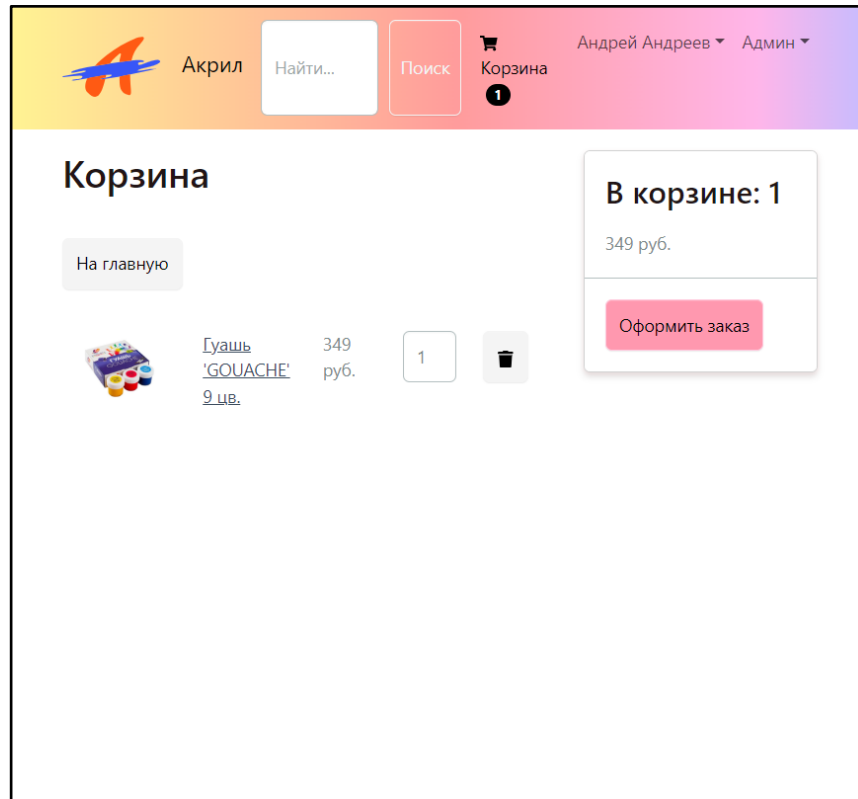


Рисунок 5 – Вид корзины на планшетной версии сайта

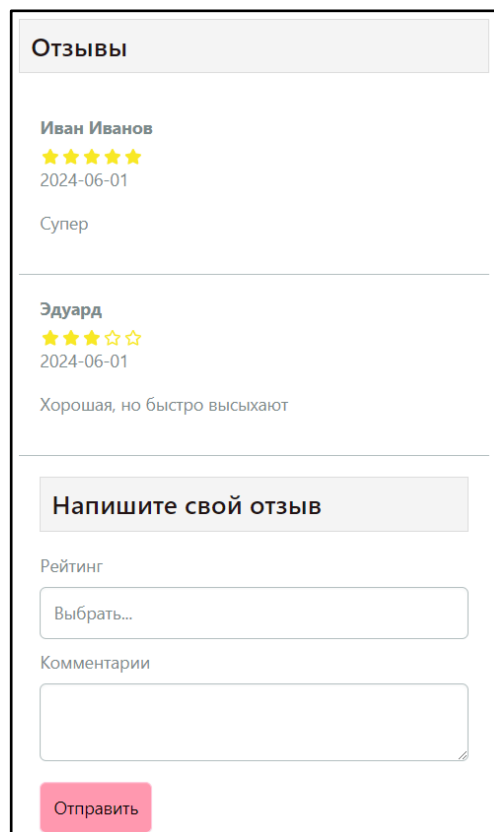


Рисунок 6 – Отображение отзывов на мобильной версии сайта