

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка и администрирование web-приложения компании оптовой торговли фармацевтической продукцией на основе CASE-технологии»

Обучающийся

М.С. Сучков

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.ф.-м.н., О.В. Лелонд

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент С.А. Гудкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Бакалаврская работа выполнена на тему «Разработка и администрирование web-приложения компании оптовой торговли фармацевтической продукцией на основе CASE-технологии».

Объектом исследования данной бакалаврской работы является процесс разработки и администрирования web-приложения, облегчающего обработку заказов, а его предметом – эффективное применение методов, технологий и инструментов при разработке и администрировании web –приложения.

Цель исследования состоит в разработке эффективного и функционального web-приложения для фармацевтической компании.

Бакалаврская работа состоит из введения, трёх глав, заключения и списка использованной литературы.

Во введении представлена постановка задачи, освещается актуальность работы, а также предоставляется краткое описание общей структуры работы.

В первой главе рассматривается анализ предметной области, обзор и анализ существующих решений, формулирование требований к web-приложению, а также проводится постановка задачи на разработку web-приложения

Во второй главе производится детальное проектирование всех компонентов web-приложения. Это включает описание CASE-технологий, выбор средств реализации, обоснование архитектуры, проектирование пользовательских сценариев, логической и физической моделей данных.

Третья глава посвящена реализации требуемых функций web-приложения и проведению расчета экономической эффективности его внедрения.

В заключении производится обобщение результатов выполнения выпускной квалификационной работы.

Бакалаврская работа выполнена на 73 страницах, содержит 8 таблиц, 36 рисунков и 3 формулы.

Abstract

The title of the graduation work is "Development and Administration of a Pharmaceutical Wholesale Company's Web Application Based on CASE Technology".

The object of this study is the process of developing and administering a web application that facilitates order processing. The subject of the study is the effective application of methods, technologies, and tools in the development and administration of the web application.

The aim of the study is to develop an efficient and functional web application for a pharmaceutical company.

The graduation work consists of an introduction, three chapters, a conclusion, and a list of references. The introduction presents the problem statement, highlights the relevance of the work, and provides a brief description of the overall structure of the work.

The first chapter reviews the domain analysis, existing solutions, the formulation of requirements for the web application, and the problem statement for the web application development.

The second chapter involves detailed design of all web application components. This includes the description of CASE technologies, the selection of implementation tools, architecture justification, user scenarios design, and logical and physical data models.

The third chapter is devoted to the implementation of the required functions of the web application and the calculation of its economic efficiency.

The conclusion summarizes the results of the graduation project.

The graduation work is 73 pages long and contains 8 tables, 36 figures, and 3 formulas.

Оглавление

Введение.....	6
Глава 1 Анализ предметной области.....	8
1.1 Характеристика фармацевтической компании	8
1.2 Обзор и анализ существующих решений	10
1.3 Определение требований к web-приложению	14
1.4 Постановка задачи на разработку web-приложения	16
Глава 2 Проектирование компонентов по созданию web-приложения	18
2.1 Описание CASE-технологий.....	18
2.2 Выбор средств реализации.....	19
2.2.1 Выбор системы управления данными	19
2.2.2 Выбор языка программирования и фреймворка для разработки web-приложения	21
2.3 Обоснование архитектуры построения web-приложения	24
2.4 Проектирование пользовательских сценариев для web-приложения ООО «МК Перспектива».....	26
2.5 Проектирование логической модели данных в нотации idef1x .	29
2.6 Проектирование Физической модели с применением Case-технологии моделирования.....	33
Глава 3 Реализация web-приложения	36
3.1 Разработка процессов аутентификации и авторизации пользователей.....	36
3.2 Разработка административной панели	38
3.3 Разработка пользовательской панели	46
3.4 Описание web-приложения.....	50
3.4.1 Пользовательская панель	50
3.4.2 Обзор административной панели интернет-витрины	58

3.5 Тестирование web-приложения.....	62
3.6 Расчет экономической эффективности.....	64
Заключение	70
Список используемой литературы	72
Приложение А Программный код web-приложения.....	74

Введение

В наше современное информационное время Интернет становится ключевым инструментом для предоставления и поиска данных. Более половины мирового населения имеют доступ к сети, что делает создание веб-приложений неотъемлемой частью современного бизнеса. В этом контексте веб-программирование становится самостоятельной отраслью, позволяя компаниям развивать информационную поддержку, рекламу и эффективное взаимодействие с клиентами.

Проект по разработке веб-приложения для оптовой торговли фармацевтической продукцией на основе CASE-технологии и использованием современного фреймворка является важным шагом в направлении цифровой трансформации компаний в этой отрасли. Создание веб-приложения не только предоставит компании эффективный инструмент для управления информацией и взаимодействия с клиентами, но и расширит её онлайн присутствие, что важно в условиях постоянного роста пользовательской активности в сети.

Целью данной работы является создание универсального и удобного веб-приложения, предоставляющего оптовым клиентам доступ к фармацевтической продукции, сотрудникам – эффективные инструменты управления, а компании – новые возможности для привлечения клиентов и увеличения объемов продаж.

Объектом данной бакалаврской работы является процесс разработки и администрирования web-приложения, облегчающего обработку заказов.

Предметом исследования выпускной квалификационной работы является эффективное применение методов, технологий и инструментов при разработке и администрировании web -приложения

Для достижения поставленной цели предлагается решить ряд задач:

- дать характеристику исследуемой компании;

- проанализировать существующие аналоги и определить требования к веб-приложению;
- проанализировать и выбрать технологии разработки web-приложения для компании оптовой торговли фармацевтической продукцией;
- разработать и реализовать веб-приложение с использованием CASE-технологии и фреймворка;
- оценить экономическую эффективность внедрения web-приложения.

Методы проектного исследования включают в себя теоретический анализ, сравнение, проектирование, изучение литературы и анализ существующих решений в данной области.

Практическая значимость работы заключается в создании веб-приложения, которое автоматизирует процессы оптовой торговли фармацевтической продукцией, повышает эффективность взаимодействия с клиентами и способствует расширению бизнеса в онлайн-сфере. Кроме того, разработанное приложение значительно упрощает работу сотрудников, предоставляя им возможность быстрого доступа к информации, управления заказами и клиентским, данным.

Глава 1 Анализ предметной области

1.1 Характеристика фармацевтической компании

ООО «МК ПЕРСПЕКТИВА» занимается оптовой торговлей фармацевтической продукцией. Сфера деятельности компании включает посредническую деятельность между производителями фармацевтических товаров, розничными аптеками, медицинскими учреждениями и другими клиентами. Компания «МК ПЕРСПЕКТИВА», работающая в сфере оптовых поставок фармацевтических изделий, предлагает обширный выбор товаров. В их каталоге представлены дезинфекционные средства, медицинские добавки, оборудование для здравоохранения и другие продукты, необходимые для поддержания здоровья и лечения больных.

ООО «МК Перспектива» была создано в 2008 году, имеет миссию предоставления качественных фармацевтических товаров и услуг, основано на принципах качества, профессионализма и ответственности. Компания стремится предоставить своим клиентам высококачественные услуги в сфере торговли оптовой фармацевтической продукцией.

В своей работе «МК Перспектива» руководствуется несколькими принципами. Во-первых, компания придерживается высоких стандартов качества, гарантируя, что ее фармацевтическая продукция соответствует всем требованиям и нормативам. Во-вторых, она стремится к надежным и долгосрочным партнерским отношениям с клиентами, обеспечивая им стабильное снабжение необходимой продукцией. В-третьих, «МК Перспектива» активно следит за последними тенденциями и разработками в фармацевтической сфере, чтобы предлагать своим клиентам инновационные и эффективные продукты.

Стратегия компании основывается на развитии долгосрочных партнерских отношений с поставщиками и клиентами, а также на расширении ассортимента фармацевтической продукции. «МК Перспектива» стремится

занять лидирующую позицию на рынке поставщиков фармацевтических продуктов, обеспечивая своим клиентам широкий выбор качественных препаратов.

На рисунке 1 изображена структура компании «МК Перспектива», во главе которой стоит директор. Директор координирует работу пяти ключевых подразделений. Менеджер по закупкам занимается поиском и приобретением товаров и услуг, необходимых для компании. Бухгалтер обеспечивает финансовый учет и контроль за бюджетом. Менеджер по продажам отвечает за реализацию продукции и разработку стратегий продаж. IT отдел поддерживает работу компьютеров и внутренних программ компании. Водитель выполняет задачи, связанные с транспортировкой, доставляя товары или перевозя сотрудников [3].

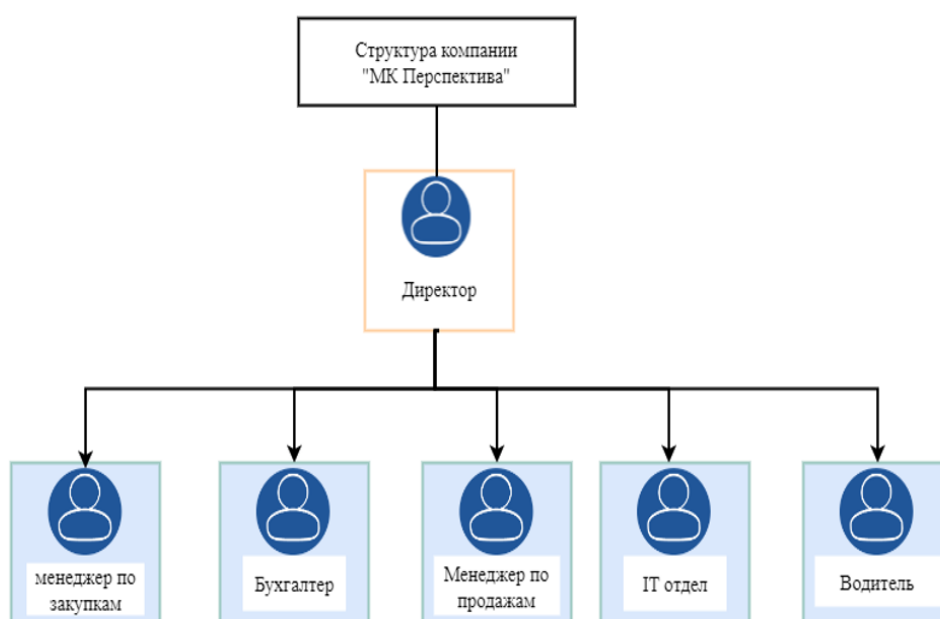


Рисунок 1 – Организационная структура «МК Перспектива»

Для обеспечения удобства потенциальных клиентов важно предоставить свободный доступ к информации о компании, ее расписанию работы и ценам

на фармацевтическую продукцию. Эта информация должна быть доступна клиентам в любое удобное для них время.

1.2 Обзор и анализ существующих решений

В современных условиях фармацевтическая отрасль сталкивается с растущим спросом на оптовые поставки, подчеркивая необходимость эффективных инструментов для продвижения и реализации продукции. Большинство имеющихся в данной области разработок представлены в формате интернет-витрины.

В данной сфере можно выделить несколько решений, которые заслуживают внимания.

Сайт «Алантера» (<https://alantera.ru/>) является важным игроком в сфере оптовой торговли фармацевтической продукцией, предоставляя клиентам доступ к широкому спектру медицинских товаров. Главная страница портала, представленная на рисунке 2, привлекает внимание своим современным дизайном и удобным навигационным интерфейсом.

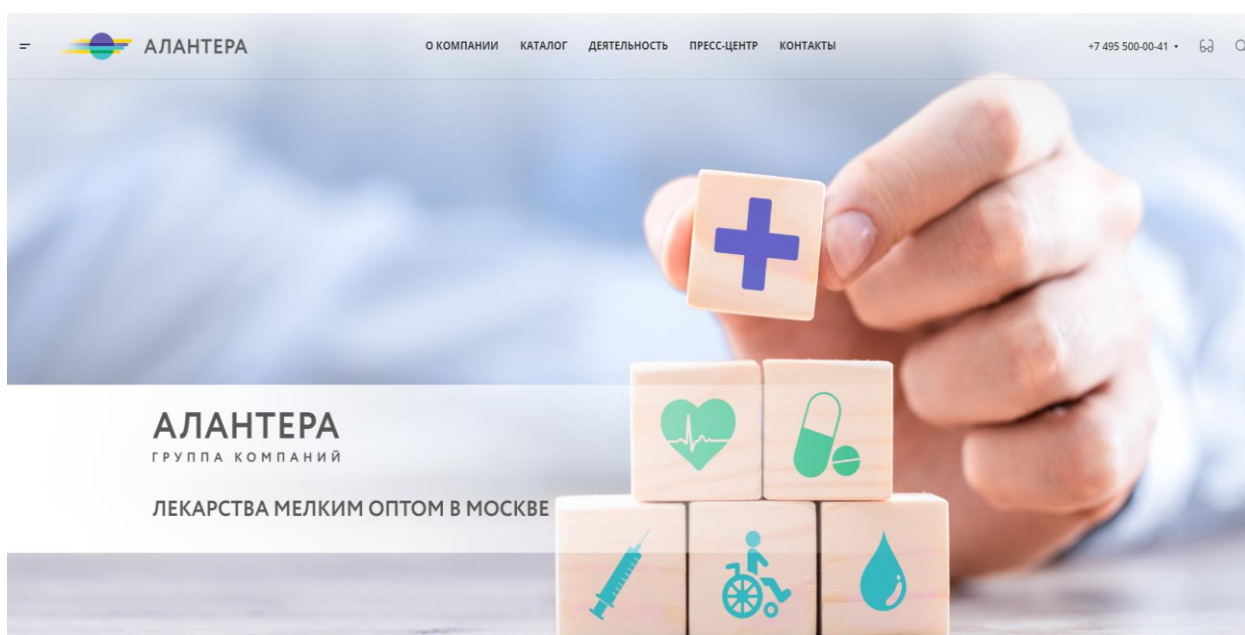


Рисунок 2 – Главная страница сайта «Алантера»

Ключевыми особенностями сайта является наличие онлайн-каталога, предоставляющего описание и изображения продукции. Это обеспечивает клиентам простой доступ к информации о медицинских товарах, их характеристиках. Но для получения информации о цене на продукцию, необходимо воспользоваться формой для обратной связи.

Сайт «Алантера» отличается высокой степенью прозрачности и информативности. На сайте предоставлена подробная информация о сертификации и качестве продукции, что важно для оптовых покупателей в сфере фармацевтики.

Дополнительным преимуществом является раздел «Акции», предлагающий клиентам информацию о текущих специальных предложениях и скидках. Это способствует стимулированию продаж и повышению интереса клиентов к продукции.

Сайт «Юмедика» (<https://u-medica.ru/>) является выдающимся онлайн-ресурсом, специализирующимся на оптовой торговле медицинским оборудованием и расходными материалами. Визуально привлекательная главная страница, представленная на рисунке 3.

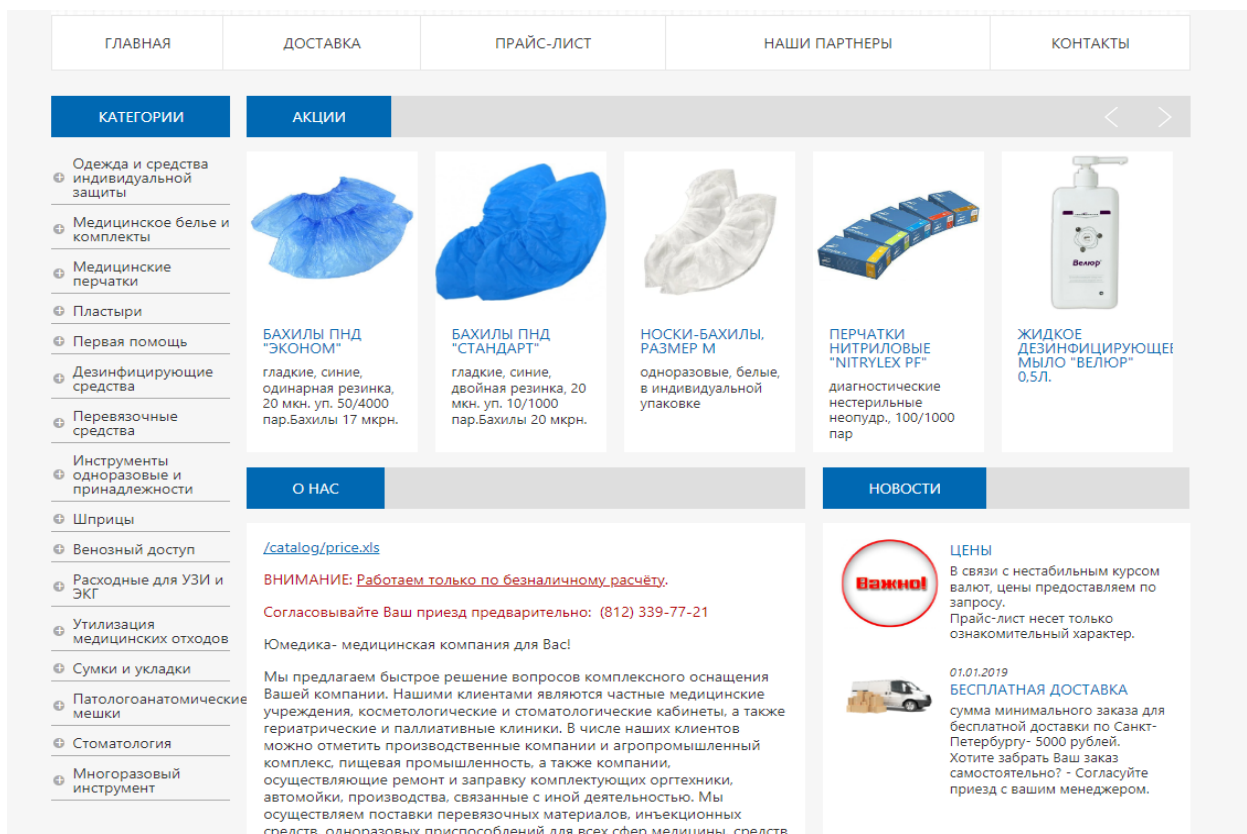


Рисунок 3 – Главная страница сайта «Юмедика»

Ключевыми особенностями сайта является подробный каталог, предоставляющий полную информацию о медицинских товарах. Посетители могут ознакомиться с детальными описаниями, изображениями и техническими характеристиками, хотя цены не предоставляются открыто. Однако, для получения прайс-листа, пользователи могут воспользоваться удобной онлайн-формой заказа.

Ресурс также предоставляет актуальную информацию о специальных предложениях и акциях, стимулируя продажи и привлекая новых клиентов.

Среди множества платформ оптовой медицинской торговли «Медмаркет» (<https://medmarket.pro/>) выделяется своей уникальной концепцией. Обзор главной страницы ресурса доступен на рисунке 4.

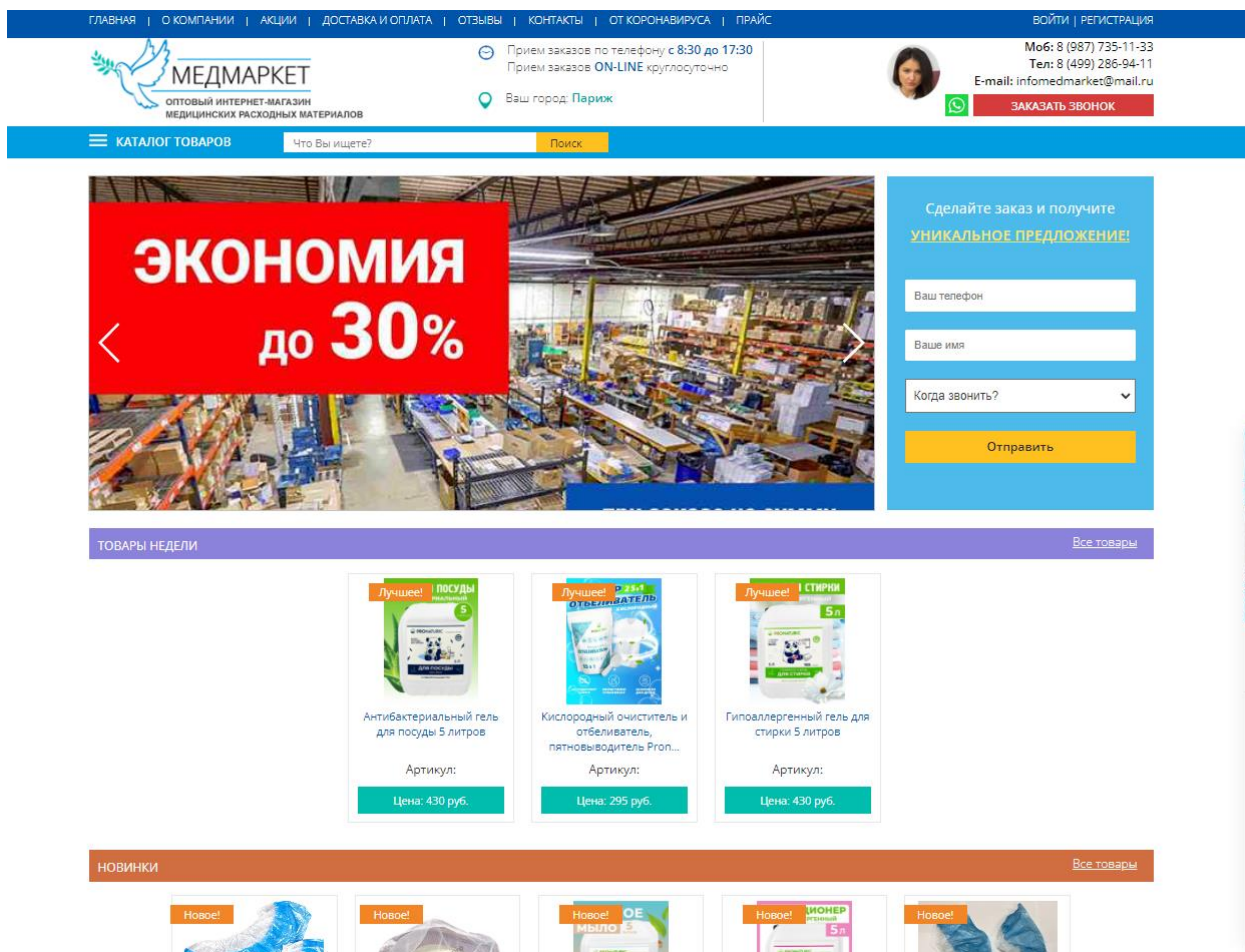


Рисунок 4– Главная страница сайта «Медмаркет»

Одной из ключевых особенностей сайта является разнообразный и детально проработанный каталог, предоставляющий обширную информацию о фармацевтической продукции. Пользователи могут ознакомиться с полными описаниями товаров, включая изображения и характеристики товаров.

Таблица 1 – Сравнительный анализ

Критерии оценки	Алантера	Юмедика	Медмаркет
Просмотр товара	+	+	+
Наличие формы для получения обратной связи от клиентов	+	+	+
Просмотр характеристик товара	+	+	+

Продолжение таблицы 1

Критерии оценки	Алантера	Юмедика	Медмаркет
Доступность цены на товар	-	-	+
Акции	-	+	+
Добавление товаров в корзину	-	-	-
Итого	3	5	5

В ходе анализа, представленного в таблице 1, стало ясно, что существующие сервисы не соответствуют требованиям нашего заказчика. На основе этого вывода представляется целесообразным разработать инновационное решение, полностью соответствующее потребностям нашего заказчика.

1.3 Определение требований к web-приложению

Глобальная сеть Интернет в настоящее время представляет собой открытое и доступное пространство, где можно решать разнообразные задачи. На этой платформе успешно решаются задачи, начиная от продвижения различных товаров и услуг через рекламу, и заканчивая интеграцией людей в области трудовой и учебной деятельности.

Конечно, широкий спектр доступных технологий значительно расширяет возможности по созданию и управлению веб-сайтами. Независимо от уровня опыта – будь то профессиональные разработчики, создающие интернет-магазины и корпоративные порталы, или новички, начинающие с простых визиток, личных страниц или работы в социальных сетях, – все они найдут подходящие инструменты. Большое разнообразие систем и технологий для веб-разработки позволяет удовлетворить потребности и предпочтения практически любого пользователя в интернете.

В веб-разработке можно выделить три ключевые категории инструментов и технологий:

- на основе стандартных решений, включая различные системы управления контентом (CMS), такие как WordPress, Joomla и Drupal. Эти системы, наряду с платформами для электронной торговли, включая Magento, Shopify и WooCommerce, предлагают широкий спектр готовых шаблонов и функциональных возможностей, что значительно упрощает процесс создания и управления сайтами;
- в сфере интеграции, решения как MuleSoft и Apache Camel, а также инструменты для работы с API, например, Swagger и Postman, играют ключевую роль. Они облегчают интеграцию различных систем, упрощая обмен данными и взаимодействие между компонентами системы;
- разработка сайтов как независимых проектов включает использование различных фреймворков для веб-разработки, таких как Django, Ruby on Rails и Laravel. Эти инструменты, в сочетании с языками программирования, такими как HTML, CSS, JavaScript, Python и PHP, предоставляют разработчикам возможности для создания индивидуальных и сложных веб-приложений «с нуля», предлагая гибкость в реализации уникальных проектов.

Учитывая детали работы рассматриваемой компании в фармацевтической сфере, постоянный мониторинг предоставленной информации имеет важное значение. Это обусловлено необходимостью поддержания высокого стандарта качества и соответствия всей предоставляемой информации законодательству и нормативам отрасли.

Основная цель, направленная на повышение эффективности работы предприятия, занимающегося продажей фармацевтических товаров, заключается в разработке веб-платформы, полностью соответствующей требованиям данного бизнеса. Веб-приложение должно быть оснащено необходимым функционалом, который способен обеспечить выполнение следующих задач:

- предоставление ключевой информации о фармацевтической компании;
- демонстрация основных продуктовых линеек, включающих разнообразную фармацевтическую продукцию;
- обеспечение возможности поиска товаров по их названию и категории;
- информирование посетителей о существующих акциях;
- наличие личного кабинета для клиентов, где они смогут отслеживать статус своего заказа;
- возможность добавления продукции в корзину.

1.4 Постановка задачи на разработку web-приложения

Необходимо разработать веб-приложение с функциональностью интернет-витрины для компании, специализирующейся на продаже фармацевтической продукции.

Для пользователей требуется создать возможность ознакомления с полным каталогом фармацевтической продукции, где каждый товар будет снабжен детальным описанием, указанием состава, инструкцией по применению и ценой. Они также должны получить возможность фильтровать товары по различным категориям и характеристикам для более удобного поиска нужных продуктов.

Для пользователей также необходимо создать систему авторизации и профилей. Клиенты смогут зарегистрироваться и войти в свои личные профили, где будут храниться их история заказов, предпочтения и контактные данные.

Для осуществления заказов пользователи должны получить возможность использовать специальную форму заказа, где будут указывать свои ФИО, адрес электронной почты, номер телефона и адрес доставки. Это

обеспечит им легкость и оперативность при оформлении заказов через интернет.

Администраторам требуется доступ к административной панели, через которую они смогут просматривать и управлять заказами. Они смогут подтверждать заказы, отслеживать статусы доставки, создавать новые карточки товаров.

Таким образом, при заполнении заявки на сайте, информация о клиенте будет автоматически заноситься в базу данных, что приведёт к сокращению времени, затрачиваемого сотрудниками на ручное формирование заявок.

Выводы по главе 1

В первой главе был проведен анализ фармацевтической компании «МК Перспектива», включая её виды деятельности, миссию и стратегию развития. Далее было исследовано существующие решения в отрасли фармацевтики с целью определения их преимуществ и недостатков. В результате анализа было выявлено, что ни одно из существующих веб-приложений не полностью удовлетворяет потребности «МК Перспектива». Таким образом, был сделан вывод о необходимости разработки нового веб-приложения, способного соответствовать требованиям компании и обеспечить её эффективное функционирование в онлайн-среде.

Глава 2 Проектирование компонентов по созданию web-приложения

2.1 Описание CASE-технологий

CASE-технологии, или CASE (Computer-Aided Software Engineering), представляют комплексный подход, включающий методы, инструменты и процессы, цель которых — автоматизировать и оптимизировать разработку программного обеспечения. Эти технологии направлены на улучшение эффективности и качества процессов создания ПО, а также на сокращение затрат и времени, требуемых для разработки сложных программных продуктов. В контексте разработки и администрирования web-приложения для оптовой торговли фармацевтической продукцией, CASE-технологии играют ключевую роль, обеспечивая всестороннюю поддержку проекта от начала до конечного этапа жизненного цикла.

Одним из основных компонентов CASE-технологий является графическое моделирование. С использованием графических инструментов разработчики могут создавать визуальные модели проекта, включая диаграммы классов, диаграммы последовательностей, диаграммы состояний и другие, что упрощает восприятие и понимание структуры приложения. Это не только способствует лучшему взаимопониманию среди членов команды, но также является мощным инструментом для анализа и проектирования [6].

Важным элементом CASE-технологий является их поддержка для документирования. Автоматизированные генераторы документации помогают создавать сопроводительную документацию к коду, что улучшает понимание работы системы как для самой команды разработчиков, так и для будущих администраторов приложения.

2.2 Выбор средств реализации

2.2.1 Выбор системы управления данными

В эпоху цифровизации бизнес-процессов и обширного роста информационных технологий, создание эффективного веб-представительства становится неотъемлемым элементом успеха любой организации. Один из ключевых компонентов при создании таких систем — это выбор эффективной и надежной системы управления базами данных (СУБД). СУБД служит фундаментом для сохранения, обработки и предоставления доступа к важнейшим данным. Она предоставляет комплекс программных средств и языковых инструментов, которые способствуют качественному управлению данными, поддержанию их целостности, а также обеспечивают безопасность и доступность данных для разнообразных приложений.

Современные системы управления базами данных предлагают широкий спектр функциональности, включая управление данными во внешней и оперативной памяти, управление буферами, обработку транзакций, поддержку языков запросов к базам данных, администрирование и журнализацию операций изменения данных. Они состоят из множества компонентов, таких как ядро для управления данными, процессор запросов для оптимизации операций чтения и записи, а также различные сервисные утилиты, упрощающие обслуживание и администрирование системы [5].

В настоящее время на рынке представлены различные СУБД, среди которых особое внимание заслуживают MySQL, PostgreSQL и MongoDB, каждая из которых имеет свои уникальные особенности и предназначена для решения специфических задач. Выбор конкретной СУБД для проекта зависит от множества факторов, включая требования к масштабируемости, надежности, производительности, простоте использования и поддержке со стороны разработчиков.

В таблице 2 представлен сравнительный анализ СУБД, который был проведен для определения наиболее подходящей системы для нашего проекта.

MySQL, являясь одной из наиболее популярных реляционных систем управления базами данных, продемонстрировала значительные преимущества по большинству рассмотренных критериев, особенно выделяясь своей масштабируемостью, высокой надежностью функционирования, отличной производительностью и широкой распространенностью среди разработчиков. Простота использования и мощная поддержка разработчиков делают MySQL отличным выбором для обеспечения стабильной и эффективной работы приложений [14], [17], [21].

Таблица 2 – Сравнительный анализ СУБД

Характеристика	MySQL	PostgreSQL	MongoDB
Масштабируемость	+	+	+
Надежность функционирования	+	-	-
Производительность системы	+	-	-
Распространенность среди разработчиков	+	-	-
Простота в использовании	+	+	+
Гибкость схемы данных	-	-	+
Поддержка разработчиков	+	+	+
Надежность транзакций	+	+	-
Итого	6	3	3

Выбор MySQL подтверждается не только ее впечатляющими техническими характеристиками, но и широким признанием в индустрии, что делает ее надежным и эффективным решением для управления данными.

2.2.2 Выбор языка программирования и фреймворка для разработки web-приложения

В настоящем аналитическом обзоре рассматриваются три популярных языка программирования, которые широко используются для разработки интернет-витрин:

- php,
- java,
- javascript.

Каждый из этих языков имеет свои особенности, преимущества и недостатки, которые следует учитывать при выборе подходящего языка для данной задачи.

PHP является одним из наиболее распространенных языков программирования для веб-разработки. Его преимущества включают широкое распространение и популярность среди разработчиков, простоту изучения и использования, а также наличие множества готовых решений в виде фреймворков и библиотек. PHP легко интегрируется с различными базами данных, что делает его удобным выбором для создания интернет-витрин с динамическим контентом. Однако некоторые критики указывают на несовершенный синтаксис и уязвимости безопасности PHP [13], [18], [19].

Java, в свою очередь, известен своей платформенной независимостью и высоким уровнем безопасности. Эти качества делают его хорошим выбором для разработки крупных и сложных интернет-витрин. Java также обеспечивает масштабируемость проектов и хорошую поддержку в отношении управления памятью. Однако разработка на Java может потребовать большого объема кода и длительного цикла разработки из-за необходимости компиляции и запуска на виртуальной машине Java.

JavaScript, с другой стороны, позволяет разработчикам создавать интерфейсы как на клиентской, так и на серверной стороне. Его асинхронная природа и отзывчивость делают его идеальным для создания интерактивных элементов на витрине и обеспечивают хороший пользовательский опыт.

JavaScript также известен своей обширной экосистемой, включающей в себя множество фреймворков и библиотек для различных задач. Однако различия в поддержке и реализации на разных браузерах могут создать проблемы с совместимостью.

Исходя из представленного анализа, можно сделать вывод, что PHP является наиболее подходящим языком программирования для разработки интернет-витрины. Несмотря на некоторые недостатки, преимущества PHP, такие как его широкое распространение, простота использования, готовые решения и интеграция с базами данных, делают его предпочтительным выбором для данной задачи. Кроме того, PHP обладает большим опытом применения в веб-разработке и имеет огромное сообщество разработчиков, что обеспечивает поддержку и ресурсы для успешной реализации проекта.

Далее, рассмотрим три популярных фреймворка:

- laravel,
- symfony,
- cakephp.

Laravel – это фреймворк с открытым исходным кодом, созданный Тейлором Отвеллом. Он предлагает элегантный синтаксис и многочисленные инструменты для упрощения веб-разработки. Одной из ключевых особенностей Laravel является его удобная ORM (Eloquent) для работы с базами данных, а также встроенная система маршрутизации, миграции и аутентификации [15], [16].

Symfony – это флагманский PHP-фреймворк с высоким уровнем расширяемости и конфигурируемости. Он ориентирован на создание сложных веб-приложений и предоставляет множество компонентов и библиотек для решения различных задач. Symfony также активно поддерживается сообществом разработчиков и имеет долгосрочную поддержку. По сравнению с Laravel, Symfony является более громоздким и менее интуитивным в использовании.

CakePHP – это фреймворк с акцентом на быстрой разработке приложений. Он предлагает много встроенных функций, таких как ORM, маршрутизация и валидация данных, что делает разработку приложений более быстрой и простой. Однако по сравнению с Laravel, CakePHP менее гибкий и масштабируемый.

В таблице 3 представлено сравнение ключевых характеристик фреймворков:

Таблица 3 – Сравнительный анализ фреймворков

Характеристика	Laravel	Symfony	CakePHP
Уровень сложности	Низкий	Средний	Средний
Скорость разработки	Высокая	Средняя	Высокая
Расширяемость	Высокая	Высокая	Средняя
Сообщество	Большое	Большое	Среднее
Производительность	Высокая	Средняя	Средняя

Проведенный анализ делает Laravel наиболее привлекательным фреймворком для разработки web-приложения. Этот выбор обоснован несколькими факторами, включая его простоту в использовании, высокую скорость разработки, гибкость и хорошую производительность. Дополнительным преимуществом является активное сообщество разработчиков, что обеспечивает надежную поддержку и доступ к обширным ресурсам. Также, мой опыт работы с Laravel делает его предпочтительным по сравнению с другими фреймворками, поскольку я имею понимание его особенностей и структуры.

2.3 Обоснование архитектуры построения web-приложения

Современное создание веб-сайтов часто включает применение многоуровневой архитектуры, где задействован отдельный сервер приложений для обработки данных. Пользователи взаимодействуют с заранее подготовленной информацией, которая может обрабатываться прямо на их компьютерах с помощью скриптов. Такой подход позволяет всем посетителям сайта получать доступ к информации и просматривать контент через браузер.

Зарегистрированные пользователи имеют дополнительные функции, такие как отслеживание состояния заказов, в то время как менеджеры могут просматривать и изменять статусы заказов, а также организовывать доставку при необходимости. Эти возможности улучшают пользовательский опыт и повышают удобство управления заказами на веб-платформах.

Оптимальной считается трехуровневая архитектура, которая позволяет эффективно распределять обработку больших объемов данных. Пользовательские запросы на информацию или изменение прав доступа обрабатываются непосредственно в браузере. Трехуровневая архитектура позволяет детализировать ключевые элементы и обеспечивает высокую производительность системы [4].

Трехзвенная архитектура представлена на рисунке 5.



Рисунок 5 – Трёхзвенная «клиент-серверная» архитектура

На сервере хранятся и обрабатываются данные, предоставленные клиентом, организованные в массивы, структурированные из различных источников. Этот метод подготовки массивов является ключевым для работы системы.

Каждый участник системы выполняет роль пользователя, имея разделённые права на управление и просмотр данных при взаимодействии с системой (Таблица 4).

Таблица 4 – Права доступа в зависимости от статуса пользователя

Роль	Особенности доступа	Рамки доступа	Разрешения
Администратор	Один избранный с полным контролем над данными	Аутентификация	Полный контроль и редактирование данных
Клиент	Требуется регистрация для вступления в группу	Аутентификация	Все разрешения гостя, так же права на управление заказами
Гость	Посетитель сайта, не входящий в группу	Вход на сайт	Права на управление корзиной, поиском по каталогу, навигацию по сайту

Пользователи подразделяются на три основные категории:

- администраторы,
- клиенты,
- посетители.

Трехзвенная архитектура обеспечивает эффективное распределение обработки информации и разграничение доступа пользователей с различными правами.

2.4 Проектирование пользовательских сценариев для web-приложения ООО «МК Перспектива»

Разработка диаграммы прецедентов является ключевым этапом в планировании и проектировании систем, в том числе и интернет-витрин. Для достижения высокой точности и удобства в проектировании используются CASE-технология моделирования, позволяющая значительно упростить и автоматизировать процесс создания различных. Одним из наиболее

популярных инструментов, реализующих подход CASE, является StarUML — мощная и гибкая платформа для моделирования, поддерживающая широкий спектр диаграмм в нотации UML (Unified Modeling Language).

UML, или Унифицированный язык моделирования, является стандартизированным языком для графического моделирования, который используется для определения, визуализации, разработки и документирования артефактов программного обеспечения. В рамках UML диаграмма прецедентов (Use Case Diagram) занимает ключевое место, так как она позволяет на высоком уровне описывать функциональные требования системы, определять ключевых участников (актеров) и их взаимодействие с системой. Это способствует глубокому пониманию задач и целей проекта [11], [12].

Запуск интернет-витрины для ООО «МК Перспектива», занимающейся распределением фармацевтических товаров, влечет за собой организацию пользователей по отдельным группам (рисунок 6).

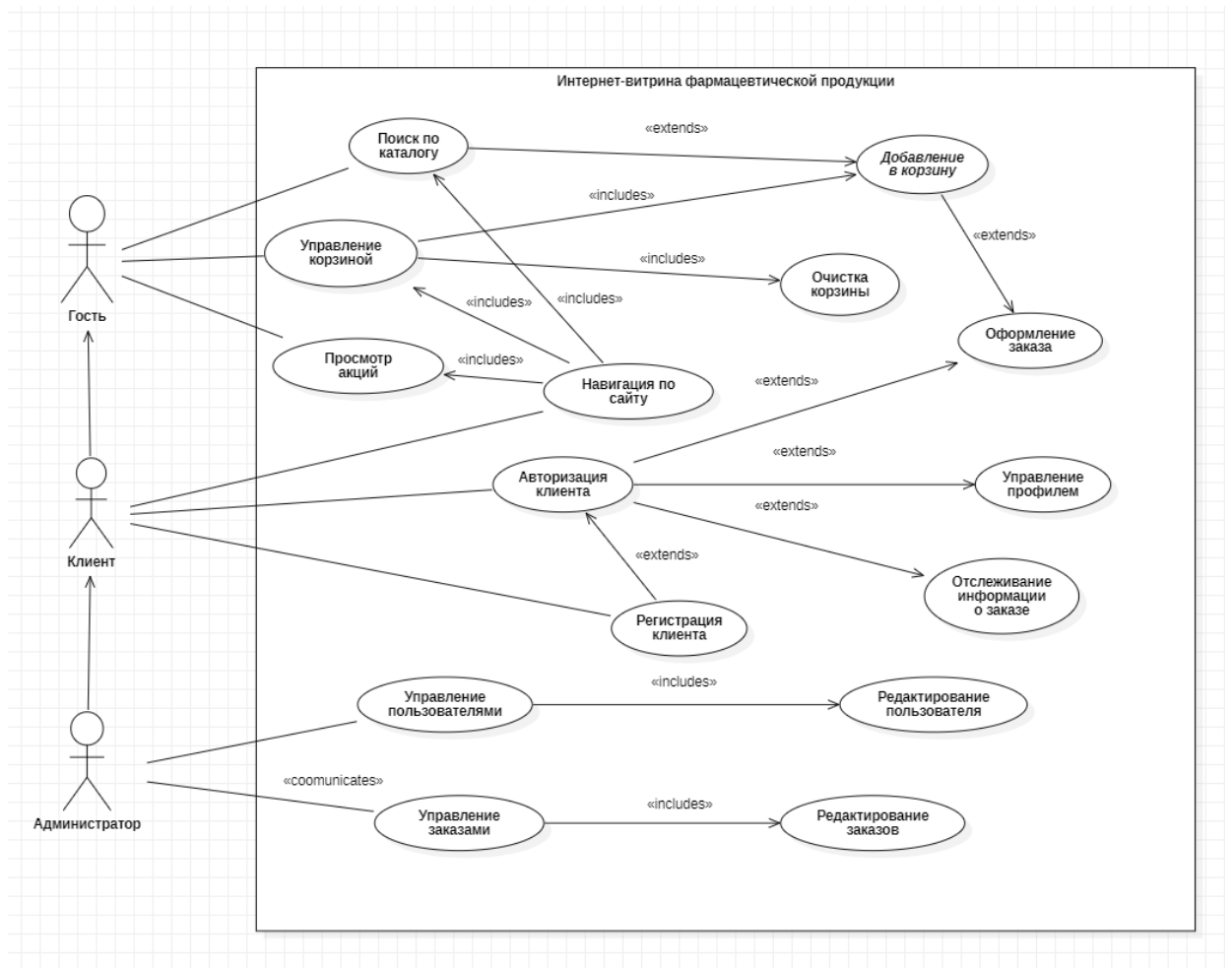


Рисунок 6 – Диаграмма прецедентов

Пользователи имеют право осуществлять контроль над следующими процессами:

- администрирование, включающее процессы управления аккаунтами пользователей;
- управление заказами, включающее добавление товаров в корзину, оформление заказа и возможность его редактирования;
- навигация по сайту, включающая использование поисковой функции для поиска продукции по различным параметрам, таким как производитель, название;

- работа с корзиной, которая служит промежуточным этапом, где товары временно сохраняются перед окончательным оформлением в заказ;
- доступ к дополнительным функциям после прохождения авторизации, например, к управлению личным профилем и мониторингу информации о статусе заказов.

Проектирование диаграммы прецедентов через применение CASE-инструмента StarUML и нотации UML, улучшает понимание требований к интернет-витрины и способствует координации между участниками проекта. Этот подход не только обогащает анализ функционала, но и укрепляет коммуникацию между всеми вовлеченными сторонами, обеспечивая тем самым более прозрачное и эффективное взаимодействие.

2.5 Проектирование логической модели данных в нотации *idef1x*

Логическая модель данных представляет собой важный этап в проектировании информационных систем, играя ключевую роль в определении структуры базы данных. В данном разделе будет рассмотрена логическая модель данных для проекта, связанного с веб-приложением оптовой торговли фармацевтической продукцией.

Для разработки логической модели данных применяется CASE-технология моделированием данных, которая предоставляет эффективные средства и методологии для проектирования баз данных. В данной работе будет использована методология IDEF1X (Integration DEFINITION for Information Modeling), предоставляющая наглядное и формализованное представление структуры данных.

В качестве CASE-средства для разработки логической модели данных выбран MySQL Workbench. Это интегрированное средство разработки баз

данных обеспечивает удобный интерфейс для создания, визуализации и администрирования баз данных с использованием методологии IDEF1X.

Выбор этих инструментов обусловлен стремлением к эффективному проектированию, управлению данными и обеспечению согласованности базы данных с требованиями бизнес-процессов компании. В следующих разделах представлено более подробное описание логической модели данных, разработанной с использованием указанных методологии и CASE-средства.

Логическая модель данных, представленная на рисунке 7, разработана в соответствии с методологией IDEF1X.

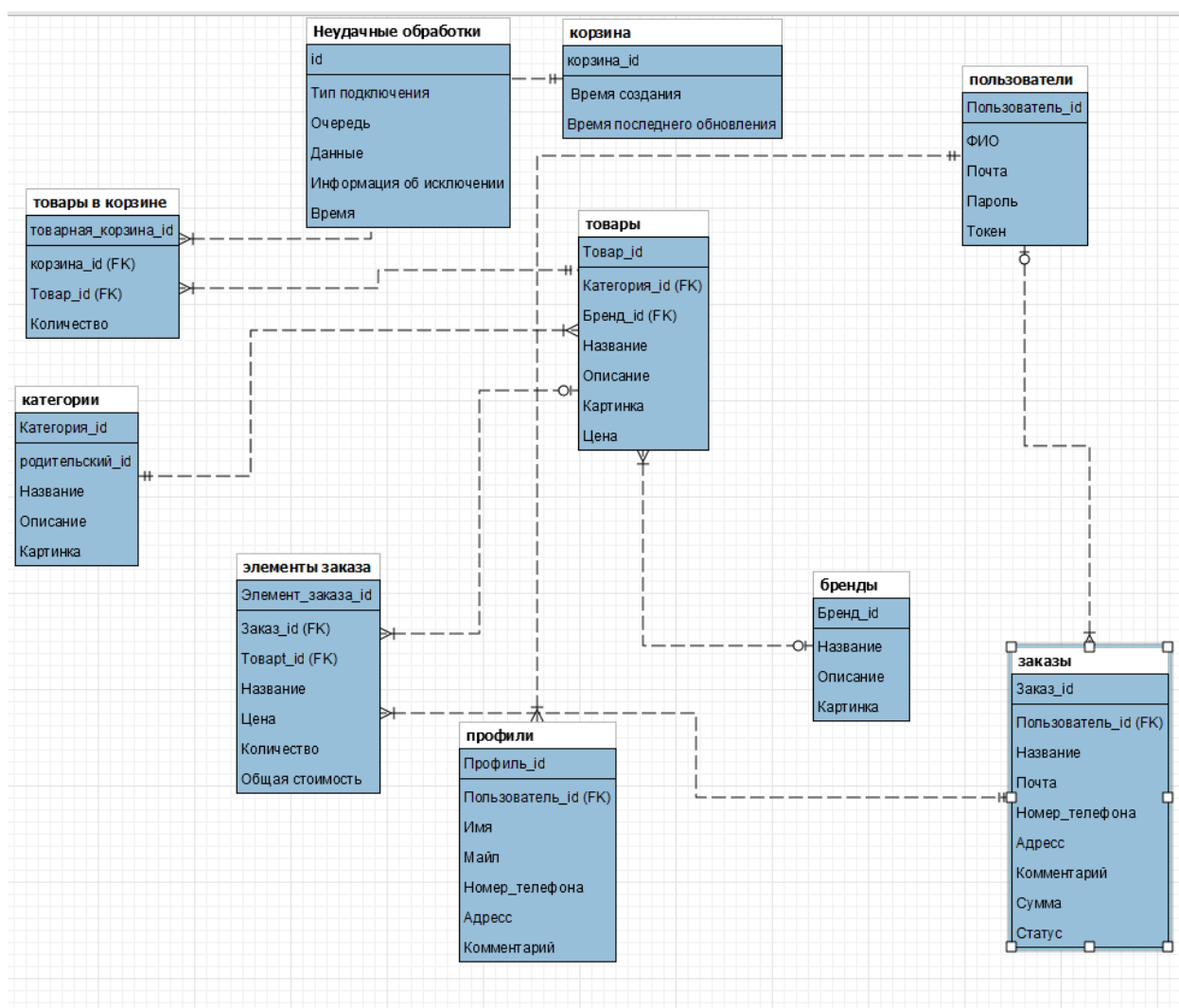


Рисунок 7 – Логическая модель

Таблица «Неудачные обработки» содержит информацию о неудачных операциях. Она включает в себя следующие атрибуты:

- идентификатор (id),
- тип подключения,
- очередь,
- данные,
- исключения,
- время.

Таблица «Корзина» представляет собой хранилище информации о корзинах. У неё есть следующие атрибуты:

- идентификатор корзины (корзина_id);
- время создания;
- время последнего обновления.

Таблица «Товары» содержит информацию о товарах, доступных в магазине. Она включает атрибуты:

- идентификатор товара,
- идентификатор категории,
- идентификатор бренда,
- название,
- описание,
- изображение,
- цена товара.

Таблица «Товары в корзине» связывает товары с конкретными корзинами. Она имеет следующие атрибуты:

- идентификатор товара в корзине (товарная_корзина_id);
- идентификатор корзины (корзина_id);
- идентификатор товара (Товар_id);
- количество товара в корзине.

Таблица «Категории» содержит информацию о категориях товаров. У неё есть следующие атрибуты:

- идентификатор категории (Категория_id);
- идентификатор родительской категории (Родительский_id);
- название;
- описание;
- изображение категории.

Таблица «Элементы заказа» содержит информацию о товарах, включенных в заказ. Она включает следующие атрибуты:

- идентификатор элемента заказа;
- идентификатор заказа;
- идентификатор товара
- название товара;
- цену;
- количество товара;
- общая стоимость.

Таблица «Профили» хранит информацию о пользователях. У неё есть следующие атрибуты:

- идентификатор профиля (Профиль_id);
- идентификатор пользователя (Пользователь_id);
- имя;
- фамилия;
- адрес электронной почты;
- номер телефона и пароль.

Таблица «Бренды» содержит информацию о брендах товаров. Она включает в себя следующие атрибуты:

- идентификатор бренда,
- название бренда,
- описание,

- изображение бренда.

Таблица «Заказы» представляет собой информацию о заказах, размещенных пользователями. Она включает следующие атрибуты:

- идентификатор заказа
- идентификатор пользователя
- название заказа;
- адрес доставки;
- комментарии к заказу;
- сумму заказа;
- статус заказа.

Логическая модель данных на основе методологии IDEF1X позволяет эффективно организовывать, хранить и управлять данными веб-приложения оптовой торговли фармацевтической продукцией

2.6 Проектирование Физической модели с применением Case-технологии моделирования

В ходе выполнения бакалаврской работы был проведен обширный анализ и тщательное теоретическое исследование аспектов

Физическое моделирование информационной системы, основанное на CASE-технологиях, становится завершающим этапом в проектировании базы данных. В ходе этого процесса, логическая модель, разработанная на предыдущих этапах, преобразуется в физическую схему, готовую к реализации на конкретных носителях информации. Для оптимальной организации данных рассматривается структура файлов, создание эффективных индексов, обеспечивающих быстрый доступ к информации, а также определяются методы поддержания целостности данных и их защиты.

Специфика физического проектирования заключается в принятии ключевых решений относительно хранения данных и обработки запросов с целью повышения производительности и расширяемости системы.

Проектирование базы данных в рамках CASE-технологий с применением инструментария MySQL Workbench позволяет глубоко интегрировать проектные решения с выбранной СУБД, что обеспечивает не только выполнение функциональных требований, но и поддержку высоких нагрузок, типичных для фармацевтической оптовой торговли.

На представленной диаграмме (рисунок 8) в MySQL Workbench зафиксированы основные сущности и связи разрабатываемой системы.

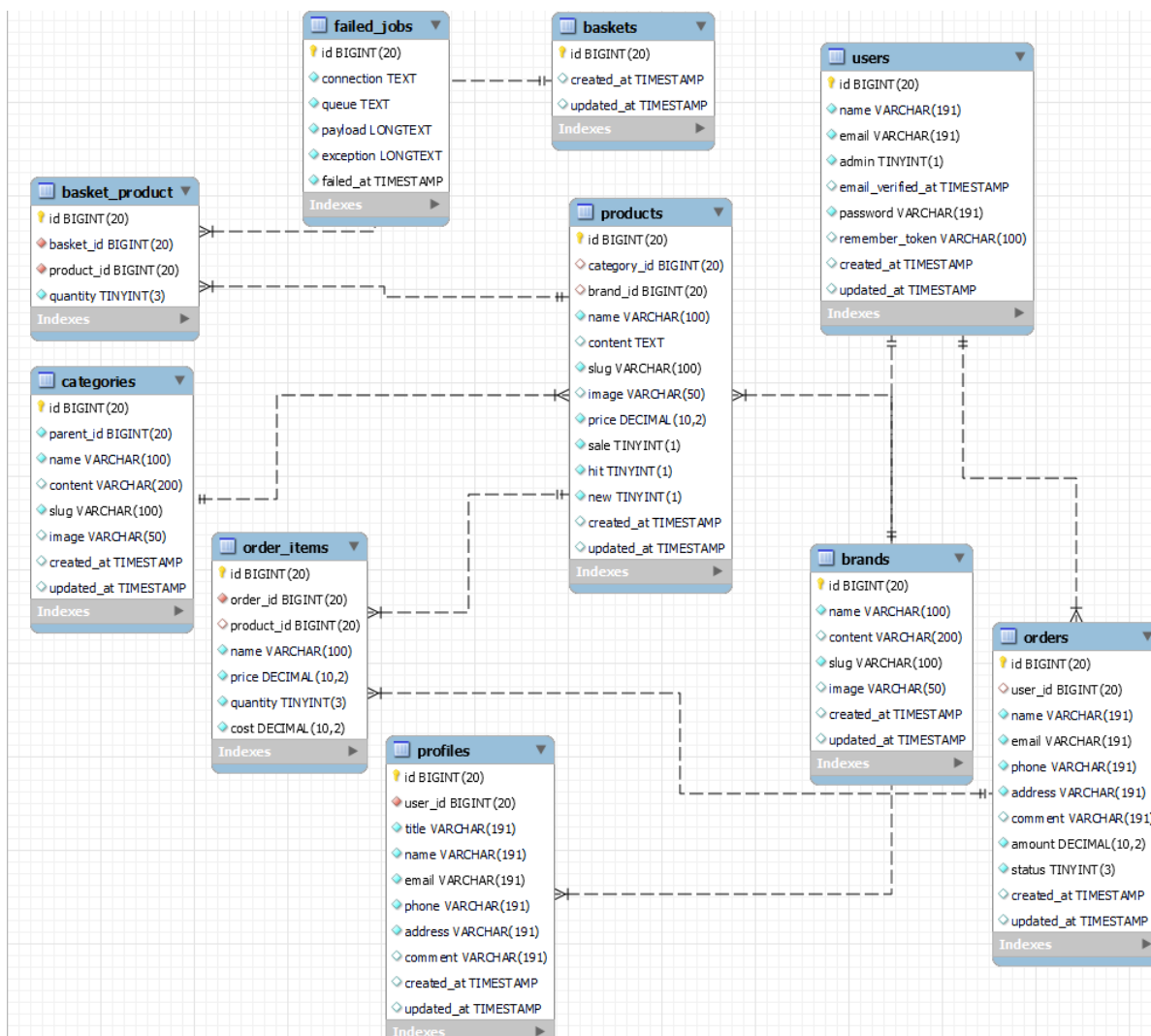


Рисунок 8 – Физическая модель данных

Таблицы users, orders, products и другие, структурированы таким образом, чтобы обеспечить надёжное и удобное управление данными о

товарах, заказах, клиентах и операциях. Каждая таблица содержит поля, отражающие атрибуты сущности и типы данных, которые были выбраны в соответствии с требованиями к хранению и обработке информации. Например, использование типа VARCHAR для наименований и DECIMAL для финансовых показателей, позволяет с балансом учесть как точность данных, так и оптимизацию занимаемого пространства.

Индексы, представленные в диаграмме, спроектированы для обеспечения быстрого действия запросов, особенно в отношении операций поиска и сортировки, что критично для эффективности работы интернет-витрины. Внимание уделяется и целостности данных, для чего определены внешние ключи и другие ограничения, что позволяет поддерживать связанность данных при различных операциях.

Таким образом, физическая модель данных является важнейшим компонентом проекта базы данных, отражающим все предыдущие этапы анализа и проектирования, и служит надёжной основой для последующей разработки и администрирования информационной системы интернет-витрины.

Выводы по главе 2

В ходе этой главы основной задачей было проектирование интернет-витрины, обеспечивающая информационную поддержку фармацевтического предприятия, включая представление компании, продукции, акций, и управление заказами.

Было проведено обоснование архитектуры системы, определяя трехуровневую архитектуру с выделением сервера приложений для обработки данных, обеспечивая удобное взаимодействие пользователей с системой.

Логическая и физическая модели данных были разработаны с использованием CASE-технологий и методологии IDEF1X, а также инструмента MySQL Workbench. Эти модели предоставляют эффективную основу для хранения и обработки информации, учитывая требования к структуре данных и производительности системы.

Глава 3 Реализация web-приложения

3.1 Разработка процессов аутентификации и авторизации пользователей

Этот раздел представляет собой подробное описание реализации механизмов аутентификации и авторизации пользователей в приложении. Аутентификация и авторизация являются ключевыми аспектами любой системы, обеспечивая безопасность и управление доступом к её ресурсам. Начнем с анализа структуры авторизации, которая приведена на рисунке 9.

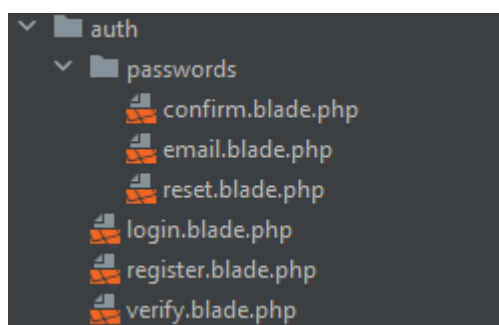


Рисунок 9 – Структура аутентификации и авторизации на сайте

Здесь представлена структура директорий, в которой содержатся файлы, относящиеся к аутентификации и авторизации пользователей. Видим, что есть директория «auth», а внутри нее - поддиректория «passwords». Это может указывать на применение аутентификации по паролю в данном приложении. Внутри этой структуры находятся файлы, такие как «confirm.blade.php», «email.blade.php», «reset.blade.php», «login.blade.php», «register.blade.php» и «verify.blade.php», каждый из которых отвечает за определенный этап процесса аутентификации или авторизации [8].

Приступая к регистрации, пользователь должен создать новый аккаунт в системе. Для этого используется страница «register.blade.php». На данной странице предусмотрена форма, где пользователь может ввести свои личные

Этот блок кода отвечает за отображение формы входа пользователя в личный кабинет. Форма содержит поля для ввода email и пароля. Когда пользователь вводит свои данные и отправляет форму, они передаются на сервер для проверки. Если данные верны, пользователь получает доступ к личному кабинету, в противном случае отображается сообщение об ошибке.

При смене или забытии пароля пользователь может воспользоваться страницей «reset.blade.php». Здесь предлагается ввести свой адрес электронной почты для получения ссылки на сброс пароля.

Для подтверждения пароля перед выполнением определенных действий используется страница «confirm.blade.php». Здесь пользователь должен ввести свой текущий пароль, прежде чем сможет продолжить операцию.

После успешной аутентификации и подтверждения пароля пользователь может быть перенаправлен на страницу «verify.blade.php», где будет предложено подтвердить адрес электронной почты [20]. Это может быть важным шагом в рамках процесса безопасности приложения.

Таким образом, каждый этап в этой структуре выполняет уникальную функцию в процессе аутентификации и авторизации пользователей в приложении, обеспечивая безопасность и удобство использования для конечных пользователей [2].

3.2 Разработка административной панели

В ходе разработки административной панели веб-приложения был реализован комплексный подход, целью которого было создание удобного и функционального интерфейса для управления ключевыми аспектами платформы. В директории Admin был создан набор контроллеров, каждый из которых предназначен для управления определенным сегментом приложения, включая товары, категории, заказы, пользователей и статичные страницы. Шаблоны представлений blade.php обеспечивают администратору интуитивно понятные инструменты для выполнения необходимых операций [9], [10].

Структура административной панели представлена на рисунке 11.

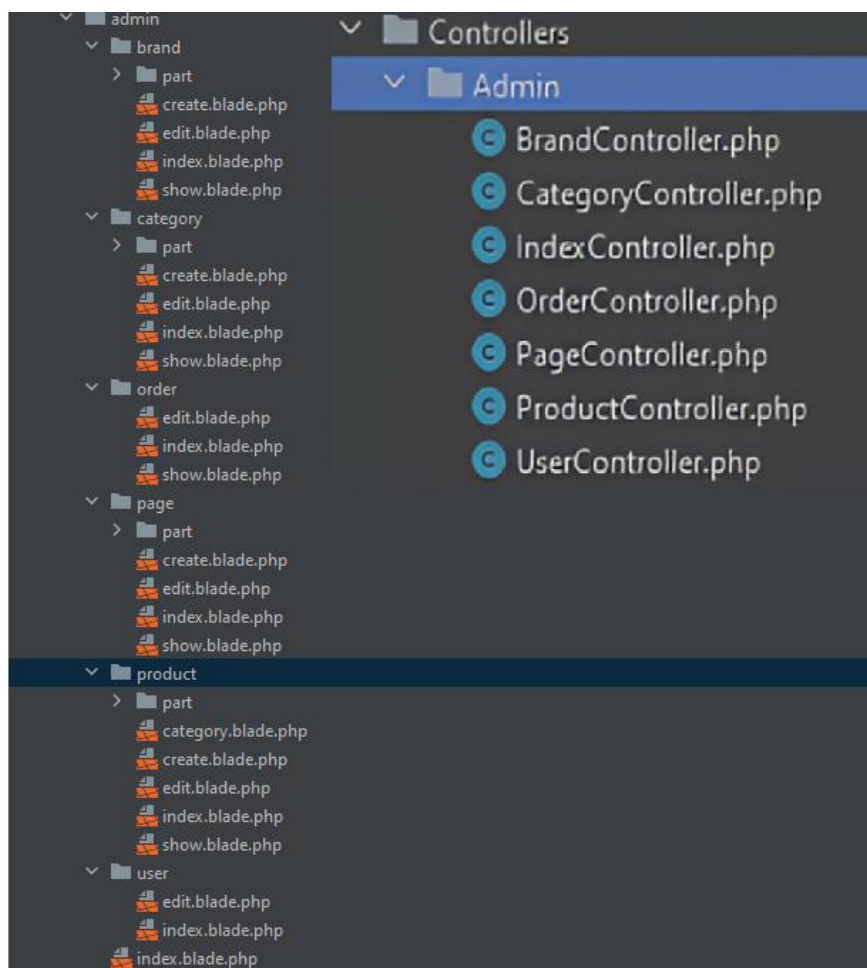


Рисунок 11 – Структура административной панели

Управление брендами обеспечивается через `BrandController.php`, который содержит функции для добавления новых брендов (`create.blade.php`), редактирования существующих (`edit.blade.php`), просмотра списка всех брендов (`index.blade.php`) и подробного отображения информации о конкретном бренде (`show.blade.php`). Эти инструменты значительно упрощают процесс взаимодействия с брендами-производителями. Пример части кода, отвечающего за добавление нового бренда, приведен на рисунке 12.

```
7
8
9
10
11
12
13
14
15
16
17
18
19

/**
 * Показывает форму для создания бренда
 *
 * @return \Illuminate\Http\Response
 */
public function create() {
    return view('admin.brand.create');
}

@extends('layout.admin', ['title' => 'Создание нового бренда'])

@section('content')
    <h1>Создание нового бренда</h1>
    <form method="post" action="{{ route('admin.brand.store') }}" enctype="multipart/form-data">
        @csrf
        @include('admin.brand.part.form')
    </form>
@endsection
```

Рисунок 12 – Добавление нового бренда

Для управления категориями товаров разработан CategoryController.php, который позволяет администратору структурировать категории товаров и актуализировать информацию о них, способствуя более удобной организации товаров и улучшенной навигации для конечного пользователя. Пример части кода для управления категориями товаров представлен на соответствующем рисунке 13.

```
CategoryController.php
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

/**
 * Показывает форму для редактирования категории
 *
 * @param \App\Models\Category $category
 * @return \Illuminate\Http\Response
 */
public function edit(Category $category) {
    // все категории для возможности выбора родителя
    $items = Category::all();
    return view('admin.category.edit', compact('category', 'items'));
}

edit.blade.php
1
2
3
4
5
6
7
8
9
10
11

@extends('layout.admin', ['title' => 'Редактирование категории'])

@section('content')
    <h1>Редактирование категории</h1>
    <form method="post" enctype="multipart/form-data"
        action="{{ route('admin.category.update', ['category' => $category->id]) }}">
        @method('PUT')
        @include('admin.category.part.form')
    </form>
@endsection
```

Рисунок 13 – Редактирование категорий товаров

Функция `edit` в `CategoryController.php` показывает форму для редактирования категорий, что является центральным элементом управления, позволяя администраторам быстро вносить изменения в уже существующие категории. Это упрощает процесс обновления информации и помогает поддерживать данные в актуальном состоянии. Как видно на рисунке, метод `edit` отвечает за передачу данных о категории в соответствующий шаблон представления `edit.blade.php`.

Шаблон `edit.blade.php` наследует основной макет административной панели и предоставляет место для внесения изменений в категорию с помощью формы HTML.

`OrderController.php` включает методы для эффективного просмотра всех заказов (`index.blade.php`), их редактирования (`edit.blade.php`) и детального просмотра (`show.blade.php`), что позволяет администратору поддерживать оперативное управление заказами и их своевременную обработку. Иллюстрация части кода для просмотра заказов также приведена на рисунке 14.

```
OrderController.php
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use ...
6
7 no usages
8
9 class OrderController extends Controller {
10     /**
11      * Просмотр списка заказов
12      *
13      * @return \Illuminate\Http\Response
14      */
15     public function index() {
16         $orders = Order::orderBy('created_at', 'desc')->paginate(5);
17         $statuses = Order::STATUSES;
18         return view('admin.order.index', compact('orders', 'statuses'));
19     }
20
21     /**
22      * Просмотр отдельного заказа
23      *
24      * @param \App\Models\Order $order
25      * @return \Illuminate\Http\Response
26      */
27     public function show(Order $order) {
28         $statuses = Order::STATUSES;
29         return view('admin.order.show', compact('order', 'statuses'));
30     }
31 }
32
33 show.blade.php
34 @extends('layout.admin', ['title' => 'Просмотр заказа'])
35
36 @section('content')
37     <h1>Данные по заказу № {{ $order->id }}</h1>
38
39     <p>
40         Статус заказа:
41         @if ($order->status == 0)
42             <span class="text-danger">{{ $statuses[$order->status] }}</span>
43         @elseif (in_array($order->status, [1,2,3]))
44             <span class="text-success">{{ $statuses[$order->status] }}</span>
45         @else
46             {{ $statuses[$order->status] }}
47         @endif
48     </p>
49
50     <h3 class="mb-3">Состав заказа</h3>
51     <table class="table table-bordered">
52         <tr>
53             <th>№</th>
54             <th>Наименование</th>
55             <th>Цена</th>
56             <th>Кол-во</th>
57             <th>Стоимость</th>
58         </tr>
59     </table>
60
61 @endsection
```

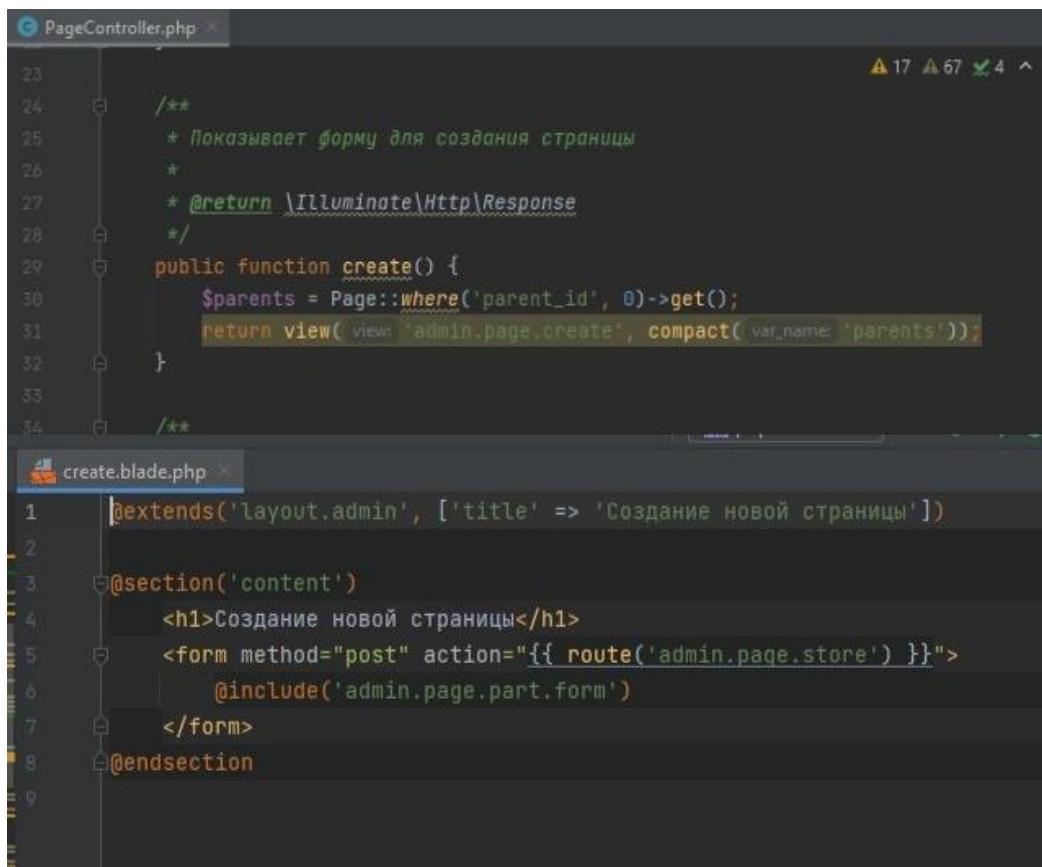
Рисунок 14 – Просмотр заказа клиента

Часть кода в OrderController.php отвечает за представление заказов в административном интерфейсе с использованием цветовой кодировки статусов, заданных в модели Order. Это цветовое кодирование является визуальным ключом, который способствует более быстрой оценке текущего статуса заказов и улучшает навигацию по панели управления.

Для детализированного просмотра информации о каждом индивидуальном заказе служит шаблон show.blade.php. Этот шаблон

позволяет администратору видеть все детали заказа, включая сведения о покупателе и полный список товаров в заказе, а также текущее состояние заказа.

Управление содержимым статичных страниц сайта выполняется через PageController.php. Этот контроллер предоставляет инструменты для создания (create.blade.php), редактирования (edit.blade.php), просмотра (index.blade.php) и детализации (show.blade.php) страниц, что играет важную роль для информационного наполнения сайта и его SEO-позиционирования. На рисунке 15 демонстрируется часть кода для создания новой страницы.



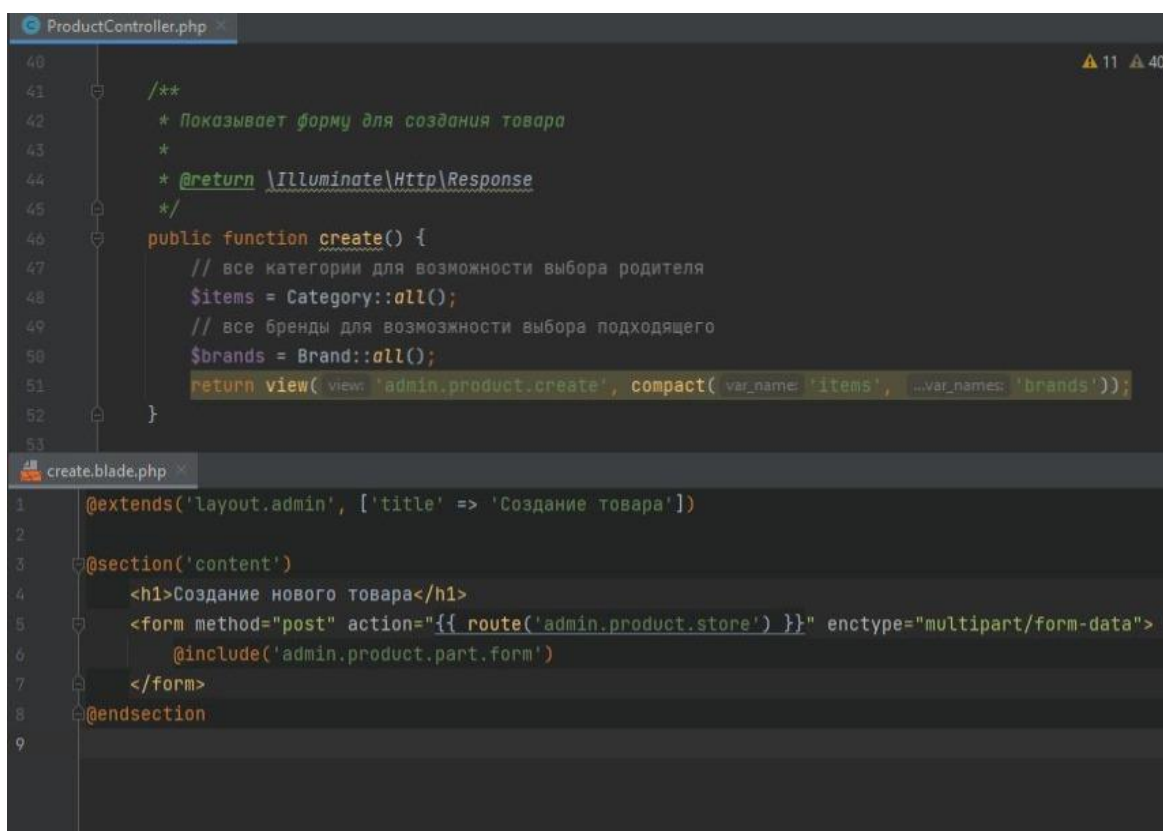
```
PageController.php
23
24  /**
25   * Показывает форму для создания страницы
26   *
27   * @return \Illuminate\Http\Response
28   */
29  public function create() {
30      $parents = Page::where('parent_id', 0)->get();
31      return view('admin.page.create', compact('parents'));
32  }
33
34  /**

create.blade.php
1  @extends('layout.admin', ['title' => 'Создание новой страницы'])
2
3  @section('content')
4      <h1>Создание новой страницы</h1>
5      <form method="post" action="{{ route('admin.page.store') }}">
6          @include('admin.page.part.form')
7      </form>
8  @endsection
9
```

Рисунок 15 – Создание новой страницы интернет-витрины

ProductController.php дает администраторам полный набор функций для управления товарами, включая добавление новых товаров, их редактирование и обновление информации об ассортименте, что является жизненно важным

для коммерческих платформ. Пример части кода для управления товарами отображен на рисунке 16.



```
ProductController.php
40
41 /**
42  * Показывает форму для создания товара
43  *
44  * @return \Illuminate\Http\Response
45  */
46 public function create() {
47     // все категории для возможности выбора родителя
48     $items = Category::all();
49     // все бренды для возможности выбора подходящего
50     $brands = Brand::all();
51     return view('admin.product.create', compact('items', 'brands'));
52 }
53

create.blade.php
1 @extends('layout.admin', ['title' => 'Создание товара'])
2
3 @section('content')
4     <h1>Создание нового товара</h1>
5     <form method="post" action="{{ route('admin.product.store') }}" enctype="multipart/form-data">
6         @include('admin.product.part.form')
7     </form>
8 @endsection
9
```

Рисунок 16 –Создание товара

В ProductController.php одной из ключевых функций контроллера является метод create, который позволяет администратору запускать процесс добавления нового товара в ассортимент.

Внутри шаблона create.blade.php задаётся форма для создания нового товара, которая обеспечивает передачу данных о товаре в ProductController через маршрут admin.product.store. Форма использует множественную загрузку файлов, что позволяет администратору добавлять изображения и другие медиафайлы к описанию товара, тем самым улучшая его презентацию для пользователей.

Наконец, UserController.php содержит все необходимые средства для редактирования профилей edit.blade.php, просмотра данных пользователей

show.blade.php, а также для удаления аккаунтов при необходимости. Пример кода, относящегося к редактированию пользователя, иллюстрируется на рисунке 17.

```
UserController.php
22 /**
23  * Показывает форму для редактирования пользователя
24  *
25  * @param \App\Models\User $user
26  * @return \Illuminate\Http\Response
27  */
28 public function edit(User $user) {
29     return view('admin.user.edit', compact('user'));
30 }

edit.blade.php
1 @extends('layout.admin', ['title' => 'Редактирование пользователя'])
2
3 @section('content')
4 <h1 class="mb-4">Редактирование пользователя</h1>
5 <form method="post" action="{{ route('admin.user.update', ['user' => $user->id]) }}">
6     @csrf
7     @method('PUT')
8     <div class="form-group">
9         <input type="text" class="form-control" name="name" placeholder="Имя, Фамилия"
10            required maxlength="255" value="{{ old('name') ?? $user->name ?? '' }}">
11     </div>
12     <div class="form-group">
13         <input type="email" class="form-control" name="email" placeholder="Адрес почты"
14            required maxlength="255" value="{{ old('email') ?? $user->email ?? '' }}">
15     </div>
16     <div class="form-group form-check">
17         <input type="checkbox" class="form-check-input" name="change_password"
18            id="change_password">
19         <label class="form-check-label" for="change_password">
20             Изменить пароль пользователя
21         </label>
22     </div>
23     <div class="form-group">
24         <input type="text" class="form-control" name="password" maxlength="255"
25            placeholder="Новый пароль" value="">
26     </div>
27     <div class="form-group">
28         <input type="text" class="form-control" name="password_confirmation" maxlength="255"
29            placeholder="Пароль еще раз" value="">
30     </div>
31     <div class="form-group">
32         <button type="submit" class="btn btn-success">Сохранить</button>
33     </div>
34 </form>
35 @endsection
```

Рисунок 17 – Редактирование профиля пользователя

Таким образом, разрабатываемая административная панель не только усиливает контроль над важнейшими аспектами веб-приложения, но и предоставляет администратору мощные и гибкие инструменты для поддержания его актуальности и эффективности.

3.3 Разработка пользовательской панели

Разработка пользовательской панели интернет-витрины для фармацевтической компании является актуальной задачей, целью которой является не только упрощение процесса выбора и покупки продукции для конечных пользователей, но и повышение эффективности бизнес-процессов самой компании.

Основные элементы пользовательской панели состоят из каталога, поиска, корзины и заявки. Рассмотрим детально каждый из компонентов (рисунок 18).

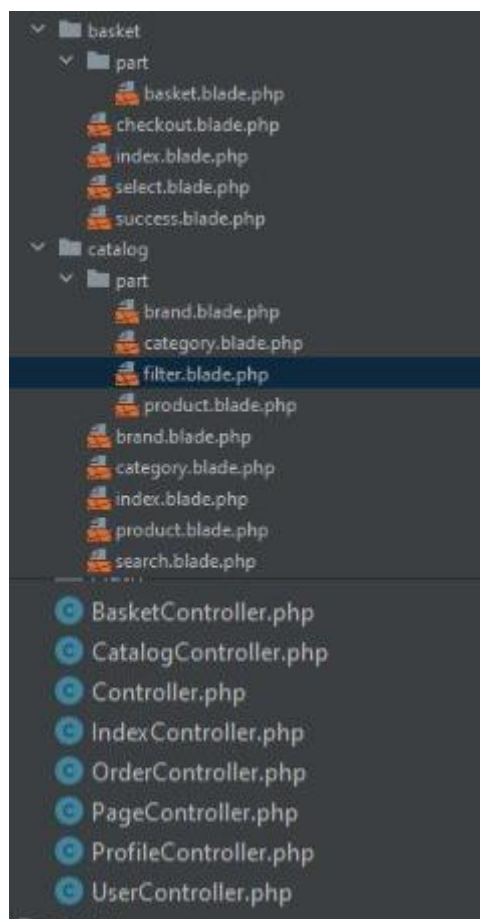
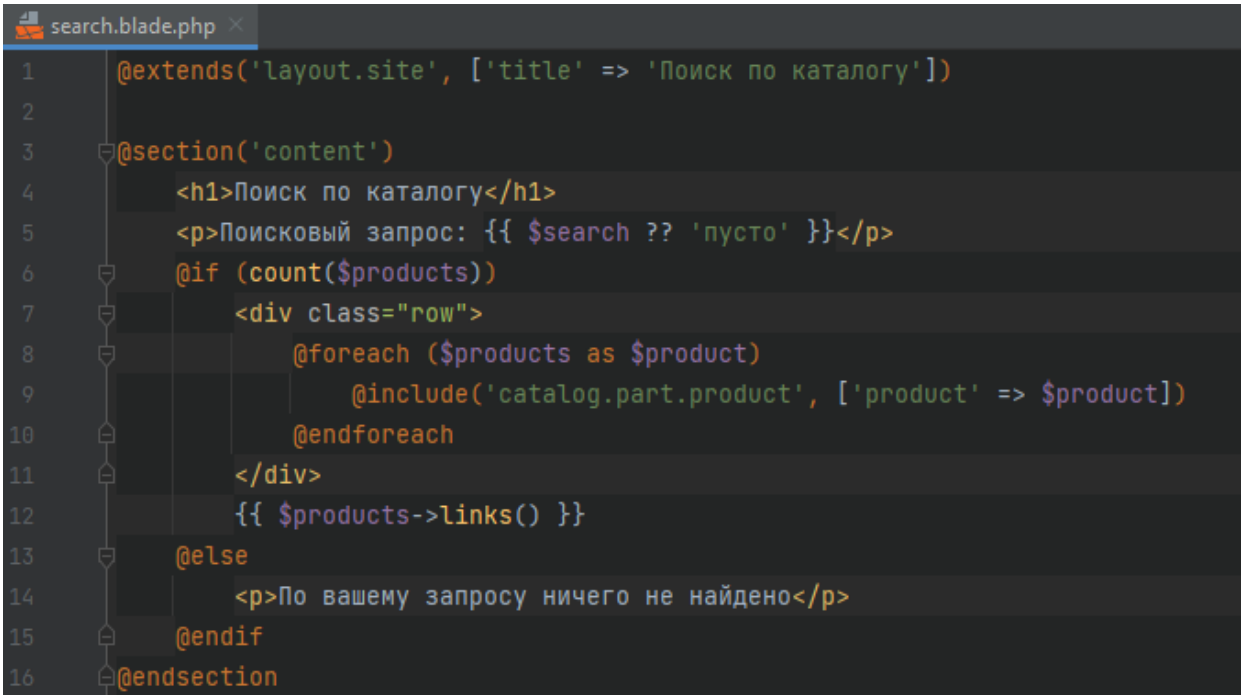


Рисунок 18 – Основные элементы пользовательской панели

Просмотр каталога реализован посредством CatalogController.php, который является основой для отображения списка товаров. Категоризация товаров по брендам и категориям через brand.blade.php и category.blade.php, а также детальное представление информации о товаре через product.blade.php, позволяют пользователям навигировать по разнообразию фармацевтической продукции. Главная страница каталога index.blade.php предоставляет сводную информацию, предлагая актуальные акции и новинки, в то время как модуль поиска search.blade.php интегрирован для облегчения поиска конкретных товаров или категорий в рамках каталога [9]. На рисунке 19 демонстрируется часть кода поиска по каталогу.



```
1 @extends('layout.site', ['title' => 'Поиск по каталогу'])
2
3 @section('content')
4     <h1>Поиск по каталогу</h1>
5     <p>Поисковый запрос: {{ $search ?? 'пусто' }}</p>
6     @if (count($products))
7         <div class="row">
8             @foreach ($products as $product)
9                 @include('catalog.part.product', ['product' => $product])
10            @endforeach
11        </div>
12        {{ $products->links() }}
13    @else
14        <p>По вашему запросу ничего не найдено</p>
15    @endif
16 @endsection
```

Рисунок 19 – Поиск по каталогу

Поиск играет важную роль в улучшении пользовательского опыта, позволяя быстро находить нужные товары без необходимости просмотра всего каталога. Реализация поиска предполагает возможность фильтрации товаров по ключевым словам, категориям, брендам или другим параметрам, что существенно экономит время пользователя.

Корзина покупок, организованная через `BasketController.php`, предоставляет пользователям легкость управления wybranными товарами: добавление и удаление продуктов, изменение количества и просмотр итоговой суммы заказа осуществляются на странице `basket.blade.php`. Переход к оформлению заказа через `checkout.blade.php` сводит процесс покупки к минимуму шагов, повышая вероятность его завершения. После подтверждения заказа, страница `success.blade.php` информирует пользователя о успешном оформлении и предоставляет детали заказа для отслеживания. На рисунке 20 демонстрируется часть кода оформления заказа.

```
checkout.blade.php
1  @extends('layout.site', ['title' => 'Оформить заказ'])
2
3  @section('content')
4      <h1 class="mb-4">Оформить заказ</h1>
5      @if ($profiles && $profiles->count())
6          @include('basket.select', ['current' => $profile->id ?? 0])
7      @endif
8      <form method="post" action="{{ route('basket.saveorder') }}" id="checkout">
9          @csrf
10         <div class="form-group">
11             <input type="text" class="form-control" name="name" placeholder="Имя, Фамилия"
12                 required maxlength="255" value="{{ old('name') ?? $profile->name ?? ' ' }}">
13         </div>
14         <div class="form-group">
15             <input type="email" class="form-control" name="email" placeholder="Адрес почты"
16                 required maxlength="255" value="{{ old('email') ?? $profile->email ?? ' ' }}">
17         </div>
18         <div class="form-group">
19             <input type="text" class="form-control" name="phone" placeholder="Номер телефона"
20                 required maxlength="255" value="{{ old('phone') ?? $profile->phone ?? ' ' }}">
21         </div>
22         <div class="form-group">
23             <input type="text" class="form-control" name="address" placeholder="Адрес доставки"
24                 required maxlength="255" value="{{ old('address') ?? $profile->address ?? ' ' }}">
25         </div>
26         <div class="form-group">
27             <textarea class="form-control" name="comment" placeholder="Комментарий"
28                 maxlength="255" rows="2">{{ old('comment') ?? $profile->comment ?? ' ' }}</textarea>
29         </div>
30         <div class="form-group">
31             <button type="submit" class="btn btn-success">Оформить</button>
32         </div>
33     </form>
34 @endsection
```

Рисунок 20 – Оформление заказа

Просмотр и управление заказами обрабатываются в OrderController, который является центральным компонентом пользовательской панели, отвечающим за отображение истории заказов и их текущего статуса. Функциональность модуля включает в себя два основных метода: index() для списка заказов и show() для просмотра деталей конкретного заказа. На рисунке 21 демонстрируется код контроллера заказа.

```
<?php

namespace App\Http\Controllers;

use App\Models\Order;

no usages
class OrderController extends Controller {

    public function index() {
        $orders = Order::whereUserId(auth()->user()->id)
            ->orderBy('created_at', 'desc')
            ->paginate(5);
        $statuses = Order::STATUSES;
        return view('user.order.index', compact('orders', 'statuses'));
    }

    public function show(Order $order) {
        if (auth()->user()->id !== $order->user_id) {
            // можно просматривать только свои заказы
            abort(404);
        }
        $statuses = Order::STATUSES;
        return view('user.order.show', compact('order', 'statuses'));
    }
}
```

Рисунок 21 – Просмотр и управление заказами

Метод index() активируется при обращении пользователя к списку своих заказов. Используя модель Order, контроллер извлекает из базы данных все заказы текущего пользователя, сортирует их по дате создания и организует постранично. Константа Order::STATUSES предоставляет набор возможных статусов заказа, что позволяет пользователям понимать, на каком этапе обработки находится их заказ. Эти данные представлены в шаблоне

index.blade.php, который формирует пользовательский интерфейс для навигации по истории заказов.

Метод show() вызывается при запросе деталей конкретного заказа. Система безопасности, реализованная в контроллере, гарантирует, что пользователи могут просматривать только свои заказы, предотвращая доступ к заказам других клиентов. Если проверка не проходит, то происходит переадресация на страницу ошибки 404. Для заказов, прошедших проверку, информация о статусе заказа извлекается и отображается вместе с детальной информацией о заказе в шаблоне show.blade.php.

В совокупности, разработанная пользовательская панель представляет собой мощный инструмент для оптимизации торговых операций на интернет-витрине, способствуя повышению уровня удовлетворенности клиентов за счет удобства использования, оперативности обработки заказов.

3.4 Описание web-приложения

3.4.1 Пользовательская панель

Основная цель пользовательской части - предоставить посетителям удобный и интуитивно понятный интерфейс для просмотра и покупки товаров, а также доступ к информации о компании и услугах. Доступ к следующим страницам открыт для всех пользователей, включая гостей:

- главная;
- авторизация;
- регистрация;
- каталог;
- о нас;
- доставка и оплата;
- корзина.

Главная страница (рисунок 22) является первым, что видят посетители при заходе на сайт. Здесь расположены актуальные предложения, акции и

новинки. Дизайн страницы создан так, чтобы привлечь внимание и побудить к покупке [1], [7].

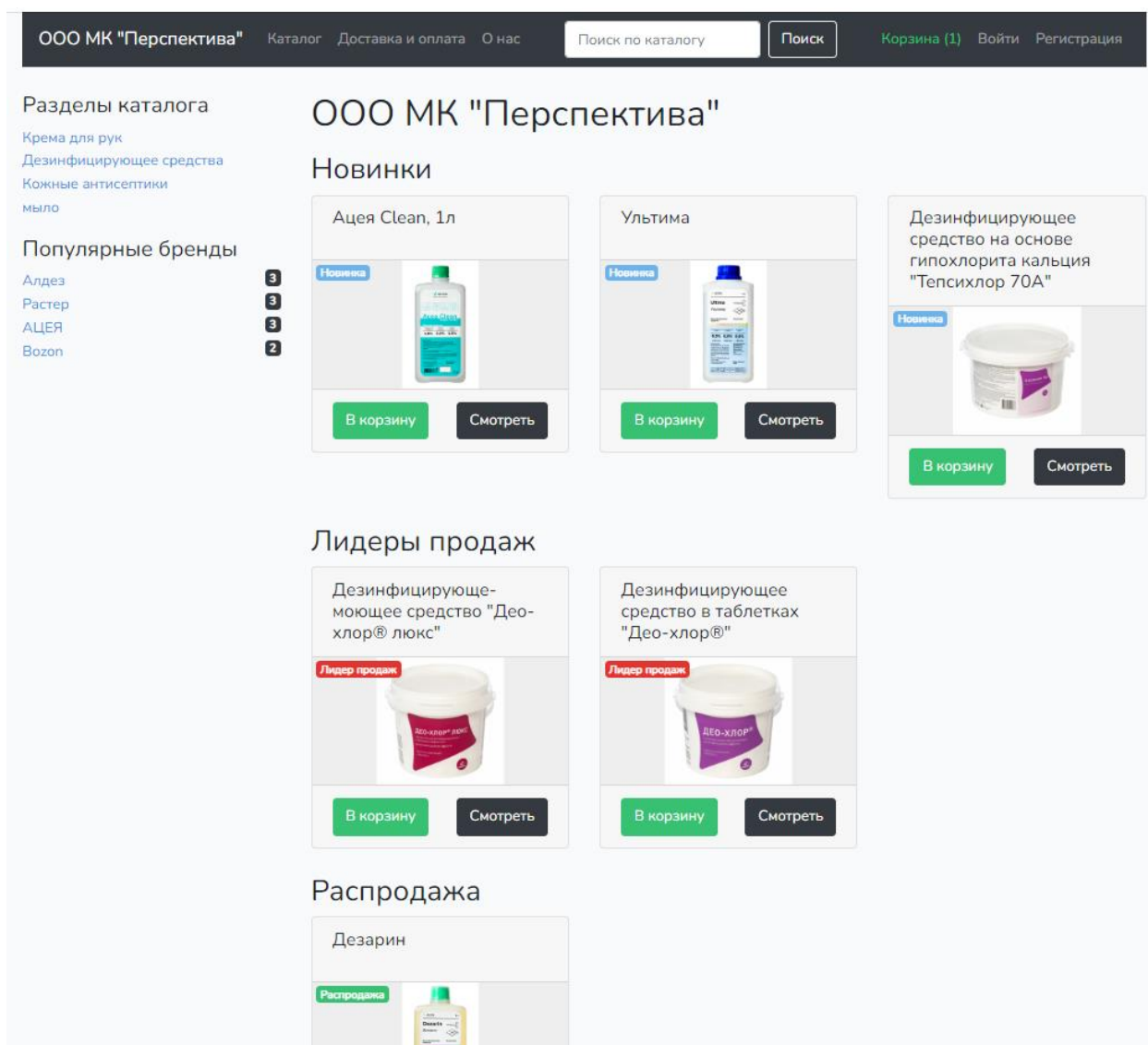


Рисунок 22 – Главная страница интернет-витрины

В центре внимания — разделы с товарами, включающие Новинки, Лидеры продаж и Распродажа. Каждый раздел состоит из карточек товаров с четкими изображениями, названиями и кратким описанием. Пользователи могут добавить товары в корзину или перейти к более подробному просмотру, используя соответствующие кнопки под каждым товаром.

Слева расположено вертикальное меню каталога, где представлены категории товаров, например, «Кремы для рук», «Дезинфицирующие средства», что облегчает поиск нужной продукции. Также присутствует список популярных брендов, представленный в виде логотипов с возможностью перехода к товарам выбранного производителя.

Каталог (рисунок 23) содержит список всех доступных товаров с разделением по категориям и фильтрами для упрощения поиска нужного продукта.

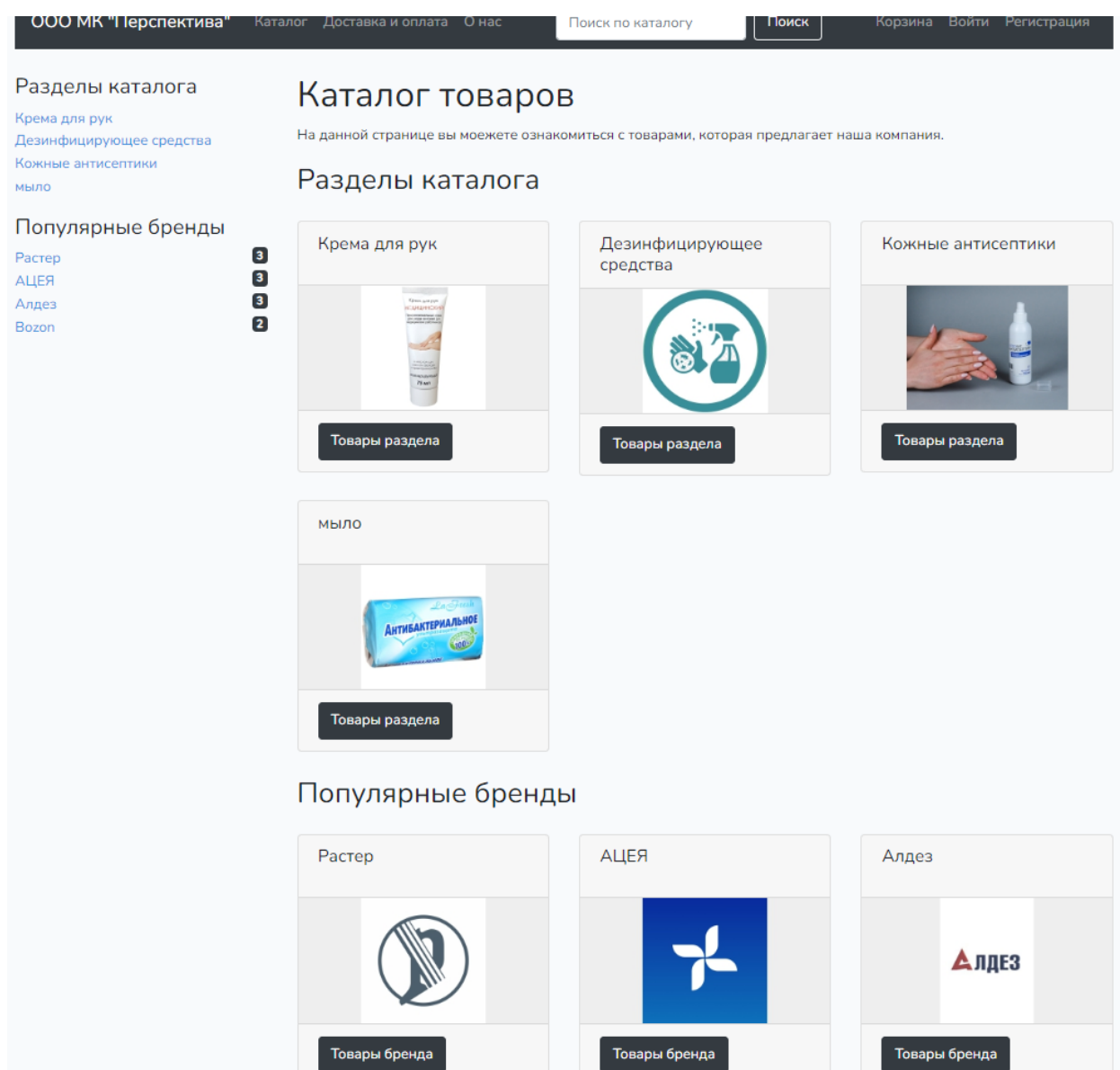


Рисунок 23 - Каталог товаров

Каталог разбит на разделы, каждый из которых включает иконку и название категории. Товары в каждом разделе представлены в виде сетки с картинками и подписями. В нижней части страницы находится дополнительный раздел, выделяющий популярные бренды с их логотипами. Вся композиция страницы выполнена в чистом и структурированном стиле, предназначенном для удобства покупателей в навигации по различным товарам.

Страница «О нас» (рисунок 24) рассказывает посетителям о компании и своих возможностях, помогая создать доверительные отношения с клиентами.

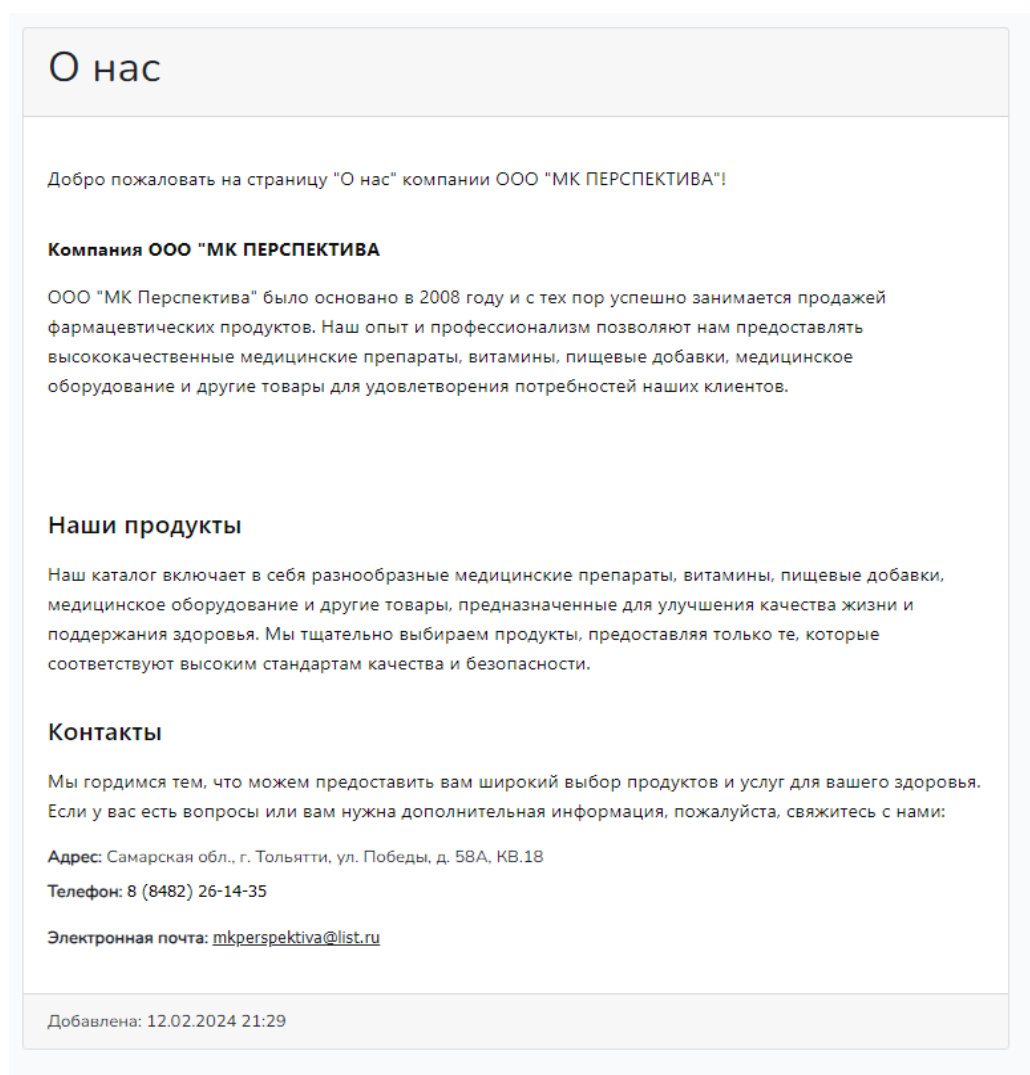



Рисунок 24 – Страница «О нас»

На странице «О нас» информация о дате добавления играет важную роль. Она не только помогает пользователям оценить актуальность представленной информации, но и обеспечивает прозрачность и достоверность контента. Знание даты добавления позволяет пользователям делать информированные решения и убеждаться в актуальности представленной информации.

На странице «Доставка и оплата» (рисунок 25) представлена информация о способах доставки товаров клиентам. Этот раздел важен для покупателей, так как он помогает им понять, каким образом они получают свои покупки и какие варианты доставки доступны. Это создает удобство для клиентов и помогает им выбрать наиболее подходящий способ получения товара, что в свою очередь может повлиять на решение о совершении покупки.

Доставка и оплата



Способы доставки

Доставка

При заказе от 15 000 р. делаем БЕСПЛАТНУЮ доставку до терминала транспортной компании!

Доставка происходит с нашего склада до терминала транспортной компании в Вашем городе или до указанного Вами местоположения. Мы можем сократить расходы, связанные с перевозкой товара, организовав доставку попутным грузом. Этот способ выгоден при перевозке небольшого количества товара.

Перевозка груза оплачивается по факту его прибытия – в местном терминале ТК или заранее по безналичному расчёту. Мы можем взять на себя организацию и оплату перевозки. В данном случае стоимость доставки включается в стоимость товара. Цена перевозки зависит от веса и объёма товара.

Доставка по России осуществляется через транспортные компании .

Доставка по региону и транспортной компанией является бесплатной при суммарной стоимости заказа от 15 тыс. рублей.

Сроки доставки при желании могут быть точно определены при оформлении заказа с менеджером.

Оплата

Оплата осуществляется безналичным расчётом или другим выбранным способом оговоренным при оформлении заказа с менеджером.

Добавлена: 12.02.2024 21:06

Рисунок 25 – страница «Доставка и оплата»

На странице «корзина» (рисунок 26) представлена информация о товарах, добавленных в корзину покупателем. Этот раздел играет ключевую роль в процессе покупки, поскольку он позволяет клиентам видеть выбранные товары, их количество и общую стоимость. Здесь покупатели могут легко управлять содержимым корзины, изменять количество товаров или удалять их, а также оценить общую стоимость заказа. Такая информация помогает покупателям принимать информированные решения о покупке и осуществлять заказы с уверенностью.

The screenshot shows the 'Ваша корзина' (Your cart) page. At the top, there is a navigation bar with the company name 'ООО МК "Перспектива"', a search bar, and links for 'Каталог', 'Доставка и оплата', 'О нас', 'Корзина (2)', 'Войти', and 'Регистрация'. On the left, there are sections for 'Разделы каталога' (Hand cream, Disinfectant, Skin antiseptics, Soap) and 'Популярные бренды' (Aldez, Raster, ACEA, Vozon). The main area contains a table with two items:

№	Наименование	Цена	Кол-во	Стоимость	
1	АБСОЛЮЦИД лайт	220.00	2	440.00	
2	АБСОЛЮСЕПТ 75	250.00	1	250.00	
Итого				690.00	

Buttons for 'Очистить корзину' (Clear cart) and 'Оформить заказ' (Place order) are visible.

Рисунок 26 – Корзина пользователя

Когда пользователь нажимает кнопку «Оформить», он переходит к форме для оформления заказа. Здесь пользователю предложено ввести свои личные данные, такие как имя, электронная почта, номер телефона, адрес доставки и любые комментарии. Эта форма необходима для того, чтобы пользователь мог завершить процесс покупки, предоставляя всю необходимую информацию для организации доставки и для связи. После того как все данные будут введены и пользователь снова нажмет кнопку "Оформить", заказ будет успешно сформирован, и пользователь получит подтверждение о выполнении покупки (см. рисунок 27).

Оформить заказ

Рисунок 27 – Оформление заказа

Страница авторизации (рисунок 28) предоставляет пользователям возможность входа в личный кабинет. Это важный элемент безопасности, позволяющий защитить данные пользователя.

Регистрация

Войти

 Запомнить меня
 [Забыли пароль?](#)

Рисунок 28 -Форма авторизации и регистрации в системе

После авторизации у пользователя есть возможность перейти в свой личный профиль, где он может просмотреть информацию о своих заказах (рисунок 29).

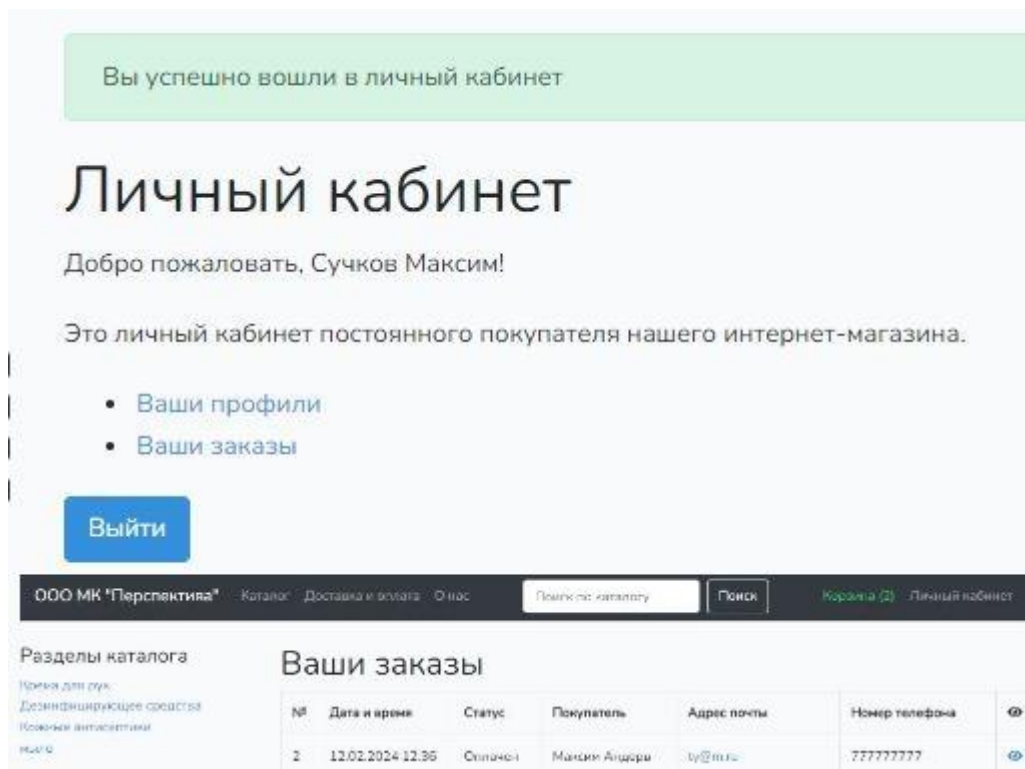


Рисунок 29 – Личный кабинет пользователя

На странице личного кабинета предоставлены следующие опции: «Ваш профиль» пользователь может просмотреть и управлять своими личными данными; «Ваши заказы» здесь пользователь может просмотреть информацию о своих заказах, включая детали каждого заказа.

Таким образом, пользовательская часть интернет-витрины представляет собой гармоничное сочетание удобства использования и информативности. От яркой главной страницы с актуальными предложениями до страницы оформления заказа, каждый элемент разработан с учетом потребностей пользователей. Каталог товаров, разделы с информацией о компании и услугах, а также интуитивно понятный интерфейс корзины позволяют

пользователям легко находить необходимую информацию и осуществлять покупки с уверенностью.

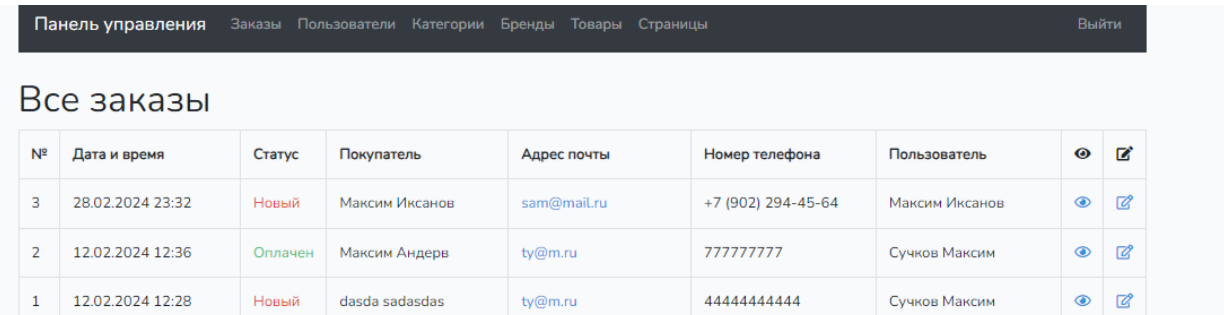
3.4.2 Обзор административной панели интернет-витрины

Эффективное управление административной частью интернет-магазина является ключевым аспектом его успешной работы. В данной работе будет представлен обзор административной панели интернет-витрины, включая анализ основных страниц и их функциональности.

Административная панель интернет-витрины включает в себя следующие страницы:

- заказы,
- пользователи,
- категории,
- бренды,
- товары,
- страницы.

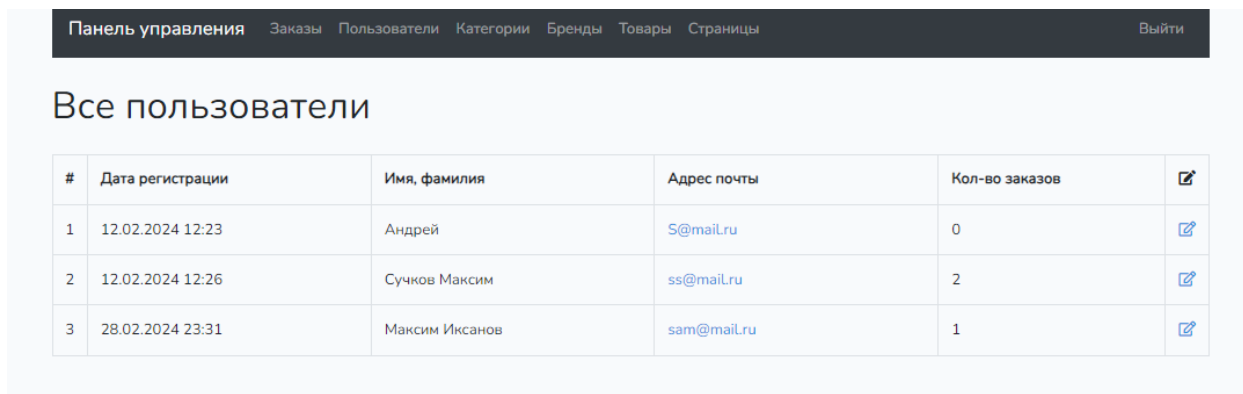
Страница «Заказы» предоставляет возможность администратору управлять всеми заказами, поступающими от пользователей. На этой странице администратор может просматривать список заказов, изменять их статусы и просматривать детальную информацию о каждом заказе (рисунок 30).



№	Дата и время	Статус	Покупатель	Адрес почты	Номер телефона	Пользователь	👁	✎
3	28.02.2024 23:32	Новый	Максим Иксанов	sam@mail.ru	+7 (902) 294-45-64	Максим Иксанов	👁	✎
2	12.02.2024 12:36	Оплачен	Максим Андерв	ty@m.ru	777777777	Сучков Максим	👁	✎
1	12.02.2024 12:28	Новый	dasda sadasdas	ty@m.ru	44444444444	Сучков Максим	👁	✎

Рисунок 30 – Управление заказами

На странице «Пользователи» (рисунок 31) администратор может управлять зарегистрированными пользователями. Здесь он может просматривать список пользователей, редактировать их данные.

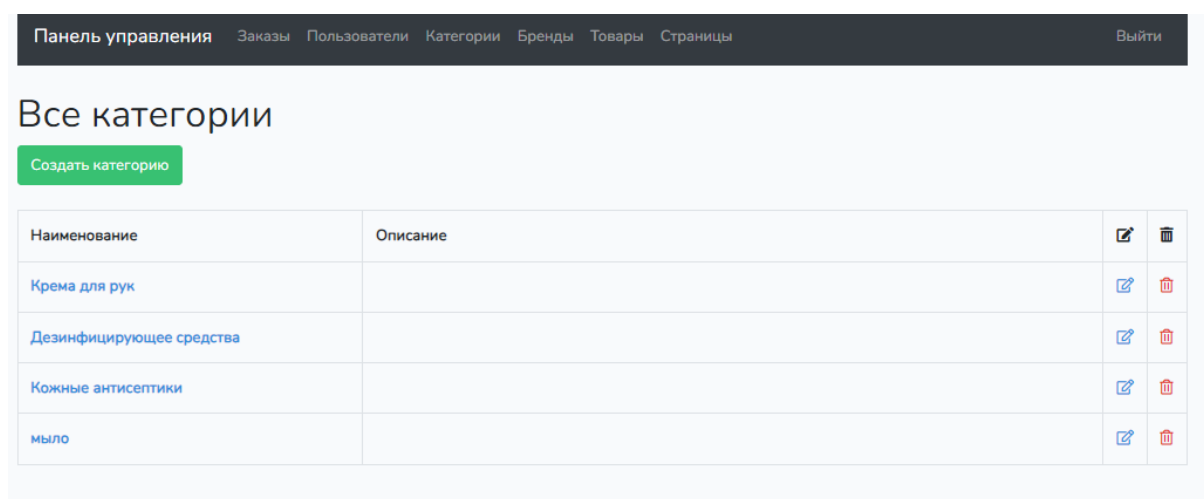


The screenshot shows the 'All Users' page in an admin interface. At the top, there is a navigation bar with 'Панель управления' (Control Panel) and several menu items: 'Заказы' (Orders), 'Пользователи' (Users), 'Категории' (Categories), 'Бренды' (Brands), 'Товары' (Goods), and 'Страницы' (Pages). A 'Выйти' (Logout) button is on the right. Below the navigation bar, the page title is 'Все пользователи' (All Users). A table lists three users with columns for '#', 'Дата регистрации' (Registration Date), 'Имя, фамилия' (Name, Surname), 'Адрес почты' (Email Address), 'Кол-во заказов' (Number of Orders), and an edit icon. The data rows are as follows:

#	Дата регистрации	Имя, фамилия	Адрес почты	Кол-во заказов	
1	12.02.2024 12:23	Андрей	S@mail.ru	0	
2	12.02.2024 12:26	Сучков Максим	ss@mail.ru	2	
3	28.02.2024 23:31	Максим Иксанов	sam@mail.ru	1	

Рисунок 31 – Управление пользователями

На рисунке 32 представлена страница управления категориями в интерфейсе администратора интернет витрины. Администратор имеет возможность добавлять новые категории товаров с помощью кнопки, предназначенной для этой цели. Каждая категория имеет свою строку в таблице, где можно просматривать и редактировать существующие категории, используя иконку редактирования, и удалять их, используя иконку удаления.



The screenshot shows the 'All Categories' page in an admin interface. At the top, there is a navigation bar with 'Панель управления' (Control Panel) and several menu items: 'Заказы' (Orders), 'Пользователи' (Users), 'Категории' (Categories), 'Бренды' (Brands), 'Товары' (Goods), and 'Страницы' (Pages). A 'Выйти' (Logout) button is on the right. Below the navigation bar, the page title is 'Все категории' (All Categories). A green button labeled 'Создать категорию' (Create Category) is visible. Below the button, a table lists categories with columns for 'Наименование' (Name), 'Описание' (Description), an edit icon, and a delete icon. The data rows are as follows:

Наименование	Описание		
Крема для рук			
Дезинфицирующее средства			
Кожные антисептики			
мыло			

Рисунок 32 – Управление категориями

На рисунке 33 представлен раздел управления брендами на платформе интернет-витрины. Администратор сайта может выполнять несколько действий. С помощью кнопки для создания новых брендов можно добавлять новые записи в каталог. В таблице с брендами администратор может просматривать список брендов с их названиями и описаниями. Редактирование деталей бренда доступно через иконку в виде карандаша, расположенную в каждой строке, а удаление бренда осуществляется с помощью иконки корзины для мусора.

Панель управления Заказы Пользователи Категории Бренды Товары Страницы Выйти

Все бренды каталога

Создать бренд

Наименование	Описание		
Растер	производитель дезинфицирующих средств и оборудования для аэрозольной дезинфекции (аэрозольных генераторов)		
АЦЕЯ	производственное объединение, которое создаёт дезинфицирующие и гигиенические средства, соответствующие мировым стандартам качества.		
Bozon	производственно-дистрибьюторская компания, с 2008 года успешно работающая на российском рынке.		
Алдез	занимается разработкой, производством и реализацией средств дезинфекции, косметики и гигиены под собственными торговыми марками.		

Рисунок 33 – Управление пользователями

На рисунке 34 отображается раздел управления товарами интернет-витрины, где администратор может выполнять различные операции. В списке, расположенном над кнопкой «Создать товар», указаны категории, которые администратор может выбрать для добавления нового товара. Под этим списком и кнопкой располагается таблица, в которой указаны названия и описания товаров, предоставляющая возможность редактировать информацию о товаре или удалять товар из каталога с помощью соответствующих иконок. Также присутствуют элементы управления для навигации по страницам таблицы товаров.

Панель управления [Заказы](#) [Пользователи](#) [Категории](#) [Бренды](#) [Товары](#) [Страницы](#) Выйти

Все товары

- [Крема для рук](#)
- [Дезинфицирующее средства](#)
- [Кожные антисептики](#)
- [мыло](#)

[Создать товар](#)

Наименование	Описание		
Дезинфицирующее средство в таблетках "Део-хлор®"	90 таблеток по 3,4 г Дезинфицирующее средство на основе натриевой соли дихлоризоциануровой кислоты, содержание активного хлора в средстве 44,5+4,5%		
Дезинфицирующе-моющее средство "Део-хлор® люкс"	Объем:90 таблеток по 3,4 г,50 ТАБЛЕТОК 1,7 Г, 300 ТАБЛЕТОК 3,4 Г,1 КГ Таблетированное дезсредство с моющим эффектом. Обладает вирулицидным, бактерици		
Дезинфицирующее средство на основе гипохлорита кальция "Тепсихлор 70А"	Объем: 2кг, 25кг Гранулированное дезинфицирующее средство. Может использоваться для обеззараживания воды, используемой для питья, а также как пр		
Ультима	Объем: 1 л 5 л Универсальный концентрат на основе перекиси водорода, кислот Универсальный концентрат на основе перекиси водорода и кислот Ул		
Дезарин	Объем: 1 л 5 л Эффективный концентрат для дезинфекции поверхностей, ИМН Многофункциональный дезинфицирующий препарат с моющим эффектом. На осн		

« 1 2 3 »

Рисунок 34 –Управление товарами

На рисунке 35 представлен раздел «Страницы» отображает список всех страниц сайта в виде таблицы. Каждая строка таблицы содержит следующую информацию: порядковый номер страницы, название страницы и ЧПУ (человекопонятный URL) страницы. На данном изображении присутствуют две страницы: «Доставка и оплата» и «О нас».

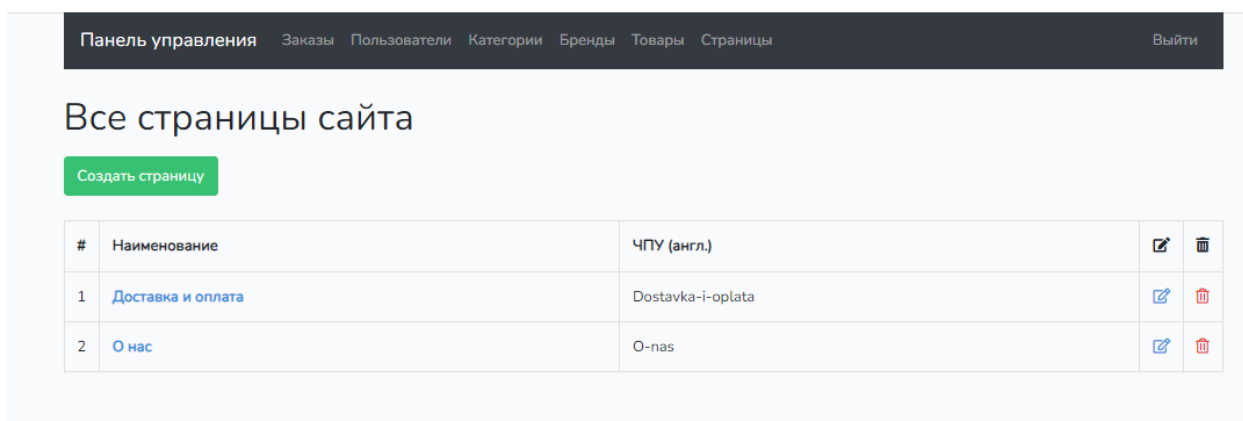


Рисунок 35 – Редактирование и создание новых страниц на интернет-витрине

Обзор административной панели интернет-витрины показал, что она предоставляет обширные возможности для эффективного управления интернет-магазином. Страницы, такие как «Заказы», «Пользователи», «Категории», «Бренды», «Товары» и «Страницы», предоставляют администратору все необходимые инструменты для управления контентом, заказами, пользователями и категориями товаров.

Каждая страница имеет интуитивно понятный интерфейс, что делает процесс управления витриной более удобным и эффективным. Различные функциональные возможности, такие как просмотр, редактирование и удаление, доступны на всех страницах, что обеспечивает полный контроль над содержимым магазина.

Таким образом, обзор административной панели подчеркивает ее важную роль в успешной работе интернет-магазина и демонстрирует, что она обладает всеми необходимыми инструментами для эффективного управления различными аспектами онлайн-бизнеса.

3.5 Тестирование web-приложения

В процессе функционального тестирования web-приложения были осуществлены проверки следующих основных функциональностей:

- а) процедуры авторизации и аутентификации пользователей:
- 1) проверка успешной возможности входа в систему с использованием зарегистрированного аккаунта,
 - 2) проверка обработки ошибок при вводе неверных учетных данных.
 - 3) проверка процесса регистрации нового профиля и успешной регистрации пользователя,
 - 4) проверка доступа к конкретным функциям в соответствии с назначенными правами пользователя,
 - 5) проверка корректного отображения информации о пользователе после авторизации;
- б) создание и отправка формы заказа:
- 1) проверка корректного отображения формы заказа на странице,
 - 2) проверка возможности пользователя заполнить форму заказа с необходимой информацией,
 - 3) проверка отправки формы заказа и сохранения данных в базе данных,
 - 4) проверка процедуры отправки уведомления администратору при получении новой заявки на заказ;
- в) рассмотрение и внесение изменений в информацию о пользователе:
- 1) проверка доступа к просмотру информации о пользователе, включая имя, почта и т. д.,
 - 2) проверка корректного отображения информации о пользователе на странице,
 - 3) проверка возможности внесения изменений и сохранения обновленных данных пользователя,
 - 4) проверка сохранения изменений и обновления информации о пользователе в базе данных;
- г) обновление статуса заказа:

- 1) проверка доступности функции изменения статуса заказа с «новый» на «обработан», «оплачен», «доставлен» и «завершен»,
 - 2) проверка того, что уведомление о изменении статуса заказа было успешно отправлено пользователю,
 - 3) проверка сохранения изменений статуса заказа в базе данных;
- д) обработка ошибок и исключений:
- 1) проверка отображения информативных сообщений об ошибках при неправильном вводе данных в формы (например, неверное имя пользователя или пароль),
 - 2) проверка корректного отображения страницы с сообщением об ошибке при возникновении исключительных ситуаций, таких как ошибки сервера или недоступность базы данных.

3.6 Расчет экономической эффективности

В настоящее время существует множество методов для определения экономической выгоды проектов и инвестиций. Среди наиболее распространенных подходов можно выделить оценку экономической целесообразности инвестиций, анализ приведенных издержек, а также сравнение финансовых результатов работы системы с необходимыми для ее создания и внедрения расходами.

Конкретно, для оценки рентабельности интернет-витрины предпочтение отдается методу, который учитывает сопоставление доходов от работы системы с инвестициями в ее разработку и внедрение. Этот подход особенно актуален при разработке программного обеспечения, представляющего собой модифицированную версию существующего продукта, когда вносятся определенные улучшения и дополнения в первоначальную версию.

В таблице 5 представлены расчеты затрат на рабочие часы.

Таблица 5 – Постепенный план внедрения и развития системы

Этап работы	Трудозатраты, дни
Исследование технологических решений	4
Выработка новой стратегии	4
Формирование требований к интернет-витрине	6
Создание интернет-витрины	45
Реализация проекта в эксплуатацию	3
Итого:	66

Таким образом, на создание web-приложения потребуется 66 дней, что приблизительно равно трем месяцам.

Рассчитаем затраты на проектирование и реализацию web-приложения для фармацевтической компании и оценим ее рентабельность.

В таблице 6 приведены данные о заработной плате членов команды, занимающейся разработкой web-приложения.

Таблица 6 – Расчет заработной платы сотрудников

Роль в проекте	Ставка в час, руб.	Часы работы	Итоговый доход, руб.
разработчик веб-проекта	300	240	72 000
Координатор продаж	200	60	12 000
Итого	-	300 ч.	84 000

Бюджет, потраченный на разработку интернет-витрины для фармацевтической компании представлен в таблице 7.

Таблица 7 – Издержки на разработку и интеграцию интернет-витрины

Расходы	Сумма, руб.
Затраты на оборудование	4 000
Основная заработная плата сотрудников	84 000
Дополнительная заработная плата сотрудников	14 400
Расходы на социальные отчисления	33 120
Электропитание	1 000
Расходы на прочие нужды	3 000
Итого	139 520

Эффективность предоставления информации о продукции компании покупателям, измеряемая в экономии рабочего времени, определяется с использованием формулы (1):

$$C_{\text{ч}} = K \times Y_{\text{а}} \times 12, \quad (1)$$

где $C_{\text{ч}}$ – количество часов, сэкономленное благодаря предоставлению информации;

K – число сотрудников, занятых учетом продаж;

$Y_{\text{а}}$ – время, требуемое для проведения учета.

Поскольку каждый день два работника заняты учетом, общее сокращение времени будет:

$$C_{\text{ч}} = 2 \times 8 \times 12 = 192 \text{ч.}$$

Эффективность использования рабочего времени сотрудников, приводящая к экономии, определяется с помощью формулы 2:

$$C_э = C_ч/P_ч/P_д \times C_{зп}, \quad (2)$$

где $C_э$ – величина экономии;

$P_ч$ – количество отработанных часов за 1 день;

$P_д$ – количество отработанных часов за 1 месяц;

$C_{зп}$ – среднемесячная заработная плата специалиста.

Тогда:

$$C_э = 192/8/22 \times 75000 = 78\,545 \text{ руб./год.}$$

В итоге, экономия достигает 78 545 рублей в год.

Формула для расчета NPV (чистой приведенной стоимости) рассчитывается по формуле 3:

$$NPV = \sum_{t=1}^n \frac{CF_t}{(1+i)^t} - I, \quad (3)$$

где t , n – количество лет;

CF_t – денежный поток за период t ;

i – ставка дисконтирования;

I – первоначальные инвестиции в проект.

В таблице 8 рассчитаем NPV за 5 лет

Таблица 8 – Расчет NPV

Период	Первоначальные инвестиции, руб.	Денежный доход, руб.	Коэффициент дисконтирования	Дисконтированный доход	Дисконтированный денежный поток
1	139 520				-139 520
2		78 545	1	78 545	-60 975
3		78 545	0,95	74 617	13 642
4		78 545	0,90	70 690	84 332
5		78 545	0,85	66 763	151 095

На пятом году эксплуатации web-приложения фармацевтической компании прогнозируется достижение чистой приведенной стоимости (NPV) в размере 151 095 рублей. При этом ожидается, что инвестиции в web-приложение начнут приносить прибыль уже на третий год её функционирования.

Для определения фактического коэффициента экономической эффективности можно воспользоваться следующей формулой:

$$E_{\phi} = 1/0,92 = 1,087$$

Изображение, демонстрирующее результаты NPV, показано на рисунке 36.

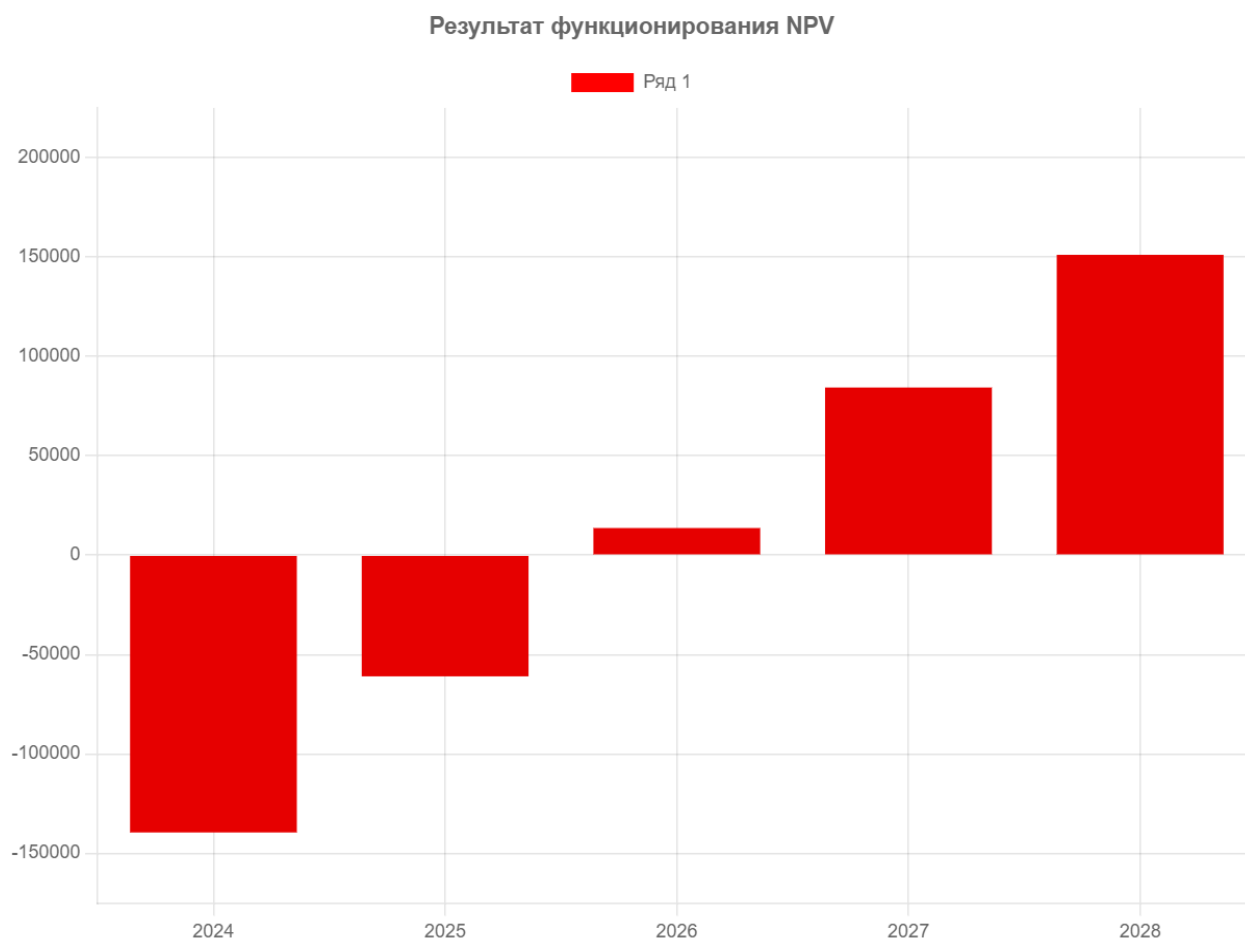


Рисунок 36 – Результат функционирования NPV

Выводы по Главе 3

В третьей главе работы был выполнен процесс реализации интернет-витрины. На начальных этапах были осуществлены выбор средств реализации, таких как система управления данными и фреймворк для разработки. Затем были разработаны административная и пользовательская панели, каждая из которых прошла детальное описание. После этого был проведен расчет экономической эффективности проекта.

Эта глава является ключевой для понимания процесса создания интернет-витрины и включает в себя как технические аспекты разработки, так и экономические расчеты, необходимые для оценки эффективности проекта.

Заключение

В ходе выполнения бакалаврской работы по разработке и администрированию интернет-витрины для ООО МК «Перспектива», специализирующегося на производстве фармацевтической продукции, были успешно решены все основные задачи.

Обзор существующих решений на рынке программного обеспечения показал, что большинство из них не предоставляют удобных инструментов для клиентов, таких как мгновенное ознакомление с ассортиментом и акциями, возможность добавления всех интересующих товаров в корзину и оформление заявки для обратной связи. Однако, наша разработанная интернет-платформа для ООО МК «Перспектива» решает эти проблемы, обеспечивая клиентам простой и удобный способ просмотра продукции, быстрое добавление товаров в корзину и возможность оформления заявки для обратной связи. Это значительно повышает удобство использования платформы для клиентов и способствует улучшению их опыта покупок.

В процессе разработки интернет-витрины для ООО МК «Перспектива» были применены CASE-технологии моделирования, чтобы обеспечить более системный и структурированный подход к проектированию и реализации системы. Использование CASE-средств помогло создать диаграммы прецедентов для описания основных функциональных возможностей системы и их взаимодействия с различными типами пользователей. Это предоставило ясное представление о том, как различные актеры будут взаимодействовать с системой и какие функциональные требования должны быть реализованы. Кроме того, CASE-средства помогли создать логическую схему данных, которая определяет структуру данных системы на более абстрактном уровне, а затем физическую схему данных, определяющую способ хранения и организации данных в базе данных. Это обеспечило эффективное проектирование базы данных, соответствующей требованиям проекта. Использование CASE-технологий моделирования вместе с методологией

разработки позволило эффективно реализовать web-приложение. Web-приложение было разделено на несколько категорий пользователей: гость, администратор и клиент.

Для реализации проекта используется база данных, созданная с использованием системы управления базами данных MySQL. Веб-приложение отображается как веб-сайт, на котором страница создается с использованием HTML и CSS в стиле. Серверная логика обрабатывается с использованием языка программирования PHP, а для более удобной и эффективной разработки использовался фреймворк Laravel.

Проведенные расчеты эффективности внедрения интернет-витрины для ООО МК «Перспектива» демонстрируют значительные перспективы для компании. Оценка финансовой привлекательности проекта основывается на его периоде окупаемости, который составляет примерно три года. Учитывая затраты на разработку и интеграцию интернет-витрины, а также ожидаемые экономии в затратах на рабочее время персонала, проект является экономически выгодным. Кроме того, коэффициент экономической эффективности проекта превышает единицу, что подтверждает его способность принести значительную прибыль компании и улучшить ее финансовое положение

Список используемой литературы

1. Адамчик, М. В. Дизайн и основы композиции в дизайнерском творчестве и фотографии / М. В. Адамчик. – Минск : Харвест, 2017. 192 с.
2. Бенкен, Е. С. PHP, MySQL, XML: программирование для Интернета. – 3-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2011. 287 с.
3. ВЕБ-представительство компании: сущность и методы создания. [Электронный ресурс]. URL: http://www.aselibrary.ru/press_center/journal/irr/irr3648/irr36483710/irr364837103714/irr3648371037143724 (дата обращения: 15.02.2024)
4. Водяхо, А. И. Архитектурные решения информационных систем/ А. И. Водяхо, Л. С. Выговский. – М: Лань, 2017. 356 с
5. Гаврилов, М. В. Информатика и информационные технологии: учеб. для студентов вузов / М. В. Гаврилов, В. А. Климов. – 3-е изд. перераб., и доп.; гриф УМО. Москва :Юрайт, 2013. 378 с.
6. Гвоздева, Т. В. Проектирование информационных систем: технология автоматизированного проектирования. Лабораторный практикум : учебное пособие / Т. В. Гвоздева, Б. А. Баллод. – 2-е изд., стер. – СанктПетербург : Лань, 2020. 156 с
7. Гультяев, А. К. Проектирование и дизайн пользовательского интерфейса / А. К. Гультяев, В. А. Машин. – СПб. : Корона-Принт, 2018. 350 с.
8. Ерохин, В. В., Прогонышева Д. А. Безопасность информационных систем: учебное пособие/В. В. Ерохин, Д. А. Погонышева. – Флинта, 2015. 184 с.
9. Зельдман, Д. Web-дизайн по стандартам / Д. Зельдман. – М. : НТ Пресс, 2019. 440 с.
10. Круг, С. Как сделать сайт удобным. Юзабилити по методу Стива Круга / С. Круг. – СПб. : Питер, 2019. – 207 с.

11. Купер, А. Об интерфейсе. Основы проектирования взаимодействия / А. Купер, Р. Рейман, Д. Кронин. – М. : Символ-Плюс, 2019. - 688 с.
12. Фаулер, М. UML в кратком изложении: применение стандартного языка объектного моделирования: пер. с англ. / М. Фаулер, К. Скотт. – М.: Вильямс, 2010. 580 с.
13. Alan Forbes. The Joy of PHP / 2020. 128 p.
14. Database Systems Concepts, Languages and Architectures [Электронный ресурс]. URL: <https://users.dimi.uniud.it/~angelo.montanari/logicDesign.pdf> (дата обращения: 10.02.2024).
15. Laravel documentation // Laravel: Model View Controller [Электронный ресурс]. URL: <https://laravel.com/docs> (Дата обращения 18.01.2024).
16. Laravel - The PHP Framework For Web Artisans [Электронный ресурс]: Документация по Laravel. URL: <https://laravel.com/> (дата обращения: 19.05.2022).
17. MySQL Workbench [Электронный ресурс]. URL: <https://www.mysql.com/products/workbench/> (дата обращения: 21.01.2024).
18. PHP Language Specifications [Электронный ресурс]. URL: <https://github.com/php/php-langspeg> (дата обращения: 29.01.2024).
19. PHP Programming Cookbook. Exelixis Media P.C., 2016. 71 p.
20. Scott T. Systems Analysis and Design. // Т. Scott, Harry J. Rosenblatt. 2016. 752p.
21. What is MySQL [Электронный ресурс]. URL: <https://www.oracle.com/mysql/what-is-mysql/> (дата обращения: 05.02.2024).

Приложение А

Программный код web-приложения

```
class MainController extends Controller
{
    public function home()
    {
        $newArrivals = Product::where('is_new', true)->latest()->limit(6)->get();
        $bestSellers = Product::where('is_best_seller', true)->limit(6)->get();
        $sales = Product::where('is_on_sale', true)->limit(6)->get();

        return view('main.home', [
            'newArrivals' => $newArrivals,
            'bestSellers' => $bestSellers,
            'sales' => $sales,
        ]);
    }
}

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ООО МК "Перспектива"</title>
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
    <link rel="stylesheet" href="{{ asset('css/styles.css') }}">
</head>
<body>
    <header class="header">
```

```

<div class="container">
  <h1>ООО МК "Перспектива"</h1>
  <nav class="nav">
    <ul>
      <li><a href="{{ route('home') }}">Главная</a></li>
      <li><a href="{{ route('catalog.index') }}">Каталог</a></li>
      <li><a href="{{ route('about') }}">О нас</a></li>
      <li><a href="{{ route('contact') }}">Контакты</a></li>
    </ul>
  </nav>
</div>
</header>

<main class="main-content">
  <div class="container">
    <section class="section">
      <h2>Новинки</h2>
      <div class="products-grid">
        @foreach($newArrivals as $product)
          <div class="product-card">
            name }}">
            <h3>{{ $product->name }}</h3>
            <p>{{ $product->price }} руб.</p>
          </div>
        @endforeach
      </div>
    </section>

    <section class="section">

```

```

<h2>Лидеры продаж</h2>
<div class="products-grid">
  @foreach($bestSellers as $product)
    <div class="product-card">
      name }}">
      <h3>{{ $product->name }}</h3>
      <p>{{ $product->price }} руб.</p>
    </div>
  @endforeach
</div>
</section>

<section class="section">
  <h2>Распродажа</h2>
  <div class="products-grid">
    @foreach($sales as $product)
      <div class="product-card">
        name }}">
        <h3>{{ $product->name }}</h3>
        <p>{{ $product->price }} руб.</p>
      </div>
    @endforeach
  </div>
</section>
</div>
</main>

<footer class="footer">

```

```

<div class="container">
  <p>&copy; 2024 ООО МК "Перспектива". Все права защищены.</p>
</div>
</footer>
</body>
</html>
<div class="form-element">
  <label for="profile_title">Название профиля</label>
  <input type="text" id="profile_title" class="input-field" name="title"
placeholder="Название профиля"
value="{{ old('title', $profile->title ?? ' ' ) }}" maxlength="255" required>
</div>
<div class="form-element">
  <label for="profile_name">Имя и Фамилия</label>
  <input type="text" id="profile_name" class="input-field" name="name"
placeholder="Имя и Фамилия"
value="{{ old('name', $profile->name ?? ' ' ) }}" maxlength="255" required>
</div>
<div class="form-element">
  <label for="profile_email">Адрес электронной почты</label>
  <input type="email" id="profile_email" class="input-field" name="email"
placeholder="Email"
value="{{ old('email', $profile->email ?? ' ' ) }}" maxlength="255" required>
</div>
<div class="form-element">
  <label for="profile_phone">Телефон</label>
  <input type="text" id="profile_phone" class="input-field" name="phone"
placeholder="Телефон"
value="{{ old('phone', $profile->phone ?? ' ' ) }}" maxlength="255" required>
</div>

```

```

<div class="form-element">
    maxlength="255" rows="3">{{ old('comment', $profile->comment ?? " )
    <button type="submit" class="submit-button">Сохранить</button>
@csrf
<div class="input-wrapper">
    <label for="product_title">Наименование</label>
    <input type="text" id="product_title" class="input-field" name="name"
    <label for="product_slug_field">ЧПУ (на англ.)</label>
    <input type="text" id="product_slug_field" class="input-field" name="slug"
placeholder="Введите ЧПУ"
    <label for="product_price_field">Цена (руб.)</label>
    <input type="text" id="product_price_field" class="input-field short-input"
name="price" placeholder="Введите цену"
        value="{{ old('price', $product->price ?? " ) }}" required>
    <div class="checkbox-wrapper">
        <input type="checkbox" id="is_new_product" name="new" class="checkbox-
input"
            {{ (isset($product) && $product->new) || old('new') ? 'checked' : " }}
value="1">
        <label for="is_new_product">Новинка</label>
    </div>
    <div class="checkbox-wrapper">
        <input type="checkbox" id="is_hit_product" name="hit" class="checkbox-
input"
            {{ (isset($product) && $product->hit) || old('hit') ? 'checked' : " }}
value="1">
        <label for="is_hit_product">Лидер продаж</label>
    </div>
    <div class="checkbox-wrapper">

```

```



```

```

    }
}
class CartController extends Controller
{
    public function index()
    {
        $cart = session()->get('cart');
        return view('cart.index', compact('cart'));
    }
    public function update(Request $request)
    {
        if ($request->id && $request->quantity) {
            $cart = session()->get('cart');
            $cart[$request->id]["quantity"] = $request->quantity;
            session()->put('cart', $cart);
            return redirect()->back()->with('success', 'Cart updated!');
        }
    }
    public function remove(Request $request)
    {
        if ($request->id) {
            $cart = session()->get('cart');
            if (isset($cart[$request->id])) {
                unset($cart[$request->id]);
                session()->put('cart', $cart);
            }
            return redirect()->back()->with('success', 'Product removed!');
        }
    }
    public function clear()

```



```

    {
        session()->forget('cart');
        return redirect()->back()->with('success', 'Cart cleared!');
    }
}
<body>
    <div class="container">
        <h1>Ваша корзина</h1>
        @if(session('success'))
            <div class="alert alert-success">
                {{ session('success') }}
            </div>
        @endif
        <table class="table">
            <thead>
                <tr>
                    <th>№</th>
                    <th>Наименование</th>
                    <th>Цена</th>
                    <th>Кол-во</th>
                    <th>Стоимость</th>
                    <th></th>
                </tr>
            </thead>
            <tbody>
                @php $total = 0 @endphp
                @if(session('cart'))
                    @foreach(session('cart') as $id => $details)
                        @php $total += $details['price'] * $details['quantity'] @endphp
                    <tr>

```

```

<td>{{ $loop->iteration }}</td>
<td>{{ $details['name'] }}</td>
<td>{{ $details['price'] }}</td>
<td>
    <form action="{{ route('cart.update') }}" method="POST">
        @csrf
        <input type="hidden" name="id" value="{{ $id }}">
        <input type="number" name="quantity" value="{{
$details['quantity'] }}" min="1">
        <button type="submit">Update</button>
    </form>
</td>
<td>{{ $details['price'] * $details['quantity'] }}</td>
<td>
    <form action="{{ route('cart.remove') }}" method="POST">
        @csrf
        <input type="hidden" name="id" value="{{ $id }}">
        <button type="submit">Remove</button>
    </form>
</td>
</tr>
@endforeach
@endif
</tbody>
</table>
<div>
    <strong>Итого: {{ $total }}</strong>
</div>
<div>
    <form action="{{ route('cart.clear') }}" method="POST">

```

```
        @csrf
        <button type="submit">Очистить корзину</button>
    </form>
</div>
<div>
    <form action="{{ route('cart.checkout') }}" method="POST">
        @csrf
        <button type="submit">Оформить заказ</button>
    </form>
</div>
</div>
</body>
```