СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc169258204)

[1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc169258205)

[1.1. Обзор рынка зоомагазинов 6](#_Toc169258206)

[1.2. Эффективное frontend-проектирование для зоомагазина 7](#_Toc169258207)

[1.3. Исследование потребностей и ожиданий целевой аудитории 10](#_Toc169258208)

[2. ПОСТАНОВКА ЗАДАЧИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА 13](#_Toc169258209)

[2.1. Постановка задачи 13](#_Toc169258210)

[2.2. Диаграммы UML 14](#_Toc169258211)

[2.3. Диаграммы IDEF0 16](#_Toc169258212)

[2.4. Стиль дизайна и цветовые схемы 18](#_Toc169258213)

[3. РЕАЛИЗАЦИЯ. ОПИСАНИЕ АЛГОРИТМОВ РАБОТЫ ПРОГРАММНОГО СРЕДСТВА 21](#_Toc169258214)

[3.1. Обоснование выбора инструментов разработки 21](#_Toc169258215)

[3.2. Каталог и страницы товаров 22](#_Toc169258216)

[3.3. Раздел корзины 23](#_Toc169258217)

[3.4. Раздел авторизации 25](#_Toc169258218)

[3.5. Поиск 26](#_Toc169258219)

[3.6. Раздел администрирования 26](#_Toc169258220)

[3.7. Карта сайта 27](#_Toc169258221)

[4. ТЕСТИРОВАНИЕ 29](#_Toc169258222)

[4.1. Тест-план 29](#_Toc169258223)

[4.2. Чек-листы 32](#_Toc169258238)

[4.3. Тест-кейсы программного средства 33](#_Toc169258239)

[4.4. Вывод о результатах тестирования 35](#_Toc169258240)

[5. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ 36](#_Toc169258241)

[5.1. Исходная информация 36](#_Toc169258242)

[5.2. Определение себестоимости программного обеспечения (ПО), как базы для .формирования цены 36](#_Toc169258243)

[5.3. Выбор схемы монетизации программного продукта 43](#_Toc169258244)

[ЗАКЛЮЧЕНИЕ 46](#_Toc169258245)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 47](#_Toc169258246)

[ПРИЛОЖЕНИЕ 1 ИСХОДНЫЙ ТЕКСТ ПРОГРАММНОГО СРЕДСТВА 48](#_Toc169258247)

[ПРИЛОЖЕНИЕ 2 БЛОК-СХЕМА АЛГОРИТМА РАБОТЫ ПРОГРАММНОГО СРЕДСТВА 57](#_Toc169258248)

[ПРИЛОЖЕНИЕ 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 58](#_Toc169258249)

[ПРИЛОЖЕНИЕ 4 ПРОТОКОЛ ИСПЫТАНИЯ ПРОГРАММНОГО СРЕДСТВА. 62](#_Toc169258250)

# ВВЕДЕНИЕ

Современный рынок онлайн-торговли демонстрирует постоянный рост популярности, в том числе и в сегменте товаров для животных. Хозяева все чаще обращаются к онлайн-магазинам для удовлетворения потребностей своих питомцев, что подчеркивает значимость разработки удобных и функциональных интернет-магазинов для зоотоваров.

Тема данного дипломного проекта – "Разработка frontend сайта зоомагазина".

Целью дипломного проектирования является реализация современного, удобного и информативного сайта для зоомагазина.

Чтобы достичь поставленной цели, необходимо решить следующие задачи:

* изучить основные принципы разработки frontend сайтов, включая аспекты дизайна, пользовательского опыта и производительности;
* провести анализ требований к функционалу и дизайну сайта зоомагазина, учитывая специфику товаров для животных и потребности целевой аудитории;
* спроектировать структуру сайта, разработать дизайна и создать интуитивно понятный пользовательский интерфейс для удобства использования покупателями;
* реализовать frontend части сайта с применением современных технологий веб-разработки, таких, как HTML, CSS и JavaScript;
* провести тестирование и оптимизацию работы сайта с целью обеспечения высокой производительности, быстрой загрузки страниц, адаптивности к различным устройствам и браузерам.

Реализация проекта позволит создать удобный и информативный онлайн-ресурс для зоомагазина, что поспособствует привлечению новых клиентов и повышению конкурентоспособности. Плюсом программного средства, разработанного в рамках дипломного проекта, будет являться соответствие современным требованиям, таким как адаптивный дизайн, высокая производительность, удобство использования и безопасность данных.

1. ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ
   1. Обзор рынка зоомагазинов

Статистика и тренды в индустрии зоомагазинов отражают растущую популярность этой отрасли. Согласно исследованию NielsenIQ за 2023 год, 72% потребителей в странах СНГ владеют домашними животными. Большинство владельцев питомцев (82%) ухаживают за кошками, а почти половина (44%) держит собак. На третьем месте по популярности — рыбки (13%). И на фоне того, что за последние годы количество питомцев в семьях увеличивалось, сегодня индустрия ухода за ними, получив мощный импульс, продолжает развиваться двузначными темпами.Владельцы питомцев все больше обращают внимание на натуральные и экологически чистые продукты, а также на разнообразие предлагаемых услуг, включая груминг и медицинский уход. В целом, тренды в индустрии зоомагазинов свидетельствуют о растущей значимости домашних животных в жизни людей и стремлении обеспечить им лучшие условия и заботу.

Рост популярности онлайн-торговли зоотоварами – значительный тренд в индустрии зоомагазинов. Привлекательность онлайн-покупок объясняется их удобством, широким выбором товаров и возможностью сравнения цен. Улучшение услуг доставки также способствует этому росту, удовлетворяя потребности владельцев питомцев.

На рынке существуют основные игроки, включая крупные корпорации и международные бренды, а также местные предприятия и стартапы. Например, рассмотрим официальный сайт зоомагазина "Zoobazar" (<https://zoobazar.by>):

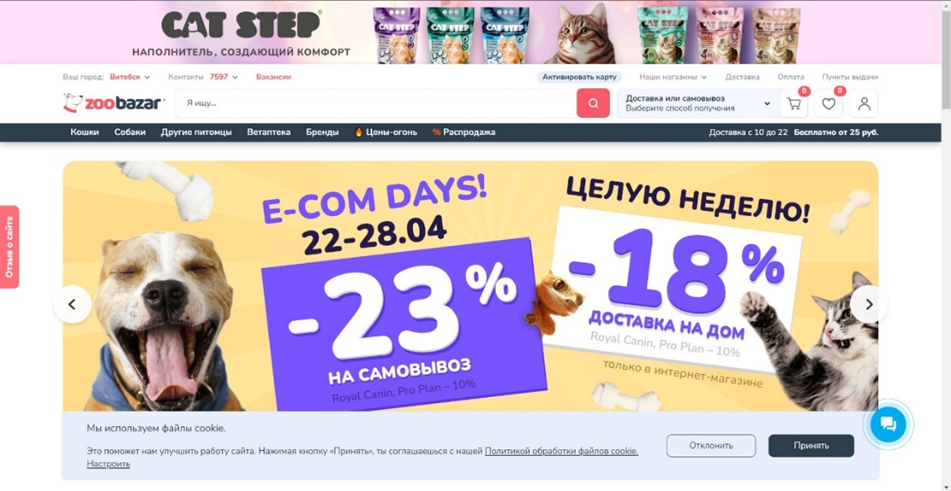


Рисунок 1.1 – Внешний вид при переходе на сайт "Zoobazar"

Дизайн и пользовательский интерфейс официального сайта "Zoobazar" сочетают привлекательность и удобство использования. Стильный и интуитивно понятный дизайн обеспечивает легкую навигацию по категориям товаров, удобство поиска необходимых продуктов, а также предоставляет информацию о специальных предложениях и услугах магазина, создавая приятный и эффективный пользовательский опыт.

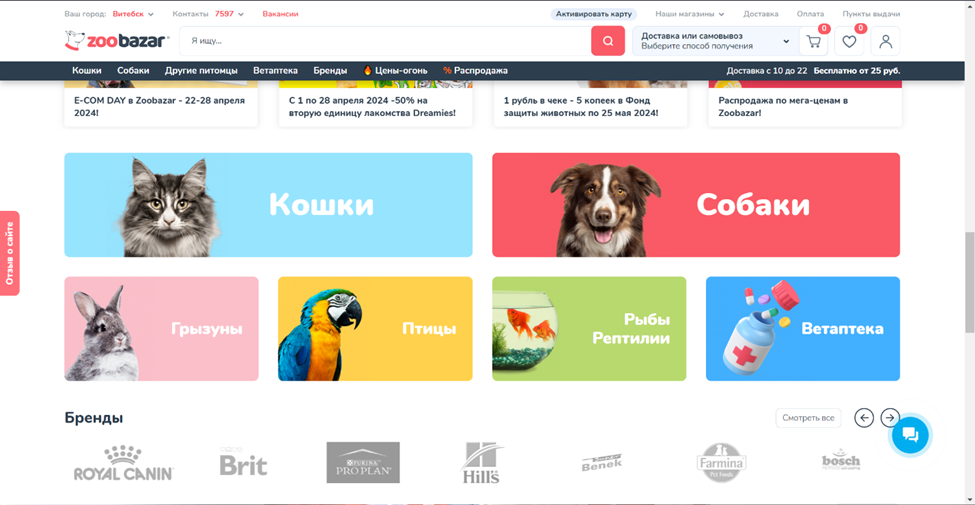


Рисунок 1.2 – Категории товаров на сайте "Zoobazar"

Официальный сайт "Zoobazar" предоставляет широкий спектр функциональности и возможностей, включая:

* каталог товаров;
* поиск и фильтрация;
* детальное описание товаров;
* онлайн-заказ и доставка;
* акции и специальные предложения;
* регистрация и личный кабинет;
* система отзывов и рейтингов.

Эти функции и возможности делают сайт "Zoobazar" удобным и полезным ресурсом для покупателей.

* 1. Эффективное frontend-проектирование для зоомагазина

Влияние пользовательского опыта на успех онлайн-торговли не может быть недооценено. Пользовательский опыт, или UX (User Experience), описывает взаимодействие пользователей с веб-сайтом, мобильным приложением или любым другим цифровым продуктом. Качество пользовательского опыта может существенно повлиять на решение пользователей совершить покупку, вернуться к ней в будущем и рекомендовать ее другим.

Вот несколько способов, как пользовательский опыт влияет на успех онлайн-торговли:

* первое впечатление: хороший дизайн создает положительное впечатление у пользователей;
* удобство использования: интуитивный интерфейс и легкость взаимодействия помогают пользователям быстро находить товары и оформлять заказы;
* скорость загрузки: быстрая загрузка страниц поддерживает удовлетворение пользователей и повышают вероятность покупки;
* удовлетворение потребностей: детальная информация о товарах, понятные описания и отзывы покупателей помогают пользователям принять решение о покупке.

В целом, положительный пользовательский опыт создает доверие, удовлетворение и лояльность пользователей, что способствует успеху онлайн-торговли. Он помогает привлечь новых клиентов, удерживать существующих и формировать положительный имидж бренда.

Ключевые аспекты, влияющие на эффективность frontend-сайта зоомагазина, включают удобство навигации, привлекательный дизайн и информативность контента (ссылка на ресурс: <https://e-zoo.by>).

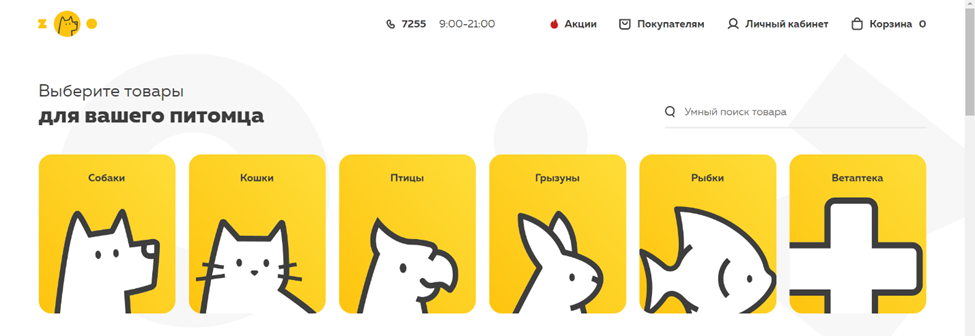


Рисунок 1.3 – Главное меню сайта "E-zoo" с категориями товаров

Дизайн сайта должен быть привлекательным и соответствовать тематике зоомагазина, чтобы привлекать внимание посетителей и создавать положительное впечатление. Контент должен быть информативным, содержать подробную информацию о товарах, их характеристиках, ценах и наличии.

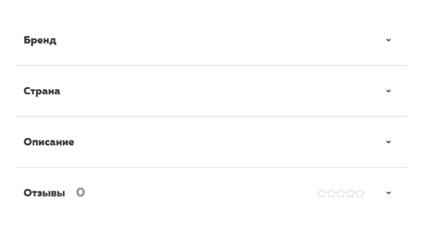
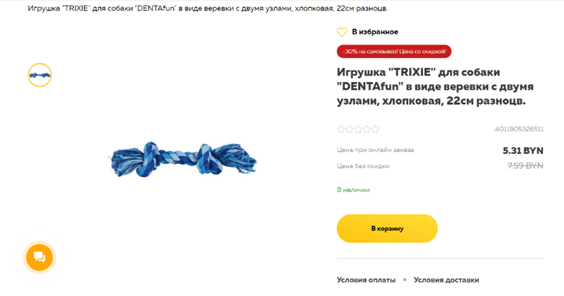


Рисунок 1.4 – Страница товара на сайте "E-zoo"

Комбинация этих аспектов обеспечит удобство использования и привлекательность frontend-сайта зоомагазина, что поможет привлечь и удержать посетителей.

Лучшие практики в разработке frontend-интерфейсов для зоомагазинов включают адаптивный дизайн и оптимизацию производительности.

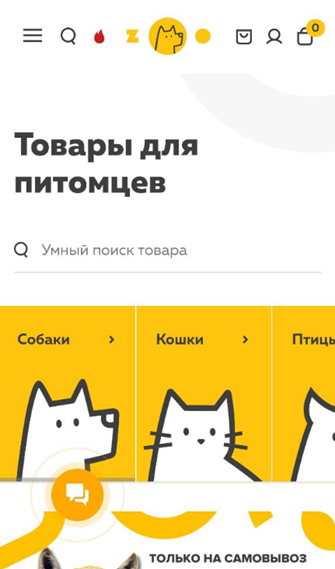


Рисунок 1.5 – Мобильная версия сайта "E-zoo"

В современном мире мобильные устройства становятся все более популярными, поэтому важно создать интерфейс, который будет корректно отображаться и удобно использоваться на различных экранах и разрешениях.

Оптимизация производительности также является важным аспектом разработки frontend-интерфейсов для зоомагазинов. Это включает минимизацию размера файлов, сжатие и кэширование изображений, использование эффективных техник загрузки данных и другие меры, которые позволяют ускорить время загрузки страницы и обеспечить быструю и плавную работу интерфейса. Это особенно важно в онлайн-магазинах, где пользователи ценят оперативность и отзывчивость интерфейса.

Применение данных лучших практик в разработке frontend-интерфейсов поможет создать удобную и эффективную платформу для зоомагазина, повысить удовлетворенность пользователей и улучшить результаты бизнеса.

* 1. Исследование потребностей и ожиданий целевой аудитории

Идентификация целевой аудитории зоомагазина играет важную роль в успешной маркетинговой стратегии. В первую очередь необходимо определить, кто является основной целевой группой клиентов: владельцы собак, кошек, грызунов или птиц. Это позволит более точно настроить маркетинговые кампании и предложения, а также подобрать ассортимент товаров.



Рисунок 1.6 – Распределение продаж по категориям товаров для домашних животных

Данная диаграмма представляет процентное распределение продаж по различным категориям товаров для домашних животных. Видно, что категория "Товары для кошек" занимает самую большую долю в общем объеме продаж и составляет 42%. Товары для собак также имеют значительную долю и составляют 22% от общего объема. Остальные категории, такие как товары для рыб, грызунов, птиц и рептилий, имеют меньшую долю в общем объеме продаж, составляя 6%, 6%, 3% и 1% соответственно. Эти данные могут быть полезны при планировании ассортимента и маркетинговых стратегий в зоомагазине, позволяя сосредоточиться на наиболее востребованных категориях товаров.



Рисунок 1.7 – Распределение продаж по категориям товаров для животных в различных каналах продаж

Данная диаграмма представляет процентное распределение продаж по категориям товаров для животных в различных каналах продаж: современная розница, зооспециалисты и онлайн-канал. Онлайн-канал является основным каналом для продаж товаров для животных, превышая доли продаж в современной рознице и у зооспециалистов во всех категориях, таких как товары для животных, лакомства, наполнители, корм для собак и корм для кошек. Эти данные могут помочь в планировании и принятии решений о стратегиях продаж и маркетинговых активностях в зависимости от предпочтений и поведения покупателей в разных каналах.

Для анализа потребностей и предпочтений пользователей можно использовать аналитические инструменты, такие как сбор данных о поведении пользователей на сайте, отзывы и оценки товаров, а также опросы и анкеты. Это позволит получить количественные данные о том, какие функции и особенности интерфейса наиболее востребованы, а также выявить проблемные места, которые требуют улучшения.

Важно также принимать во внимание обратную связь от пользователей через различные каналы коммуникации, включая электронную почту, социальные сети и онлайн-чаты.

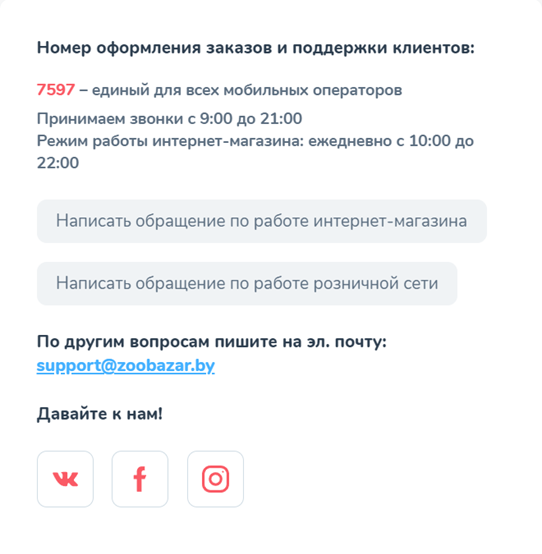


Рисунок 1.8 – Обратная связь на сайте " Zoobazar "

Активное взаимодействие с клиентами позволит получить ценные идеи и рекомендации, а также непосредственно отреагировать на их потребности.

Анализ потребностей и предпочтений пользователей является непрерывным процессом, который требует постоянного мониторинга и обновления интерфейса зоомагазина. Это позволит создать удовлетворительный пользовательский опыт и удерживать клиентов на сайте.

Важные факторы, влияющие на пользовательское взаимодействие с сайтом зоомагазина, включают навигацию, удобство использования и доступность контента. Хорошо спроектированная навигационная система с логической структурой и интуитивно понятными категориями продуктов помогает пользователям быстро и легко находить необходимые товары и информацию.

Учет этих важных факторов в разработке интерфейса зоомагазина позволит создать приятное и удобное пользовательское взаимодействие, что способствует привлечению и удержанию клиентов на сайте.

1. ПОСТАНОВКА ЗАДАЧИ И ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

* 1. Постановка задачи

В ходе работы над дипломным проектом требуется разработать frontend сайта, включая создание вёрстки. В рамках разработки дипломного проекта должно быть сделано основное ударение на реализацию функциональных требований, которые обеспечат эффективное и удобное взаимодействие пользователей с сайтом.

Важным аспектом является адаптивное отображение сайта на различных устройствах, включая компьютеры и мобильные устройства. Это означает, что веб-сайт должен корректно и красиво отображаться на экранах разных размеров, а его элементы и компоненты должны быть оптимизированы для удобного использования на разных устройствах. Будут использованы технологии и подходы, позволяющие создать отзывчивый интерфейс, который автоматически адаптируется к различным экранам.

Для обеспечения удобной навигации по сайту должна быть создана структура страниц и обеспечена возможность легкого перехода между ними. Должны быть разработаны различные страницы, включая главную страницу, страницы с подробным описанием товаров, корзина и оформления заказа.

Для удобства пользователей и наглядности структуры сайта должна быть создана карта сайта. Это графическое представление, которое поможет визуализировать иерархию страниц и логические связи между ними. Карта сайта облегчит пользователям поиск нужной информации и быстрый переход между страницами.

При разработке сайта также должны быть учтены следующие требования к отображаемой информации, дизайну, стилю и цветовым решениям.

* все необходимые данные о товарах, такие как название, описание, цена, наличие, фотографии и характеристики;
* категории товаров с фильтрацией по различным параметрам;
* информация о доставке;
* контактная информация, включая телефон и электронную почту;
* привлекательный и современный дизайн, соответствующий тематике зоомагазина;
* четкая и интуитивно понятная навигация, обеспечивающая легкость перемещения по сайту;
* читабельный шрифт для облегчения восприятия информации пользователем;
* качественные и эстетически привлекательные фотографии товаров, которые наглядно демонстрируют их внешний вид и характеристики;
* стиль, отражающий заботу о животных и их благополучие, создающий дружелюбную и доверительную атмосферу;
* использование приятных, но не раздражающих глаз цветовых палитр.

Так же должна быть реализована функция регистрации пользователей для удобного оформления заказов, получения персональной информации или использования других функций сайта.

Все эти требования должны быть реализованы с учетом создания удобного и приятного пользовательского опыта, стимулирующего привлечение и удержание посетителей сайта.

* 1. Диаграммы UML

Диаграмма вариантов использования является одной из наиболее часто встречающихся типов диаграмм UML и используется для визуализации возможностей, взаимодействия субъектов (пользователей) и систем. Она позволяет определить и описать основные сценарии использования систем или основных требований, которые должны быть удовлетворены для достижения определенных целей.



Рисунок 2.1 – Диаграмма вариантов использования для клиента

Данная диаграмма вариантов использования представляет собой визуализацию основных возможностей системы, связанной с клиентом. В центре диаграммы находится актер «Клиент», который представляет собой пользовательскую систему. Линии, исходящие от клиента, предлагают различные варианты использования или требования к поставщикам услуг, которые могут быть выполнены клиентом. Эти варианты использования предоставляют общее представление о возможностях системы связи с клиентом и осуществления различных действий, которые клиент может выполнять при использовании системы.

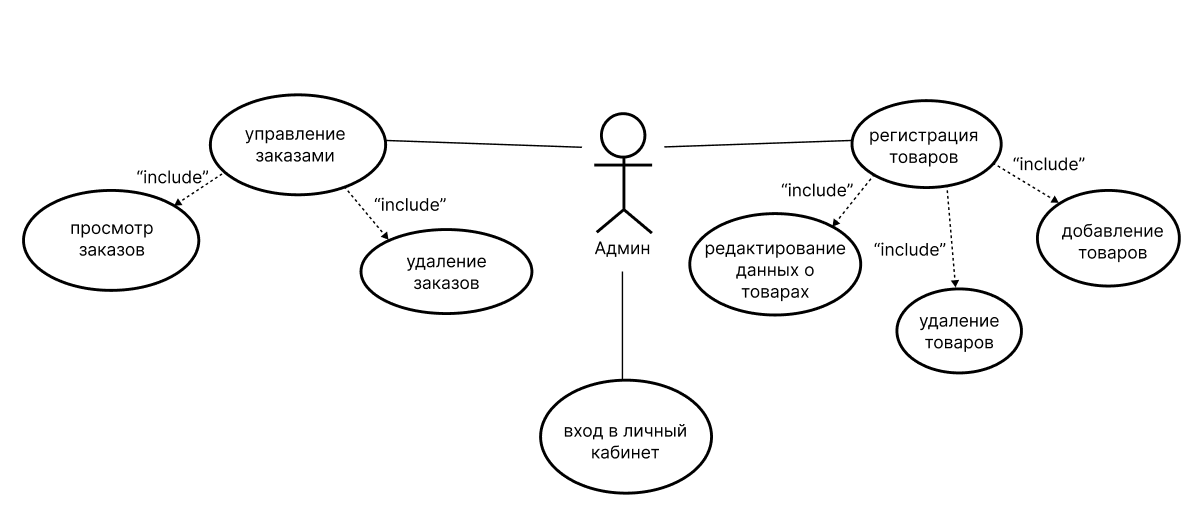


Рисунок 2.2 – Диаграмма вариантов использования для администратора

Данная диаграмма вариантов использования представляет собой визуализацию функциональных возможностей системы, связанных с администратором. В центре диаграммы находится актер "Администратор", который имеет возможность управлять заказами и регистрировать товары. Первый вариант использования, "Управление заказами", позволяет администратору редактировать данные о заказах, удалять и просматривать их. Второй вариант использования, "Вход в личный кабинет", предоставляет администратору возможность войти в систему (личный кабинет). Третий вариант использования, "Регистрация товаров", позволяет администратору добавлять новые товары в систему. Связанные с управлением заказами функции включают удаление заказов и их просмотр. Связанные с регистрацией товаров функции включают редактирование данных о товарах, удаление товаров и добавление новых товаров. Администратор может эффективно управлять заказами и товарами в системе, обеспечивая актуальность данных и предоставляя полный контроль над функциональностью системы.

Диаграмма последовательности является одним из типов диаграмм UML и используется для визуализации взаимодействия между объектами или компонентами системы в рамках определенной последовательности событий. Она позволяет лучше понять, как объекты взаимодействуют друг с другом и в каком порядке происходят операции. В диаграмме последовательности объекты системы представлены вертикальными линиями, называемыми обычно активационными линиями или жизненными линиями. Горизонтальные стрелки между активационными линиями представляют собой сообщения, передаваемые между объектами. В целом, диаграмма последовательности является инструментом для анализа взаимодействия в системе и позволяет легко визуализировать логику и последовательность выполнения операций или сообщений.

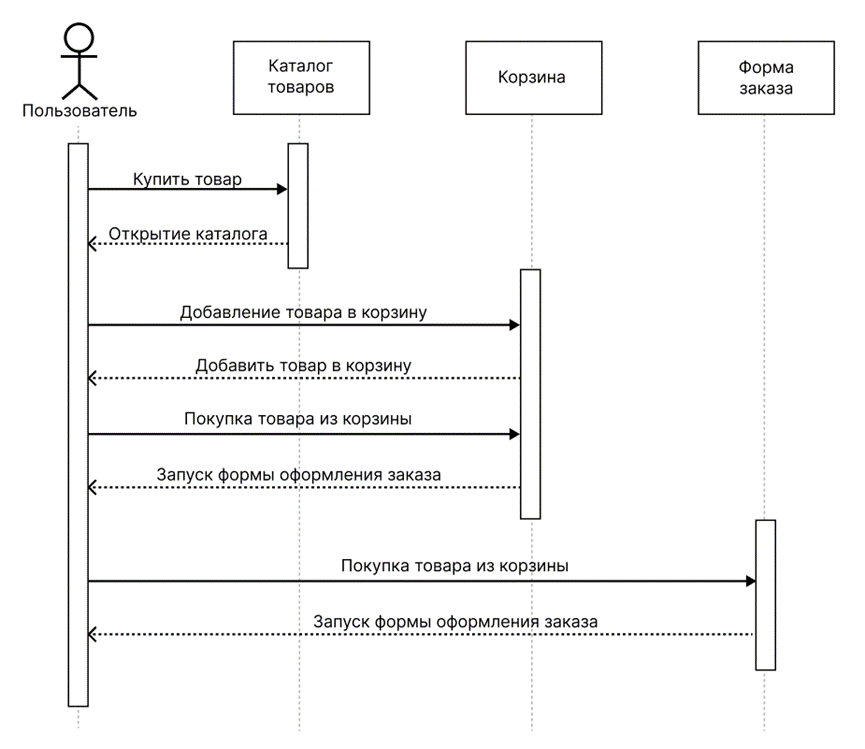


Рисунок 2.3 – Диаграмма последовательности

* 1. Диаграммы IDEF0

Диаграмма IDEF0 представляет собой графическое представление функций системы и их взаимосвязей. Она состоит из блоков, стрелок и текстовых описаний. Блоки представляют функции системы, а стрелки показывают потоки данных или управления между функциями. Текстовые описания предоставляют дополнительную информацию о каждой функции. Диаграмма IDEF0 помогает визуализировать процессы, задействованные в системе, и понять, как функции взаимодействуют друг с другом.

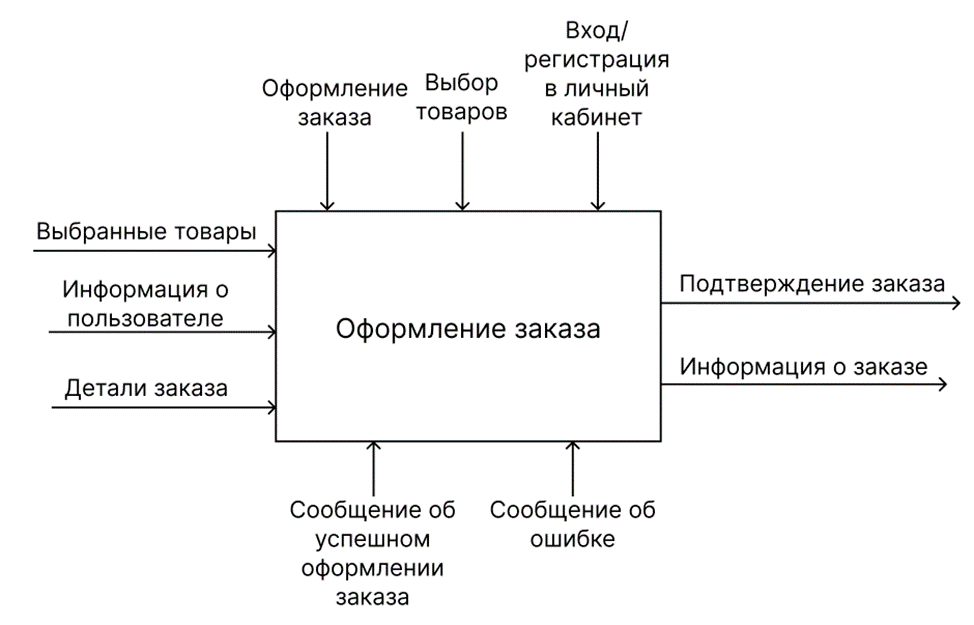


Рисунок 2.4 – Контекстная диаграмма IDEF0

Данная диаграмма IDEF0 включает центральный блок "Оформление заказа". Сверху входят стрелки, обозначающие этапы "Вход/регистрация в личный кабинет", "Выбор товаров" и "Оформление заказа", в которых пользователь входит в систему, выбирает нужные товары и передает данные для оформления заказа. Снизу расположены стрелки "Сообщение об успешном оформлении заказа" и "Сообщение об ошибке", указывающие на результаты процесса. Справа находятся стрелки, обозначающие "Выбранные товары", "Информацию о пользователе" и "Детали заказа", в которых передаются соответствующие данные. Слева расположены стрелки "Подтверждение заказа" и "Информация о заказе", указывающие на этапы, связанные с подтверждением заказа и предоставлением информации о заказе соответственно.

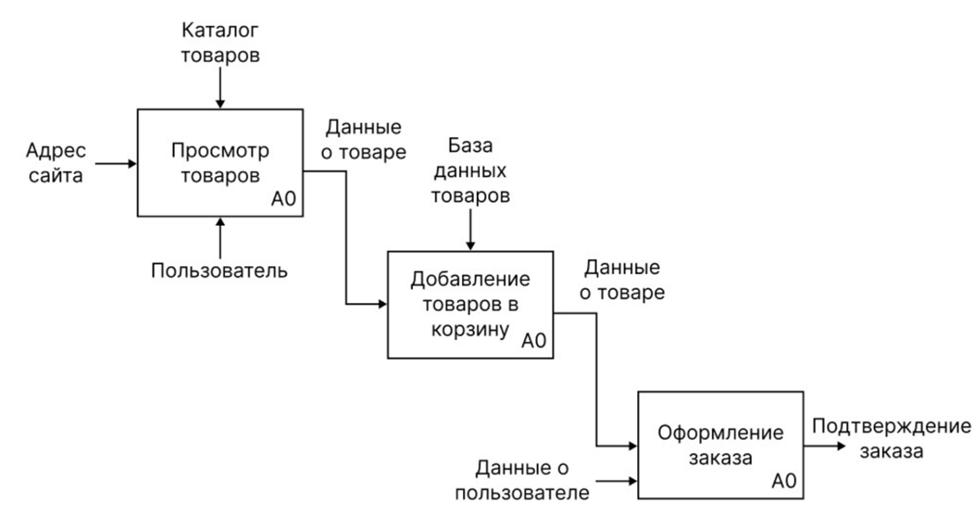


Рисунок 2.5 – Декомпозиция уровня А1

* 1. Стиль дизайна и цветовые схемы

Стиль дизайна является ключевым аспектом в создании визуального облика зоомагазина и передачи его характера и целей. При выборе стиля следует учесть, что он должен наилучшим образом отражать уникальность и профессионализм магазина, привлекать внимание посетителей и создавать определенное настроение.

Один из возможных вариантов стиля дизайна - игривый и яркий. Яркие цвета, игривые иллюстрации животных или мультяшные персонажи могут быть использованы для создания привлекательного и дружелюбного интерфейса и помогут привлечь внимание посетителей и создать позитивное настроение. Также простые формы, чистые линии и понятная навигация помогут пользователям быстро ориентироваться на сайте и делать покупки с удовольствием.

Цветовая схема также является важным аспектом в создании атмосферы и визуальной идентичности зоомагазина.

При выборе цветовой палитры для зоомагазина можно рассмотреть использование фиолетового цвета. Широкий спектр фиолетовых оттенков может придать зоомагазину уникальный и загадочный облик. Фиолетовый цвет ассоциируется с тайной, роскошью и креативностью, что может привлечь внимание посетителей и создать особую атмосферу.

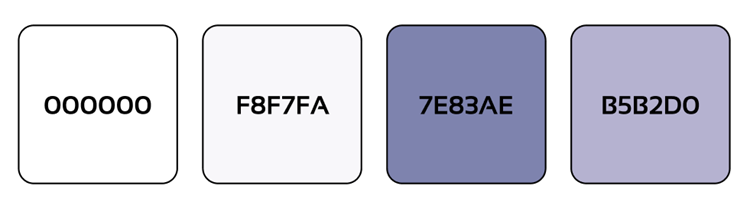


Рисунок 2.6 – Цветовая палитра сайта

Одним из вариантов использования фиолетовой палитры в дизайне зоомагазина является комбинация различных оттенков фиолетового с цветами, дополняющими его. Например, можно сочетать фиолетовый с нейтральными серыми или белыми оттенками, чтобы создать элегантный и современный вид. Такая цветовая схема может быть особенно эффективной для представления премиум-товаров для животных или для подчеркивания особого качества продукции.

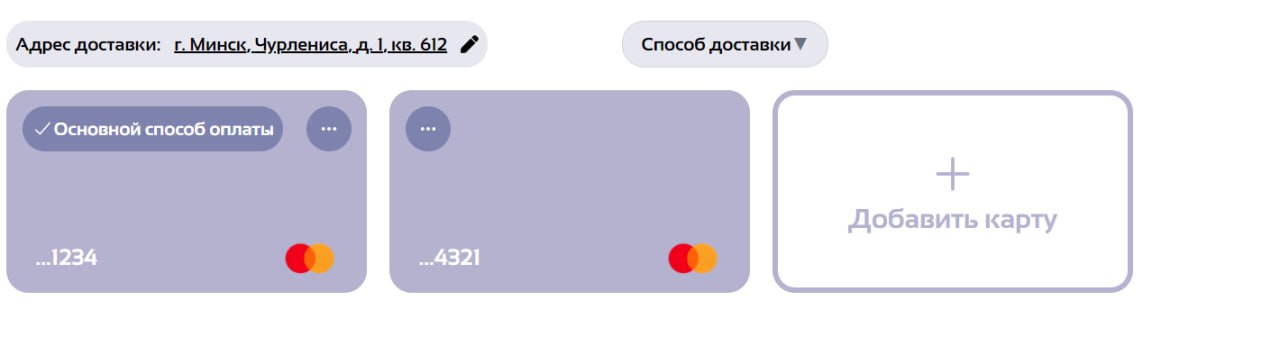


Рисунок 2.7 – Часть макета сайта зоомагазина

Выбор подходящих шрифтов играет важную роль в создании единого стиля и обеспечении читаемости текста на веб-сайте. При выборе шрифтов необходимо учитывать их читабельность в различных размерах и совместимость с различными устройствами и браузерами. Важно выбрать шрифты, которые будут одновременно эстетически привлекательными и функциональными.

Для заголовков можно выбрать более выразительные и уникальные шрифты, чтобы привлечь внимание посетителей и выделить важные разделы на сайте. Такие шрифты могут иметь уникальные формы и стилизацию, которая соответствует общей концепции бренда или атмосфере зоомагазина. Однако важно убедиться, что шрифты сохраняют хорошую читаемость даже в больших размерах и на разных устройствах.

Для основного текста рекомендуется выбирать шрифты, которые обладают хорошей читабельностью. Это обеспечит комфортное чтение информации на сайте и предотвратит утомление пользователей. Шрифты с простым и четким дизайном, такие как Sansation могут быть хорошим выбором для основного текста.

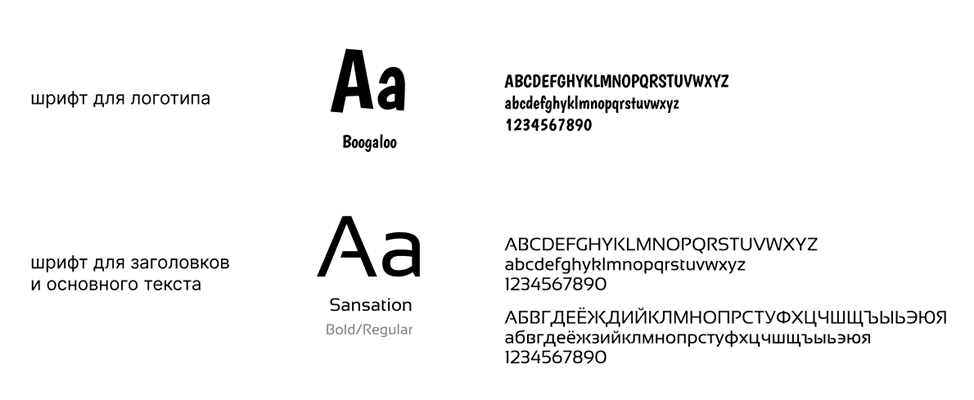


Рисунок 2.8 – Шрифтовая пара сайта

Предложенные решения смогут эффективно поддерживать цели и задачи зоомагазина, привлекая внимание целевой аудитории и создавая положительный опыт взаимодействия с сайтом.

1. РЕАЛИЗАЦИЯ. ОПИСАНИЕ АЛГОРИТМОВ РАБОТЫ ПРОГРАММНОГО СРЕДСТВА
   1. Обоснование выбора инструментов разработки

При разработке современного и функционального frontend-сайта зоомагазина в качестве основных средств разработки были тщательно подобраны ключевые технологии, которые позволят реализовать все необходимые требования к проекту.

В первую очередь, были выбраны стандарты HTML5 и CSS3 - базовые языки для создания структуры и визуального оформления web-страниц. HTML5 предоставляет широкие возможности для верстки документа, что дает возможность четко определить смысловую структуру различных разделов сайта. Использование последней версии каскадных таблиц стилей CSS3 позволяет реализовать современный и привлекательный дизайн с учетом принципов адаптивности и отзывчивости. Данные технологии обеспечивают кроссбраузерную совместимость и поддерживаются всеми актуальными веб-браузерами, что особенно важно для обеспечения широкого доступа к сайту зоомагазина.

Для реализации интерактивных и динамических элементов пользовательского интерфейса был выбран язык программирования JavaScript. Использование современного стандарта JavaScript открывает доступ к расширенному функционалу языка, включая стрелочные функции, классы, модули и другие возможности, повышающие эффективность разработки. Применение JavaScript позволит реализовать на клиентской стороне такие интерактивные компоненты, как всплывающие окна, формы обратной связи, добавление товаров в корзину и прочее, что является важным требованием для коммерческого сайта.

В качестве основного JavaScript-фреймворка для разработки был выбран React.js. Данная библиотека предоставляет эффективный механизм работы с компонентами пользовательского интерфейса, что значительно упрощает процесс разработки и поддержки сложных веб-приложений. Использование React позволил организовать структуру сайта зоомагазина в виде иерархии повторно используемых компонентов, обеспечивая при этом высокую производительность и простоту обновления пользовательского интерфейса.

Выбранные ключевые технологии - HTML5, CSS3 и React.js - являются современными, гибкими и высокоэффективными инструментами, которые позволили реализовать frontend-сайт зоомагазина, отвечающий актуальным требованиям пользователей к дизайну, функциональности и производительности.

* 1. Каталог и страницы товаров

При разработке frontend сайта зоомагазина было уделено особое внимание структуре и формату страниц, отображающих товары. Каждая страница товара имеет четкую и интуитивно понятную структуру, которая позволяет пользователям быстро ориентироваться и получать необходимую информацию.

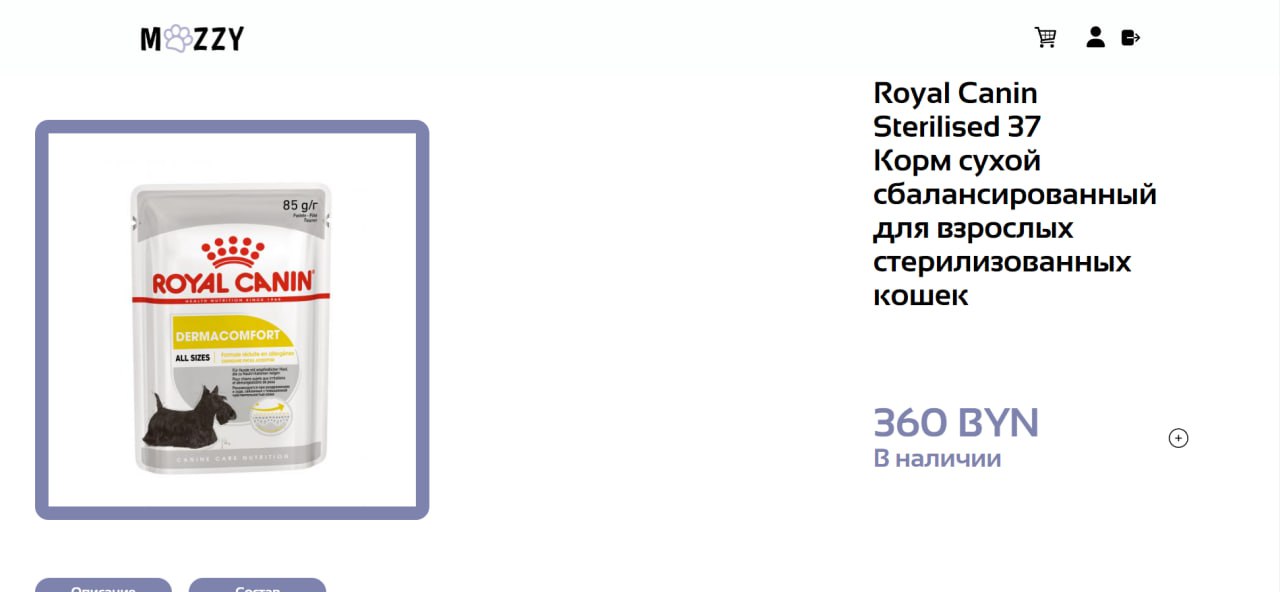


Рисунок 3.1 – Страница товара

Страницы, отображающие товары, содержат основную информацию о товаре, такую как название, изображение, цену и наличие. Также предоставляются дополнительные детали о товаре, такие как описание и состав. Кроме того, на страницах товаров предусмотрены функциональный элемент, позволяющий пользователям добавлять товары в корзину.

Также в процессе разработки была добавлена функция фильтрации товаров, чтобы пользователи могли быстро и удобно находить необходимые им товары. Это значительно повышает удобство использования сайта и ускоряет процесс выбора и покупки товаров.

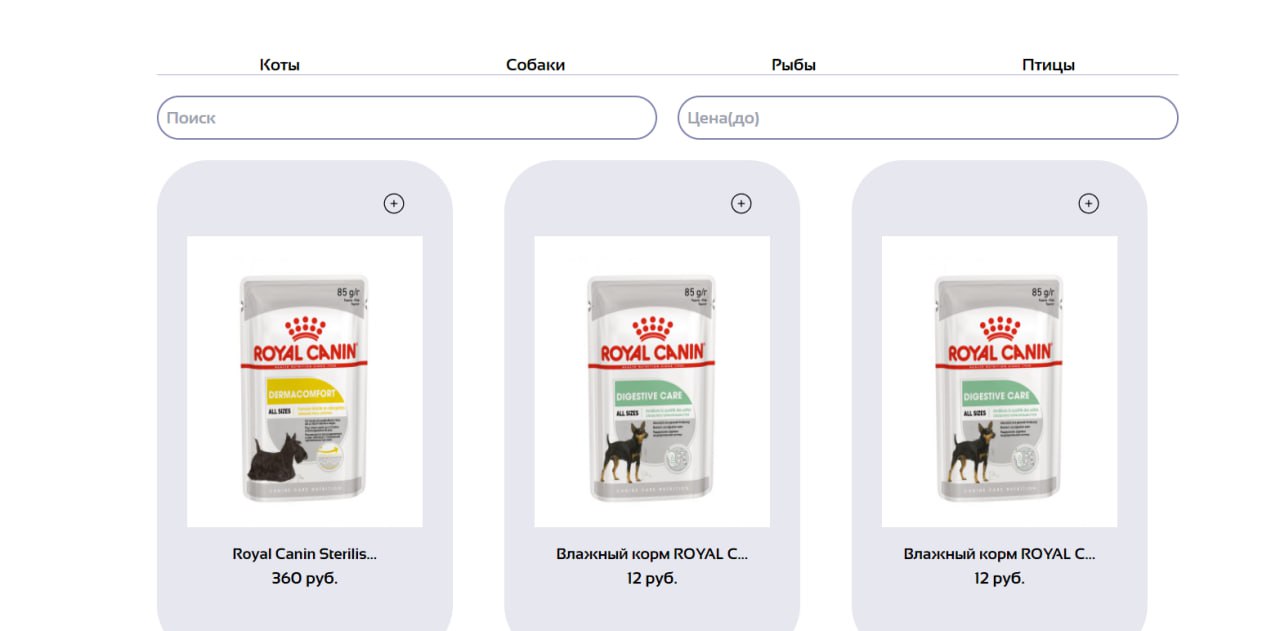


Рисунок 3.2 – Фильтры для поиска товаров

Вдобавок, при разработке была реализована возможность на добавление товаров в корзину. Пользователям предоставляется возможность выбрать интересующий их товар и добавить его в корзину.

При нажатии на кнопку "плюса", соответствующий товар добавляется в корзину пользователя. После нажатия на кнопку "плюса" появляется всплывающее уведомление, сообщающее о добавлении товара в корзину.

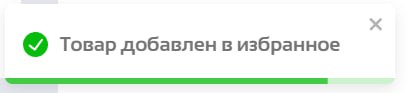
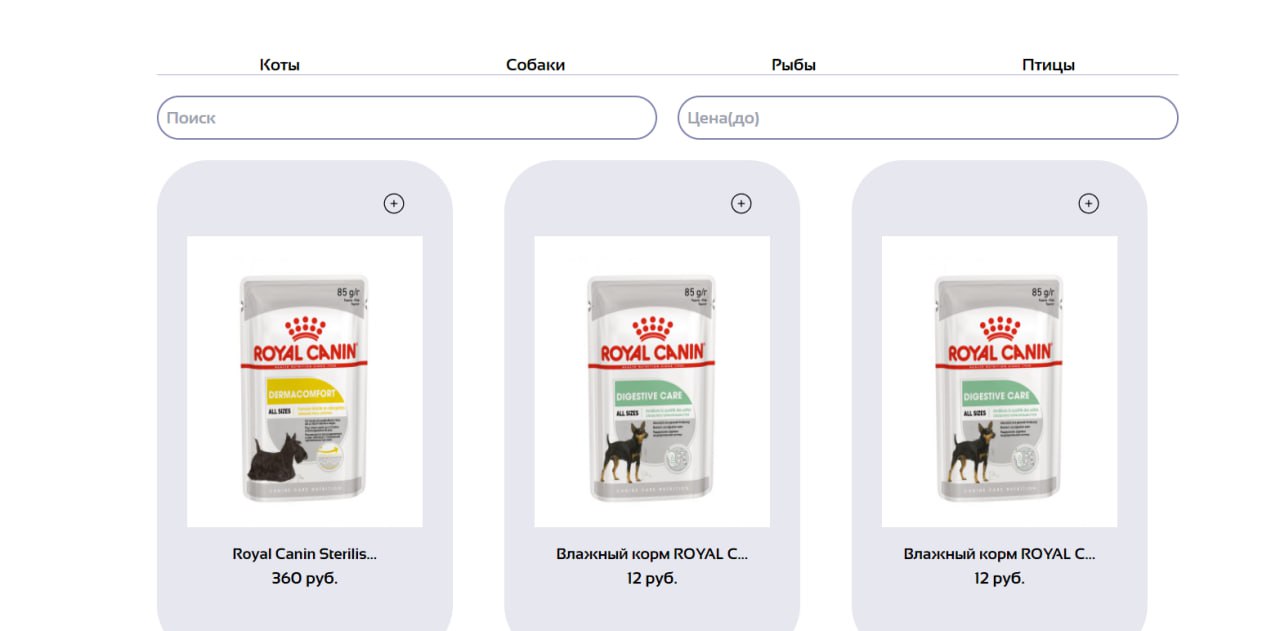


Рисунок 3.3 – Элементы, отображающие добавление товара в корзину

* 1. Раздел корзины

В процессе разработки fronted сайта зоомагазина, особое внимание было уделено организации и внешнему виду страницы корзины. Страница корзины представляет собой удобный и информативный интерфейс, который позволяет пользователям просматривать, редактировать и оформлять свои покупки.

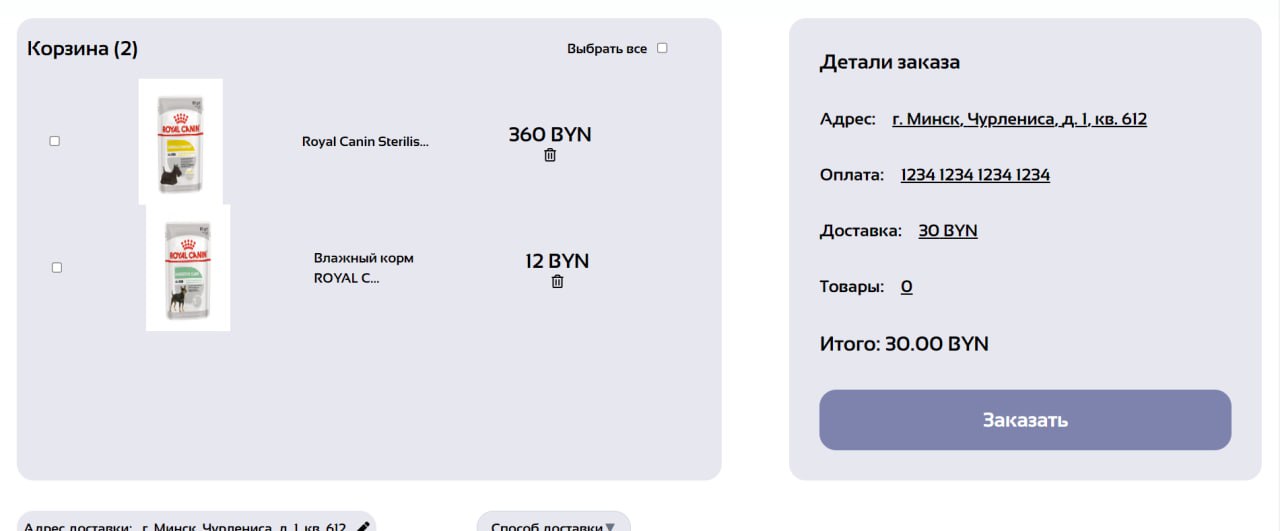


Рисунок 3.5 – Страница корзины

На странице корзины пользователи могут видеть полный список товаров, добавленных ими для покупки. Каждый товар представлен со своим названием, изображением и ценой. Кроме того, для каждого товара предоставляется возможность удалить товар из корзины.

Пользователям предоставляется возможность выбрать удобный способ доставки, выбрав подходящий способ из выпадающего списка. А также выбрать карту, с которой будет производиться оплата или добавить новую.

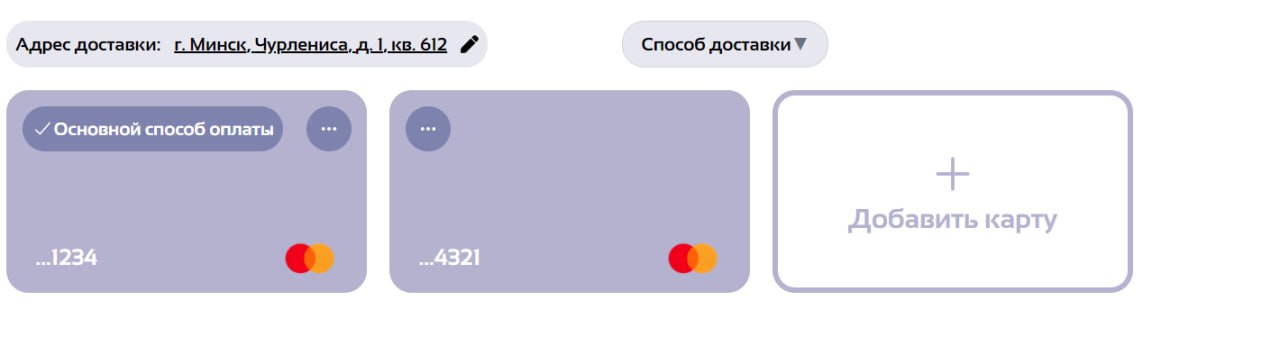


Рисунок 3.6 – Элементы оформления заказа на странице корзины

Страница корзины также содержит информацию о суммарной стоимости всех товаров, выбранных пользователем. То есть, каждый раз, когда пользователь добавляет или удаляет товары в корзине, общая стоимость всех товаров автоматически пересчитывается. На странице также присутствуют функциональные элементы для оформления заказа.

* 1. Раздел авторизации

При создании frontend сайта зоомагазина был создан удобный и интуитивно понятный способ авторизации. Окно авторизации предоставляет пользователям возможность войти в свой аккаунт с помощью личных данных, таких как логин и пароль. Структура окна авторизации включает поля ввода для логина и пароля, а также кнопку "Войти".

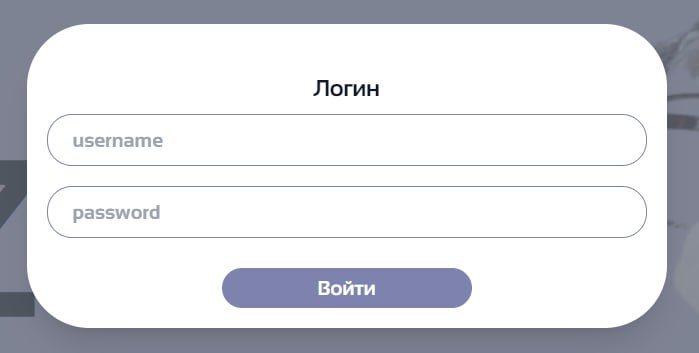


Рисунок 3.7 – Окно авторизации

Окно регистрации предоставляет пользователям возможность создать новую учетную запись в зоомагазине. Оно содержит поля для ввода необходимых данных, таких как логин, пароль и почта. В окне регистрации присутствует кнопка "Зарегистрироваться", которую пользователи могут нажать для завершения процесса регистрации.

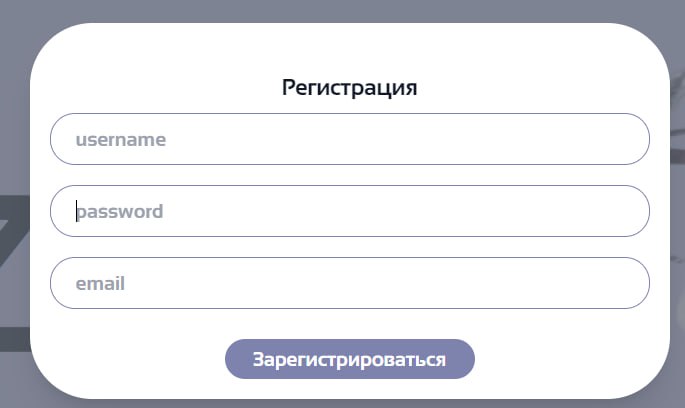


Рисунок 3.8 – Окно регистрации

Структура и формат окон авторизации и регистрации разработаны таким образом, чтобы обеспечить простоту использования и безопасность для пользователей зоомагазина. Пользователи могут легко войти в свой личный кабинет или создать новый с минимальными усилиями, что способствует хорошему пользовательскому опыту и повышает удовлетворенность клиентов.

* 1. Поиск

В процессе создания интерфейса также был реализован эффективный алгоритм поиска товаров. Пользователь нажимает на поле с надписью "Поиск" и вводит название нужного товара. Затем алгоритм обрабатывает введенные данные и производит поиск товаров. Результаты поиска выводятся на странице с информацией о найденных товарах.

Разработанный алгоритм поиска товаров обеспечивает удобство и эффективность для пользователей зоомагазина. Он позволяет пользователям находить нужные товары быстро и точно, учитывая их предпочтения и требования. Это важный функционал сайта, который способствует удовлетворенности клиентов и увеличению числа совершаемых покупок.

На странице с результатами поиска располагаются карточки товаров. Каждая карточка содержит изображение товара, название, цену и кнопку добавления в корзину. Визуальное представление карточек товаров позволяет пользователям быстро просматривать результаты поиска.

Таким образом, разработанная структура и формат страницы с результатами поиска обеспечивает удобство и информативность для пользователей зоомагазина. Она позволяет быстро просматривать и оценивать найденные товары. Это важный компонент функциональности сайта, который способствует удовлетворенности пользователей и повышению эффективности поиска товаров.

* 1. Раздел администрирования

В процессе разработке пользовательского интерфейса сайта зоомагазина была создана структура и формат страницы, доступной администратору.

На этой странице администратор может управлять товарами, а также просматривать и обрабатывать заказы. Структура страницы включает навигационное меню с разделами "Товары" и "Заказы", которые позволяют администратору легко и быстро переключаться между различными функциональными областями.

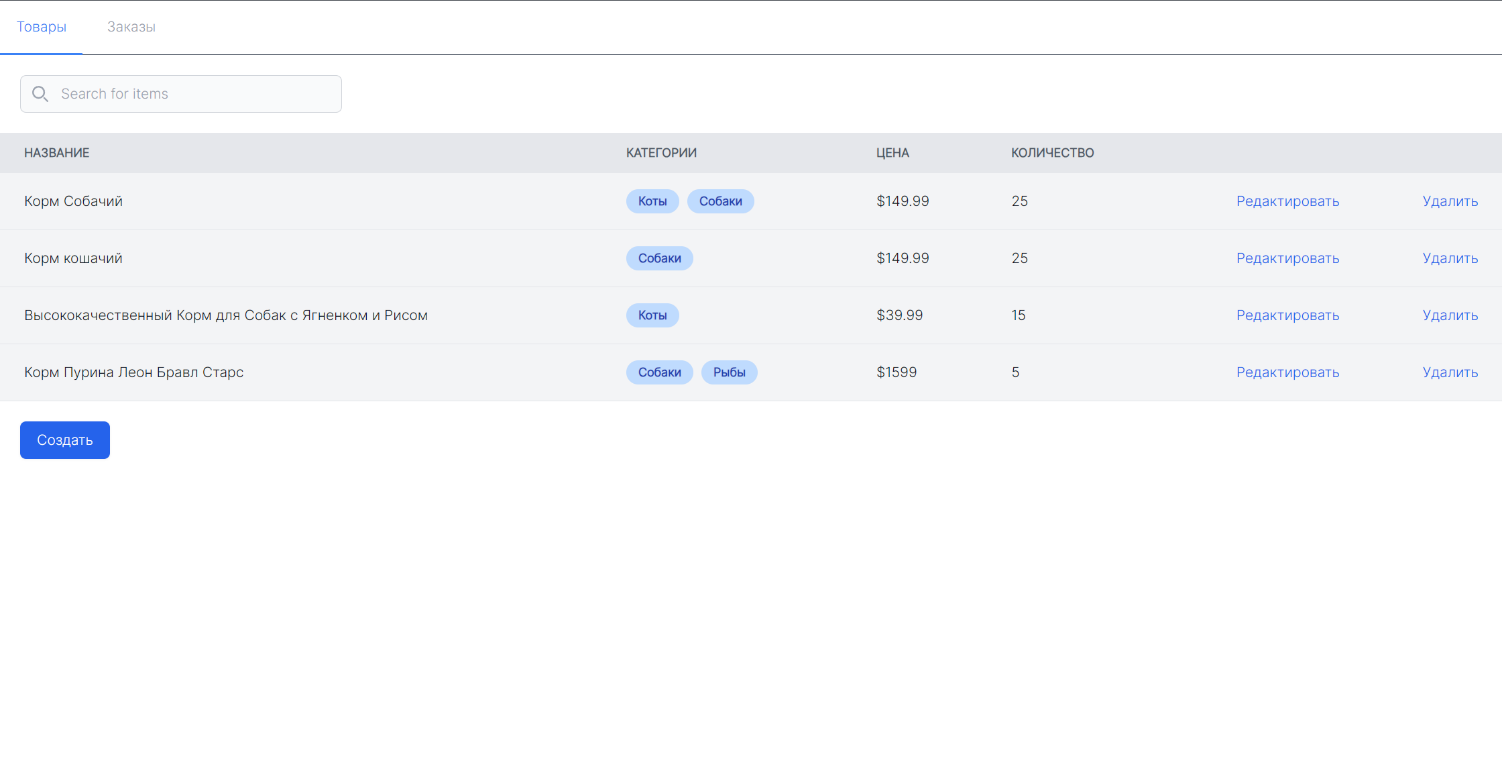


Рисунок 3.11 – Страница администратора

Таким образом, структура и формат страницы, доступной администратору зоомагазина, обеспечивают удобный и интуитивно понятный интерфейс. Это позволяет администратору эффективно управлять товарами и заказами. Этот функционал является важным компонентом разработанного сайта и способствует эффективному управлению и администрированию зоомагазина.

* 1. Карта сайта

Карта сайта — это визуальное представление структуры и взаимосвязей различных разделов и страниц сайта. Она играет важную роль в организации удобной и интуитивно понятной навигации по сайту. Ниже на рисунке 3.12 приведена карта разработанного в рамках дипломного проекта сайта.

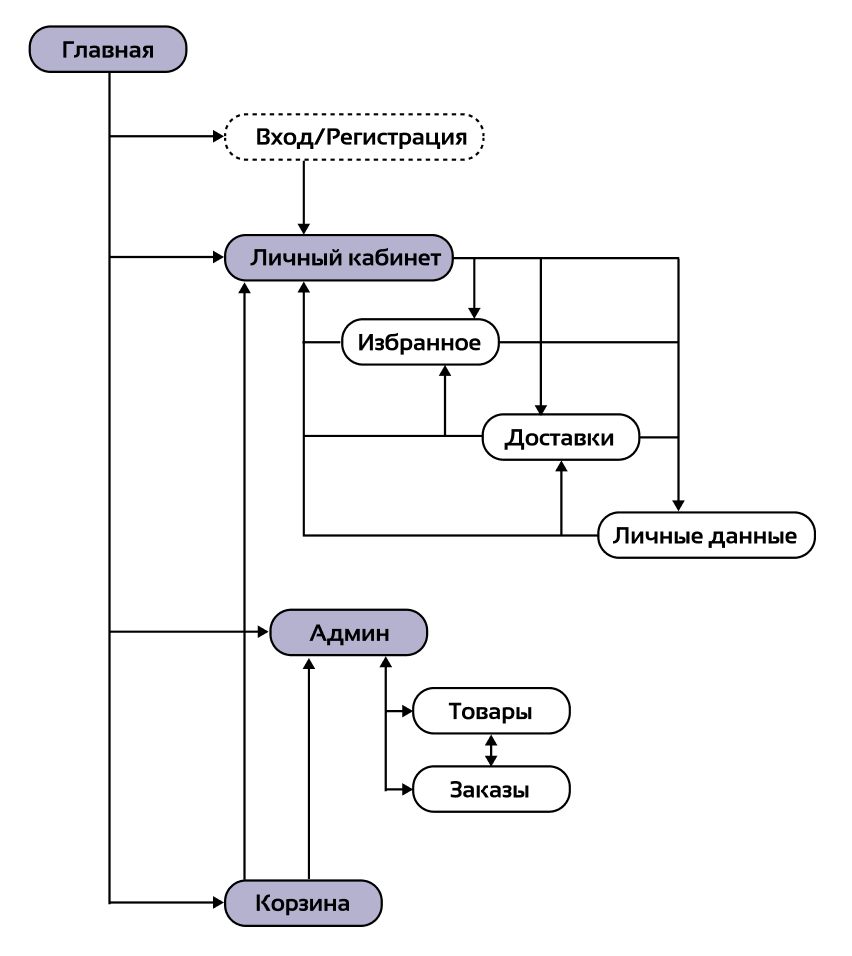


Рисунок 3.12 – Карта сайта

Карта сайта позволяет четко продумать и структурировать все разделы и подразделы, связи между ними, а также определить ключевые точки входа и логические переходы. Карта демонстрирует иерархию страниц сайта, показывая, как разные разделы связаны между собой.

Хотя сама карта сайта обычно скрыта от глаз покупателей, ее структура лежит в основе всей архитектуры сайта. Таким образом, тщательная проработка карты сайта - важный этап в разработке эффективного и удобного ресурса.

1. ТЕСТИРОВАНИЕ

Темой дипломного проекта является "Разработка fronted сайта зоомагазина", основное назначение которого заключается в возможности предоставления пользователям удобного интерфейса для поиска, покупки и управления товарами для домашних животных. Цель проведения тестирования разработанного дипломного проекта – подтверждение качественных характеристик данного программного продукта:

* функциональная пригодность;
* уровень производительности;
* совместимость;
* удобство использования (юзабилити).

В соответствии с темой дипломного проекта необходимо разработать план тестирования. Ниже приведен план тестирования для Web-приложения.

### 4.1. Тест-план

Тестирование Web-приложения по теме "Разработка fronted сайта зоомагазина".

### 4.1.1. Введение

Цель данного документа – создание тест плана Web-приложения. Представленный тест-план оценивает подходы к тестированию данного продукта. Основное назначение данного тест-плана является усвоение навыков проверки качества созданного программного продукта.

### 4.1.2. Объекты тестирования

Объектами тестирования являются:

* функциональные;
* нефункциональные:
* [производительност](https://ru.wikipedia.org/w/index.php?title=%D0%A3%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8&action=edit&redlink=1)ь;
* [совместимость](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B2%D0%BC%D0%B5%D1%81%D1%82%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C);
* [удобство использования](https://ru.wikipedia.org/wiki/%D0%AE%D0%B7%D0%B0%D0%B1%D0%B8%D0%BB%D0%B8%D1%82%D0%B8).

### 4.1.3. Что будет тестироваться

По методике чек-листа тестируем удобство использования, совместимость, уровень производительности. Функциональная пригодность будет протестирована по методике тест-кейсов.

### 4.1.4. Что не будет тестироваться

Тестирование не будет затрагивать надежность, защищенность, сопровождаемость, переносимость, так как данные качественные характеристики затрагивают серверную часть приложения.

### 4.1.5. Подход к тестированию

По доступу к исходному коду – Black Box

По запуску кода – динамическое тестирование

По степени автоматизации – ручное

По объекту тестирования:

* функциональное;
* нефункциональное.

По требованиям

* позитивное.

По степени подготовленности - интуитивное тестирование.

### 4.1.6. Критерии прохождения тестов

Все тест-кейсы, имеющие высокий приоритет, закрыты с результатом “пройдено”.

Тестовое покрытие проверено и являются достаточным, где критерий достаточности составляет не менее 99% покрытия требований тестами.

### 4.1.7. Результаты проведения тестирования

Тест-план, чек-лист, тест-кейсы, отчёт о дефектах.

### 4.1.8. Задачи для проведения тестирования

Таблица 4.1 – Задачи для проведения тестирования

|  |  |
| --- | --- |
| Задача | Расположение |
| Написание тест-плана | Создание тест плана, обязанности |
| Написание чек-листа | Объекты тестирования, обязанности |
| Написание тест-кейсов | Объекты тестирования, обязанности |
| Проведение тестирования и оценка результатов | Подход к тестированию, обязанности |
| Создание отчетов о результатах тестирования | Результаты проведения тестирования |

### 4.1.9. Технические требования

Тестирование Web-приложения будет происходить на следующей операционной системе – Windows 10. В следующих браузерах: Chrome, FireFox.

### 4.1.10. Обязанности

Обязанности по проведению тестирования возложены на разработчика ПП Рудак Ксению Дмитриевну.

### 4.1.11. Необходимые компетенции и тренинги

Для выполнения поставленных у задач необходимо обладать следующими знаниями и умениями:

- знания и умения применения на практике стандарта IEEE-829;

- знания и умения применить на практике основных техник тест дизайна;

- знание различных типов тестирования, в том числе функционального и нефункционального.

### 4.1.12. Расписание / срок сдачи

Срок окончания всех работ и сдачи проекта – 10/06/2024 09:00:00

### 4.1.13. Риски

Возможные риски во время тестирования:

* недостаточное количество знаний для тестирования приложения в установленные сроки;
* задержки в сроках выполнения работ, вызванные другими обстоятельствами.

### 4.1.14. Утверждение

|  |  |
| --- | --- |
| Тестирование выполнил учащаяся группы 9ПЗ-55 Рудак Ксения Дмитриевна | Дата «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

## 4.2. Чек-листы

Удобство использования:

* интуитивно понятный и простой интерфейс пользователя (навигация главного меню доступна, все оформлено в виде простых и интуитивно понятных иконок);
* читаемость выводимой информации из базы данных (шрифт, размер текста, выравнивание ячеек);
* все элементы находятся в поле зрения и симметрично расположены относительно друг друга;
* гибкость и удобство использование (у пользователей есть возможность совершать часто повторяющиеся действия как можно быстрее и с наименьшими усилиями).

Совместимость:

* браузер Chrome;
* браузер Opera.

Уровень производительности:

* скорость загрузки сайта (с учетом отправки запроса на сервер) с интернетом, превышающим 5 мб/с, составляет менее 1 с;
* скорость загрузки сайта с интернетом от 1 мб/с – 5 мб/с составляет менее 2 с;
* скорость загрузки сайта с интернетом менее 1 мб/с составляет до 4 с.

## 4.3. Тест-кейсы программного средства

Таблица 4.2 – Тест-кейсы приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Описание | Шаги, входные данные | Ожидаемый результат | Фактический результат | Статус |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 4.2.1 | Запуск программы | В адресной строке браузера ввести url: http://localhost:3000 | Загрузилась главная страница | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.2 | Переход на форму регистрации | Нажимаем на кнопку регистрации | Загрузилась форма регистрации | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.3 | Переход на форму авторизации | Нажимаем на кнопку авторизации | Загрузилась форма авторизации | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.4 | Переход на страницу товара | Нажимаем на товар | Загрузилась страница выбранного товара | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.5 | Переход на профиль пользователя | Нажимаем на иконку пользователя после чего на кнопку профиль | Загрузилась форма профиля пользователя | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.6 | Выполнение поиска | Нажимаем на поле поиска, вводим название товара, нажимаем на Enter | Отображение товаров, подходящих под поиск | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.7 | Получение товара по категории | Нажимаем на категорию | Отображение товаров из заданной категории | Ожидаемый результат подтвержден | Пройден успешно |

Продолжение таблицы 4.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 4.2.8 | Переход на страницу заказа | Нажимаем на кнопку "Корзина" | Страница корзины пользователя | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.9 | Создание заказа | На странице "Корзина" выбираем товары и нажимаем заказать | Уведомление об успешном заказе | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.10 | Переход на страницу администратора | На главной странице нажать на кнопку "Admin" | Открывается страница администратора | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.11 | Ошибка входа обычного пользователя на страницу администратора | Ввод в адресную строку http://localhost:3000/admin | Ошибка с кодом 404 | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.12 | Создание товара администратором | Нажать на кнопку "Создать" на вкладке "Товары" с корректными данными (название-строка, описание-строка, цена-число, фото в формате картинки меньше 5Мб) | товар создаться и клиент получит статус код 200 | Ожидаемый результат подтвержден | Пройден успешно |
| 4.2.13 | Ошибка создания товара администратором | Нажать на кнопку "Создать" на вкладке "Товары" с не корректными данными(цена-строка) | товар создаться и клиент получит статус код 400 | Ожидаемый результат подтвержден | Пройден успешно |

## 4.4. Вывод о результатах тестирования

При проведении тестирования Web-приложения на тему «Разработка fronted сайта зоомагазина» дефектов не обнаружено, программный продукт готов к эксплуатации.

1. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

## Исходная информация

Качество и глубина проработки экономических вопросов в дипломном проекте определяется тем, насколько полно собраны данные, необходимые для экономического анализа и расчетов. Эту работу необходимо выполнять в период прохождения преддипломной практики в соответствии с методическими указаниями, содержащимися в программе преддипломной практики.

Исходная информация для расчета экономического раздела получена в Локомотивном депо Витебск УП "Витебское отделение Белорусской железной дороги" указана в таблице 5.1.

Таблица 5.1 - Исходная информация

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель | Условное обозначение | Единицы измерения | Значение, рублей |
| Стоимость компьютера | К | Рублей | 2400 |
| Мощность, потребляемая компьютером | МК | КВт/час | 0,310 |
| Полезный фонд времени работы компьютера за год | FK | Час | 2032 |
| Стоимость принтера | П | Рублей | 900 |
| Мощность, потребляемая принтером | МП | КВт/час | 1,12 |
| Полезный фонд времени работы принтера за год | FП | Час | 1016 |
| Стоимость 1 кВт/час (с учетом НДС) | С | Рублей | 0,239 |
| Норма амортизации для оборудования годовая | На | Проценты | 20 |
| Часовая тарифная ставка 13 разряда | ЧТС13 | Рублей | 2,37 |

Руководитель дипломного проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

* 1. Определение себестоимости программного обеспечения (ПО), как базы для формирования цены

В современных рыночных экономических условиях ПО выступает преимущественно в виде продукции научно-технических организаций, представляющей собой функционально завершенное и имеющее товарный вид ПО, реализуемое покупателям по рыночным отпускным ценам, установленным в результате анализа рыночных условий и переговоров с заказчиком и возможностей тиражирования. Все завершенные разработки ПО являются научно-технической продукцией.

Широкое применение вычислительной техники требует постоянного обновления и совершенствования ПО. Выбор эффективных проектов ПО связан с их экономической оценкой и расчетом экономического эффекта, который может определяться как у разработчика, так и у пользователя.

У разработчика экономический эффект выступает в виде чистой прибыли, остающейся в распоряжении предприятия от реализации ПО, а у пользователя – в виде экономии трудовых, материальных и финансовых ресурсов, получаемой за счет:

* снижения трудоемкости расчетов и алгоритмизации программирования и отладки программ, за счет использования ПО в процессе разработки автоматизированных систем обработки данных;
* сокращения расходов на оплату машинного времени и других ресурсов на отладку программ;
* снижения расходов на материалы;
* ускорение ввода в эксплуатацию новых систем;
* улучшения показателей основной деятельности предприятий в результате использования ПО.

Стоимостная оценка ПО у разработчиков предполагает составление сметы затрат, которая включает следующие статьи:

* затраты на материалы;
* стоимость машинного времени;
* основная заработная плата;
* дополнительная заработная плата;
* отчисление в фонд социальной защиты населения;
* обязательное страхование от несчастных случаев на производстве и профессиональных заболеваний;
* прочие затраты;
* расходы на сопровождение и адаптацию.

На основании сметы затрат рассчитывается себестоимость ПО. Расчет ведется в следующей последовательности:

* + 1. Расчет затрат на материалы

По статье «Материалы» отражаются расходы на магнитные носители, бумагу, красящие ленты и другие материалы, необходимые для разработки ПО. Расчет затрат на материалы (М), необходимые для разработки программного обеспечения (ПО) осуществляется по нормативу (Нр) на 100 команд (принимаем равным 1 руб.), с учетом общего объема команд (VкПО ) (берется учащимся по факту, исходя из полученного им количества команд при разработке ПО).

Определение затрат на материалы осуществляется по формуле, в рублях:

М= (5.1)

М=

* + 1. Обоснование трудоемкости разработки ПО

Обоснование трудоемкости разработки ПО ведется следующим образом: на основании опытно-статистического (суммарного) метода, в целом на все разработку, исходя из того, что ее выполняет один программист 2-ой категории. Данные о трудоемкости разработки, определенные опытно-статистическим путём, подтверждаю, трудоемкость – 130 часов.

Общий объём программного обеспечения – количество команд (VkПО) – 146.

Руководитель дипломного проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

* + 1. Расчет стоимости машинного времени

Определение стоимости потребляемой электроэнергии осуществляется по формуле, в рублях:

Сэл= (МК×FK + МП ×FП) × С × Квр×Кс, (5.2)

где Квр - коэффициент, учитывающий использование, повремени (Квр=0,8)

Кс - коэффициент, учитывающий потери в сети (Kc=1,05)

Сэл= (0,310×2032 + 1,12 ×1016**)** × 0,239 × 0,8 ×1,05= 354,9 (руб.)

Определение суммы амортизационных отчислений осуществляется по формуле, в рублях:

Сао= (К+П) ×На/100, (5.3)

Сао= (2400+900) ×20/100=660 (руб.)

Результаты расчетов сводим в таблицу 2.

Таблица 5.2 - Определение стоимости электроэнергии и амортизационных отчислений:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Условное | Значение, рублей | |
| Показатель | обозначение | На год  (ЗМг) | На 1  машино-час (СМч) |
| Стоимость потребляемой электроэнергии | Сэл | 354,9 | 0,17 |
| Сумма ежегодных амортизационных отчислений | Сао | 660 | 0,32 |
| Итого: | ЗМг | 1014,9 | 0,49 |

Расчет затрат на 1 машино-час ведется исходя из стоимости машинного времени на год (графа 3 таблицы 1) деленный на полезный фонд времени работы компьютера (FK.) (из таблицы «Исходная информация»).

Определение затрат на 1 машино-час осуществляется по формуле, в рублях:

СМч=ЗМг/FK , (5.4)

СМч=1014,9 / 2032 = 0,49 (руб.)

Определение стоимости машинного времени осуществляется по формуле, в рублях:

* + 1. СМвр=Тм×СМч, (5.5)

где Тм - машинное время работы в расчёте на программу, час. (70% от трудоемкости разработки программного обеспечения)

Смги - стоимость одного машино-часа, руб. (итог графы 4 таблицы 2)

СМвр= 91 × 0,49 = 44,59 (руб.)

* + 1. Расчет затрат на оплату труда

Расчёт основной заработной платы ведётся исходя из трудоемкости программного обеспечения.

Определение прямой заработной платы осуществляется по формуле, в рублях:

Зпр=Тi×ЧТС13, (5.6)

где Тi – трудоемкость программного обеспечения, ч.

ЧТС13 – часовая тарифная ставка 13 разряда, руб.

Зпр=130×2,37 = 308,1 (руб.)

Премия определяется как 30 % от основной заработной платы.

Определение премий осуществляется по формуле, в рублях:

П = Зпр×30/100, (5.7)

П = 308,1×30/100 =92,4 (руб.)

Дополнительная заработная плата (Зд) включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей) и определяется по нормативу в процентах к основной заработной плате.

Определение дополнительной заработной платы осуществляется по формуле, в рублях:

, (5.8)

где Зос – основная заработная плата;

ЗД – дополнительная заработная плата на конкретное ПО в руб.

НД – норматив дополнительной заработной платы принимается 10% от основной заработной платы.

= 40,05 (рублей)

Отчисления в фонд социальной защиты населения (3C3) определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной зарплаты (Зд).

Определение отчислений в фонд социальной защиты населения осуществляется по формуле, в рублях:



(5.9)

где Нсз – норматив отчислений в фонд социальной защиты населения (процент действующий норматив на момент расчета).

Обязательное страхование от несчастных случаев на производстве и профессиональных заболеваний (Зстр) берётся по нормативу, установленному в целом по организации (можно принять Нстр– 0,6 %).

= 149,78 (рублей)

Определение отчислений на обязательное страхование от несчастных случаев на производстве и профессиональных заболеваний осуществляется по формуле, в рублях:



(5.10)

= 2,64 рубля

* + 1. Расчет полной себестоимости программного обеспечения

Прочие затраты (Пз) включают расходы, связанные с необходимостью содержания аппарата управления, организационные расходы и расходы на общехозяйственные нужды. Принимаются (Нпз) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации (Нпз принимаем 99%).

Определение прочих затрат осуществляется по формуле, в рублях:

, (5.11)

= 396,52

Полная себестоимость без учета расходов на сопровождение и адаптацию (Сп) рассчитывается как сумма затрат на материалы, стоимости машинного времени, затрат на основную, дополнительную заработную плату, стоимости отчислений в фонд социальной защиты населения и на обязательное страхование от несчастных случаев на производстве и профессиональных заболеваний, прочих затрат.

Определение полной себестоимости осуществляется по формуле, в рублях:

Сп=М+СМвр+Зос+Зд+Зсз+Зстр+ПЗ (5.12)

Сп=1,46+44,59+400,5+40,05+149,78+2,64+396,52=1035,5 (рублей)

Кроме того, организация-разработчик осуществляет затраты на сопровождение и адаптацию ПО (РСА), которые определяются по нормативу НРСА (принимаем 4%), от полной себестоимости без учёта расходов на адаптацию (Сп).

Определение затрат на сопровождение и адаптацию ПО осуществляется по формуле, в рублях:



(5.13)

где Сп – полная себестоимость ПО без учёта расходов на сопровождение и адаптацию, руб.

Определение общей суммы расходов на разработку (с затратами на сопровождение и адаптацию) (Спл) осуществляется по формуле, в рублях:

 , (5.14)

Спл = 1035,5 + 41,42 = 1076,92 (рублей)

Результаты всех расчётов сводим в таблицу 3.

Таблица 5.3 – Определение полной себестоимости программного обеспечения

|  |  |  |
| --- | --- | --- |
| Наименование статей затрат | Сумма, руб. | Обоснование расчета |
| 1. Затраты на материалы | 1,46 | пункт 1.1. |
| 2. Стоимость машинного времени | 44,59 | пункт 1.3. |
| 3. Основная заработная плата | 400,5 | пункт 1.4 |
| 4.Дополнительная заработная плата | 40,05 | 10% от ст.3 (пункт 1.4) |
| 5. Итого, фонд оплаты труда (ФОТ) | 440,55 | (ст.3 + ст.4) (пункт 1.4) |
| 6. Отчисление в фонд социальной защиты населения | 149,78 | 34% от ст.5 (пункт 1.4) |
| 7. Обязательное страхование от несчастных случаев на производстве и профессиональных заболеваний | 2,64 | 0,6 % от ст.5 (пункт 1.4) |
| 8. Итого: отчисления на социальные нужды | 152,42 | (ст. 6 + ст.7) (пункт 1.4) |
| 9. Прочие затраты | 396,52 | 99% от ст.3 (пункт 1.5) |
| 10. Полная себестоимость без учёта расходов на сопровождение и адаптацию | 1035,54 | (ст.1+ст.2+ст.5+ст.8+ст.9) таблицы 3 (пункт 1.5) |
| 11. Расходы на сопровождение и адаптацию. | 41,42 | 4% от ст.10 таблицы 3 (пункт 1.5) |
| 12. Полная себестоимость с учётом расходов на сопровождение и адаптацию | 1076,96 | (ст.10 + ст.11) таблицы 3 (пункт 1.5) |

* 1. Выбор схемы монетизации программного продукта

Рассмотрим основных конкурентов с указанием их функций, стоимости, достоинств и недостатков:

1. Garfield.by:

* функции: просмотр каталога товаров с сортировкой по категориям, брендам и другим параметрам; доставка заказов и личный кабинет для управления информацией о питомцах и заказах;
* стоимость: цены варьируются от 3 до 200 белорусских рублей в зависимости от товара;
* достоинства: широкий ассортимент товаров для различных видов домашних животных; возможность управления информацией о питомцах в личном кабинете; контактную информацию для связи;
* недостатки: некорректная информация о способах и стоимости доставки.

1. E-zoo.by:

* функции: широкий ассортимент товаров для домашних животных, включая корма, аксессуары, средства по уходу и ветеринарные препараты; просмотр каталога товаров, фильтрация и сортировка по различным параметрам; доставка заказов по Беларуси; личный кабинет для управления информацией о питомцах и заказами;
* стоимость: цены варьируются от 2 до 400 белорусских рублей в зависимости от товара;
* достоинства: подробное описание товаров, возможность оставлять отзывы, удобная и интуитивно понятная навигация по сайту;
* недостатки: отсутствие возможности оплаты заказа онлайн, недостаток информации о доставке.

1. Zoobazar.by:

* функции: информация о товарах для животных, акциях и скидках, доставке, оплате; заказ товаров с доставкой по Беларуси; личный кабинет;
* стоимость: цены варьируются от 2 до 300 белорусских рублей в зависимости от товара;
* достоинства: большой ассортимент товаров для различных домашних животных; наличие скидок и акций; возможность самовывоза; регулярные обновления ассортимента;
* недостатки: медленный отклик службы поддержки на вопросы клиентов; сложности с возвратом товаров; отсутствие мобильной версии сайта.

Основными конкурентными преимуществами разрабатываемого fronted-сайта зоомагазина являются:

* подробные описания товаров с характеристиками и фотографиями.
* понятная и удобная навигация по сайту;
* возможность фильтрации товаров;
* информация о наличии товаров на складе;
* возможность добавления в "Избранное" или "Корзину";
* простой и интуитивно понятный процесс оформления заказа с различными способами оплаты и доставки;
* мобильная версия сайта.

Для монетизации fronted-сайта зоомагазина могут быть использованы следующие схемы:

* комиссионная модель: взимание комиссии с каждой совершенной через сайт покупки;
* подписная модель: предложение платной подписки для доступа к расширенному функционалу;
* модель "реклама": размещение рекламных баннеров и объявлений сторонних брендов на страницах сайта.

С учетом особенностей проекта и конкурентной среды наиболее оптимальной моделью монетизации представляется комиссионная.

Комиссионная модель предполагает, что продавец платит комиссию с каждой совершенной сделки платформе, на которой он размещает свои предложения. Размер комиссии обычно варьируется от 5% до 25% от общей суммы сделки в зависимости от категории товара, объема продаж продавца, и других факторов.

Преимущества комиссионной модели:

* платформа получает доход, пропорциональный объему продаж, что стимулирует ее развивать сервис и привлекать новых продавцов и покупателей;
* продавцам не нужно платить фиксированные взносы за размещение, что снижает входной барьер для новых продавцов;
* платформа может дифференцировать размер комиссии в зависимости от категории товара, что позволяет ей монетизировать свои услуги оптимальным образом.

Недостатки комиссионной модели:

* для платформы требуется выстраивание эффективной системы контроля и взимания комиссий;
* могут возникать споры относительно прозрачности и справедливости размера комиссионных отчислений.

В целом, комиссионная модель является одной из наиболее распространенных и гибких бизнес-моделей для онлайн-платформ электронной коммерции и позволяет получать стабильный доход с каждой совершенной покупки.

# ЗАКЛЮЧЕНИЕ

В ходе работы над дипломным проектом были получены следующие результаты:

* проведено исследование предметной области, включая изучение особенностей рынка и анализ целевой аудитории;
* выполнены постановка задачи и проектирование программного средства посредствам методологий UML и IDEF0;
* выбраны и использованы инструменты fronted-разработки, такие как HTML, CSS, JavaScript, библиотека React;
* реализован функционал, включая каталог товаров, корзину, личный кабинет пользователя;
* проведено тестирование программного средства;
* выполнены все необходимые экономические расчеты;
* разработана проектная документация.

Разработанный в рамках дипломного проекта frontend-сайт зоомагазина полностью удовлетворяет поставленным целям и задачам. Он обладает современным дизайном, удобным и интуитивно понятным интерфейсом, обеспечивает эффективное взаимодействие с целевой аудиторией и предоставляет широкий спектр функциональных возможностей. В ходе работы над проектом были развиты навыки веб-разработки, получен дополнительный профессиональный опыт.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Книга "Руководство по Figma" [Электронный ресурс]. Режим доступа:

<https://medium.com/slashdesigner/figma-guide-5235b8a8ab4f>

1. Справочник по HTML [Электронный ресурс]. Режим доступа:

<https://htmlbook.ru/html>

1. Справочник по CSS [Электронный ресурс]. Режим доступа:

<http://htmlbook.ru/css>

1. Привет, мир – React [Электронный ресурс]. Режим доступа:

<https://ru.legacy.reactjs.org/docs/getting-started.html>

1. Библиотека CSS-компонентов [Электронный ресурс]. Режим доступа:

<https://flowbite.com/>

# ПРИЛОЖЕНИЕ 1

ИСХОДНЫЙ ТЕКСТ ПРОГРАММНОГО СРЕДСТВА

//AdminPage.jsx

import React from "react"

import ProductLayout from "../components/layouts/ProductLayout";

import OrdersLayout from "../components/layouts/OrdersLayout";

import { useState, useEffect } from "react";

import Cookies from "js-cookie";

import Tabs, {Tab} from "../components/tabs/AdminTabs";

const AdminPage = () => {

    const [user, setUser] = useState()

    useEffect(() => {

        const cookieUser = JSON.parse(Cookies.get('user'))

        console.log(cookieUser);

        setUser(cookieUser)

    }, [])

    if (!user) {

        return <>loading</>

    }

    if (!user.is\_super) {

        return <>404 not found</>

    }

    return (

        <div className="">

            <Tabs>

                <Tab title="Товары">

                    <ProductLayout />

                </Tab>

                <Tab title="Заказы">

                    <OrdersLayout />

                </Tab>

            </Tabs>

        </div>

    )

}

export default AdminPage;

// Bucket.jsx

import React, { useEffect, useState } from "react"

import BucketProduct from "../components/BucketProduct";

import Cookies from "js-cookie";

import { baseUrl } from "../constants";

import { FaPen } from "react-icons/fa";

import { useToggle } from "../hooks/useToggle";

import AddressEditModal from "../modals/AddressEditModal";

import CardsProfile from "../components/CardsProfile";

import { ToastContainer, toast } from "react-toastify";

import 'react-toastify/dist/ReactToastify.css';

export const Bucket = (props) => {

    const [products, setProducts] = useState()

    const [selectedProducts, setSelectedProducts] = useState([])

    const [isLoad, setIsLoad] = useState(false)

    const price = 30

    const [all, setAll] = useState(false)

    const [address, setAddress] = useState({ city: "г. Минск", street: "Чурлениса", house: "д. 1", flat: "кв. 612" })

    const { isOpen, toggle } = useToggle()

    const [cards, setCards] = useState([

        { cardID: "1234 1234 1234 1234", holderName: "valera pika", isMain: true },

        { cardID: "1234 1234 1234 4321", holderName: "valera pika", isMain: false },

    ])

    const [delivery, setDelivery] = useState([

        { label: "Доставка на дом", price: 30 },

        { label: "Самолично", price: 0 },

    ])

    const [activeDelivery, setActiveDelivery] = useState(null)

    useEffect(() => {

        if (Cookies.get('user')) {

            const userID = JSON.parse(Cookies.get('user'))?.id

            const fetchData = async () => {

                let response = await fetch(`${baseUrl}/v1/products?user\_id=${userID}`)

                let jsonResponse = await response.json()

                setProducts(jsonResponse)

                setIsLoad(prev => !prev)

            }

            fetchData()

        }

    }, [])

    const handleClick = (products) => {

        if (!Cookies.get('user')) {

            toast.error('Вы не вошли в систему', {

                position: "bottom-right",

                autoClose: 3000,

                hideProgressBar: false,

                closeOnClick: true,

                pauseOnHover: true,

                draggable: true,

                progress: undefined,

            })

        } else {

            products.forEach(val => {

                const fetchData = async () => {

                    let id = JSON.parse(Cookies.get('user')).id

                    let request = { status\_id: 1, product\_id: val.id, user\_id: id }

                    await fetch(`${baseUrl}/v1/orders`, {

                        method: "POST",

                        body: JSON.stringify(request)

                    }).then(res => toast.success('Заказ принят', {

                        position: "bottom-right",

                        autoClose: 3000,

                        hideProgressBar: false,

                        closeOnClick: true,

                        pauseOnHover: true,

                        draggable: true,

                        progress: undefined,

                    })).catch(err => toast.error('Не удалось принять заказ', {

                        position: "bottom-right",

                        autoClose: 3000,

                        hideProgressBar: false,

                        closeOnClick: true,

                        pauseOnHover: true,

                        draggable: true,

                        progress: undefined,

                    }))

                }

                fetchData()

            })

        }

    }

    if (!Cookies.get('user')) {

        return <div className="flex justify-center items-center h-full">ВЫ НЕ ЗАРЕГИСТРИРОВАНЫ</div>

    }

    if (!isLoad) {

        return (

            <>

                <svg aria-hidden="true" role="status" class="inline w-4 h-4 me-3 text-white animate-spin" viewBox="0 0 100 101" fill="none" xmlns="http://www.w3.org/2000/svg">

                    <path d="M100 50.5908C100 78.2051 77.6142 100.591 50 100.591C22.3858 100.591 0 78.2051 0 50.5908C0 22.9766 22.3858 0.59082 50 0.59082C77.6142 0.59082 100 22.9766 100 50.5908ZM9.08144 50.5908C9.08144 73.1895 27.4013 91.5094 50 91.5094C72.5987 91.5094 90.9186 73.1895 90.9186 50.5908C90.9186 27.9921 72.5987 9.67226 50 9.67226C27.4013 9.67226 9.08144 27.9921 9.08144 50.5908Z" fill="#E5E7EB" />

                    <path d="M93.9676 39.0409C96.393 38.4038 97.8624 35.9116 97.0079 33.5539C95.2932 28.8227 92.871 24.3692 89.8167 20.348C85.8452 15.1192 80.8826 10.7238 75.2124 7.41289C69.5422 4.10194 63.2754 1.94025 56.7698 1.05124C51.7666 0.367541 46.6976 0.446843 41.7345 1.27873C39.2613 1.69328 37.813 4.19778 38.4501 6.62326C39.0873 9.04874 41.5694 10.4717 44.0505 10.1071C47.8511 9.54855 51.7191 9.52689 55.5402 10.0491C60.8642 10.7766 65.9928 12.5457 70.6331 15.2552C75.2735 17.9648 79.3347 21.5619 82.5849 25.841C84.9175 28.9121 86.7997 32.2913 88.1811 35.8758C89.083 38.2158 91.5421 39.6781 93.9676 39.0409Z" fill="currentColor" />

                </svg>

            </>

        )

    }

    return (

        <div className="flex w-full flex-col">

            <div className="flex gap-20 w-full">

                <div className="flex flex-col gap-5 bg-[#b5b2d051] pr-16 pt-5 pb-5 rounded-[20px] ml-5 w-[60%]">

                    <div className="flex justify-between">

                        <h1 className="font-bold text-2xl ml-3">Корзина ({products?.length})</h1>

                        <div className="flex items-center gap-3">

                            Выбрать все

                            <input type="checkbox" value={all} onChange={() => setAll(pre => !pre)} />

                        </div>

                    </div>

                    <div className="flex flex-col ml-3 ">

                        {products ? products.map((val) => (

                            <BucketProduct product={val} setSelected={setSelectedProducts} all={all} />

                        )) : <div className="flex justify-center items-center h-full">ТОВАРОВ В КОРЗИНЕ НЕТУ </div>}

                    </div>

                </div>

                <div className="bg-[#b5b2d051] rounded-[20px] flex flex-col w-[40%] h-[550px] p-9 justify-between mr-5">

                    <h1 className="font-bold text-2xl">Детали заказа</h1>

                    <div className="flex gap-5 text-xl">

                        <label>Адрес: </label>

                        <p className="underline cursor-pointer  ">{address.city}, {address.street}, {address.house}, {address.flat}</p>

                    </div>

                    <div className="flex gap-5 text-xl">

                        <label>Оплата: </label>

                        <p className="underline cursor-pointer  ">{cards.filter(val => val.isMain === true)[0].cardID}</p>

                    </div>

                    <div className="flex gap-5 text-xl">

                        <label>Доставка: </label>

                        <p className="underline cursor-pointer  ">{price} BYN</p>

                    </div>

                    <div className="flex gap-5 text-xl">

                        <label>Товары: </label>

                        <p className="underline cursor-pointer  ">{selectedProducts.length}</p>

                    </div>

                    <h1 className="text-2xl font-bold">Итого: {selectedProducts.reduce((prev, curr) => prev + curr.price, price).toFixed(2)} BYN</h1>

                    <button

                        className="bg-primary rounded-[20px] p-5 text-white text-2xl font-bold"

                        onClick={() => handleClick(selectedProducts)}

                    >Заказать</button>

                </div>

            </div>

            <div className="flex flex-col w-4/5 ml-5 mt-9 gap-5">

                <div className="flex justify-between w-3/5">

                    <AddressPicker address={address} toggle={toggle} />

                    <DeliveryPicker deliveryVariants={delivery} setActive={setActiveDelivery} />

                </div>

                <CardsProfile cards={cards} setCards={setCards} />

            </div>

            {isOpen && <AddressEditModal show={isOpen} onHide={toggle} initialAddress={address} onSave={setAddress} />}

            <ToastContainer />

        </div>

    )

};

export const AddressPicker = ({ address, toggle }) => {

    return (

        <div className="flex gap-3 bg-[#b5b2d051] p-2 rounded-[20px] items-center">

            <label>Адрес доставки: </label>

            <p

                className="underline cursor-pointer"

                onClick={toggle}

            >

                {address.city}, {address.street}, {address.house}, {address.flat}

            </p>

            <FaPen />

        </div>

    )

};

export const DeliveryPicker = ({ deliveryVariants, setActive }) => {

    const [isOpen, setIsOpen] = useState(false);

    const [selectedOption, setSelectedOption] = useState(null);

    const toggleDropdown = () => {

        setIsOpen(!isOpen);

    };

    const handleOptionClick = (option) => {

        setSelectedOption(option);

        setActive(option);

        setIsOpen(false);

    };

    return (

        <div className="relative select-none">

            <div

                className=" bg-[#b5b2d051] border border-gray-300 rounded-[20px] px-4 py-2 flex justify-between items-center cursor-pointer"

                onClick={toggleDropdown}

            >

                {selectedOption ? selectedOption.label : 'Способ доставки'}

                <span

                    className={` text-gray-500 transition-transform duration-300 ${isOpen ? 'transform rotate-180' : ''

                        }`}

                >

                    ▼

                </span>

            </div>

            {isOpen && (

                <div className="delivery-picker\_\_options absolute z-10 bg-[#b5b2d051] border border-gray-300 rounded-[20px] mt-1 w-full">

                    {deliveryVariants.map((option, index) => (

                        <div

                            key={index}

                            className={`px-4 py-2 cursor-pointer `}

                            onClick={() => handleOptionClick(option)}

                        >

                            {option.label}

                        </div>

                    ))}

                </div>

            )}

        </div>

    );

};

// PageNotFound.jsx

import React from "react"

const PageNotFound = (props) => {

  return (

    <div>

      404 page not found

    </div>

  )

};

export default PageNotFound;

// ProductPage.jsx

import React, { useEffect, useState } from "react"

import { useParams } from "react-router-dom";

import ProductComponent from "../components/ProductComponent";

import { baseUrl } from "../constants";

const ProductPage = (props) => {

  const { id } = useParams()

  const [product, setProduct] = useState({})

  const [isLoad, setIsLoad] = useState(false)

  useEffect(() => {

    const abortController = new AbortController()

    const { signal } = abortController

    async function fetchData() {

      try {

        const response = await fetch(`${baseUrl}/v1/products/${id}`, {

          signal: signal

        });

        const jsonData = await response.json();

        setProduct(jsonData);

        setIsLoad(!isLoad)

      } catch (error) {

        console.error(error);

      }

    }

    fetchData();

    return () => {

      abortController.abort()

    }

  }, []);

  return (

    <div className="">

      {isLoad && <ProductComponent product={product}/>}

    </div>

  )

};

export default ProductPage;

// ProfilePage.jsx

import React, { useState, useEffect } from "react"

import { UserContext } from "../components/contexts/UserContext";

import Cookies from "js-cookie";

import Profile from "../components/layouts/Profle";

import ProfileTabs from "../components/tabs/ProfileTabs";

import ProfileTab from "../components/tabs/ProfileTabs";

import Favourites from "../components/layouts/Favourites";

import { Tab } from "../components/tabs/AdminTabs";

import Delivery from "../components/layouts/Delivery";

import { useParams, useSearchParams } from "react-router-dom";

const ProfilePage = () => {

  const [param, setParam] = useSearchParams()

  const tab = param.get('tab')

  return (

    <div className="w-full">

      <ProfileTabs tab={tab ? +tab : 0}>

        <Tab className="flex w-full" title="Личные данные"><Profile/></Tab>

        <Tab className="flex w-full" title="Корзина"><Favourites/></Tab>

        <Tab className="flex w-full" title="Доставка"><Delivery/></Tab>

      </ProfileTabs>

    </div>

  )

};

export default ProfilePage;

// StartPage.jsx

import { useEffect, useState } from 'react'

import styled from "styled-components";

import dog from "../recources/images/Prelude\_to\_Reality-removebg-preview 1.png"

import CustomButton from "../components/UI/CustomButton";

import { CategoryBlock } from "../components/CategoryBlock";

import { ProductsContainer } from "../components/ProductsContainer";

import i1 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_190838-removebg-preview (1) 1.png"

import i2 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_191800-removebg-preview 1.png"

import i3 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_192137-removebg-preview 1.png"

import i4 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_193028-removebg-preview 1.png"

import i5 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_193550-removebg-preview 1.png"

import i6 from "../recources/images/products/Снимок\_экрана\_2023-12-17\_194131-removebg-preview 1.png"

import { Paginator } from "../components/Paginator";

import DeliveryPost from "../components/DeliveryPost";

import delivery1 from "../recources/images/delivery/logi57n 1.png"

import { DeliveryPostsContainer } from "../components/DeliveryPostsContainer";

import { baseUrl } from '../constants';

export function StartPage() {

  const [category, setCategory] = useState([])

  const [data, setData] = useState([])

  useEffect(() => {

    const abortController = new AbortController()

    const { signal } = abortController

    async function fetchData() {

      try {

        const products = await fetch(`${baseUrl}/v1/products`, {

          signal: signal

        });

        const categories = await fetch(`${baseUrl}/admin/categories`)

        const productsJsonData = await products.json();

        const categoriesJsonData = await categories.json()

        setData(productsJsonData.sort((a, b) => a.id - b.id))

        setCategory(categoriesJsonData.sort((a, b) => a.id - b.id))

      } catch (error) {

        console.error(error);

      }

    }

    fetchData();

    return () => {

      abortController.abort()

    }

  }, []);

  return (

    <Main>

      <PreviewContainer>

        <TextContainer>

          <ZooShop>

            ЗООМАГАЗИН

          </ZooShop>

          <Mozy>

            MOZZY

          </Mozy>

          <Container>

            <Rectangle />

            <Description>Товары для вашего питомца</Description>

          </Container>

        </TextContainer>

        <Dog src={dog} />

      </PreviewContainer>

      <CategoryContainer>

        {/\* <CategoryBlock categories={categories} values={values} /> \*/}

        <ProductsContainer categories={category} products={data} />

      </CategoryContainer>

      {/\* <Paginator /> \*/}

    </Main>

  )

}

const CategoryContainer = styled.div`

  display: flex;

  justify-content: space-around;

  margin-top: 62px;

`

const StyledButton = styled(CustomButton)`

  align-self: center;

`

const Description = styled.div`

  font-weight: 400;

  font-size: 20px;

  line-height: 25px;

  margin-left: 10px;

`

const Container = styled.div`

  display: flex;

  flex-direction: row;

  align-items: center;

`

const Rectangle = styled.div`

  width: 80px;

  height: 1.5px;

  background-color: black;

  margin-left: 15px;

`

const Mozy = styled.h1`

  font-weight: 400;

  font-size: 180px;

`

const ZooShop = styled.span`

  font-weight: 400;

  font-size: 24px;

  line-height: 30px;

  margin-left: 15px;

`

const TextContainer = styled.div`

  margin-top: 138px;

  display: flex;

  flex-direction: column;

`

const Dog = styled.img`

  @media (max-width: 768px) {

    display: none !important;

  }

`

const PreviewContainer = styled.div`

  display: flex;

  flex-direction: row;

  justify-content: space-around;

  background: rgba(181, 178, 208, 1);

  border-radius: 20px;

  margin: 0 20px;

`

const Main = styled.div`

  margin-top: 10px;

  display: flex;

  flex-direction: column;

`

# ПРИЛОЖЕНИЕ 2

БЛОК-СХЕМА АЛГОРИТМА РАБОТЫ ПРОГРАММНОГО СРЕДСТВА

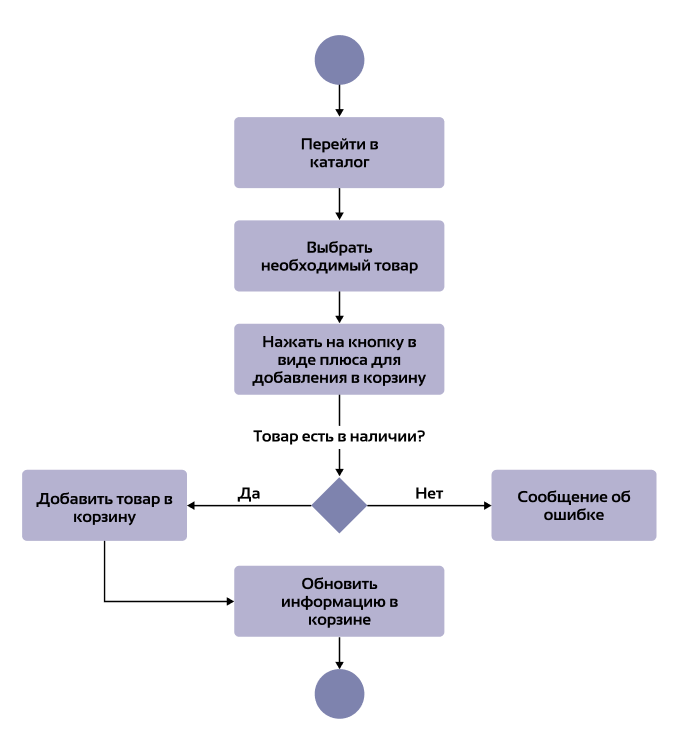


Рисунок П2.1 – Блок-схема "Добавление товара в корзину"

# ПРИЛОЖЕНИЕ 3

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Сайт представляет собой frontend-приложение для зоомагазина. Он состоит из 7 страниц, каждая из которых содержит различную информацию. Два основных элемента сайта - "Шапка" и "Подвал" - присутствуют на всех страницах и располагаются в начале и конце каждой из них.

Навигационная панель расположена в верхней части каждой страницы сайта. Через нее можно перейти на другие страницы сайта, такие как "Главная" (при нажатии на логотип), "Admin", "Корзина", "Личный кабинет".



Рисунок П3.1 – Навигационная панель

Подвал сайта располагается в нижней части каждой страницы. В нем содержится телефон и почта зоомагазина.



Рисунок П3.2 – Подвал сайта

На главной странице сайта расположены категории товаров, предоставляемых зоомагазином.

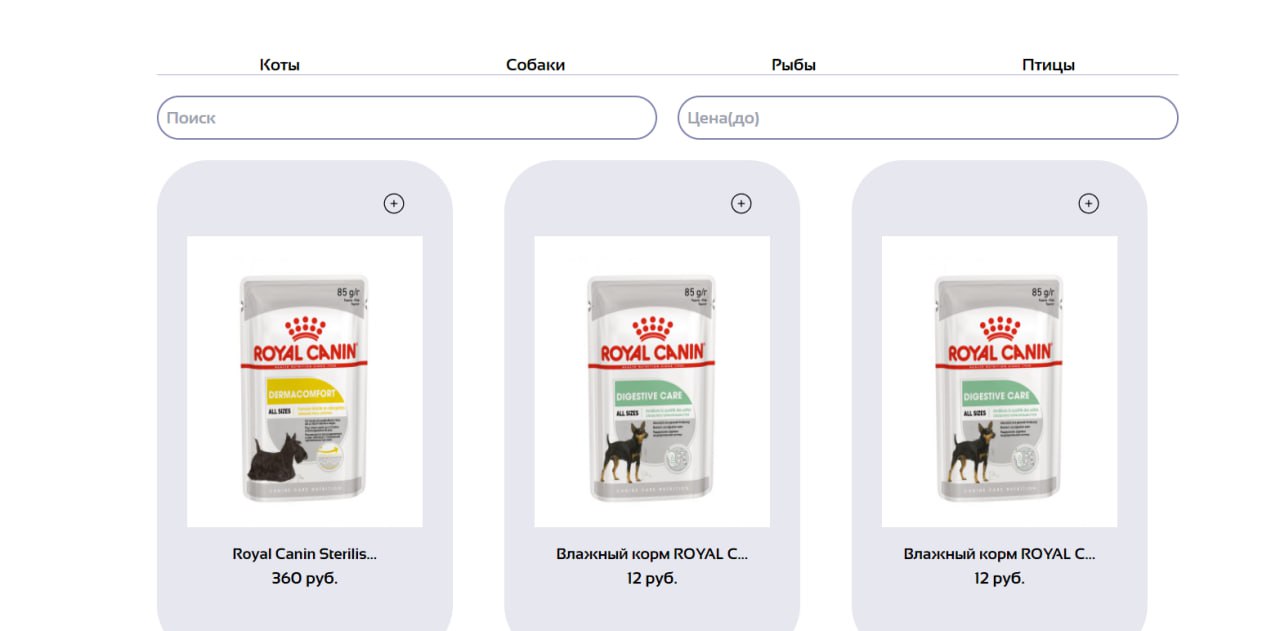


Рисунок П3.3 – Категории товаров на главной странице

При регистрации на сайте пользователь получает доступ к личному кабинету. Здесь он может управлять персональными данными, просматривать свои заказы и товары в корзине.

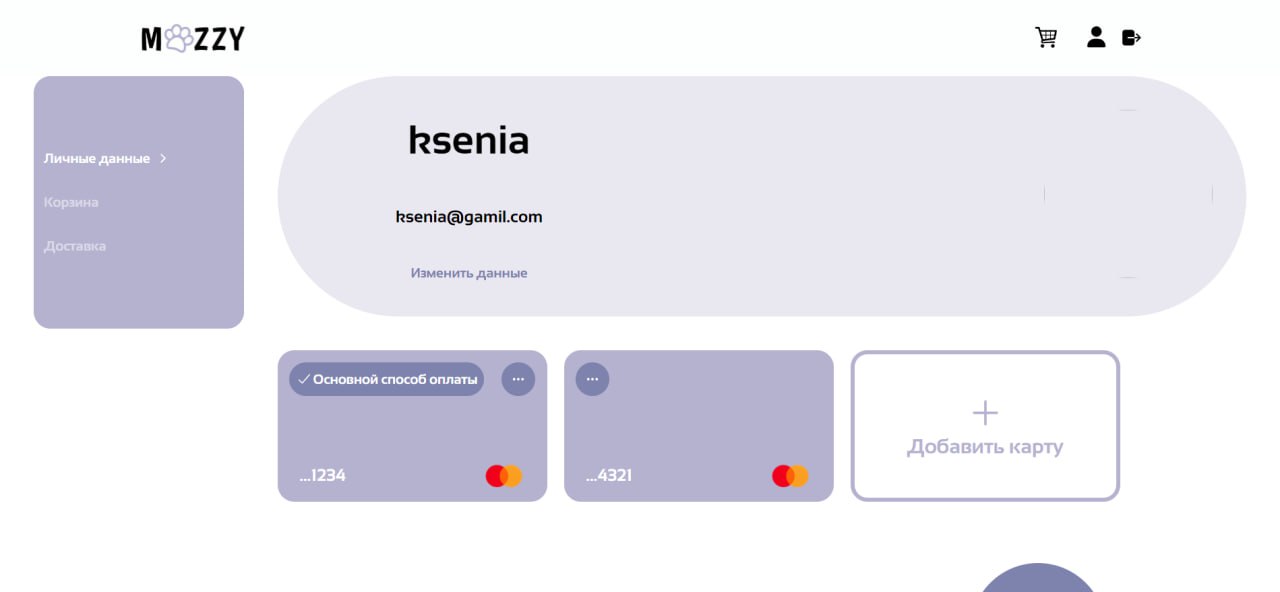


Рисунок П3.4 – Личный кабинет

В процессе выбора товаров на сайте пользователь может добавлять их в корзину. Корзина располагается в навигационном меню. При переходе на страницу корзины пользователь может просмотреть список добавленных товаров и удалять их. А также выбрать адрес, способ доставки, способ оплаты и перейти к оформлению заказа.

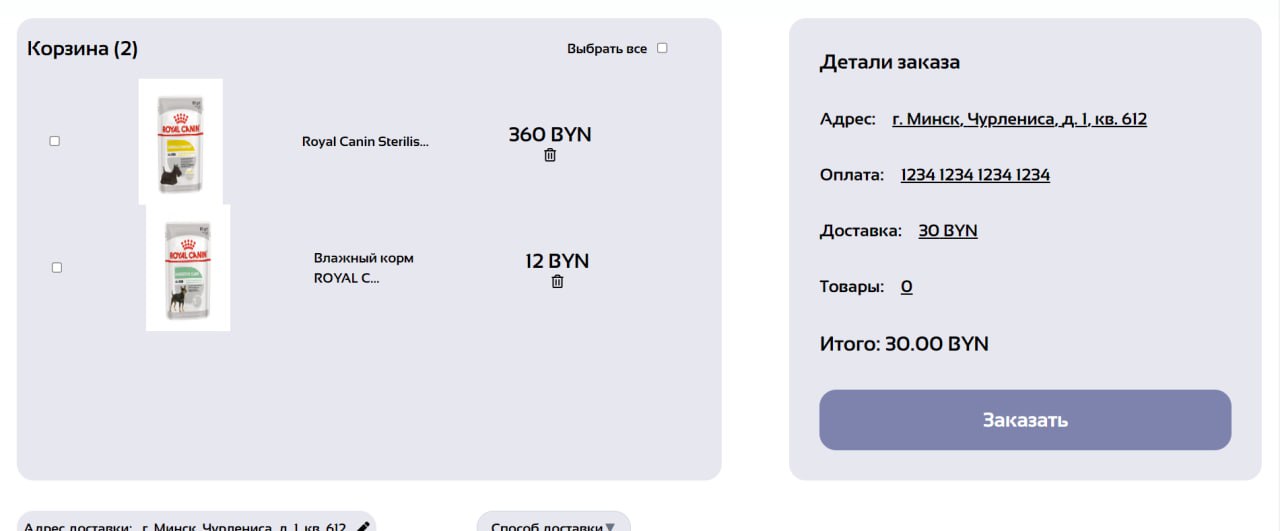


Рисунок П3.5 – Список товаров и детали заказа в корзине

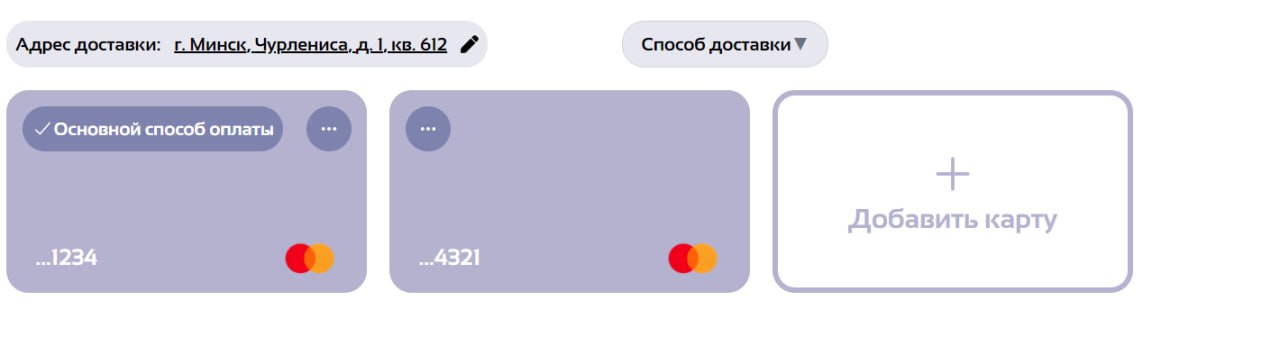


Рисунок П3.6 – Выбор способа оплаты, доставки и адреса доставки

Страница товара содержит изображение, название, цену, наличие на складе и кнопки, показывающие описание и состав, а также возможность добавить товар в корзину.

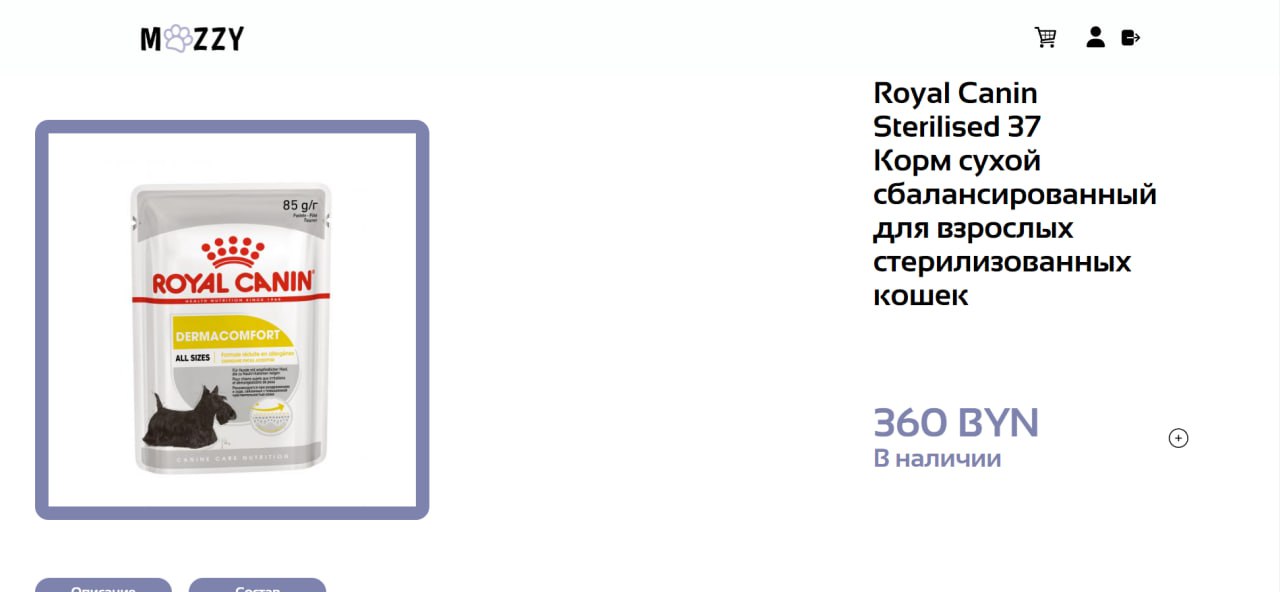


Рисунок П3.7 – Страница товара

Страница администрирования предоставляет администратору возможности добавления, редактирования и удаления товаров и управление заказами.

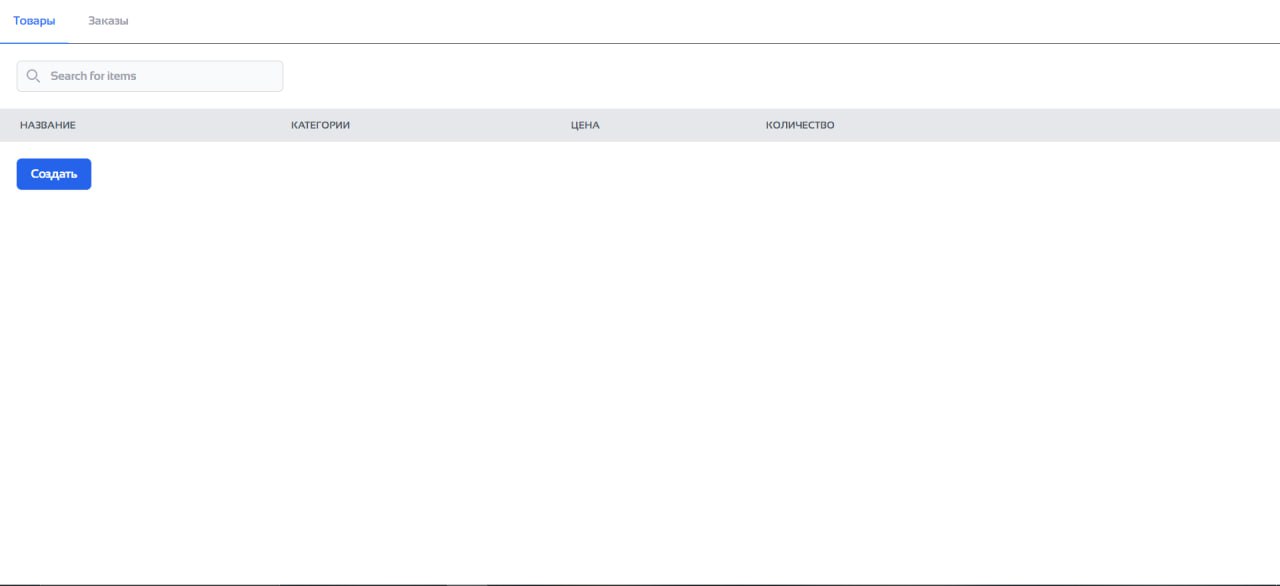


Рисунок П3.8 – Страница администрирования

# ПРИЛОЖЕНИЕ 4

ПРОТОКОЛ ИСПЫТАНИЙ ПРОГРАММНОГО СРЕДСТВА

Комиссия в составе

* руководитель дипломного проекта – С.Г. Сутович
* председатель цикловой комиссии – С.Г. Тыбербай
* преподаватель – А.А. Брылева

произвела проверку работоспособности программного средства, разработанного в порядке выполнения дипломного проекта.

1. Информация о программном средстве:
   1. Название программного средства: Fronted-сайта зоомагазина.
   2. Номер версии программного средства: 1.0.
   3. Дата разработки версии программного средства: 11.06.2024.
   4. Назначение программного средства: Программное средство предназначено для предоставления пользователям удобного интерфейса для поиска, покупки и управления товарами для домашних животных.
   5. Функции программного средства: Программное средство включает в себя каталог товаров, корзину и оформление заказа, личный кабинет пользователей и администратора.
   6. Язык программирования: HTML5, CSS3, React.js.
   7. Наличие программы инсталляции: нет.
   8. Тип компьютера(ов): ноутбук.
   9. Требования к компьютеру:
      1. Оперативная память: не менее 128 Mb.
      2. Ёмкость жесткого диска(ов) 500 Mb.
      3. Монитор(ы) с разрешением 800х600, с поддержкой 256 цветов.
      4. Дополнительное периферийное оборудование: манипулятор типа «мышь», принтер струйный или лазерный.
   10. Тип операционной системы (систем): Windows 10 и выше.
2. Информация об организации-заказчике:
   1. Полное официальное наименование: "Витебский государственный колледж электротехники".
   2. Место нахождения (почтовый адрес): 210015, г. Витебск, пр-т Черняховского, 14.
   3. Номер контактного телефона(ов): (8-0212)64-89-16.
   4. Номер факса: (8-0212)64-89-16.
   5. Адрес электронной почты: priem@[vgke.by](mailto:vgpt@vitebsk.by).
3. Информация о разработчике программного средства:
   1. Полное официальное наименование: "Витебский государственный колледж электротехники".
   2. Место нахождения (почтовый адрес): 210015, г. Витебск, пр-т Черняховского, 14.
   3. Номер контактного телефона(ов): +375-29-739-08-60.
   4. Номер факса:
   5. Адрес электронной почты: rudakkseniya99@gmail.com.
   6. Фамилия, имя, отчество учащегося, разработавшего программное средство: Рудак Ксения Дмитриевна.

Работоспособность программного средства проверена на контрольных примерах, предусматривающих все возможные варианты исходных и промежуточных данных.

Программное средство признано работоспособным, соответствующим техническому заданию и позволяет решать поставленные перед ним производственные задачи.

Руководитель дипломного проекта С.Г. Сутович

Председатель цикловой комиссии С.Г. Тыбербай

Преподаватель А.А. Брылева