**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| **ДП.09.02.03.22.182.11.ПЗ** | УТВЕРЖДАЮ  Зам. директора по УР, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_Е.А. Коробкова |

ИНФОРМАЦИОННАЯ СИСТЕМА «СЕРВИС ОБЪЯВЛЕНИЙ»

|  |  |  |
| --- | --- | --- |
| Нормоконтролер: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.Ю. Безносова) |
| Консультант по экономической части: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.А. Белова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.А. Матвеева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (В.А. Ильин) |

Иркутск 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc73484155)

[1 Предпроектное исследование 4](#_Toc73484156)

[1.1 Исследование предметной области 4](#_Toc73484157)

[1.2 Анализ инструментов разработки 5](#_Toc73484158)

[1.3 Обоснование выбора программного продукта 7](#_Toc73484159)

[2 Проектирование 10](#_Toc73484160)

[2.1 Архитектура программного обеспечения 10](#_Toc73484161)

[2.2 Функциональное проектирование 12](#_Toc73484162)

[2.3 Проектирование базы данных 14](#_Toc73484163)

[2.4 Проектирование пользовательского интерфейса 19](#_Toc73484164)

[3 Реализация программного обеспечения 24](#_Toc73484165)

[3.1 Кодирование программного обеспечения 24](#_Toc73484166)

[4 Документирование программного обеспечения 27](#_Toc73484167)

[4.1 Руководство пользователя программного обеспечения 27](#_Toc73484168)

[5 Стоимость разработки и внедрения программного продукта 38](#_Toc73484179)

[5.1 Организационно-экономическое обоснование проекта 38](#_Toc73484180)

[5.2 Расчет затрат на разработку программного продукта 38](#_Toc73484181)

[6.1 Расчет затрат на внедрение программного продукта 42](#_Toc73484182)

[6.2 Основные выводы 42](#_Toc73484183)

[Заключение 43](#_Toc73484184)

[Список использованных источников 44](#_Toc73484185)

[Приложение А - Листинг маршрутов 46](#_Toc73484186)

ВВЕДЕНИЕ

Темой дипломного проекта является информационная система «Сервис объявлений».

В современном мире множество людей пытаются завести пассивный доход, открывая свой личный бизнес, или быть самозанятыми. В любом из случаев им необходима реклама, какое-то объявление, которое поможет привлечь новых клиентов или сотрудников.

Объявлением называют официальное сообщение, уведомление о чём-либо, сделанное в устной (реже письменной) форме.

Самое главное при работе с объявлениями – это правильное хранение всех данных. Все необходимые данные удобно хранить в таблицах. Информация в базе данных (далее БД) тоже представляется в виде таблиц, что позволяет организовать хранение всех необходимых данных объявлений и пользователей. Качество и надежность результатов вышеназванных действий служит основой эффективной работы информационной системы. Актуальным является выставление любых объявлений, а не только определенной тематики. Данную базу можно применить в любом сервисе объявлений, также имеется возможность её доработать. Для этого нужна общая база данных, включающая всю необходимую информацию. И программа, которая будет понятна и автоматизирует работу с базой данных.

Цель: разработать информационную систему «Сервис объявлений».

Задачи:

* + Провести предпроектное исследование.
  + Составить техническое задание на разработку программного продукта в соответствии с ГОСТ.
  + Провести проектирование программного продукта.
  + Реализовать программный продукт.
  + Разработать документы для программного продукта.
  + Рассчитать стоимость разработки и внедрения программного продукта.

База данных должна быть спроектирована так, чтобы обеспечить хранение всех необходимых данных, имея при этом максимально упрощенную структуру. Структура базы данных должна обеспечивать целостность и корректность информации.

1. Предпроектное исследование
   1. Исследование предметной области

Предметной областью дипломного проекта является информационная система «Сервис объявлений» (далее ИС).

Рекламное объявление — это краткое изложение основной информации о товаре или услуге, которое содержит ключевые фразы и способствует привлечению новых клиентов. Рекламное объявление распространяет тексты с предложением товаров и услуг.

Сервис объявлений (электронная доска объявлений) — сайт, на котором размещаются объявления. Сервис объявлений функционально подобна обыкновенной: это сайт, где каждый желающий может вывесить своё объявление, а все посетители сайта — прочитать его. Электронная доска объявлений, как правило, поделена на несколько тематических разделов, согласно содержанию объявлений.

Информационная система — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Все пользователи информационной системы желающие выставить объявление, должны пройти регистрацию в этой ИС, но просматривать объявления может и не зарегистрированный пользователь. Также пользователь имеет право просматривать, редактировать и удалять свои объявления, поэтому в ИС имеется личный кабинет пользователя.

Таким образом, в функционирование сервиса объявлений входит:

- регистрация пользователей;

- данные об объявлениях;

- манипулирования данными объявлений;

- панель администратора;

- личный кабинет пользователя.

* 1. Анализ инструментов разработки

Для разработки приложения можно использовать следующие программные продукты: SQL, Django, PyCharm, Python, РНР, Ruby, JavaScript, JetBrains PhpStorm, JetBrains WebStorm, Notepad++, NodeJS, MySQL, Oracle, PostgreSQL, Microsoft Visual Studio, MySQL Workbench, PhpMyAdmin, Microsoft Visio, Draw IO.

SQL — «Язык структурированных запросов» декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Python — это высокоуровневый язык программирования, который используется в различных сферах IT, таких как машинное обучение, разработка приложений и другие.

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования МVС.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

РНР: Hypertext Preprocessor — «Инструменты для создания персональных веб-страниц» -серверный язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

Notepad++ — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки, а также языков описания аппаратуры VНDL и Verilog.

JetBrains PhpStorm — представляет собой интеллектуальный редактор для РНР, НТМL и JavaScript, с возможностями анализа кода на лету, предотвращения ошибок в коде, автоматизированными средствами рефакторинга для РНР и JavaScript.

JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & НТМL от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями

JavaScript — это полноценный динамический язык программирования, который применяется к НТМL документу, и может обеспечить динамическую интерактивность на веб-сайтах.

NodeJS — программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.j s добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на С++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кoдa.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке РНР и представляющее собой веб-интерфейс для администрирования CYБД MySQL.

PostgreSQL — свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIХ-подобных платформ, включая AIX, различные ВSD-системы, НР-UХ, IRIX, Linux, macOS, Tru64, QNX, а также для Microsoft Windows.

Oracle Database или Oracle RDBMS — объектно-реляционная система управления базами данных компании Oracle

MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Draw.io — это сервис, предназначенный для формирования диаграмм и схем. Сервис разделён на три части — меню, панель объектов и сам документ.

* 1. Обоснование выбора программного продукта

Для разработки ИС было решено использовать PHPStorm, PHPMyAdmin и MySQL.

Для выбора инструментов разработки были произведены сравнения, которые показаны в таблице 1, таблице 2, таблице 3, и таблице 4.

Таблица 1 — Сравнение IDE для разработки программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название IDE | Visual studio code | Notepad++ | JetBrains PhpStorm | JetBrains  WebStorm |
| Стоимость | Свободное ПО | Свободное ПО | Студенческая лицензия | Студенческая лицензия |
| Автосохранение | + | - | + | + |
| Автодополнение | + | - | + | + |
| Интеграция с системами управления версиями (git) | -  (возможно добавить плагин) | -  (возможно добавить плагин) | + | + |
| Поиск по функции в коде | - | - | + | + |
| Встроенные языки программирования | -  (возможно добавить плагин) | -  (возможно добавить плагин) | PHP, HTML, JavaScript, CSS, SQL | HTML, JavaScript, CSS, SQL |

Таким образом, из представленных IDE можно выделить несколько инструментов, способных облегчить и ускорить разработку программного обеспечения, а именно JetBrains PhpStorm, JetBrains WebStorm, так как программный продукт будет реализован с помощью web - технологий. Исходя из того, что для разработки необходим язык программирования PHP, то в качестве IDE был выбран JetBrains PhpStrorm.

Таблица 2 — Сравнение языков программирования для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название языка программирования | PHP | Ruby | Python |
| Простота обучаемости | + | - | + |
| Читабельность | + | - | + |
| Динамическая типизация | + | + | + |
| Интеграция баз данных | + | + | - |
| Объектно-ориентированные  возможности | + | + | + |

Вывод, из представленных языков программирования можно выделить РНР, так как именно этот язык более перспективный в разработке web -приложения.

Таблица 3 — Сравнение СУБД для разработки программного продукта.

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | Oracle | MySQL | PostgreSQL |
| Язык программирования | С, С++, Java, Ruby, Objective | Delphi, С, С++, Java, Perl, РНР, Pvthon, Ruby, SQL | pgSQL, РНР, Python, Ruby и др. |
| Скорость обработки | Быстрая | Быстрая | Медленная |
| Операционная система Windows | Поддерживается | Поддерживается | Поддерживается |
| Исходный код | Закрытый | Открытый | Открытый |
| Лицензия | Коммерческая | Свободная | Свободная |

В результате сравнения, из представленных СУБД можно выделить MySQL, так как оно имеет свободную лицензию и подходит для создания web – приложения.

Таблица 4 — Сравнение средств проектирования

|  |  |  |  |
| --- | --- | --- | --- |
| Название средства проектирования | Draw.io | Visio | NClass |
| Поток данных диаграмм | + | + | - |
| Er - диаграмма | + | + | - |
| Проверка связей | + | + | - |
| Удобство использования | + | + | - |
| Лицензия | свободная | коммерческая | свободная |

Следовательно, из представленных средств проектирования, было принято решение использовать Draw.io, так как оно имеет свободную лицензию и большое количество UМL средств.

1. Проектирование

Перед проектированием информационной системы было разработано техническое задание, на основе которого производилось проектирование информационной системы «Система объявлений» Техническое задание приложено к пояснительной записке.

* 1. Архитектура программного обеспечения

Архитектура программного обеспечения – совокупность важнейших решений об организации программной системы. Архитектура включает:

1. выбор структурных элементов и их интерфейсов, с помощью которых составлена система, а также их поведения в рамках сотрудничества структурных элементов;
2. соединение выбранных элементов структуры и поведения во всё более крупные системы;
3. архитектурный стиль, который направляет всю организацию – все элементы, их интерфейсы, их сотрудничество и их соединение.

Информационная система использует клиент-серверную архитектуру, что говорит о том, что все действия на клиентской части пользователя над данными обрабатываются сервером на серверной части.

Клиент-серверная архитектура позволяет значительно снизить нагрузку у пользователя, что позволяет запускать информационную систему практически на любом устройстве.

На рисунке 1 демонстрируется архитектура веб-приложений, которая соответствует разрабатываемой информационной системе.

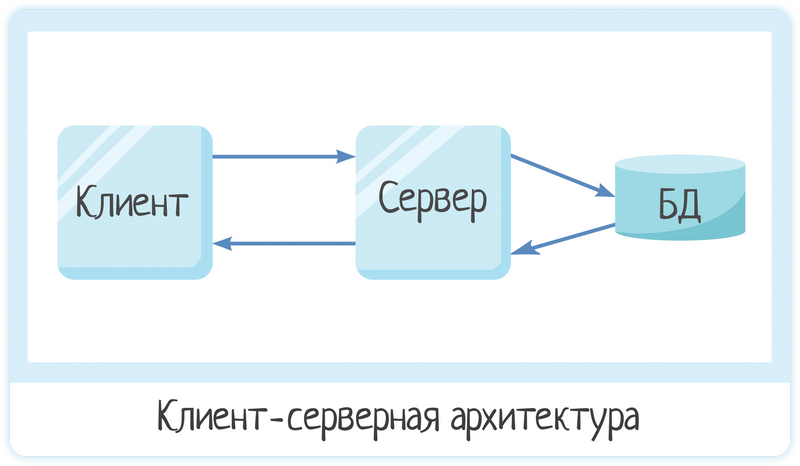


Рисунок 1 – Клиент-серверная архитектура веб-приложения

Браузер клиента отправляет запрос на подключение к странице веб-сервер. Веб-сервер отправляет запрос в веб-приложение, которое запрашивает обработку PHP. PHP может запросить подключение к MySQL, на что Apache отправляет запрос к MySQL, который по запросу берет данные из БД и отправляет обратно на веб-сервер, после чего PHP продолжает обработки. Когда обработка закончит, Apache генерирует необходимую URL и отправляет ответ клиенту, где браузер выполняет скрипты.

Информационная система использует архитектурную модель MVC (Model View Controller) представленную на рисунке 2.

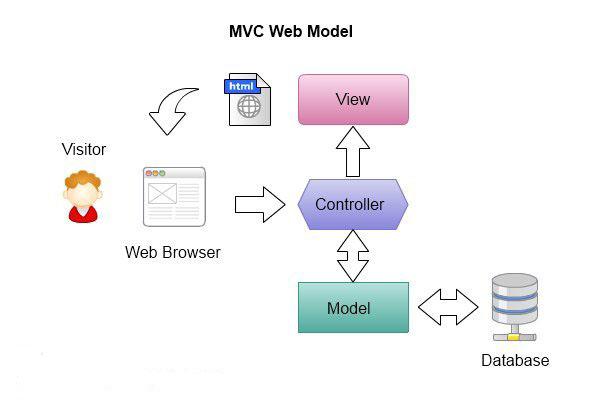


Рисунок 2 – Архитектурная модель MVC

Архитектура MVC описывает простой способ построения структуры программного продукта, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, программный продукт легче масштабируется, тестируется, сопровождается и конечно же реализуется.

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Модель — содержит бизнес-логику приложения и включает методы выборки, обработки и предоставления конкретных данных.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение.

* 1. Функциональное проектирование

Функциональное проектирование – это подход к проектированию, который нацелен, прежде всего, на создание эффективно работающего объекта. Функциональное проектирование позволяет раскрыть все функции системы.

Выполнение требуемой функции – главная цель и основа разработки объекта. Во внимание принимаются, прежде всего, функциональные показатели качества и показатели надёжности.

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На контекстной диаграмме на рисунке 3 показаны входные данные, управление механизм, выходные данные и функция.



Рисунок 3 – Контекстная диаграмма

Для демонстрации деталей в процессе необходимо спроектировать диаграмму декомпозиции.

Диаграмма декомпозиции – это разбиение функции, т.е. работа информационной системы «Сервис объявлений».

На рисунке 4 показана диаграмма декомпозиции, которая расписывает работу раздела объявления информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список объявлений», «Манипулирование объявлениями». А также на этой диаграмме присутствуют входные данные, выходные данные, управления и механизмы.

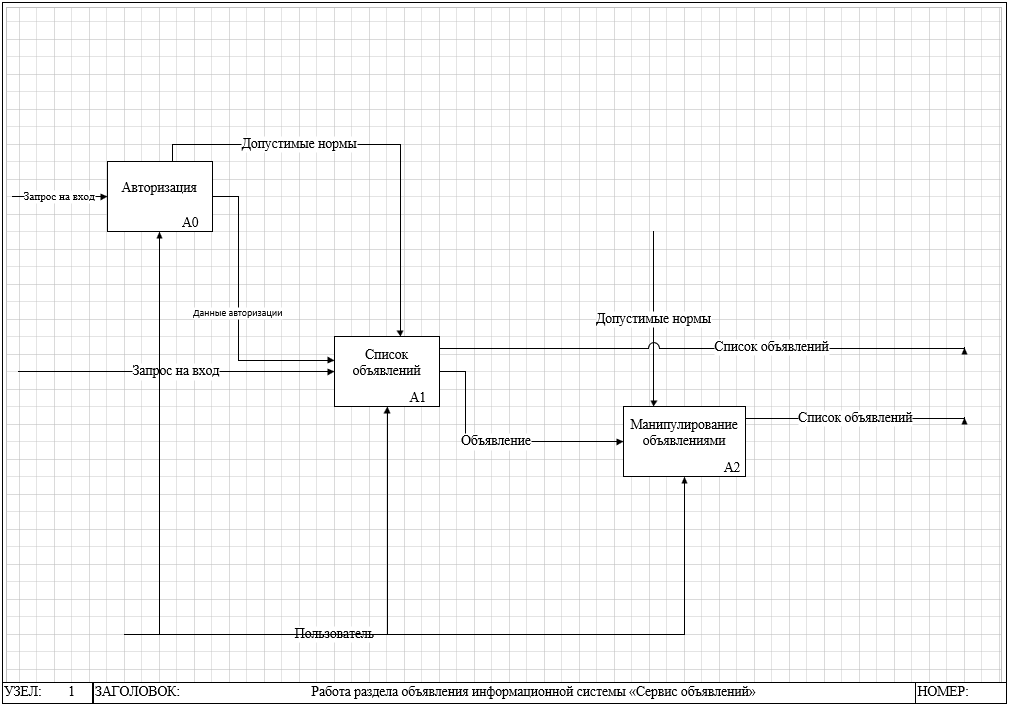


Рисунок 4 – Диаграмма декомпозиций работы раздела объявления

На рисунке 5 показана диаграмма декомпозиции, которая расписывает работу раздела города информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список городов», «Манипулирование данными модели города».



Рисунок 5 – Диаграмма декомпозиций работы раздела города

На рисунке 5 показана диаграмма декомпозиции, которая расписывает работу раздела города информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список пользователей», «Манипулирование данными пользователей».

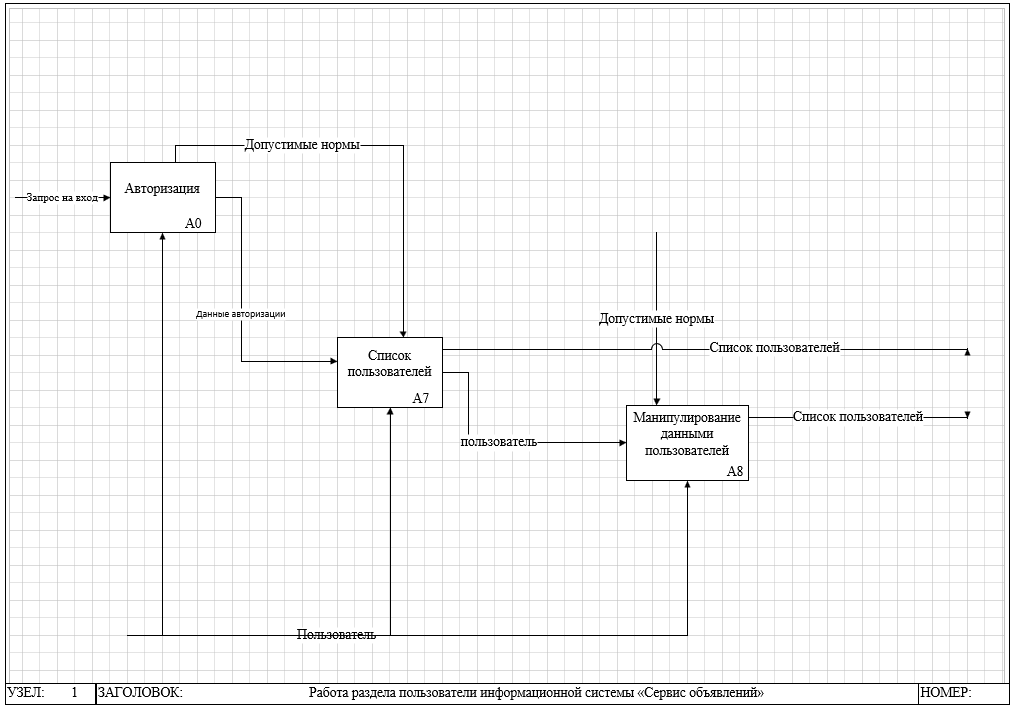


Рисунок 6 – Диаграмма декомпозиций работы раздела пользователи

На рисунке 7 показана диаграмма IDEF3, которая расписывает работу приложения из рисунка 4.

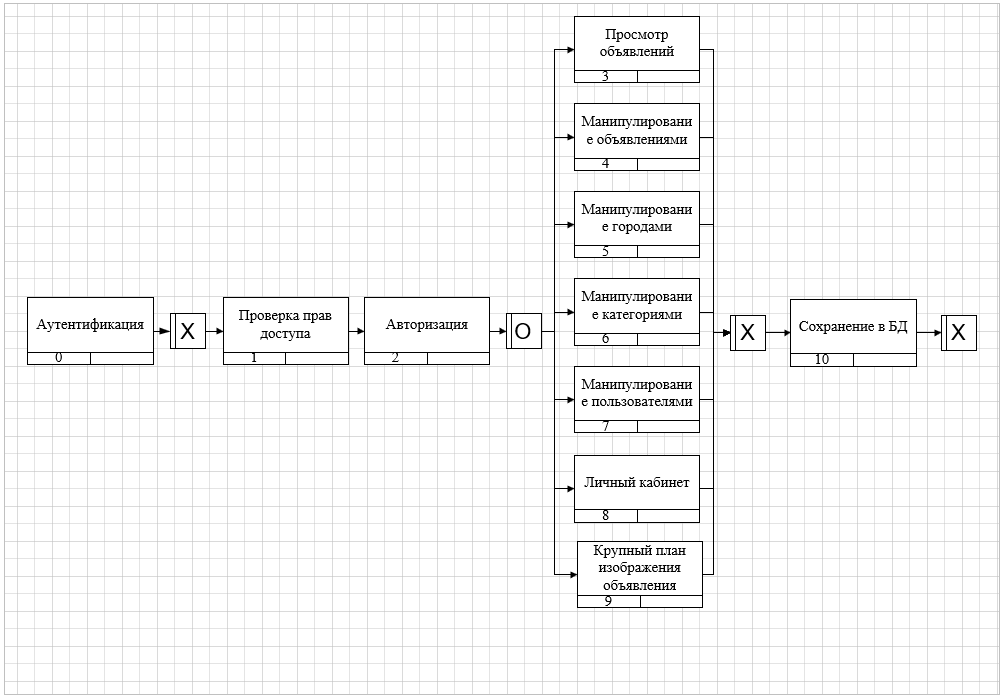


Рисунок 7 –IDEF3 диаграмма

На рисунке 8 показана диаграмма потоков данных, которая описывает передачу данных между разными функция и объектами приложения.

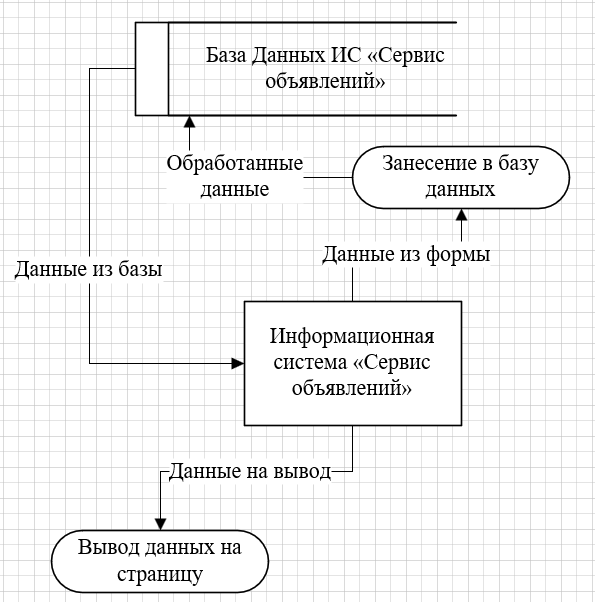


Рисунок 8 – Диаграмма потоков данных

Также во время функционального проектирования была построенна диграмма прецендентов, или как ещё называют диграмма вариантов использования, для выявления ролей пользователей и их права доступа. На рисунке 9 представлена диаграмма прецендентов информационной системы «Сервис объявлений».

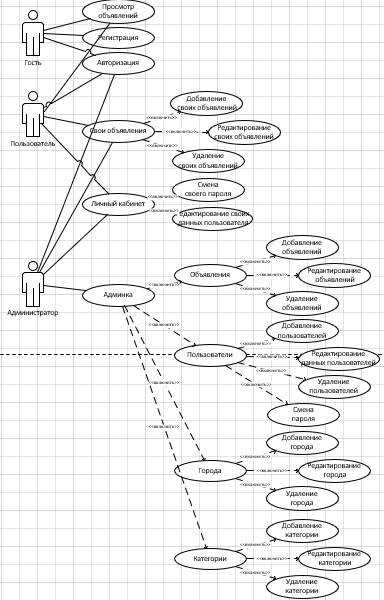


Рисунок 9 – Диаграмма прецендентов

Роль пользователя «Гость» имеет доступ только к таким действиям как: просмотр объявлений, регистрация и авторизация (рисунок 10).

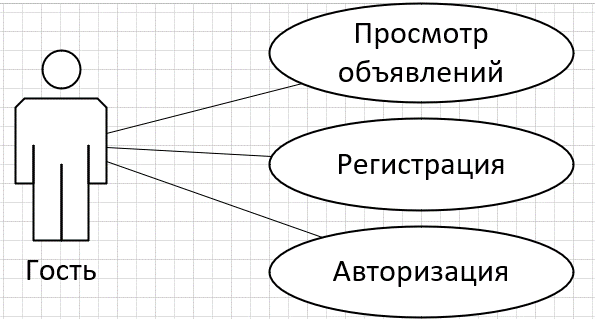


Рисунок 10 – Диаграмма прецендентов - Гость

Роль пользователя «Пользователь» имеет доступ к более широкому спектру лействий, чем «Гость», а именно к таким как: просмотр объявлений, авторизация, свои объявления, личный кабинет. Свои объявления включает в себе и другие действия, а именно: добавление своих объявлений, редактирование своих объявлений, удаление своих объявлений. Личный кабинет также включает в себя другие действия такие как: смена своего пароля и редактирование своих данных пользователя (рисунок 11).

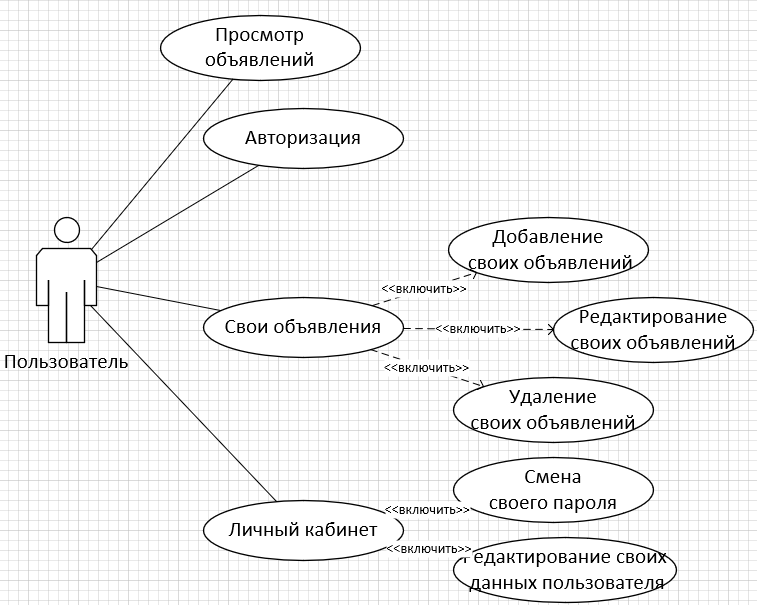


Рисунок 11 – Диаграмма прецендентов - Пользователь

Роль пользователя «Администратор» имеет доступ к тем же действия, что и «Пользователь», но добавляется группа действий «Админка». «Админка» включает в себя 4 групповых действия: объявления, пользователи, города и категории. Каждое из этих групп включает в себя действия, связанные с манипулированием данными модели группы (рисунок 12).

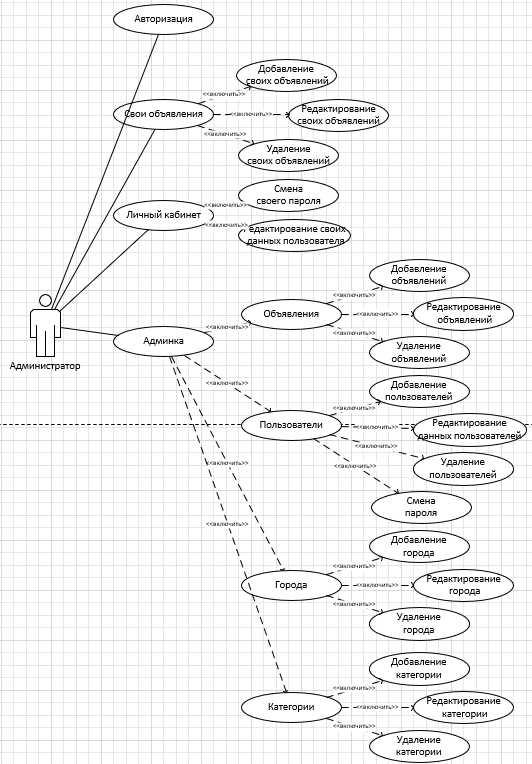


Рисунок 12 – Диаграмма прецендентов - Администратор

Подробное функциональное проектирование позволяет глубже исследовать проектную область и цель проекта, что может позволить сэкономить время на разработке информационной системы, а также позволит избежать перенаписание кода.

* 1. Проектирование базы данных

Проектирование информационной системы происходит при помощи CASE средств, которые позволяют быстро создавать схемы и реляционные модели программ.

Для лучшего изучения предметной области и проектирования базы данных. Были построены 3 схемы модели базы данных: концептуальная, логическая и физическая схемы.

Концептуальная модель — это отражение предметной области, для которой разрабатывается база данных. На рисунке 13 представлена концептуальная модель базы данных ИС. Для построения модели были выявлены 5 сущностей: пользователи, посты (объявления), картинки постов (объявлений), города и категории.

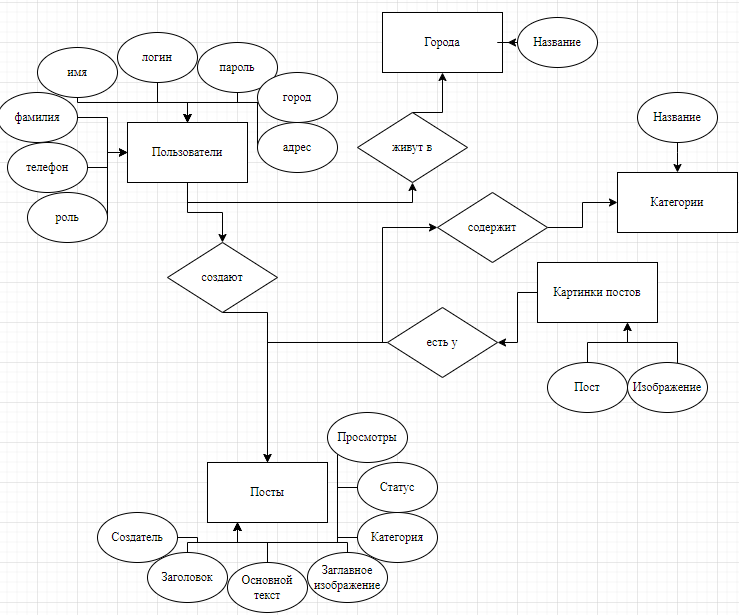


Рисунок 13 – концептуальная модель базы данных

Логическая (даталогическая) модель представляет собой модель базы данных, которая не привязана к конкретной СУБД. В ней выделяют основные объекты БД и определяют связи между этими объектами. Полученная логическая модель представлена на рисунке 14.

Модель сущность-связь или схема баз данных – модель данных, позволяющая описывать концептуальные схемы предметной области.

На рисунке 15 изображена физическая схема базы данных приложения «Интернет-магазин книг».

Рисунок 9 – Схема базы данных

База данных представляет собой 17 связанных таблиц. Для создания связей между таблицами используются внешние ключи.

Были выявлены следующие сущности и их атрибуты представленные в таблицах от 5 до 21.

Таблица 5 — Атрибуты сущности authors

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор автора |
| first\_name | varchar (100) | Имя автора |
| last\_name | varchar (100) | Фамилия автора |
| father\_name | varchar (100) | Отчество автора |
| initials | varchar (4) | Инициалы автора |

Таблица 6 — Атрибуты сущности genres

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор жанра |
| name | varchar (50) | Название жанра |

Таблица 7 — Атрибуты сущности limits

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор возрастного ограничения |
| name | varchar (3) | Название возрастного ограничения |

Таблица 8 — Атрибуты сущности publishers

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор издательства |
| name | varchar (100) | Название издательства |

Таблица 9 — Атрибуты сущности products

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор книги |
| ISBN | varchar (17) | Международный стандартный номер книги |
| name | varchar (255) | Название книги |
| pages | int | Количество страниц книги |
| price | int | Стоимость книги |
| image | varchar (255) | Путь до главного изображения книги |
| year\_release | year (4) | Год издания книги |
| description | text | Описание книги |
| limit\_id | bigint | Идентификатор возрастного ограничения |
| publisher\_id | bigint | Идентификатор издательства |
| author\_id | bigint | Идентификатор автора |

Таблица 10 — Атрибуты сущности product\_genres

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор жанра книги |
| product\_id | bigint | Идентификатор книги |
| genre\_id | bigint | Идентификатор жанра |

Таблица 11 — Атрибуты сущности product\_images

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор изображения книги |
| product\_id | bigint | Идентификатор книги |
| image | varchar (255) | Путь до изображения книги |

Таблица 12 — Атрибуты сущности users

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор пользователя |
| first\_name | varchar (255) | Имя пользователя |
| last\_name | varchar (255) | Фамилия пользователя |
| father\_name | varchar (255) | Отчество пользователя |
| phone | varchar (255) | Телефон пользователя |
| name | varchar (255) | Логин пользователя |
| email | varchar (255) | Почта пользователя |
| email\_verified\_at | timestamp | Дата и время верификации почты |
| password | varchar (255) | Пароль пользователя |
| remember\_token | varchar (100) | Токен для запоминания пользователя |

Таблица 13 — Атрибуты сущности product\_logs

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор выбора пользователем книги |
| product\_id | bigint | Идентификатор книги |
| user\_id | bigint | Идентификатор пользователя |

Таблица 14 — Атрибуты сущности statuses

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор статуса заказа |
| name | varchar (25) | Текст статуса заказа |

Таблица 15 — Атрибуты сущности orders

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор заказа |
| status\_id | bigint | Идентификатор статуса заказа |
| user\_id | bigint | Идентификатор пользователя |

Таблица 16 — Атрибуты сущности order\_has\_products

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор книги заказа |
| order\_id | bigint | Идентификатор заказа |
| product\_id | bigint | Идентификатор книги |
| count | int | Количество книг |

Таблица 17 — Атрибуты сущности user\_logs

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор действия пользователя |
| user\_id | bigint | Идентификатор пользователя |
| actions | varchar (255) | Текст действия пользователя |

Таблица 18 — Атрибуты сущности roles

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор роли |
| name | varchar (255) | Название роли |
| guard\_name | varchar (255) | Тип роли |

Таблица 19 — Атрибуты сущности permissions

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор права доступа |
| name | varchar (255) | Название права доступа |
| guard\_name | varchar (255) | Тип права доступа |

Таблица 20 — Атрибуты сущности role\_has\_permissions

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| permission\_id | bigint | Идентификатор права доступа |
| role\_id | bigint | Идентификатор роли |

Таблица 21 — Атрибуты сущности carts

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор корзины |
| user\_id | bigint | Идентификатор пользователя |
| product\_id | bigint | Идентификатор книги |

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Третья нормальная форма, которую придерживается база данных, содержит в себе следующее требование:

– Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Исходя из всего выше сказанного, уровень нормализации базы данных равен 3.

* 1. Проектирование пользовательского интерфейса

Интерфейс приложения – это одна из важнейших частей проектирования программного обеспечения.

Интерфейсом приложения «Интернет-магазин книг» является меню навигации, содержащее разделы приложения, рабочей области.

На рисунке 9 показан прототип раздела «Вход», который авторизоваться пользователю приложения.

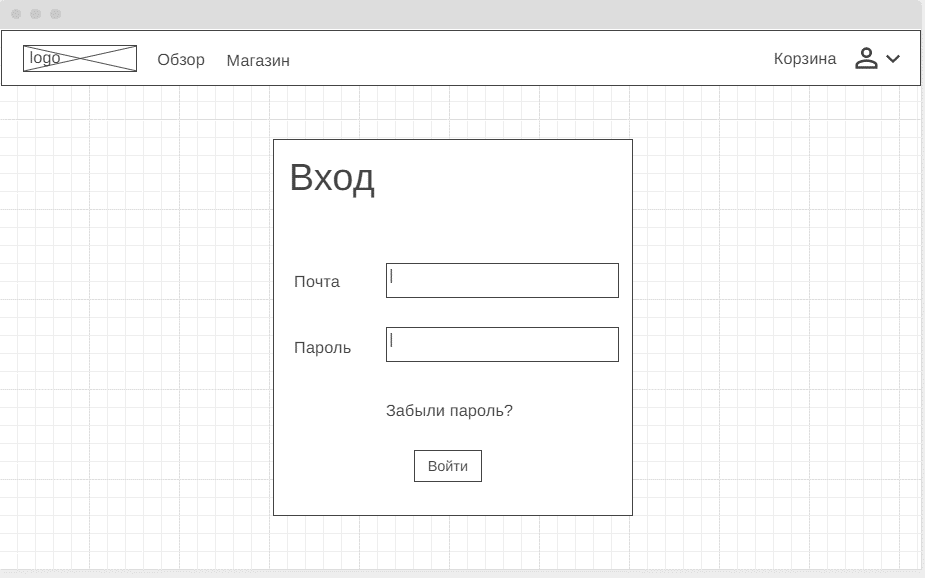


Рисунок 9 – Прототип раздела «Вход»

На рисунке 10 демонстрируется прототип основной страницы «Обзор», на которой пользователь может увидеть рекомендуемые приложением товары. На эту страницу может войти и не авторизованный пользователь приложения.

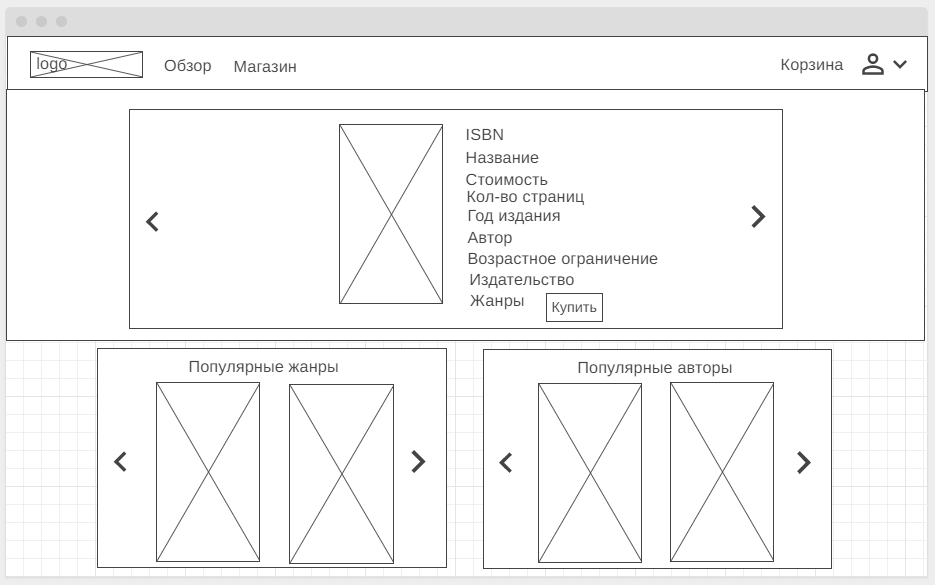


Рисунок 10 – Прототип страницы «Обзор»

На рисунке 11 демонстрируется прототип страницы «Магазин». На рабочем пространстве размещен список книг и список фильтров, которые можно применить к списку книг.

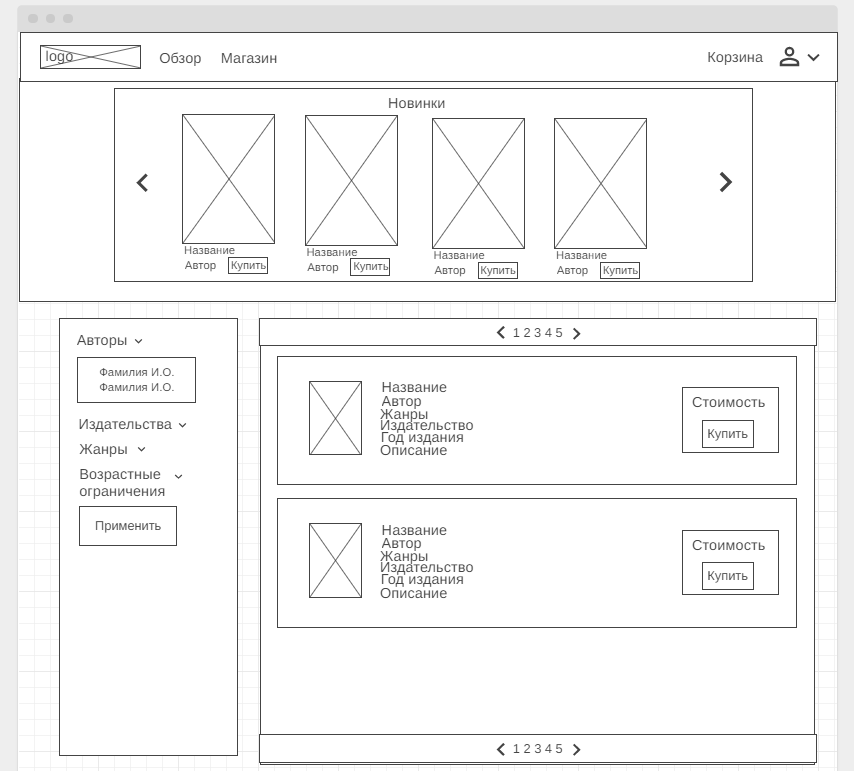


Рисунок 11 – Прототип страницы «Магазин»

На рисунке 12 демонстрируется прототип страницы «Корзина». На рабочем пространстве размещен список книг, выбранных пользователем для оформления заказа.

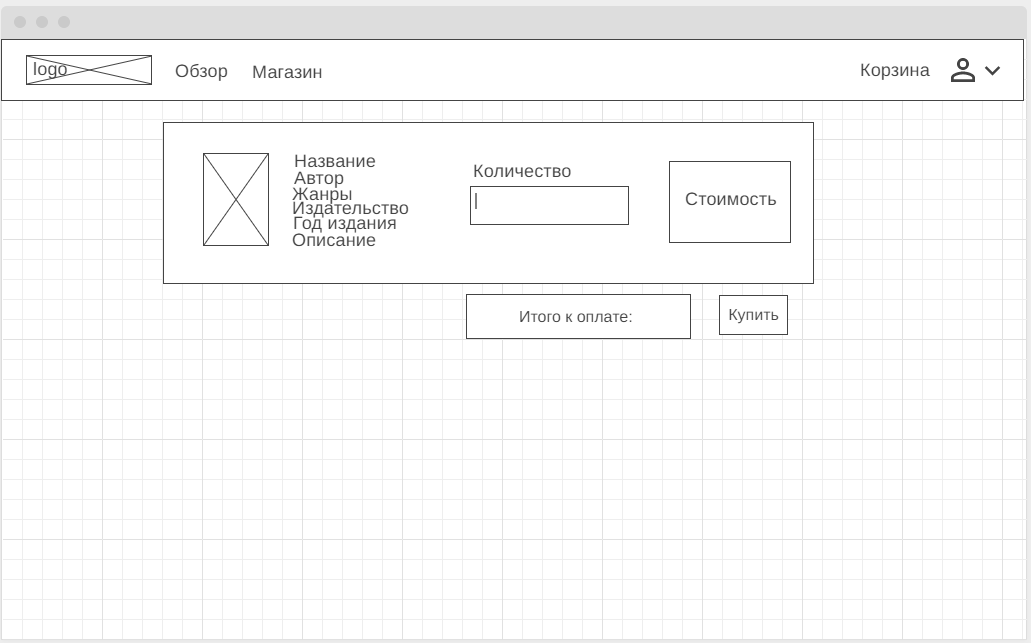


Рисунок 12 – Прототип страницы «Корзина»

На рисунке 13 демонстрируется прототип страницы «Регистрация», которая позволяет зарегистрировать нового пользователя приложения.

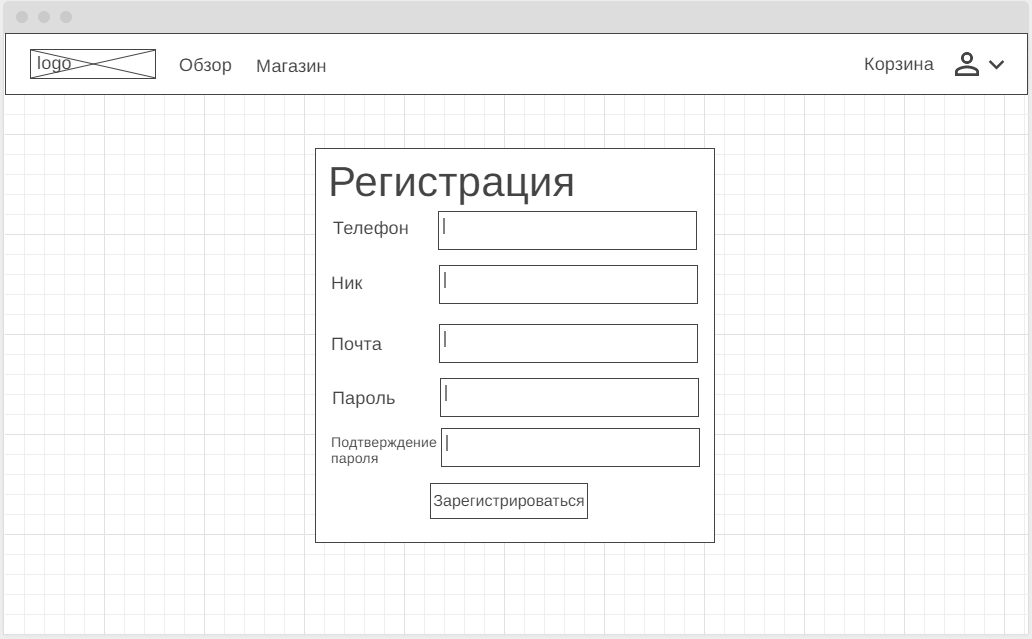


Рисунок 13 – Прототип страницы «Регистрация»

На рисунке 14 демонстрируется прототип страницы «Продукты». Данная страница доступна только пользователям с соответствующими правами доступа. На рабочем пространстве размещены форма для добавления нового товара и список форм существующих книг для редактирования или удаления данных.

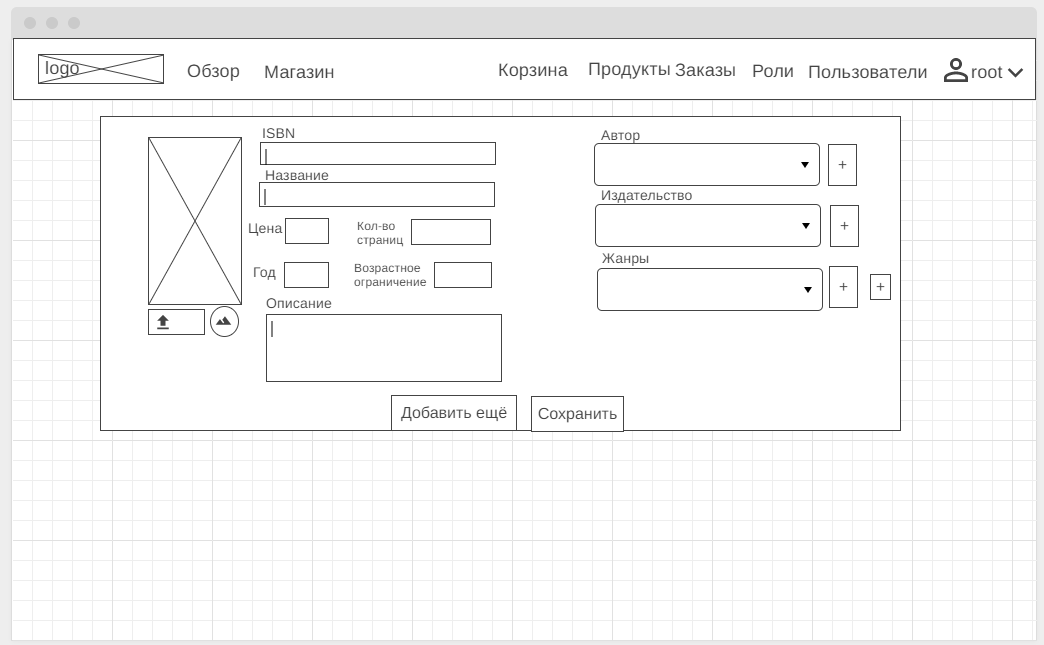


Рисунок 14 – Прототип страницы «Продукты»

На рисунке 15 демонстрируется прототип страницы «Заказы». Данная страница доступна только пользователям с соответствующими правами доступа. На рабочем пространстве размещены список заказов, которые по статусу не выданы клиенту. Страница позволяет изменить статус заказа.

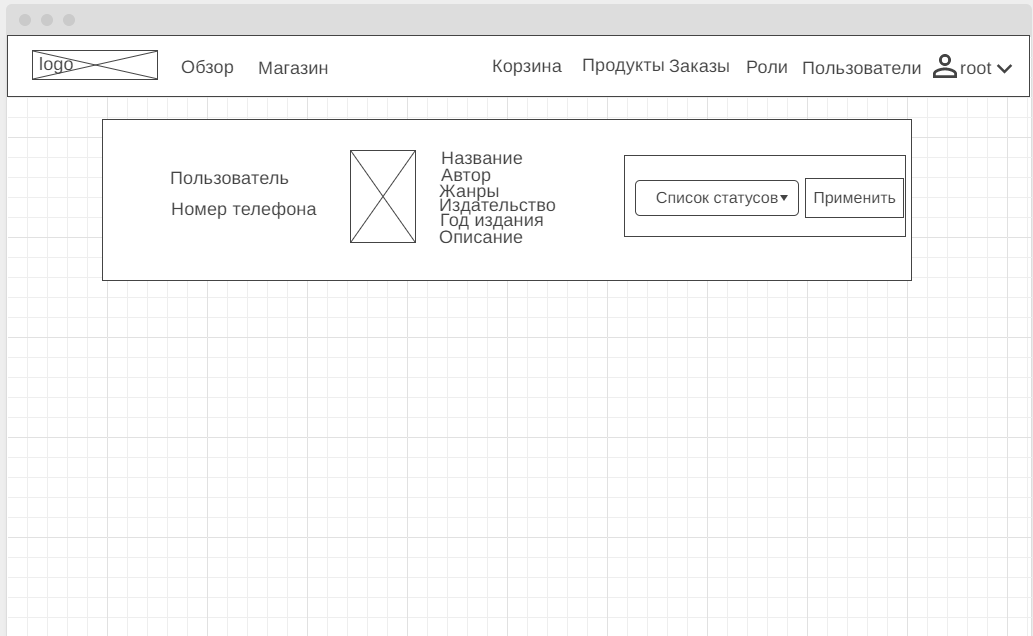


Рисунок 15 – Прототип страницы «Заказы»

Проектирование интерфейса пользователя позволяет определиться с расположение объектов и данных на рабочем пространстве. Также позволяет определиться с количеством страниц и избежать конфликтов с заказчиков.

1. Реализация программного обеспечения
   1. Кодирование программного обеспечения

Суть реализации программного обеспечения заключается в кодировании – процесс написания программного кода. В связи с тем, что разрабатываемое приложение основано на веб-технологиях, в ходе разработки было принято решение использовать:

* фреймворк tailwind css - для клиентской части;
* скриптовый язык JavaScript – для клиентской части;
* фреймворк Laravel - для серверной составляющей модуля.

Благодаря фреймворку tailwind css приложение «Интернет-магазин книг» имеет простой, но стильный внешний вид, который продемонстрирован на рисунке 16.

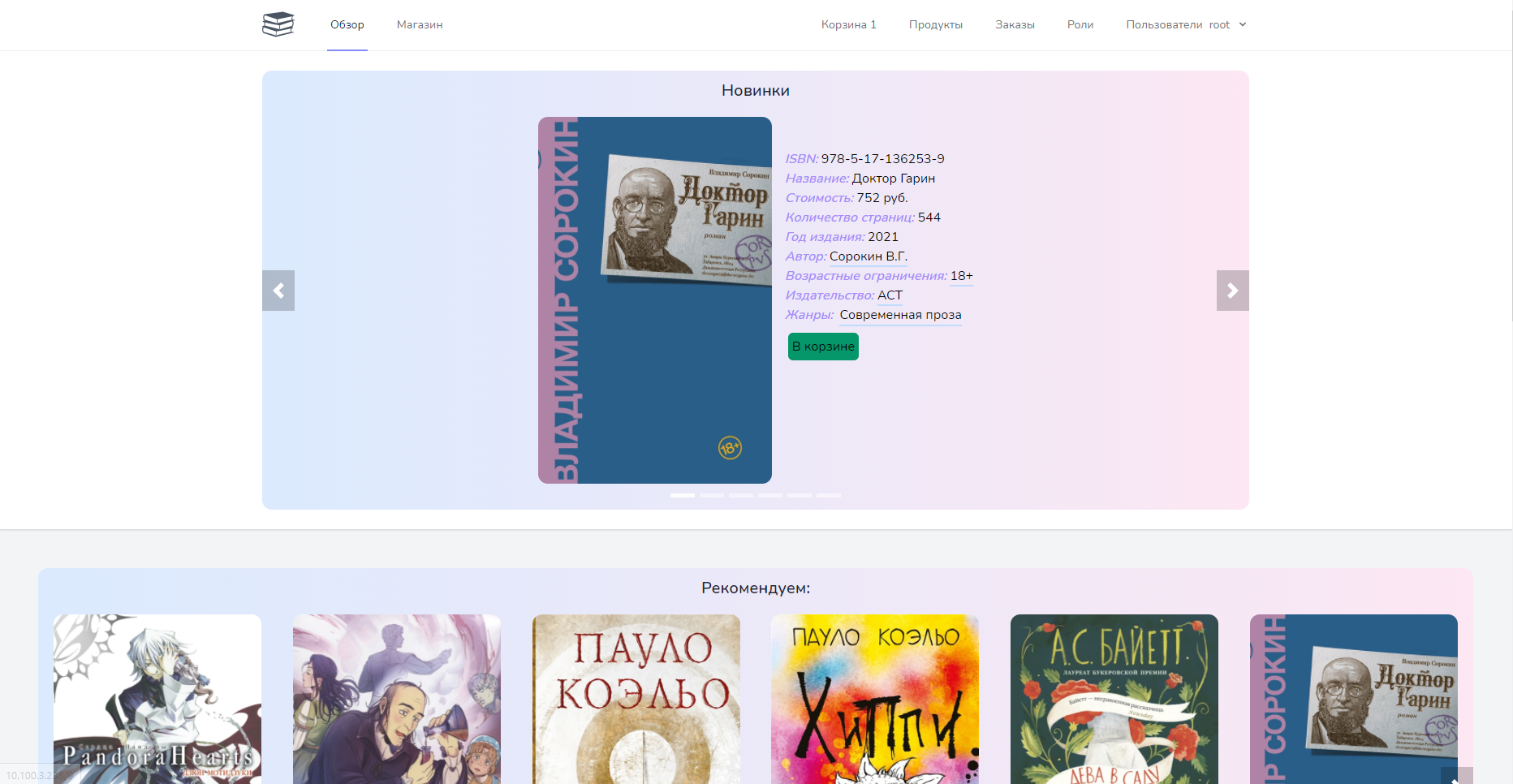


Рисунок 16 – Вкладка «Обзор»

Скриптовый язык JavaScript позволил организовать динамическое получение данных, что позволило реализовать поиск в разделах приложения.

Фреймворк Laravel, в данной разработке, позволяет осуществить защиту передачи данных и облегчить работу с БД.

Контроллеры отвечают за обработку запросов пользователя и вызов соответствующих представлений. Контроллер представляет собой класс, который содержит методы обработки функций. Все контроллеры проекта находятся в папке /app/http/Controllers. Самый крупный контроллер это ProductController. Он отвечает за обработку данных раздела «Продукты». На рисунке 17 представлен код функции create из контроллера ProductController.

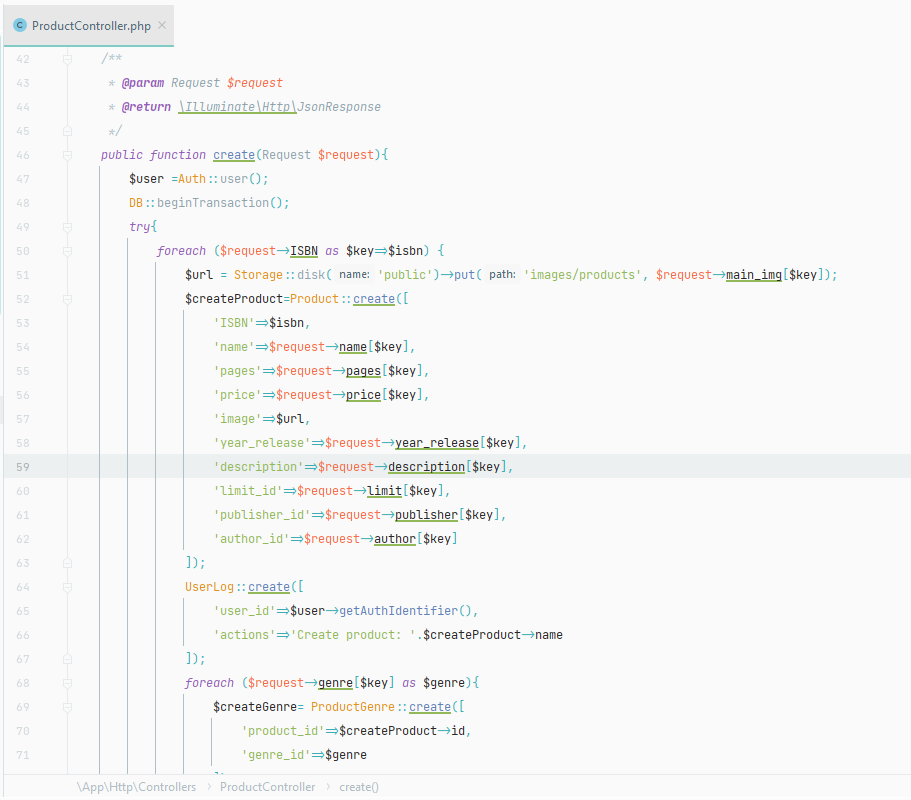


Рисунок 17 – Фрагмент контроллера ProductController

Представления отвечают за показ результатов запроса пользователю. Каждое представление – это отдельная страница, содержащая html и javascript код, а также php вставки. Все представления находятся в папке /resource/views. Представление страницы «Магазин» является одной из крупнейших представлений приложения. В ней данные из контроллера представляются в виде списка. На рисунке 18 изображен фрагмент кода этого представления.



Рисунок 18 – Фрагмент представления страницы «Магазин»

В работе приложения не малую роль играет система маршрутов, которые связывают файл представления с нужным ему контроллером. Все маршруты проекта располагаются в папке routes. Часть маршрутов проекта можно просмотреть в приложении А Листинг маршрутов.

Приложение содержит несколько крупных функций. Одна из самых крупных функций — это функция применения фильтров на странице «Магазин» (фрагмент кода на рисунке 19).



Рисунок 19 – Фрагмент функции фильтров

Эта функция применяет выбранные фильтры пользователям к странице. Функция написана на php.

1. Документирование программного обеспечения
2. Руководство пользователя программного обеспечения
   * 1. Описание системы

Приложение содержит разделы с разным уровнем доступа. При первоначальном запуске существует стандартная учетная запись, которую в последующем рекомендуется удалить. Стандартная учётная запись позволит назначить права пользователям. Работа в разделе зависит от уровня доступа учетной записи авторизированного пользователя. Подробнее смотрите в «Страница входа».

* + 1. Страницы входа и учетные записи

Страница входа позволяет войти в одну из учетных записей приложения. Страница представляет собой форму, которая демонстрируется на рисунке 20.

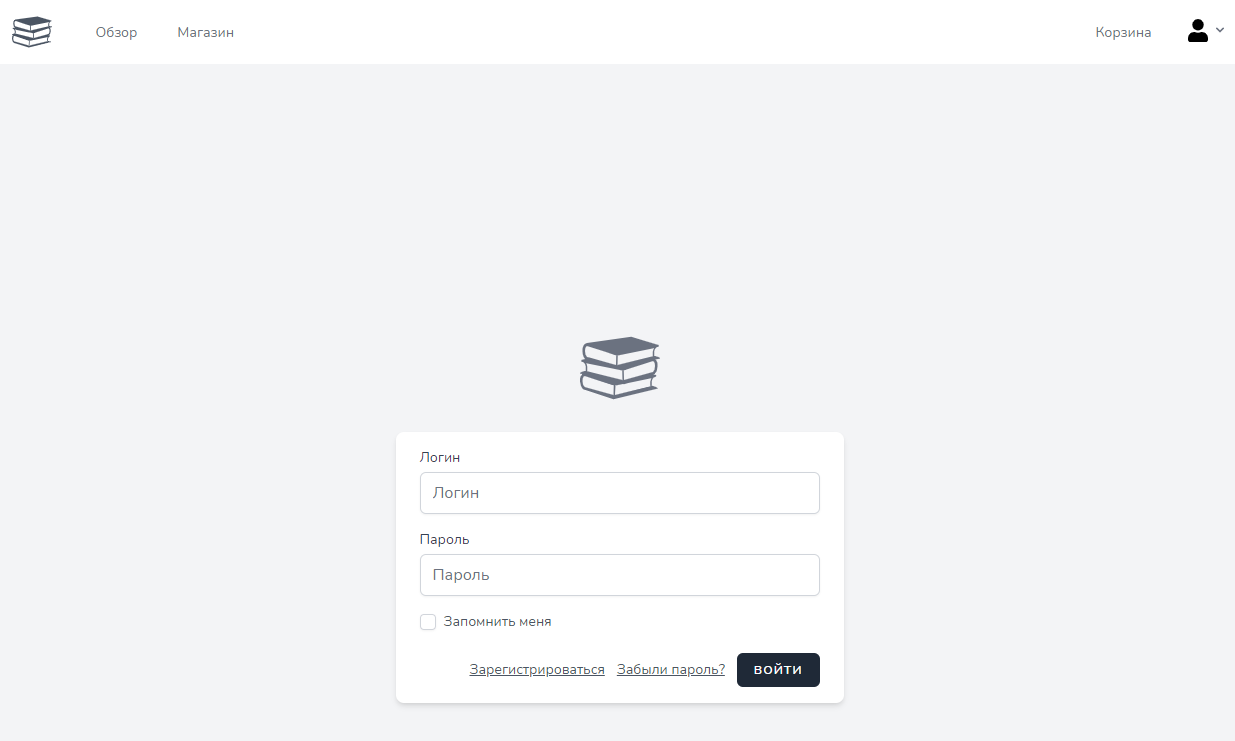


Рисунок 20 – Страница входа

В приложении существует гибкая система прав доступа к функциям разделов системы.

При первой установки приложения существует только одна учетная запись со всеми правами доступа. Ее данные смотрите в таблице 22. После входа рекомендуется создать новую учетную запись с таким уровнем доступа, войти в нее и удалить стандартную учетную запись, так как её данные доступны любому прочитавшему это руководство пользователя.

Таблица 22 – Стандартная учётная запись

|  |  |
| --- | --- |
| Поле | Данные |
| Имя пользователя | root |
| Пароль | root |

* + 1. Раздел «Обзор»

Раздел «Обзор» позволяет пользователю просмотреть и добавить в корзину представленные рекомендованные товары приложения. Раздел содержит несколько «каруселей» (списков товаров, которые можно листать): карусель «Новинки», в которую выводятся последние 10 добавленных товаров в базу, карусель «Популярные жанры», в которую выводятся по одному товару из жанра, карусель «Популярные авторы», в которую выводятся по одному товару от автора и карусель «Книги для подростков», в которую выводятся товары возрастного ограничения «16+». Фрагмент раздела смотрите на рисунке 21.

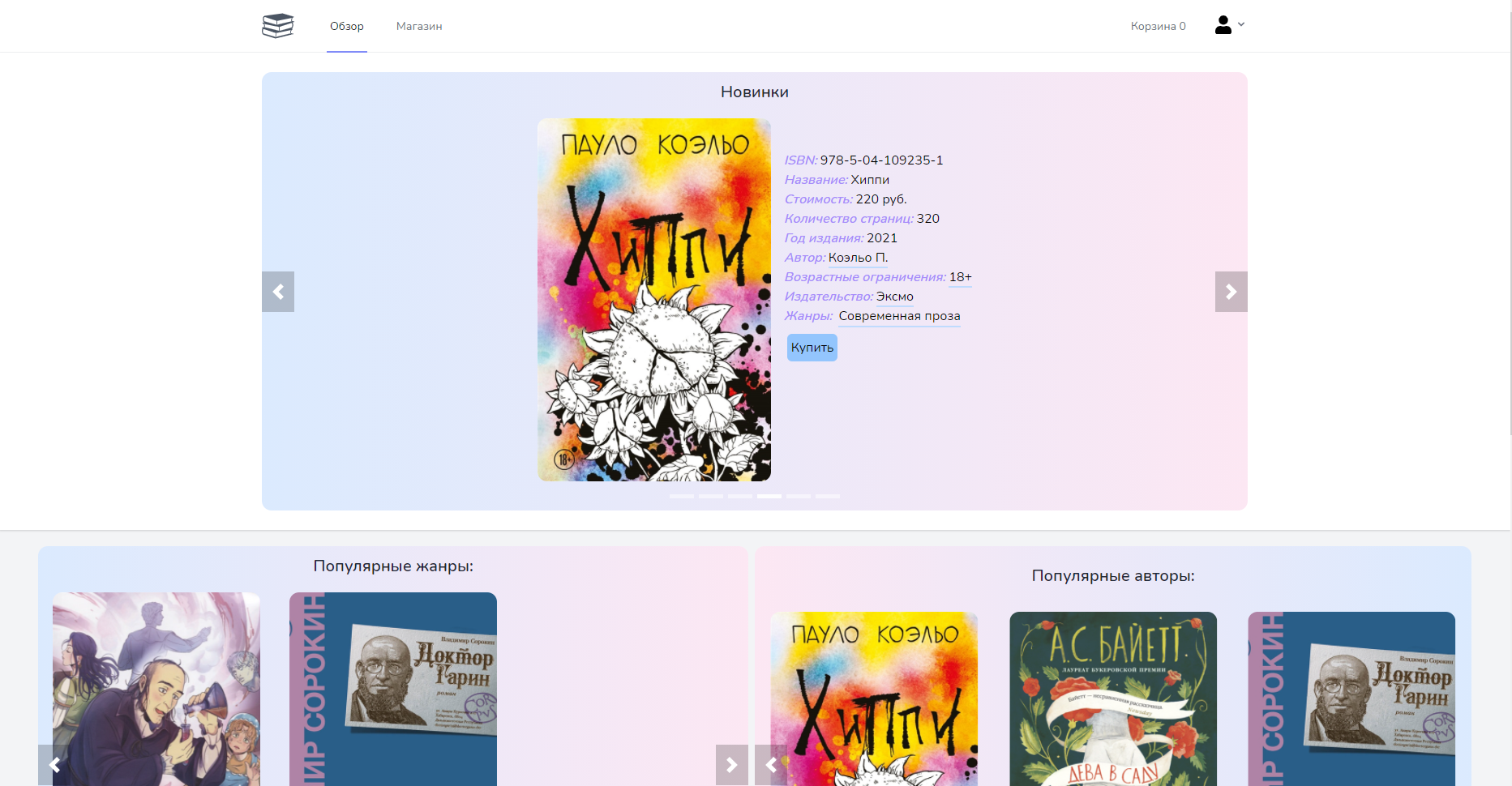


Рисунок 21 – Фрагмент раздела «Обзор»

Текст выделенный нижние голубой чертой, позволяет открыть страницы, в которой содержатся товары только этой категории.

* + 1. Раздел «Магазин»

Раздел «Магазин» позволяет просмотреть товары с описанием, а также добавить их в корзину и при помощи окна с фильтрами изменить список выводимых товаров. Внешний вид раздела представлен на рисунке 22.

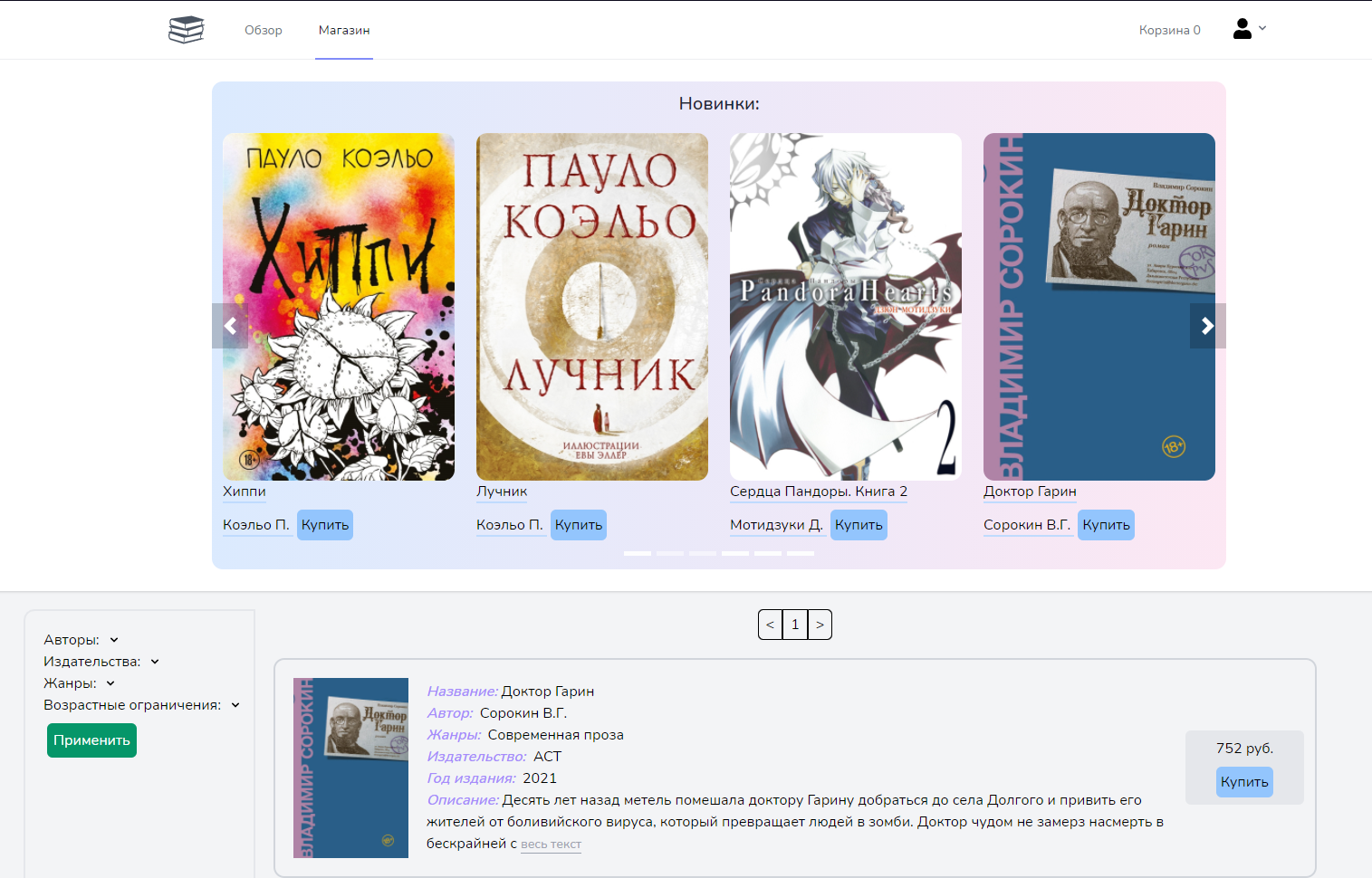


Рисунок 22 – Раздел «Магазин»

Книги выводятся на страницу в ограниченном количестве и разбиты на страницы. Это позволяет ускорить загрузку страницы.

Кнопка «Применить» позволяет применить к странице выбранные пользователем фильтры. Фильтровать данные можно только по данным, полностью относящимся к книге, таким как: авторы, издательство, жанры и возрастное ограничение. Чтобы открыть выбор фильтра по выбранной категории, необходимо нажать на стрелку, направленную вниз, находящаяся справа от названия категории.

* + 1. Раздел «Корзина»

Раздел «Корзина» позволяет просмотреть все добавленные пользователем товары в корзину, а также оформить заказ и удалить товар из корзины. Внешний вид продемонстрирован на рисунке 23.

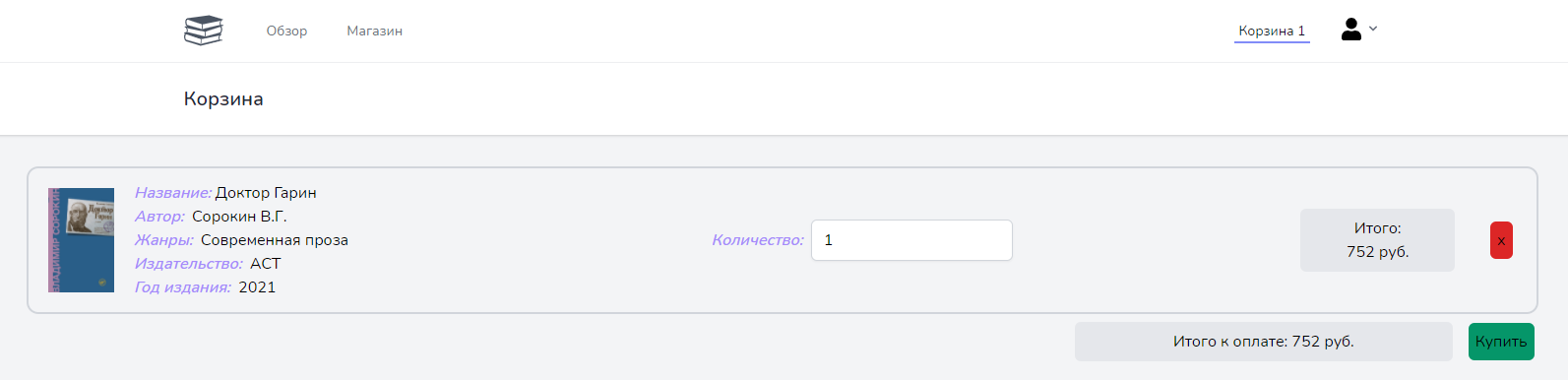


Рисунок 23 –Раздел «Корзина»

Нажав на кнопку «Купить», если пользователь был авторизован, то произойдёт оформление заказа на пользователя. Если пользователь авторизован не был, то его перебросит на страницу входа в приложение, также он может пройти регистрацию и оформить заказ повторно.

* + 1. Раздел «Личный кабинет»

Раздел «Личный кабинет» позволяет авторизированным пользователям увидеть свои личные данные, историю заказов с их статусами (рисунок 26), также этот раздел позволяет изменить личные данные пользователя (рисунок 24) и данные авторизации (рисунок 25).

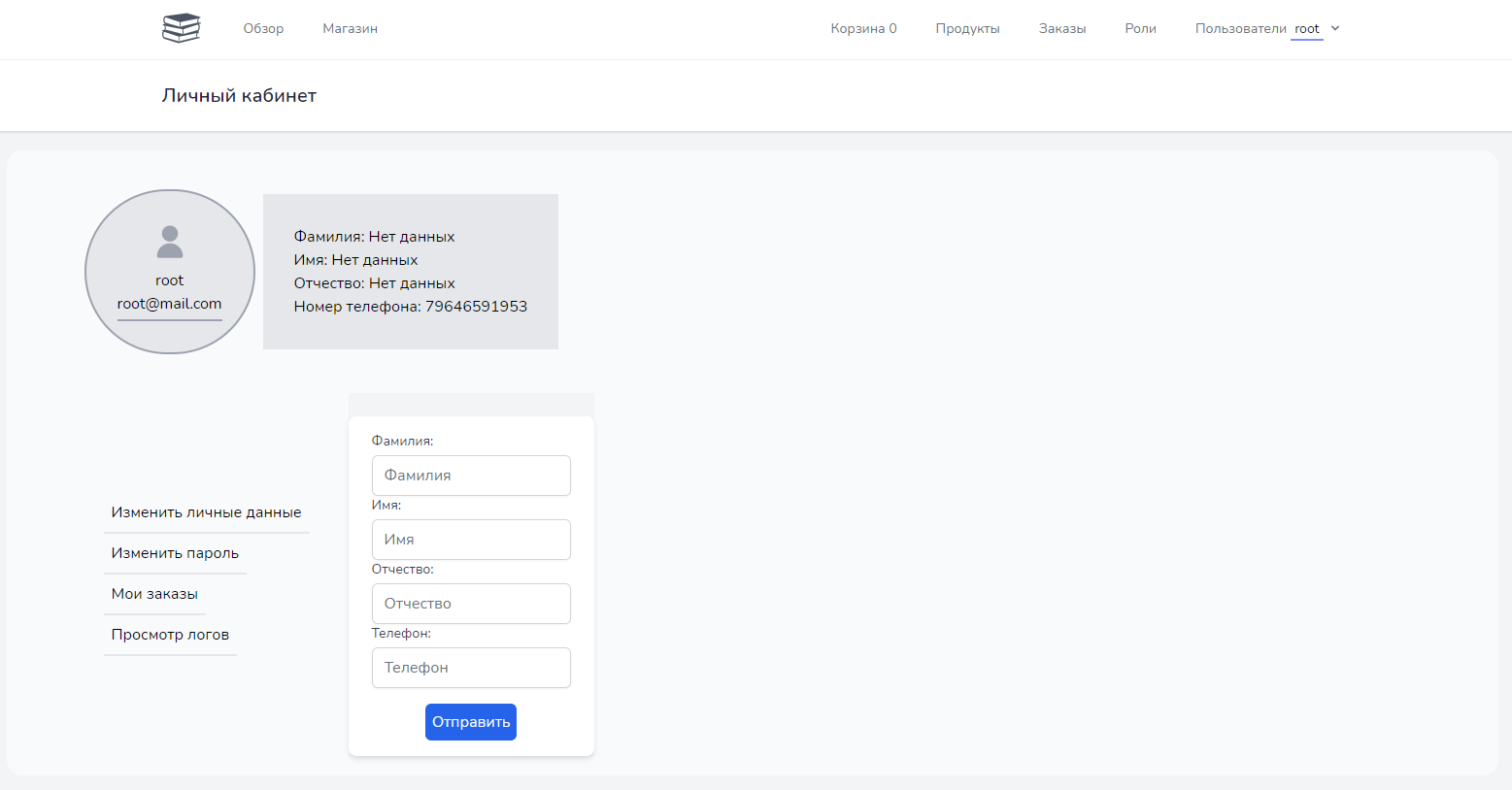


Рисунок 24 –Раздел «Личный кабинет»: Редактирование личных данных

Форма редактирование личных данных позволяет изменить фамилию, имя, отчество и номер телефона пользователя.

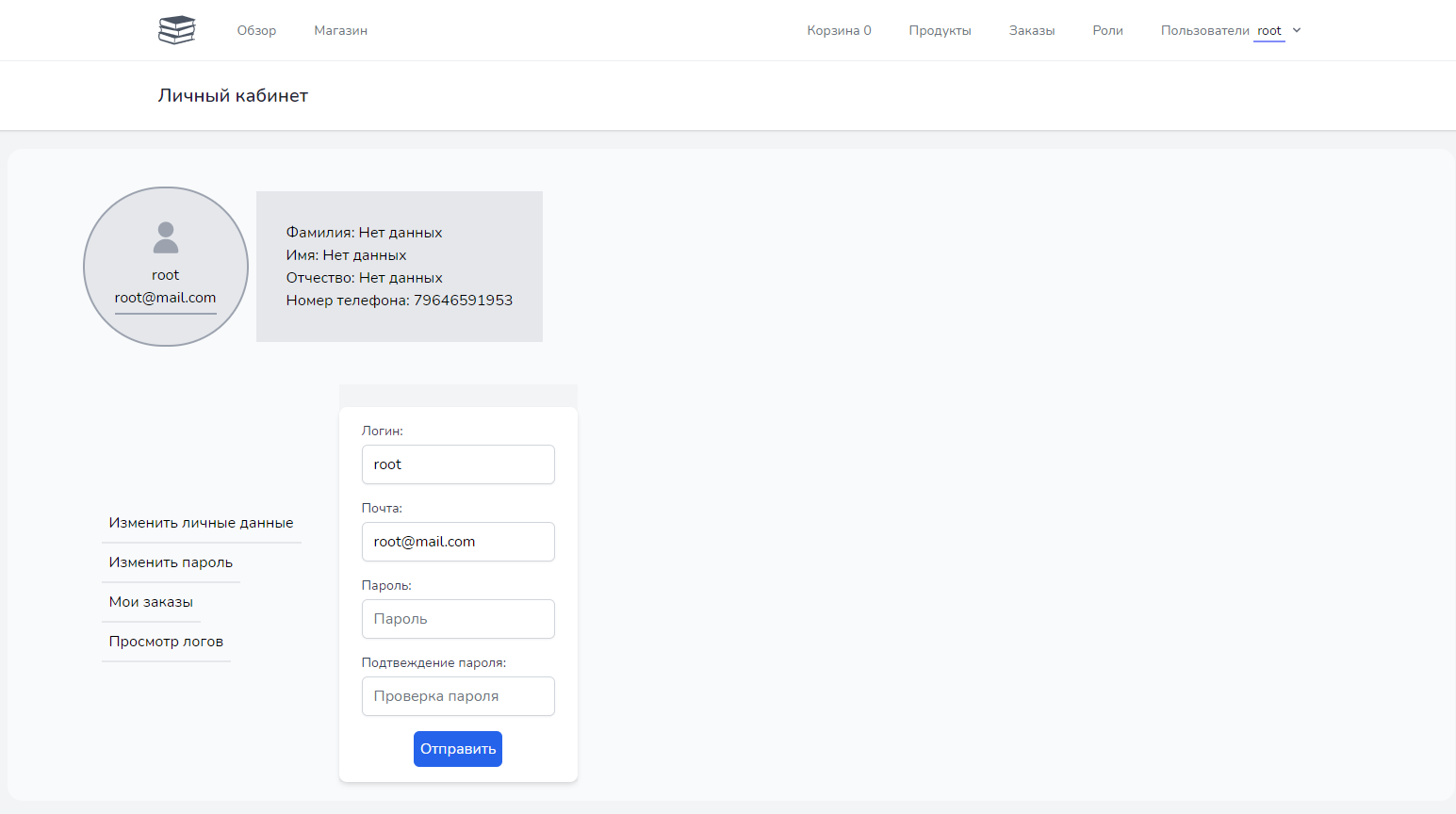


Рисунок 25 –Раздел «Личный кабинет»: Редактирование данных авторизации

Форма редактирования данных авторизации позволяет изменить логин, почту, пароль пользователя.

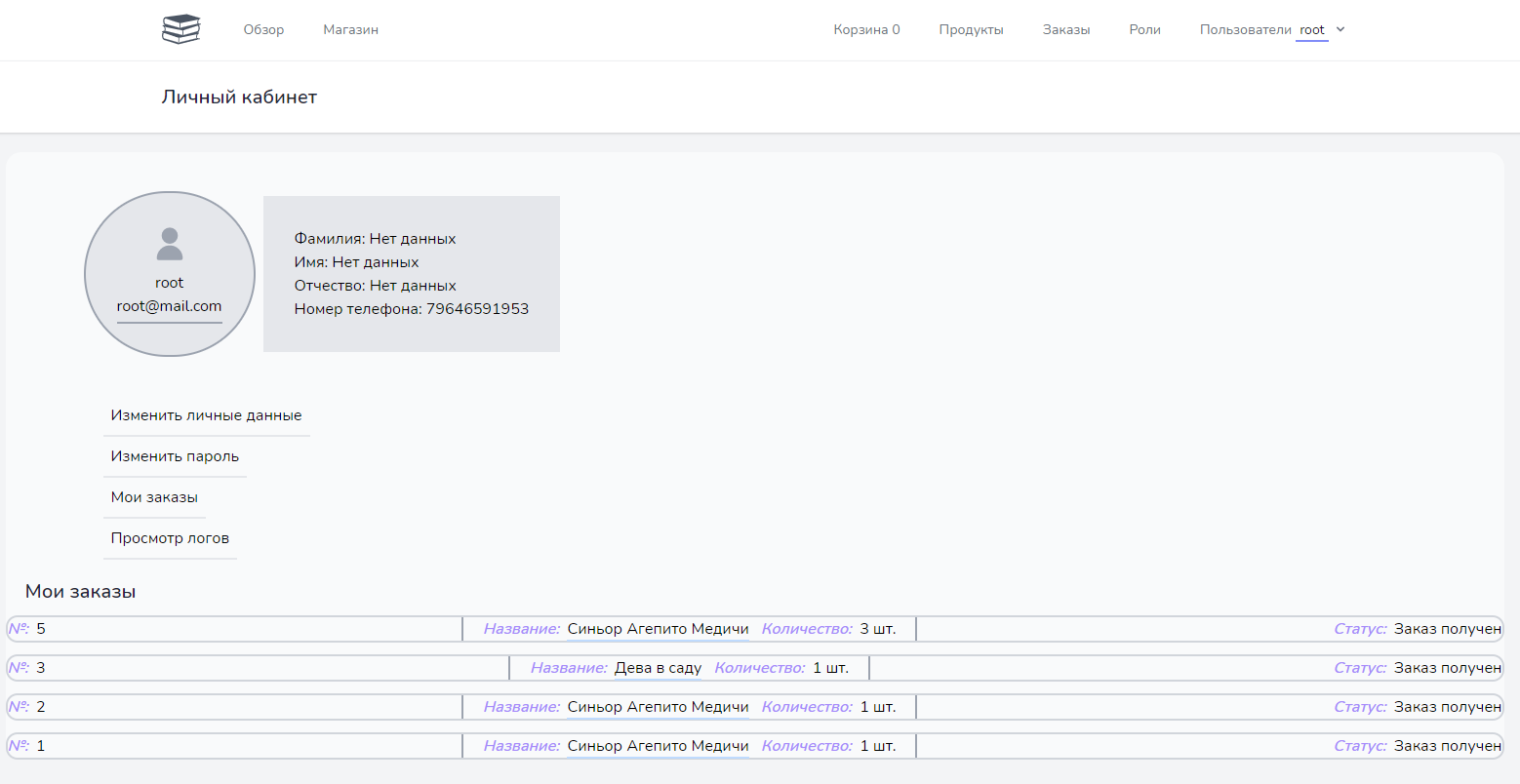


Рисунок 26 –Раздел «Личный кабинет»: Просмотр истории заказов

Просмотр истории заказов позволяет узнать пользователю статус его текущих заказов, а также узнать какие заказы у него были ранее.

* + 1. Раздел «Продукты»

Раздел «Продукты» позволяет добавить новый товар в базу, а также удалить или отредактировать данные уже имеющихся товаров в базе. В раздел «Продукты» могут зайти только пользователи с правами доступа «product-create», «product-edit» и «product-delete». От права доступа зависит какие действия может производить пользователь на странице. Внешний вид демонстрируется на рисунке 27.

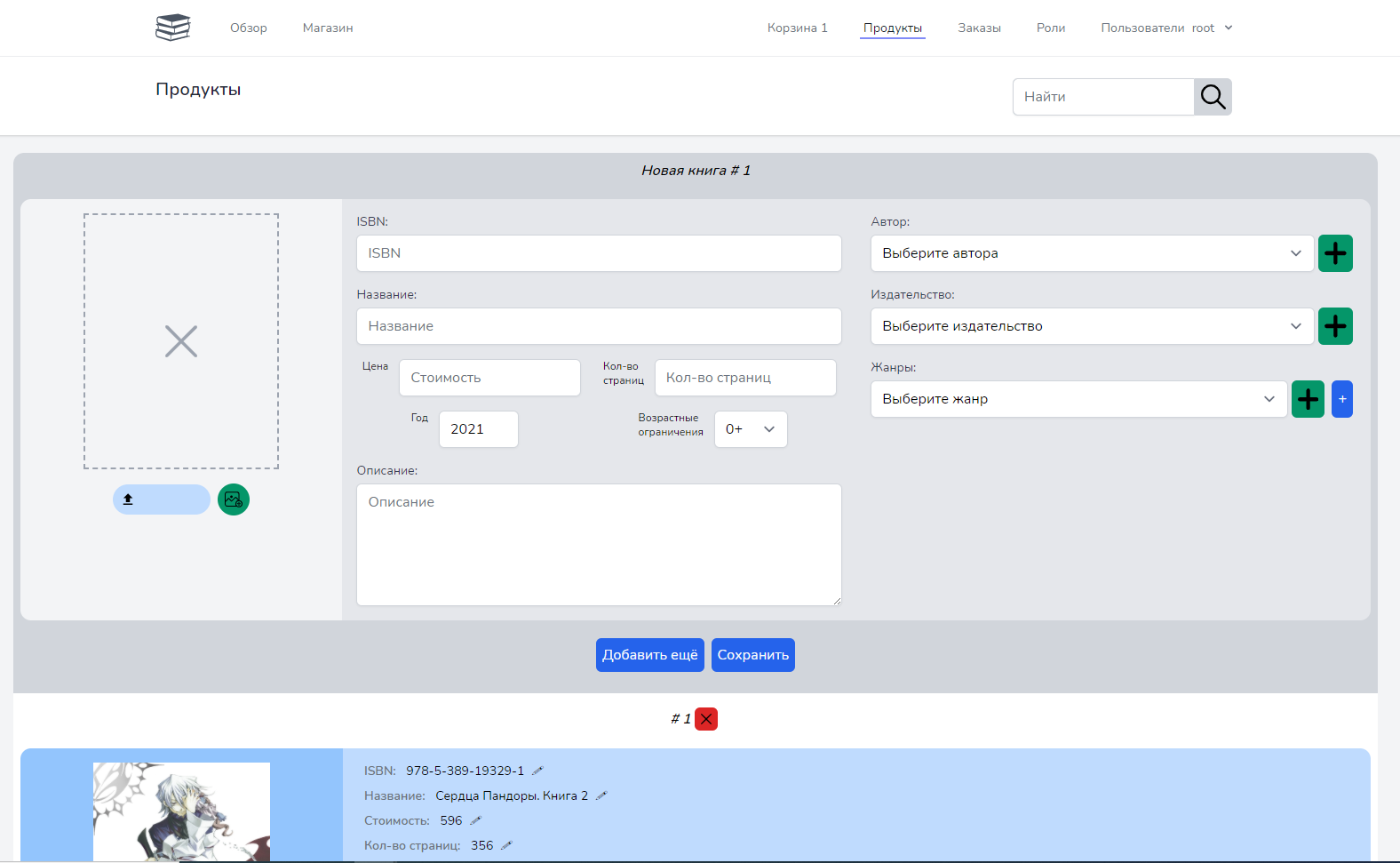


Рисунок 27 – Раздел «Продукты»

Для добавления нового товара в базу необходимо заполнить соответствующую форму с заголовком «Новая книга». Раздел позволяет продублировать форму и добавить несколько новых товаров. Также к товару можно добавить несколько жанров и дополнительных изображений (рисунок 28).

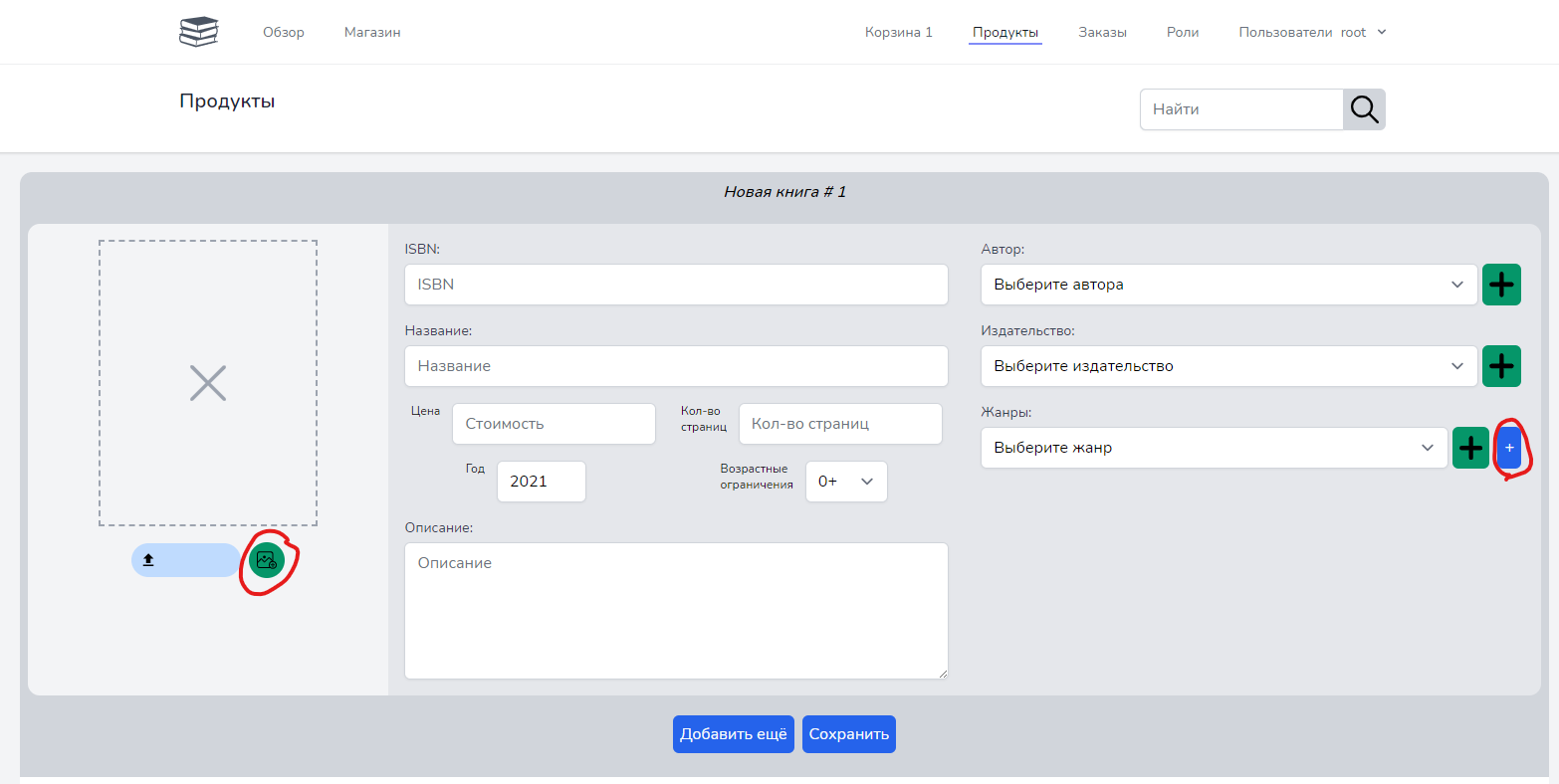


Рисунок 28 – Раздел «Продукты»: Кнопки нескольких жанров и доп. Изображений

Также на странице присутствует поиск, который позволит найти необходимый товар намного быстрее. Поле поиска ищет по всем атрибутам товара. Чтобы изменить данные книги, необходимо найти соответствующую книгу и на выбранном атрибуте, для изменения, нажать знак редактирования, который откроет форму и позволит сохранить изменения (рисунок 29).

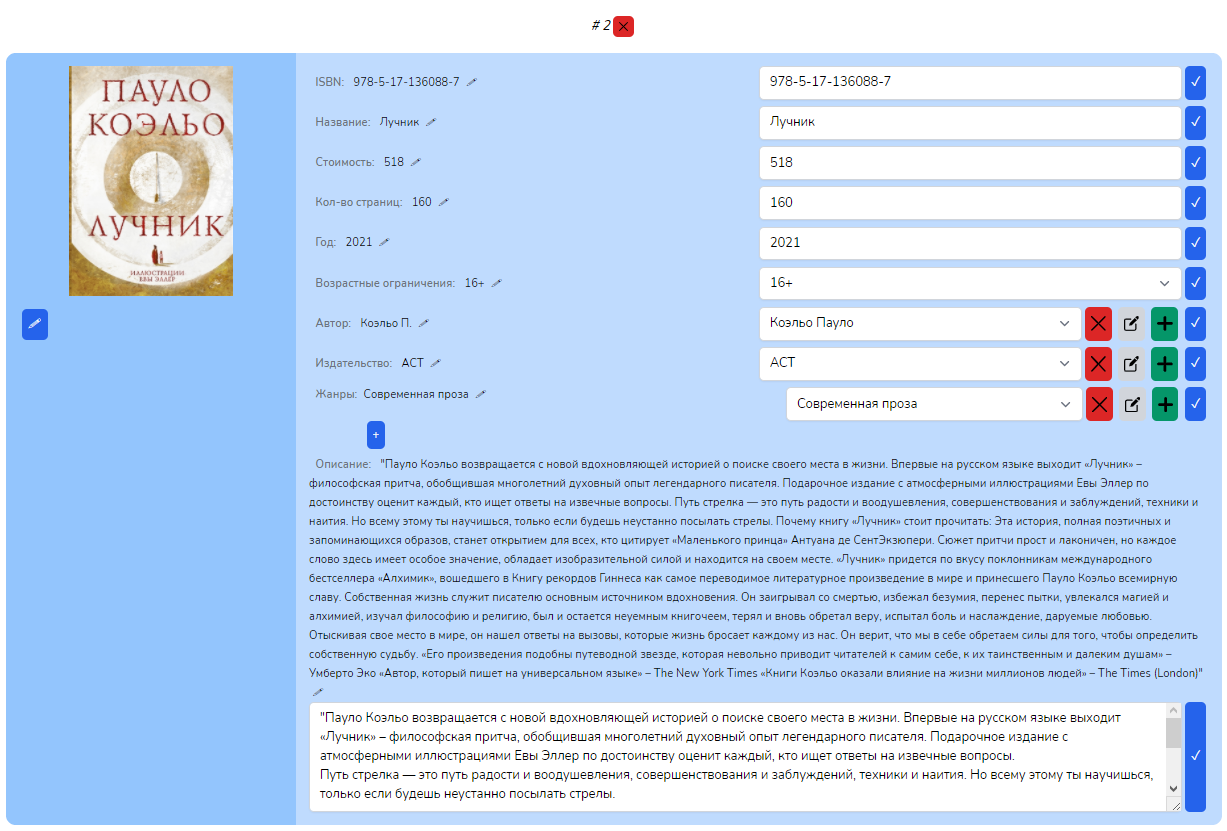


Рисунок 29 – Раздел «Продукты»: Редактирование данных

* + 1. Раздел «Заказы»

Раздел «Заказы» позволяет изменить статус невыполненных заказов. На эту страницу могут попасть пользователи только с правом доступа «order-edit». Для изменения статуса необходимо выбрать необходимый статус из списка и нажать на кнопку «Изменить». Внешний вид страницы представлен на рисунке 30.

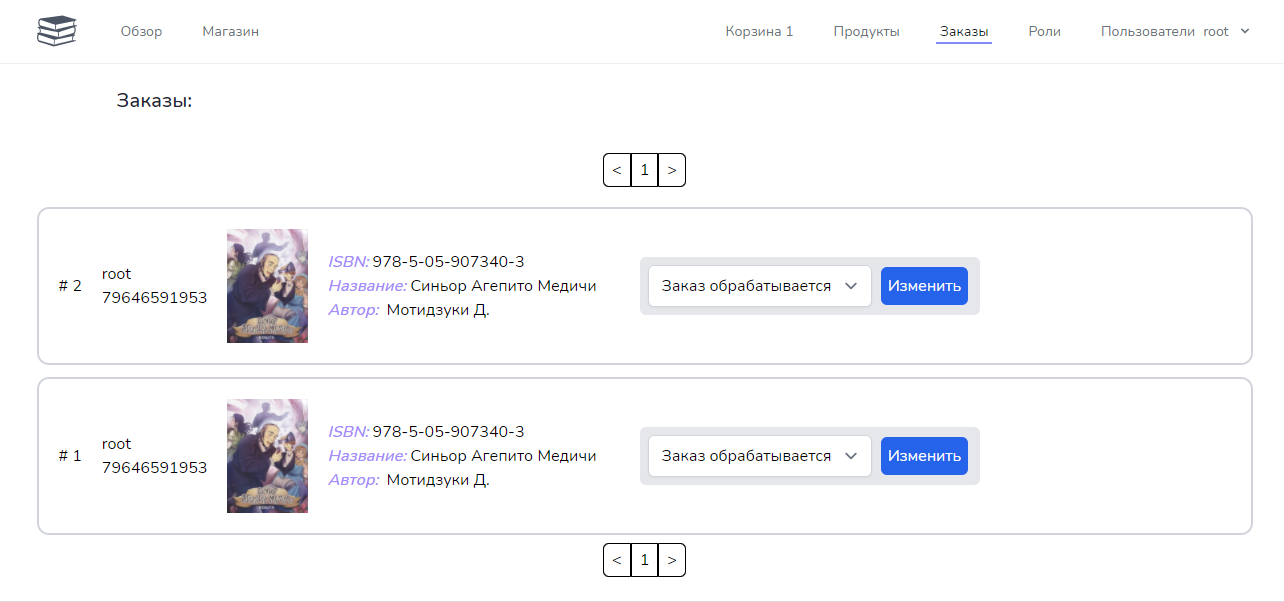


Рисунок 30 – Раздел «Заказы»

* + 1. Раздел «Роли»

Раздел «Роли» позволяет манипулировать ролями приложения. Имеется возможность просмотра прав доступа роли, редактировать права доступа роли, удалить роль, а также создать новую. Внешний вид продемонстрирован на рисунке 31.

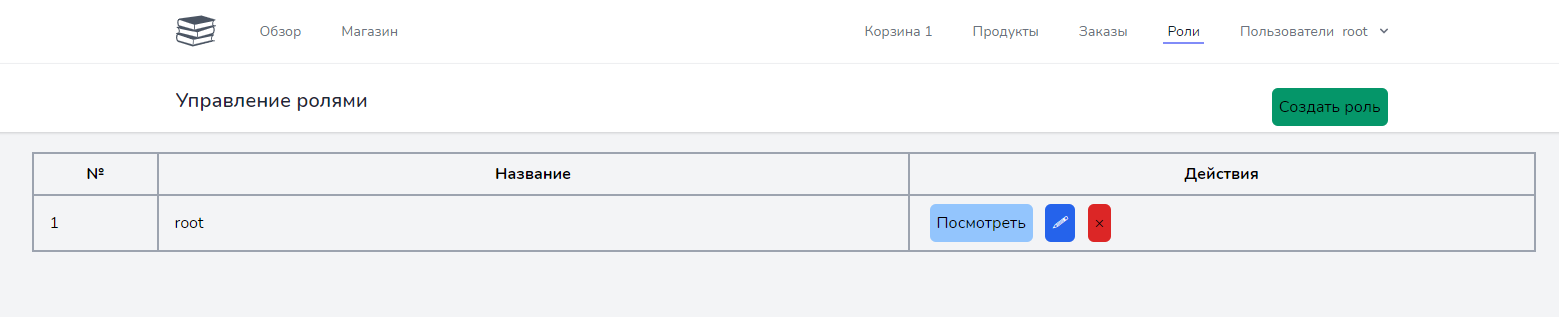


Рисунок 31 – Раздел «Роли»

Редактирование данных роли происходит после перехода на соответствующую страницу раздела. Для этого необходимо нажать на синюю кнопку со значком «карандаша». На странице редактирования имеется возможность изменить название роли, а также переназначить права доступа. Внешний вид продемонстрирован на рисунке 32.

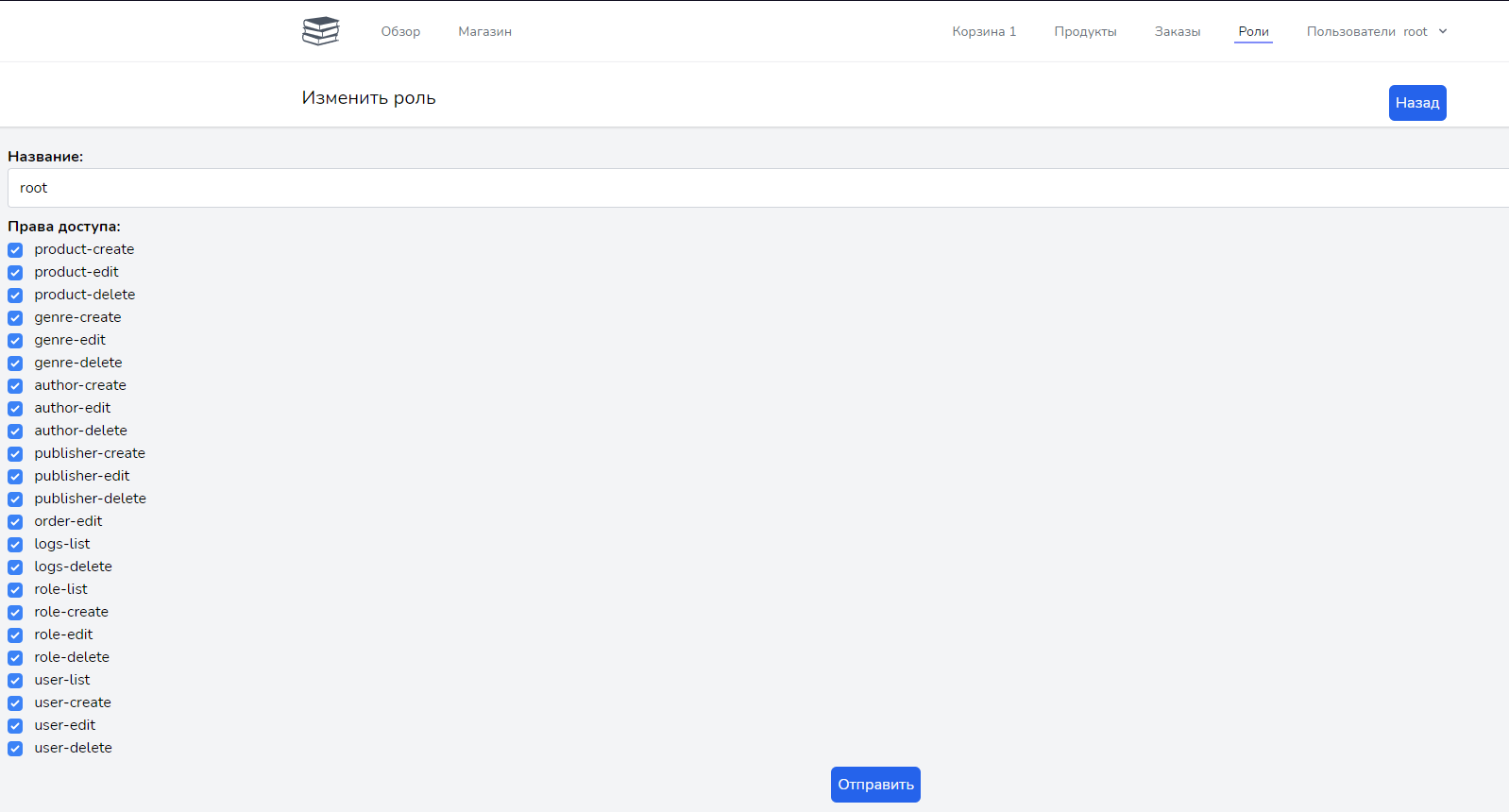


Рисунок 32 – Страница редактирования раздела «Роли»

Для создания новой роли необходимо нажать соответствующую кнопку и заполнить форму. Форма схожа с формой редактирования (рисунок 33). В форме необходимо указать название новой роли и выбрать из списка права доступа, которые будут соответствовать новой роли.

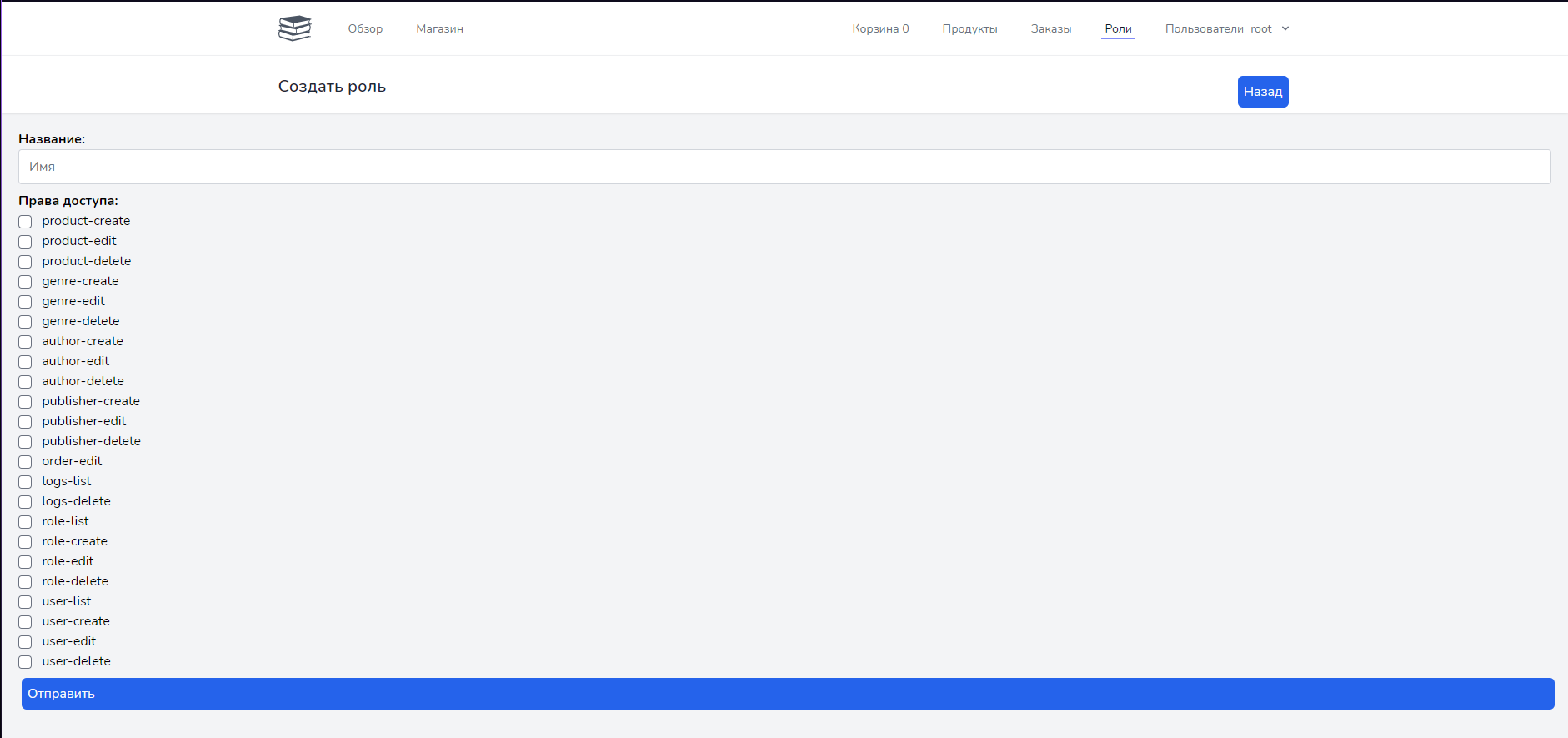


Рисунок 33 – Страница создания роли раздела «Роли»

* + 1. Раздел «Пользователи»

Раздел «Пользователи» позволяет манипулировать данными учетных записей приложения. Имеется возможность просмотра ролей учетной записи, изменить пароль, почту, логин, роль учётной записи, удалить учётную запись, а также зарегистрировать новую. Внешний вид продемонстрирован на рисунке 34.

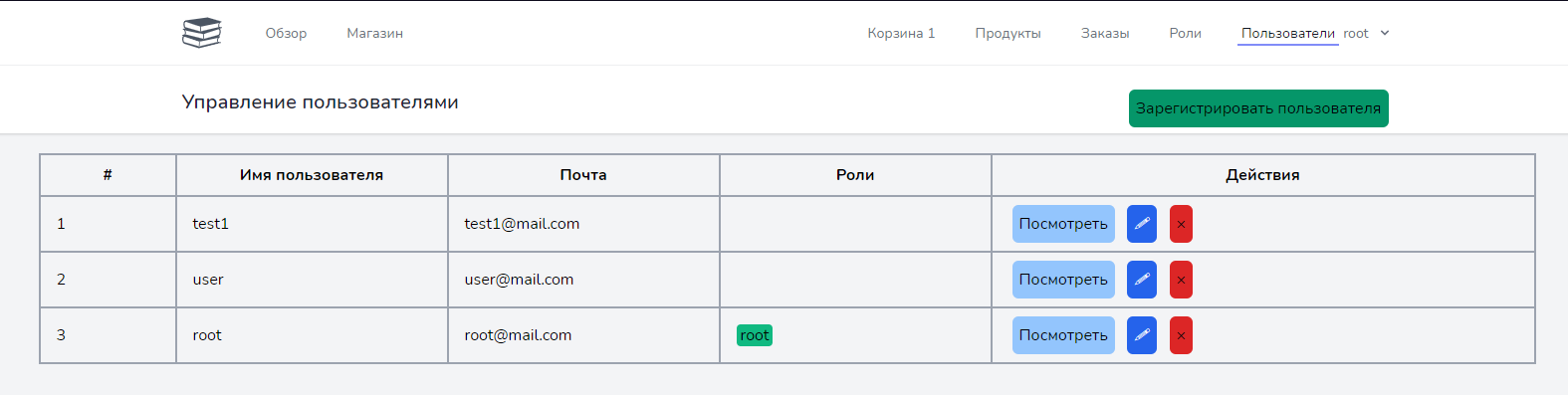


Рисунок 34 – Раздел «Пользователи»

Все действия производятся аналогично разделу «Роли», а именно редактирование данных пользователя (рисунок 35) и регистрация нового пользователя (рисунок 36).

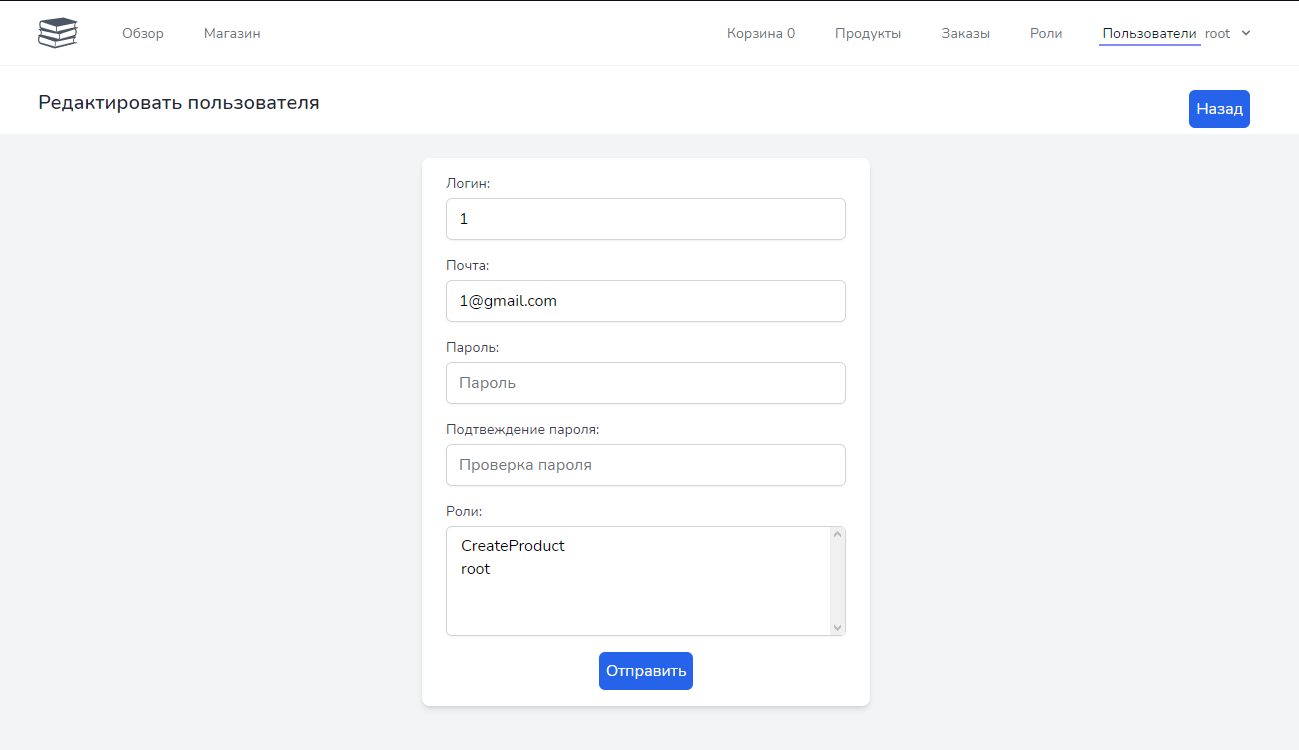


Рисунок 35 – Раздел «Пользователи»: Редактирование данных

На странице редактирования данных видна форма, которая позволяет изменить логин, почту, пароль и роль пользователя. Также пользователю возможно назначить несколько ролей.

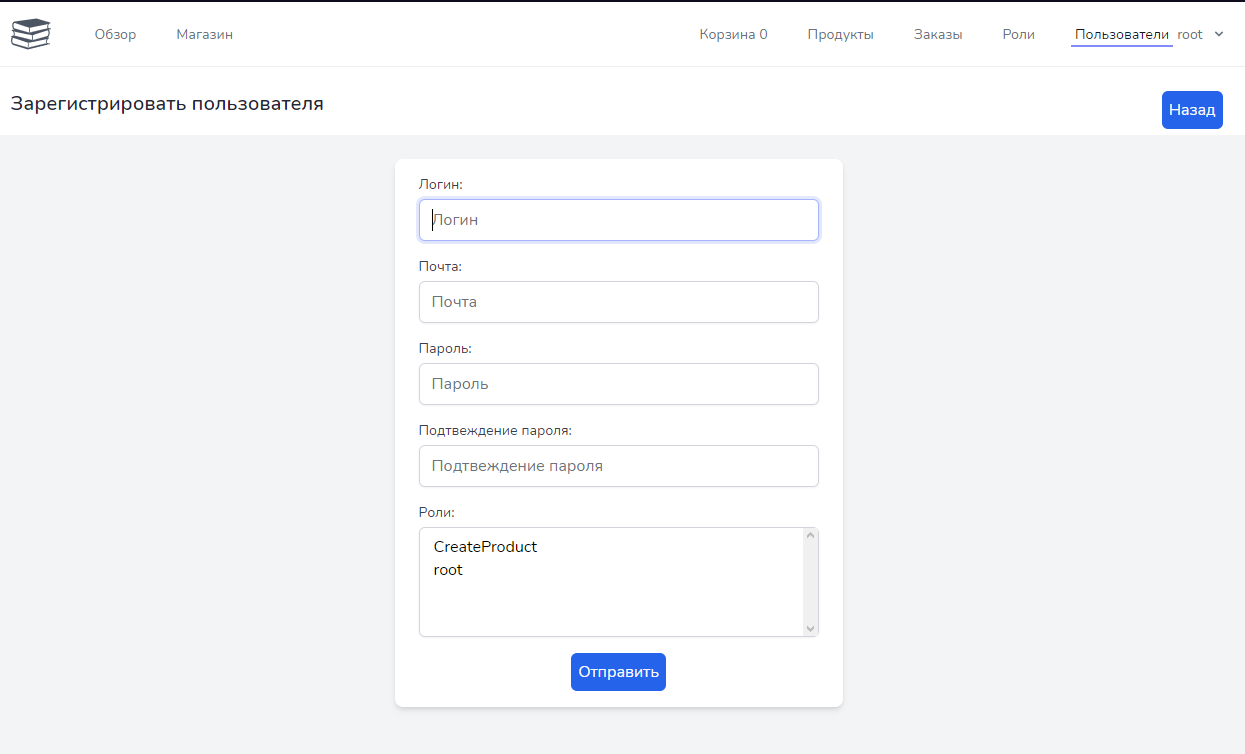


Рисунок 36 – Раздел «Пользователи»: Регистрация пользователя

На странице регистрации пользователя раздела «Пользователи» имеется форма, в которой необходимо указать логин, почту и пароль нового пользователя, а также выбрать присваиваемую ему роль.

1. Стоимость разработки и внедрения программного продукта
   1. Организационно-экономическое обоснование проекта

Разрабатываемое приложение будет использоваться для упрощения процесса оказания услуг, накопления клиентской базы предприятия, связанного c продажей книг.

Разработанный программный продукт достаточно распространённый, из-за чего все аналоги находятся в открытом доступе. Стоимость разработки интернет-магазина составляет от 50 000 рублей, в зависимости от функционала.

* 1. Расчет затрат на разработку программного продукта

Расчет полных затрат на разработку проектного решения (КРПР) осуществляется по формуле 6.1:

(6.1)

КРПР=ЗОТР+ЗЭВМ+ЗСПП+ЗХОН+Е+А,

гдеЗОТР – сумма оплаты труда разработчика/разработчиков ПП;

ЗЭВМ – затраты, связанные с эксплуатацией техники;

ЗСПП – затраты на специальные программные продукты, необходимые для разработки ПП;

ЗХОН – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.);

Е – затраты на электроэнергию, руб.;

А – амортизация ПК, руб.

Для подсчета фонда оплаты труда разработчика необходимо определить общее время разработки (таблица 23). Время, затрачиваемое на разработку проектного решения j-м разработчиком, определяется методом экспертных оценок или хронометража. Итоговое значение рассчитывается на основании приведенных исходных данных по формуле 6.2:

(6.2)

где tβ– время β-го этапа разработки проектного решения, дн.

Таблица 23 - Затраты времени на создание программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап создания | | Затраты времени  (в днях) | Затраты времени (в часах) | Машинное время работы над ПП  (в часах) |
| Разработка | Обследование объекта автоматизации | 3 | 24 | 16 |
| Анализ и уточнение требований | 3 | 24 | 8 |
| Разработка технического задания | 4 | 32 | 32 |
| Проектирование структуры | 5 | 40 | 40 |
|  | Программная реализация | 58 | 464 | 450 |
| Тестирование программного продукта | 5 | 40 | 40 |
| Отладка программного продукта | 5 | 40 | 40 |
| Разработка описания | 5 | 40 | 40 |
| ИТОГО | 88 | 704 | 666 |

Рабочий день принимается равным 8 часам.

Сумму оплаты труда разработчика за время работы над программным продуктом рассчитаем исходя из часовой тарифной ставки и фонда фактического времени, затраченного на разработку программного продукта (по формуле 6.3):

(6.3)

Зотр=Ст1\*Фвр,

где Ст1 - часовой тарифной ставки (принимаем равной 20 руб./час);

Фвр – фонд фактического времени, затраченного на разработку программного продукта, час.

Разработка программного продукта проводилась бесплатно, так как связывающий договор не требует оплаты труда.

Затраты, связанные с использованием вычислительной и оргтехники, рассчитываются по формуле 6.4:

,

(6.4)

где ТМРПР – машинное время работы над программным продуктом, час;

kГ– коэффициент готовности ЭВМ,  ;

n – количество единиц техники, равно 1;

СМ-Ч – себестоимость машино-часа, СМ-Ч=6руб.

Затраты на электроэнергию рассчитываются по следующей формуле 6.5:

Е=W×t×T,

(6.5)

где W – мощность, потребляемая ПК, кВт/час;

t – время работы ПК, час;

T – тариф электроэнергии, руб.

Тариф 1,07 рубля за киловатт.

Амортизация ПК рассчитывается по следующей формуле 6.6:

(6.6)

где – первоначальная стоимость ПК, руб;

– процент амортизации в год.

При разработке программного продукта использовались следующие свободно распространяемые специальные продукты: Tailwind CSS, PhpStorm, Laravel.

Затраты на хозяйственно-организационные нужды приводятся в таблице 24 и вычисляются по формуле 6.7:

,

(6.7)

где Цτ – цена τ-го товара, руб.;

Кτ – количество τ-го товара.

Таблица 24 - Затраты на хозяйственно-организационные нужды

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за единицу (руб.) | Кол-во (шт.) | Всего (руб.) |
| Диск CD-RWDigitex | 25 | 1 | 25 |
| Бумага | 0,5 | 110 | 55 |
| Чернила для принтера | 700 | 2 | 1400 |
| Итого | | | 1480 |

Результаты выполненных расчетов сводятся в общей таблице (таблица 25).

Таблица 25 – Затраты на разработку

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Условное обозначение | Значение |
| Оплата труда разработчика программного продукта |  | 14080 |
| Затраты, связанные с эксплуатацией техники |  | 3796,2 |
| Затраты на электроэнергию | Е | 623,38 |
| Затраты на амортизацию ПК | А | 1250 |
| Затраты на хозяйственно-операционные нужды |  | 1480 |
| Итого затрат на разработку |  | 21229,58 |

* 1. Расчет затрат на внедрение программного продукта

Приложение не внедряется в предприятие, поэтому затрат на внедрение нет. Также все используемые программные средства бесплатны и свободно распространяются.

* 1. Основные выводы

Исходя из расчетов стоимости разработки, можно прийти к выводу, что приложение обойдется предприятию (без учета оплаты труда) в размере 7149,58 рублей. Такие затраты значительно меньше стоимости рассматриваемых аналогов.

# ЗАКЛЮЧЕНИЕ

В ходе дипломного проекта было разработано приложение «Интернет-магазин книг», в котором реализовано система регистрации и авторизации, хранение данных о товарах и операциях над товарами, добавление, удаление и редактирование данных, «корзина покупателя», распределение прав учётных записей и функция подбора книг на основе совершенных покупок и просмотров.

Приложение «Интернет-магазин книг» позволяет упростить процесс оказания услуги, накапливать клиентскую базу, добавлять новые товары в базу. Это ускорит процесс оказания услуг, что позволит оказывать большее количество услуг, что приведет к росту предприятия.

Все поставленные задачи, а именно: система регистрации и авторизации, хранение данных о товарах и операциях над товарами, добавление, удаление и редактирование данных, «корзина покупателя», распределение прав учётных записей и функция подбора книг на основе совершенных покупок и просмотров.

Все цели, поставленные для разработки, были достигнуты.

В приложение были реализованы следующие функции:

* регистрация и авторизация;
* вставка, обновление и удаление данных в базе;
* разграничение прав доступа;
* корзина покупателя;
* оформление заказа.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Calltouch [Электронный ресурс] – Режим доступа: https://www.calltouch.ru/glossary/reklamnoe-obyavlenie/ – Рекламное объявление. (Дата обращения: 16.02.22).
2. CyberForum [Электронный ресурс] – Режим доступа: http://www.cyberforum.ru – (Дата обращения: 03.05.21).
3. BizzApps [Электронный ресурс] – Режим доступа: https://bizzapps.ru/p/draw-io/ – Документация (Дата обращения: 21.04.2021).
4. Hostinger Руководства [Электронный ресурс] – Режим доступа: https://www.hostinger.ru/rukovodstva/web-server-apache/ – Что такое Apache? (Дата обращения: 16.05.21).
5. Mr. Кодер работает [Электронный ресурс] – Режим доступа: http://code.mu/ru/php/book/oop/ – Учебник по ООП и MVC на PHP (Дата обращения: 07.02.22).
6. Wikipedia [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Информационная\_система – Информационная система (Дата обращения: 16.02.22).
7. PHP.NET [Электронный ресурс] – Режим доступа: https://www.php.net/ – Документация по PHP (Дата обращения: 02.02.2022).
8. PHPMyAdmin [Электронный ресурс] – Режим доступа: https://php-myadmin.ru/doc/ability.html – Руководство. Возможности PhpMyAdmin (Дата обращения: 02.05.21).
9. SQL. Полное руководство. Грофф Дж. Р., Вайнберг П.Н., Оппелъ Э. Дж. 2015 – 959 с.
10. Life-prog [Электронный ресурс] – Режим доступа: https://life-prog.ru – Документация (Дата обращения: 21.04.2021).
11. Lucidchart [Электронный ресурс] – Режим доступа: https://www.lucidchart.com/pages/ru/visio-что-это – Что такое Visio? (Дата обращения: 18.04.2021).
12. Habr [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/150267/ – Реализация MVC паттерна на примере создания сайта-визитки на PHP (Дата обращения: 17.02.2022).
13. WebForMyself [Электронный ресурс] – Режим доступа: https://webformyself.com/peredacha-znachenij-peremennyx-iz-javascript-v-php-i-naoborot/ – Документация (Дата обращения: 27.04.2021).
14. Хабр [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/ruvds/blog/539098/ – 9 самых популярных php-фреймворков (Дата обращения: 28.04.2021).
15. Web-технологии [Электронный ресурс] – Режим доступа: https://htmlweb.ru/java/example/image\_preview3.php – Предпросмотр картинки перед загрузкой на сервер на javascript (Дата обращения: 18.02.2022).

# Приложение А - Листинг маршрутов

<?php

Route::get('/', [ShopController::class, 'index'])->name('shop');

Route::get('/shop/', [ShopController::class, 'shop'])->name('shop.shop');

Route::post('/shop/', [ShopController::class, 'filter'])->name('shop.filter');

Route::get('/shop/author/{id}', [ShopController::class, 'author'])->name('shop.author');

Route::get('/shop/publisher/{id}', [ShopController::class, 'publisher'])->name('shop.publisher');

Route::get('/shop/limit/{id}', [ShopController::class, 'limit'])->name('shop.limit');

Route::get('/shop/genre/{id}', [ShopController::class, 'genre'])->name('shop.genre');

Route::post('/cart', [CartController::class, 'index'])->name('cart');

Route::group(['middleware' => ['auth']], function() {

Route::get('user/lk', [UserController::class, 'lk'])->name('lk');

Route::get('products', [ProductController::class, 'index'])->name('products');

Route::middleware('permission:order-edit')->get('orders', [OrderController::class, 'index'])->name('orders');

Route::resource('roles', RoleController::class);

Route::resource('users', UserController::class);});

require \_\_DIR\_\_.'/auth.php';

Route::get('/{id}', [ShopController::class, 'show'])->name('shop.show');