**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| ДП.09.02.03.21.173.03.ПЗ | УТВЕРЖДАЮ  Зам. директора по УР, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_Е.А. Коробкова |

ПРИЛОЖЕНИЕ «ИНТЕРНЕТ-МАГАЗИН КНИГ»

|  |  |  |
| --- | --- | --- |
| Нормоконтролер: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.Ю. Безносова) |
| Консультант по экономической части: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.А. Белова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (С.А. Удальцов) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Д.С. Зан) |

Иркутск 2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc71239671)

[1 Предпроектное исследование 5](#_Toc71239672)

[1.1 Обзор аналогов программного обеспечения 5](#_Toc71239673)

[1.2 Анализ программных продуктов 9](#_Toc71239674)

[1.3 Обоснование выбора программного продукта 11](#_Toc71239675)

[2 Проектирование 13](#_Toc71239676)

[2.1 Архитектура программного обеспечения 13](#_Toc71239677)

[2.2 Функциональное проектирование 14](#_Toc71239678)

[2.3 Проектирование базы данных 17](#_Toc71239679)

[2.4 Проектирование пользовательского интерфейса 22](#_Toc71239680)

[3 Реализация программного обеспечения 26](#_Toc71239681)

[3.1 Кодирование программного обеспечения 26](#_Toc71239682)

[4 Внедрение и сопровождение программного обеспечения 32](#_Toc71239683)

[5 Документирование программного обеспечения 33](#_Toc71239684)

[5.1 Руководство по установке программного обеспечения 33](#_Toc71239685)

[5.2 Руководство пользователя программного обеспечения 33](#_Toc71239686)

[5.3 Руководство системного администратора программного обеспечения 43](#_Toc71239696)

[6 Стоимость разработки и внедрения программного продукта 45](#_Toc71239697)

[6.1 Организационно-экономическое обоснование проекта 45](#_Toc71239698)

[6.2 Расчет затрат на разработку программного продукта 45](#_Toc71239699)

[6.3 Расчет затрат на внедрение программного продукта 49](#_Toc71239700)

[6.4 Основные выводы 49](#_Toc71239701)

[ЗАКЛЮЧЕНИЕ 50](#_Toc71239702)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 51](#_Toc71239703)

[Приложение А - Листинг маршрутов 52](#_Toc71239704)

ВВЕДЕНИЕ

В современных условиях, повышение эффективности продаж является одной из ключевых задач, стоящей перед руководством магазина книг. Чтобы повысить эффективность продаж используют множество различных способов, одним из них является добавление дополнительного место оказания услуги.

Таким местом может стать интернет-магазин, то есть появление в сети Интернет собственного веб-сайта магазина, который позволит организовать оформление заказа на товар клиентом.

В связи с этим поставлена задача в разработке приложения «Интернет-магазин книг». Приложение будет иметь доступ к базе данных, организует подбор книг по просмотрам пользователя и его заказам.

Цель дипломного проекта – разработать приложение «Интернет-магазин книг». Цель разработки – упрощение процесса оказания услуг, накопление клиентской базы.

Предполагается использование приложения магазинами книг, для добавление дополнительной платформы оказания услуг.

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. Обзор аналогов программного обеспечения

Перед тем как приступить к разработке программного продукта, необходимо выполнить обзор существующих аналогов, для определения какой функционал и в какой стилистике реализовывать программный продукт.

В результате обзора и анализа выявлены следующие аналоги будущей программной разработки соответствующей предметной области:

1. Chitai-gorod.ru – это интернет-магазин магазина книг и канцелярских товаров «Читай город». Фрагмент пользовательского интерфейса Chitai-gorod.ru можно увидеть на рисунке 1.

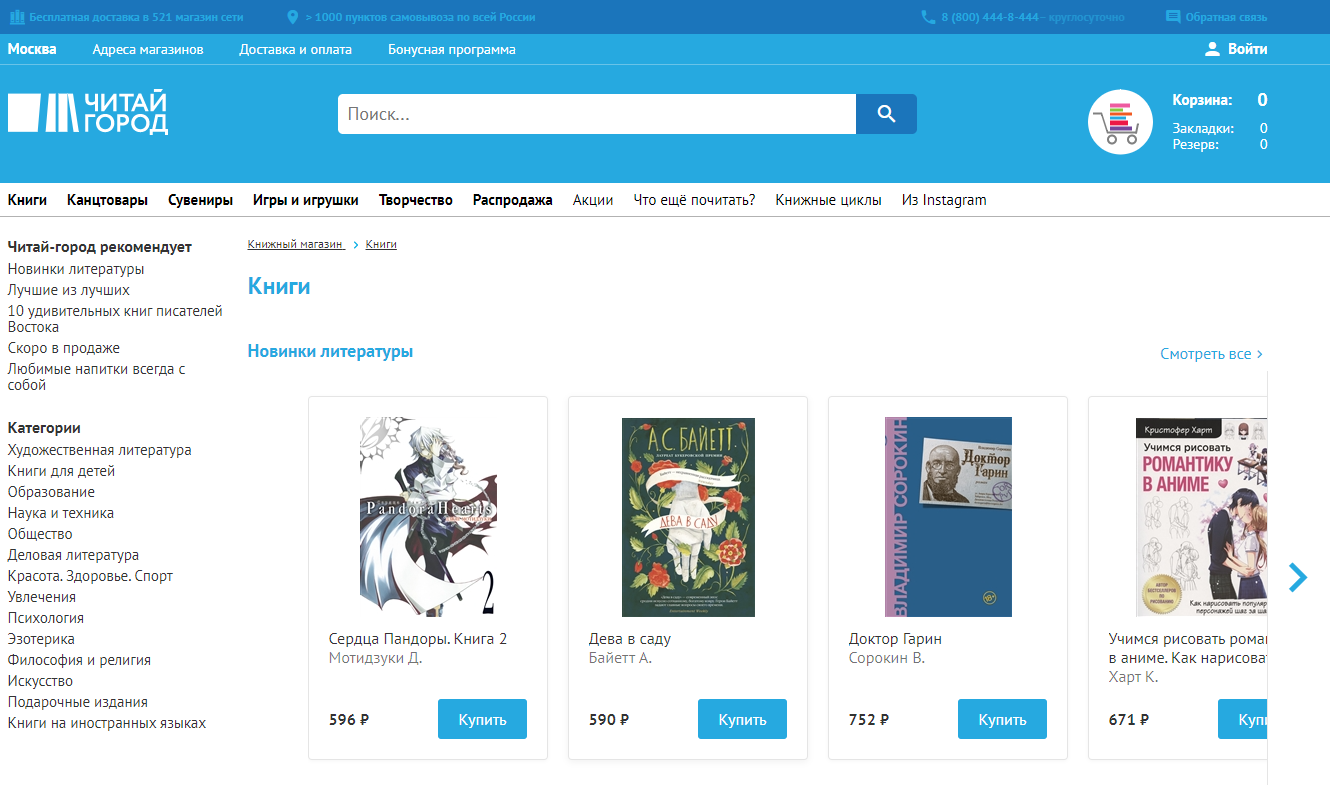


Рисунок 1 – Chitai-gorod.ru

Chitai-gorod.ru можно считать отличным программным продуктом, так как в нём грамотно разделены товары, имеется корзина и оплата заказа, а также отслеживание заказа.

1. Books.ru — старейший книжный интернет-магазин в России. К настоящему моменту в ассортименте магазина не только бумажные, но и электронные книги, а также электроника, софт, музыка и видео. Фрагмент пользовательского интерфейса books.ru на рисунке 2.

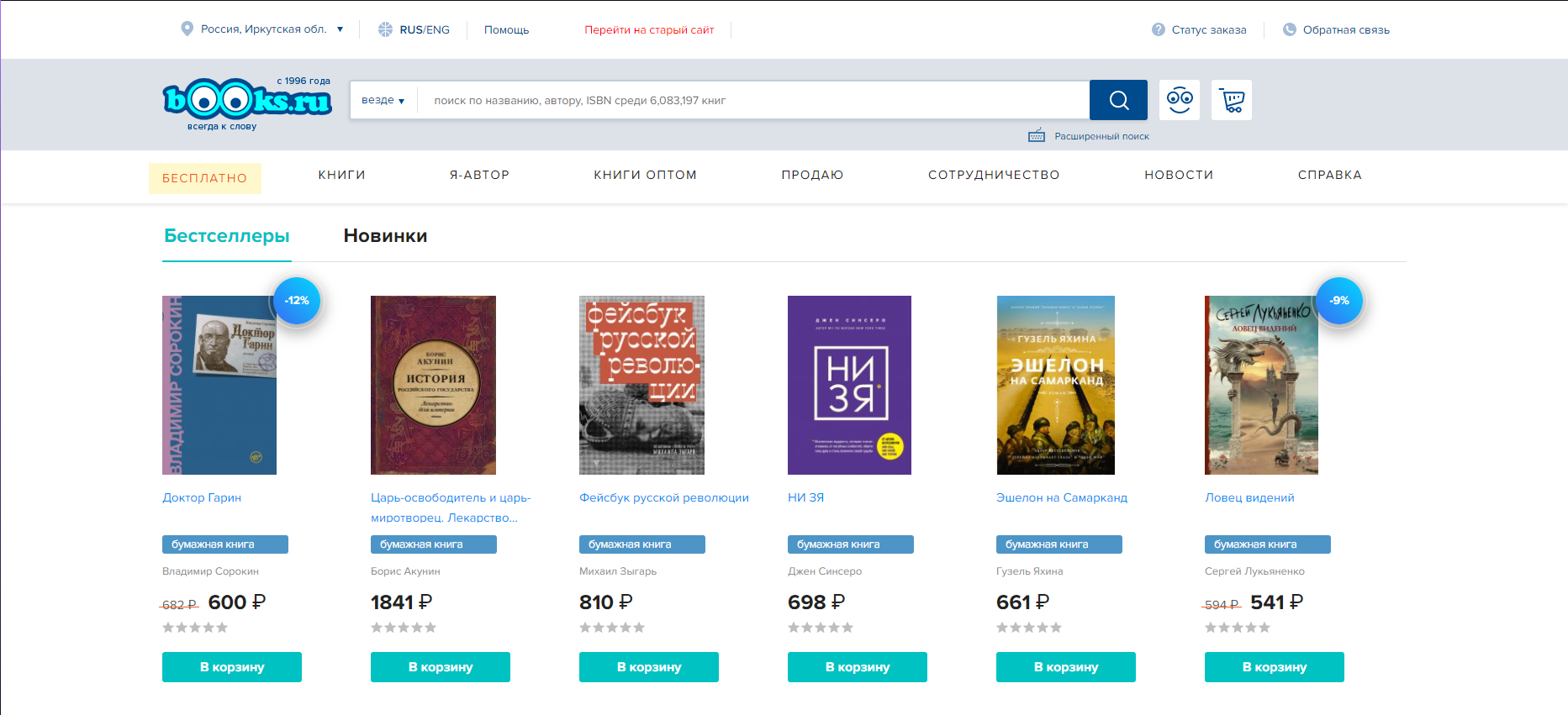


Рисунок 2 – Books.ru

Books.ru позволяет пользователю не только купить книгу, но и продать её. Также интернет-магазин имеет несколько вариаций пользовательского интерфейса.

1. Labirint.ru – интернет-магазин книг и канцелярских товаров. Предоставляет информацию о издательствах, авторах, что позволяет клиенту лучше ознакомиться с продаваемой продукцией.

Фрагмент пользовательского интерфейса Labirint.ru на рисунке 3.

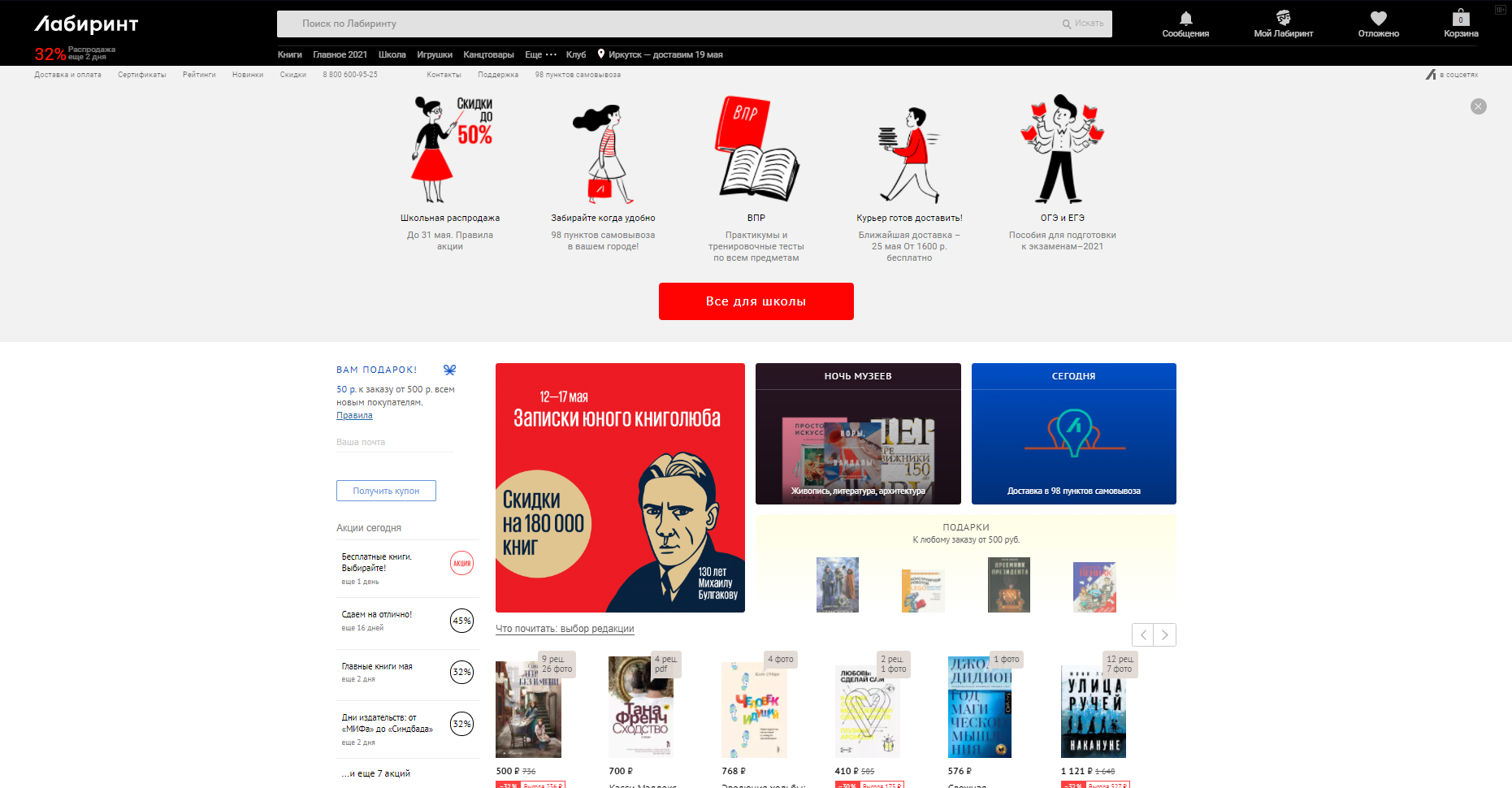


Рисунок 3 – Labirint.ru

Labirint.ru отличный интернет-магазин книг, который позволяет отлично ознакомиться с продукцией, рецензиями на неё, а также с автором и издательством.

Проанализировав аналоги можно прийти к выводу, что все интернет-магазины для лучшего представления книг используют листающийся список (слайдер), имеют фильтры по авторам, жанрам и издательствам.

* 1. Анализ программных продуктов

Приложение «Интернет-магазин книг» будет реализовано при помощи веб-технологий, так как это позволит использовать приложение без установки какого-либо дополнительного программного обеспечения.

Для работы с БД рассмотрены следующие программные средства:

1. MySQL;
2. SQLite;
3. PostgreSQL.

Для создания СУБД рассмотрены следующие программные средства:

1. PHPMyAdmin;
2. Adminer;
3. MariaDB.

Для разработки приложения рассмотрены следующие программные средства:

1. PHPStorm
2. Notepad++
3. WebStorm

MySQL – свободная реляционная система управления базами данных для сервера SQL. Это простая, эффективная и надежная система управления информацией, использующая традиционный язык SQL. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей.

SQLite - система управления базами данных. SQLite является решением для малых и средних приложений. Обычно SQLite используется в качестве сервера в автономных программах.

PostgreSQL – свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, Linux, Mac OS X, Solaris/OpenSolaris, Tru64, QNX, и даже для Microsoft Windows. PostgreSQL базируется на языке SQL и поддерживает многие из возможностей стандарта языка программирования SQL.

PostgreSQL считаются высокопроизводительной и надёжной СУБД. В ней имеются механизмы транзакций и репликация, а также легкая расширяемость программы позволяет импортировать различные языки программирования запросов.

PHPMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PHPMyAdmin позволяет через браузер, осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

Adminer (бывший phpMinAdmin) – это легковесный инструмент администрирования MySQL, PostgreSQL, SQLite, MS SQL и Oracle. Проект родился как «облегчённый» вариант phpMyAdmin. Распространяется на условиях Apache license в форме одиночного PHP-файла размером около 380 килобайт, который является результатом компиляции исходных php- и js-файлов с помощью специального PHP-скрипта.

MariaDB – ответвление от системы управления базами данных MySQL. MariaDB намерен поддерживать высокую совместимость с MySQL, обеспечивая точное соответствие с API и командами MySQL.

PhpStorm – это интегрированная среда разработки на PHP с интеллектуальным редактором, которая глубоко понимает код, поддерживает PHP 5.3-7.2 для современных и классических проектов, обеспечивает лучшее в индустрии автодополнение кода, предотвращение ошибок налету и поддерживает смешивание языков.

Notepad++ – свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки.

JetBrains WebStorm – интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями.

* 1. Обоснование выбора программного продукта

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

Основные преимущества PHPStorm:

1. автодополнение кода;
2. предотвращение ошибок;
3. выполнение запросов в среде разработки;
4. глубокий анализ кода JavaScript, его популярных библиотек и фреймворков, CSS, HTML, PHP и его популярные библиотеки и фреймворки.

Основные преимущества PHPMyAdmin:

1. простой интерфейс;
2. выполнение всех функций манипуляций с таблицами;
3. настройка прав доступа, конфигураций серверов.

Основные преимущества MySQL:

1. простота;
2. эффективность;
3. быстрота;
4. безопасность;
5. гибкость.

Таким образом, для реализации цели дипломного проекта было решено использовать следующие средства разработки:

* на этапе проектирования, для создания ER-модели использовалось CASE средства – draw io;
* для создания структурных схем использовалось CASE средства – Microsoft Visio;
* для создания контекстной и диаграмм декомпозиции использовался графический редактор – Microsoft Visio;
* на этапе разработки приложения, использовались WEB-технологии – HTML, PHP, JavaScript, CSS, tailwind css, Laravel;
* для работы с базой данных использовался метод языка программирования PHP - PDO.

1. Проектирование

Перед проектированием приложения было разработано техническое задание, на основе которого производилось проектирование приложения «Интернет-магазин книг». 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.

Приложение 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 приложение практически на любом устройстве.

На рисунке 4 демонстрируется архитектура веб-приложений, которая соответствует разрабатываемому приложению.

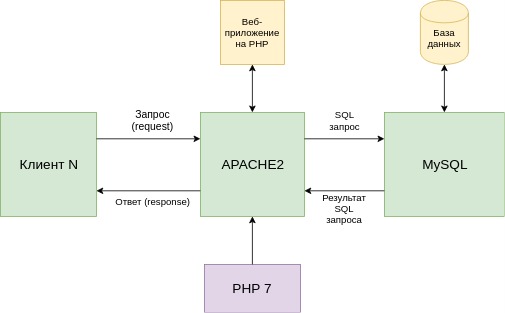


Рисунок 4 – Клиент-серверная архитектура веб-приложения

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.

* 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 5 показаны входные данные, управление механизм, выходные данные и функция.

В центре диаграммы показана приложение «Интернет-магазин книг». Левая стрелка это входные данные, т.е. «Данные с веб-страниц» и «Данные из БД». Верхняя стрелка — это управление «Алгоритм обработчика событий». Нижняя стрелка – это механизм «Пользователь». А правая - выходные данные «Обработанные данные».

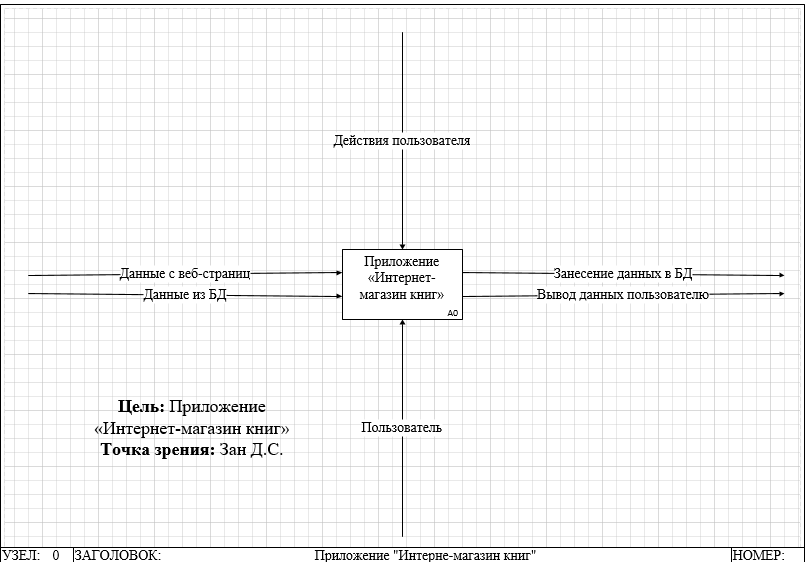


Рисунок 5 – Контекстная диаграмма модуля

Для демонстрации деталей в процессе необходимо спроектировать диаграмму декомпозиции.

Диаграмма декомпозиции – это разбиение функции, т.е. приложения «Интернет-магазин книг».

На рисунке 6 показана диаграмма декомпозиции, которая расписывает работу приложения из контекстной диаграммы. На этой диаграмме показаны 11 функций — это «Аутентификация», «Авторизация», «Обзор», «Магазин», «Корзина», «Заказы», «Личный кабинет», «Изменение БД», «Манипулирование данными ролей», «Вывод данных», «Манипулирование данными пользователей». А также на этой диаграмме присутствуют входные данные, выходные данные, управления и механизмы.

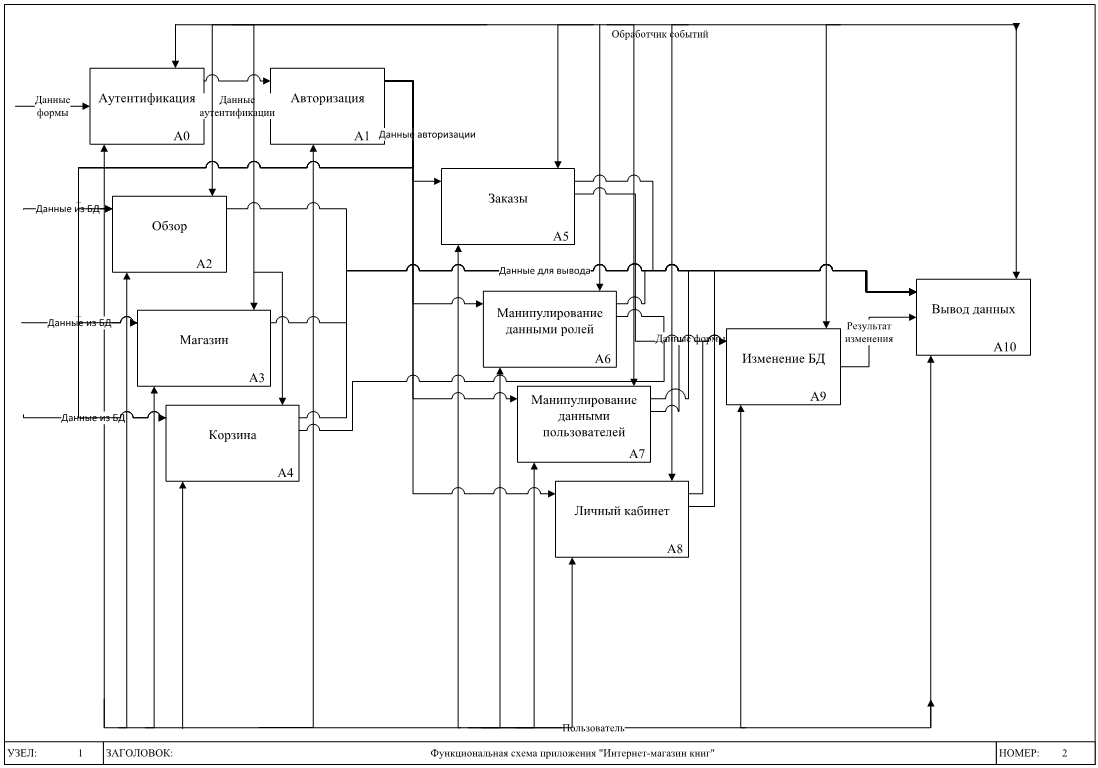


Рисунок 6 – Диаграмма декомпозиций

На рисунке 7 показана диаграмма IDEF3, которая расписывает работу приложения из рисунка 6.

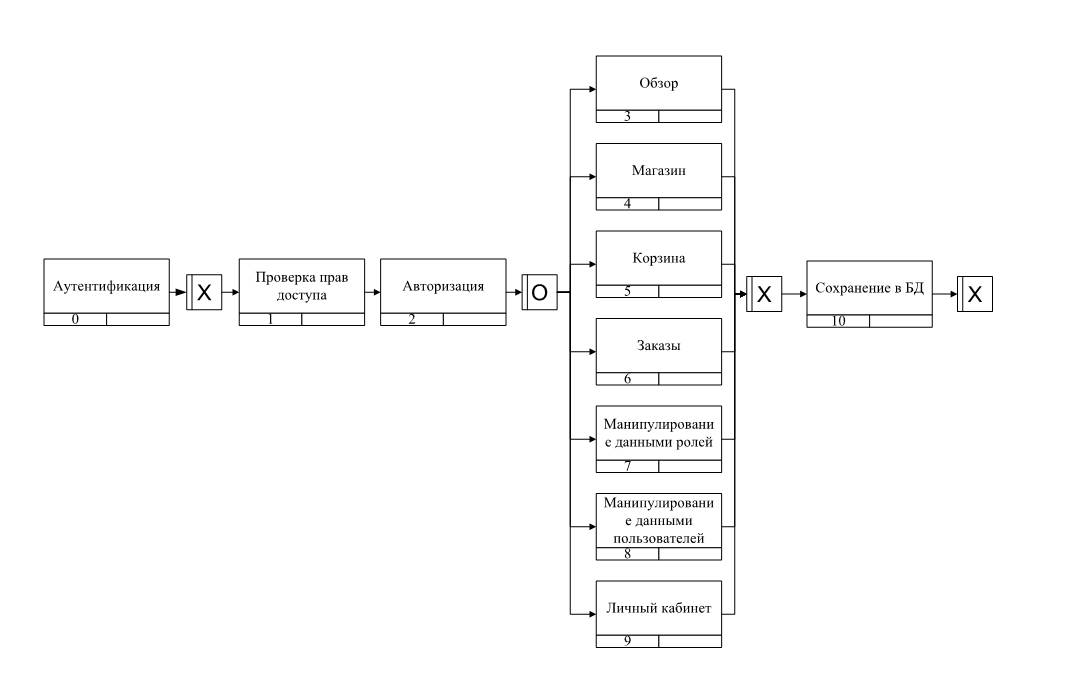


Рисунок 7 –IDEF3 диаграмма

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

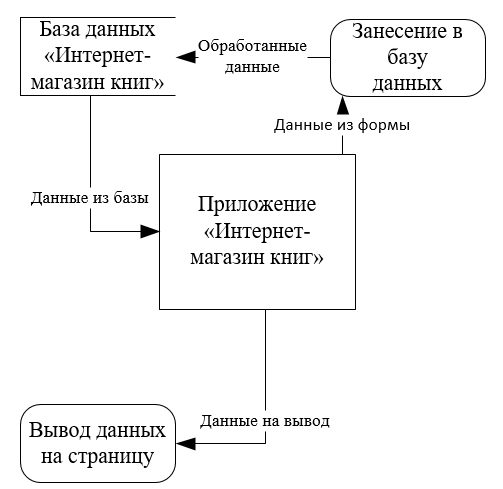


Рисунок 8 – Диаграмма потоков данных

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

* 1. Проектирование базы данных

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

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

На рисунке 9 изображена схема базы данных приложения «Интернет-магазин книг».

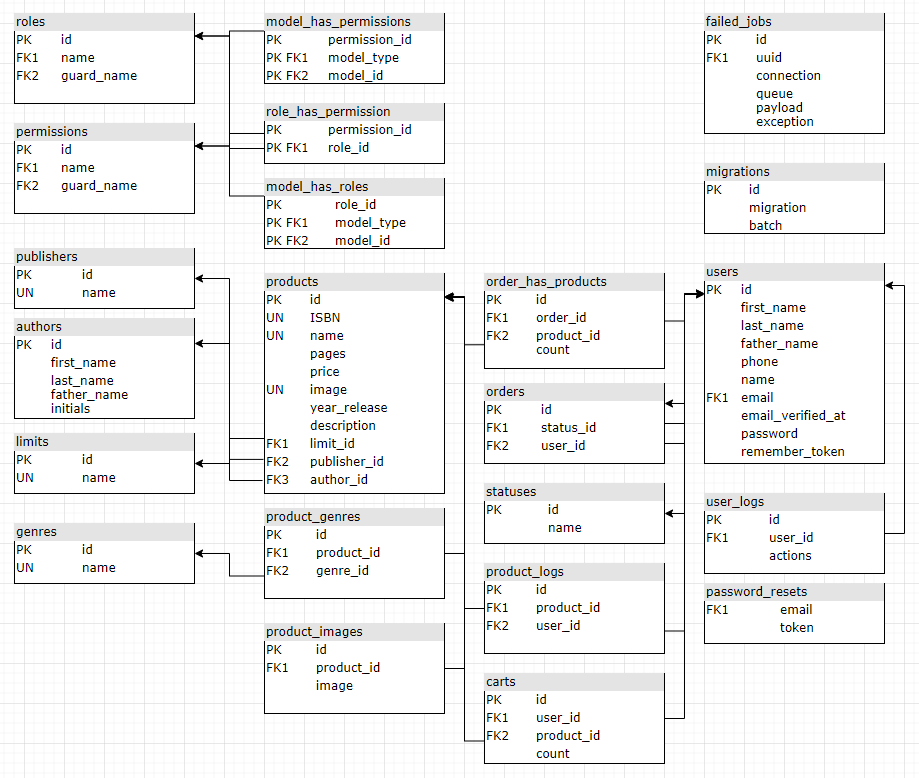


Рисунок 9 – Схема базы данных

База данных представляет собой 22 связанных таблиц. Для создания связей между таблицами используются внешние ключи.

Разработка ER-модели выполнена с помощью программы case-средств draw io. Были выявлены следующие сущности и их атрибуты:

1. Авторы:

* имя автора;
* фамилия автора;
* отчество автора
* инициалы автора.

1. Возрастные ограничения:

* название возрастного ограничения.

1. Жанры:

* название жанра.

1. Издательства:

* название издательства.

1. Жанры книг:

* книга;
* жанр.

1. Изображения книги:

* книга;
* путь до изображения.

1. Книги:

* ISBN;
* название книги;
* количество страниц;
* стоимость книги;
* главное изображение книги;
* год издания книги;
* возрастное ограничение;
* издательство;
* автор.

1. Заказы:

* пользователь;
* статус.

1. Книги заказа:

* заказ;
* книга;
* количество.

1. Статусы:

* название статуса.

1. Действия с книгами:

* пользователь;
* книга.

1. Действия пользователей:

* пользователь;
* действие.

1. Пользователи:

* никнейм;
* имя пользователя;
* фамилия пользователя;
* отчество пользователя;
* телефон пользователя;
* почта;
* пароль;
* верификация почты.

1. Восстановление пароля:

* почта;
* дата изменения.

1. Миграции:

* миграция;
* количество.

1. Ошибки:

* соединение;
* очередь;
* дата ошибки;
* нагрузка;
* исключения.

1. Роли пользователя:

* название роли;
* дата создания роли.

1. Права доступа:

* название права;
* дата создания права.

1. Модели ролей:

* роль;
* тип модели.

1. Роли прав доступа:

* право доступа;
* роль.

1. Модели прав доступа:

* право доступа;
* тип модели.

1. Корзина:

* пользователь;
* книга.

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.

Все таблицы содержат минимальный набор независимых данных. Большинство таблиц не содержат больше 5 столбцов.

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 10 показан прототип раздела «Вход», который авторизоваться пользователю приложения.

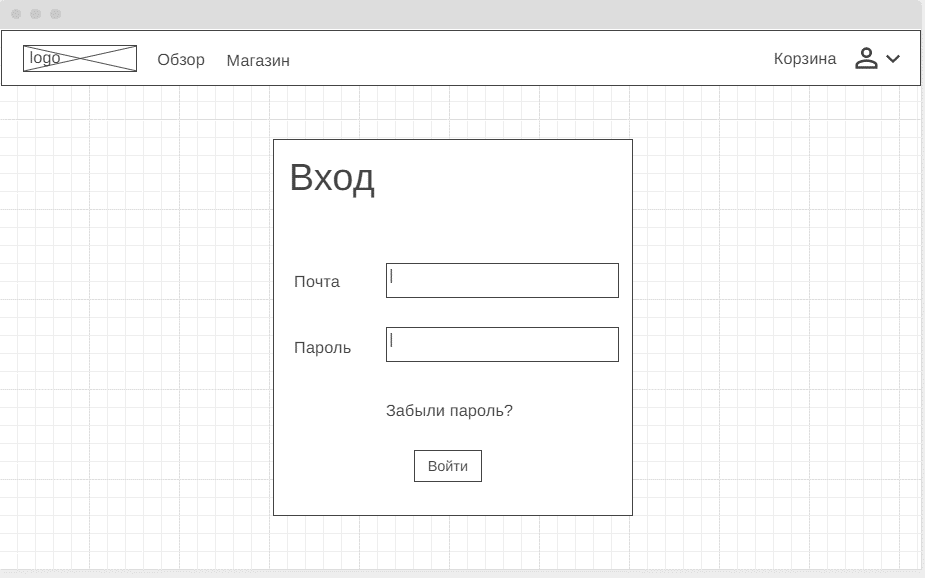


Рисунок 10 – Прототип раздела «Вход»

На рисунке 11 демонстрируется прототип основной страницы «Обзор», на которой пользователь может увидеть рекомендуемые приложением товары. На эту страницу может войти и не авторизованный пользователь приложения.

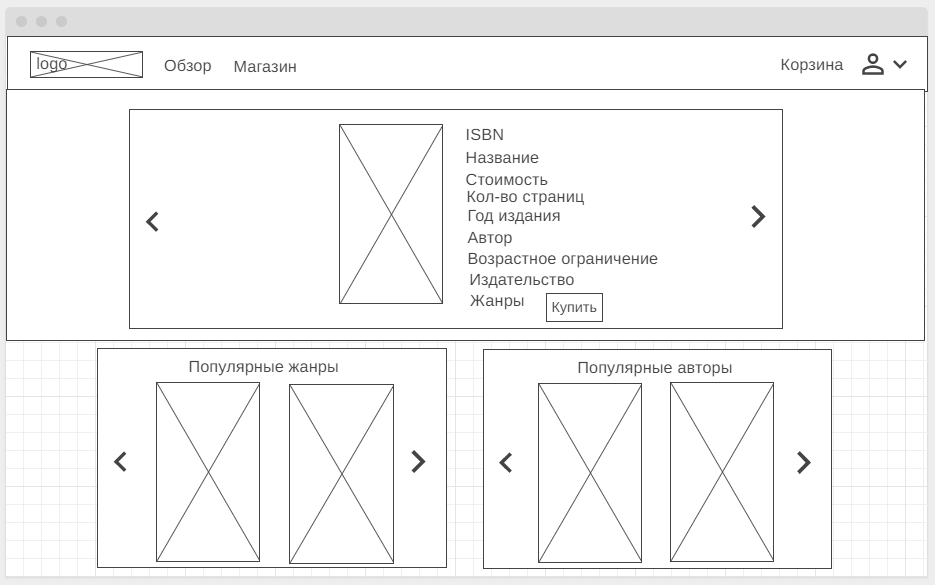


Рисунок 11 – Прототип страницы «Обзор»

На рисунке 12 демонстрируется прототип страницы «Магазин». На рабочем пространстве размещен список книг и список фильтров, которые можно применить к списку книг.

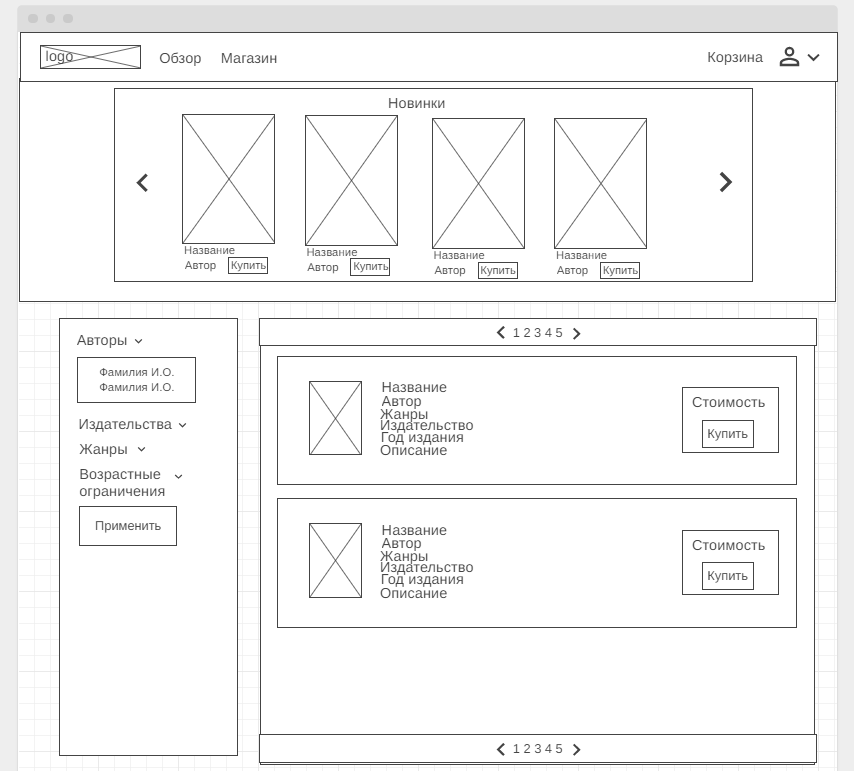


Рисунок 12 – Прототип страницы «Магазин»

На рисунке 13 демонстрируется прототип страницы «Корзина». На рабочем пространстве размещен список книг, выбранных пользователем для оформления заказа.

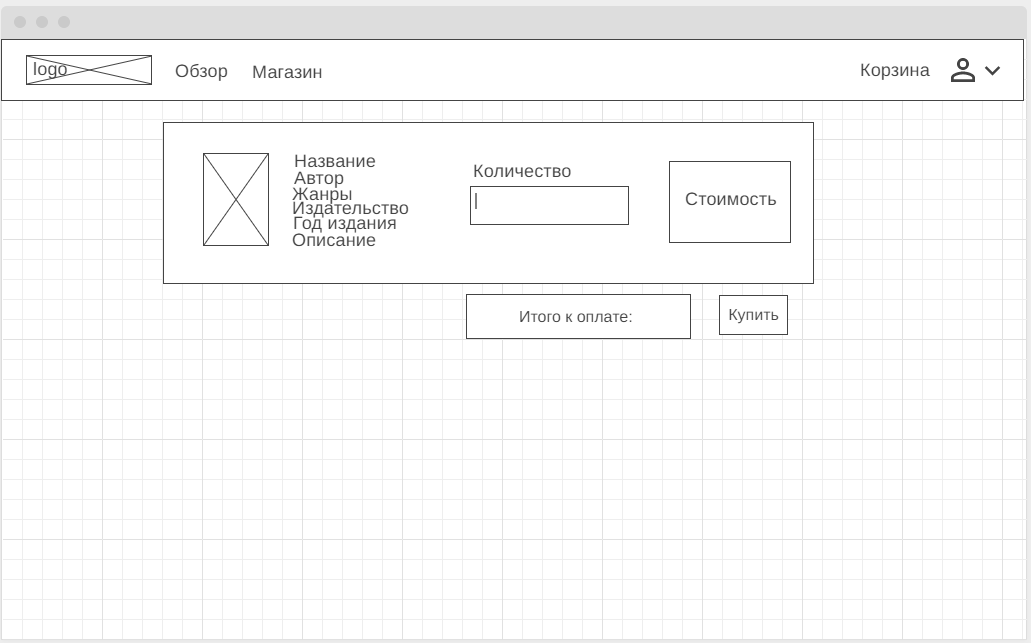


Рисунок 13 – Прототип страницы «Корзина»

На рисунке 14 демонстрируется прототип страницы «Регистрация», которая позволяет зарегистрировать нового пользователя приложения.

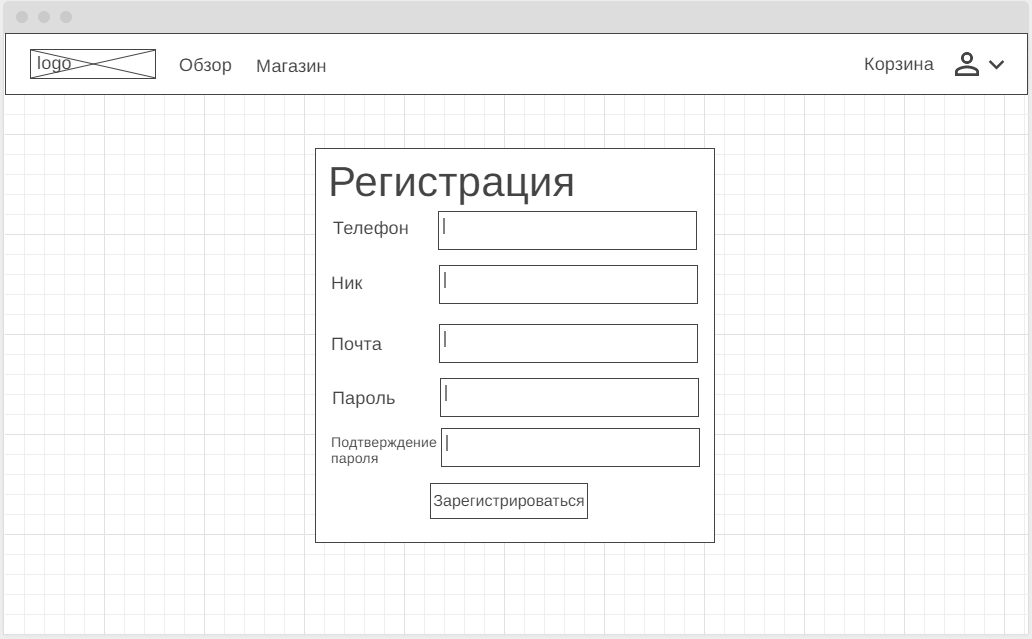


Рисунок 14 – Прототип страницы «Регистрация»

На рисунке 15 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.

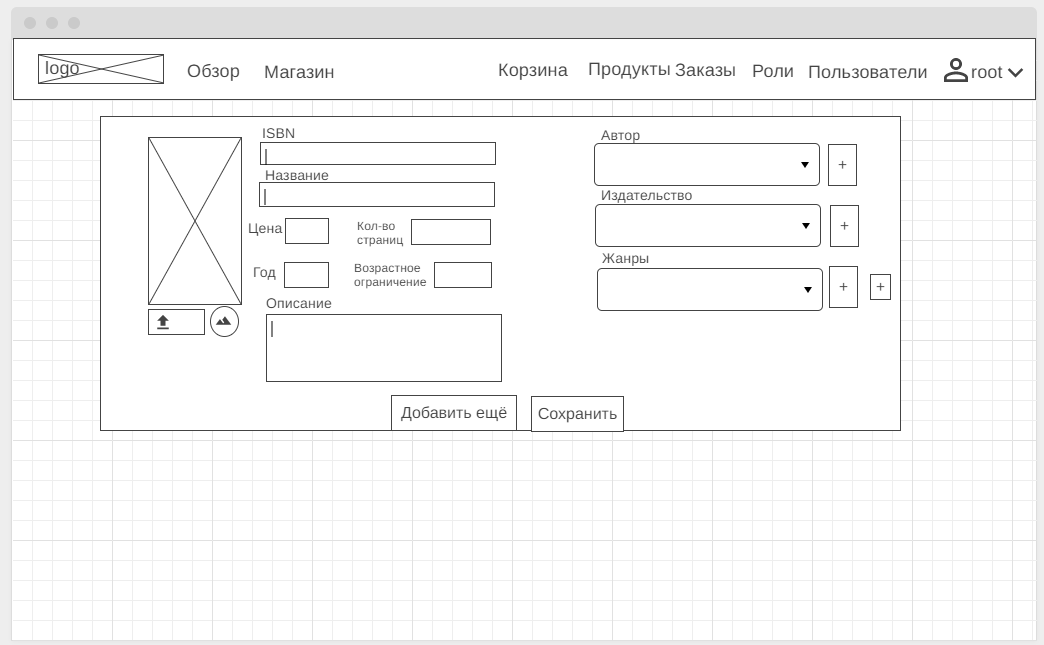


Рисунок 15 – Прототип страницы «Продукты»

На рисунке 16 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.

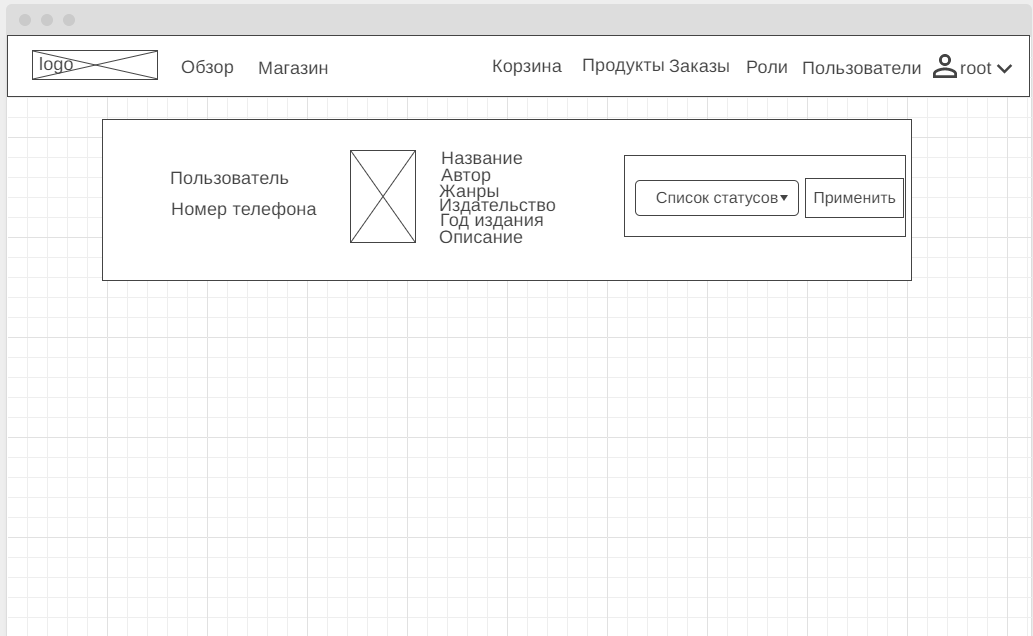


Рисунок 16 – Прототип страницы «Заказы»

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. ёРеализация 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;
* фреймворк Vanilla JS - для скриптового языка JavaScript;
* фреймворк Laravel - для серверной составляющей модуля.

Благодаря фреймворку tailwind css приложение «Интернет-магазин книг» имеет простой, но стильный внешний вид, который продемонстрирован на рисунке 17.

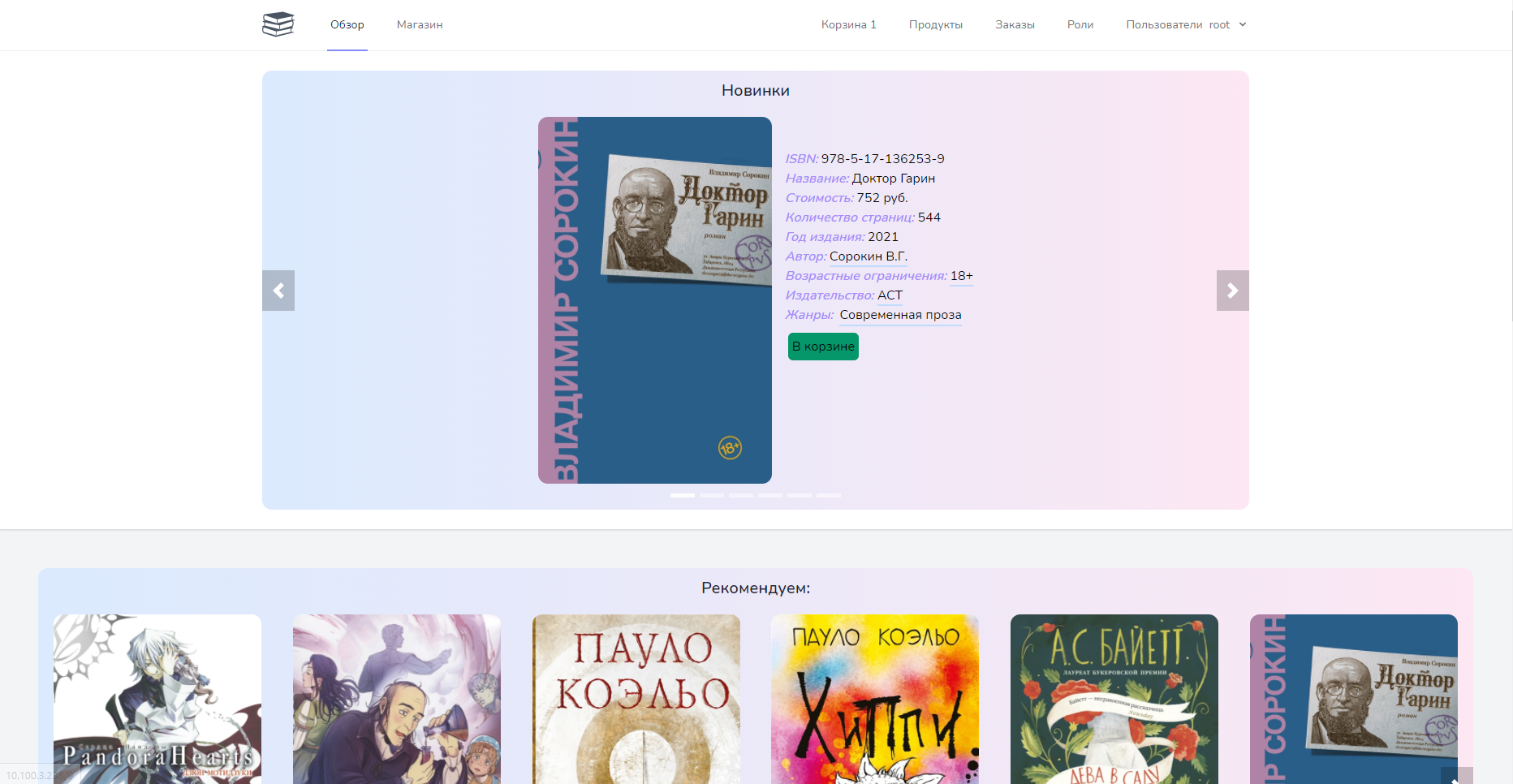


Рисунок 17 – Вкладка «Обзор»

Фреймворк Vanilla JS 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. Фреймворк использует архитектурную модель MVC (Model View Controller) представленную на рисунке 18. Принцип работы, которой необходимо создавать файл контроллера, модель и представления.

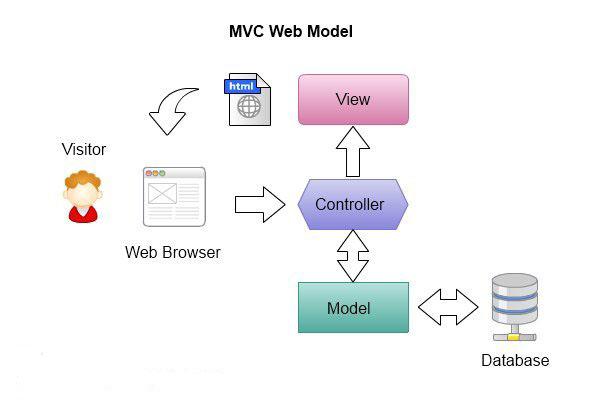


Рисунок 18 – Архитектурная модель MVC

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 19 представлен код функции create из контроллера ProductController.

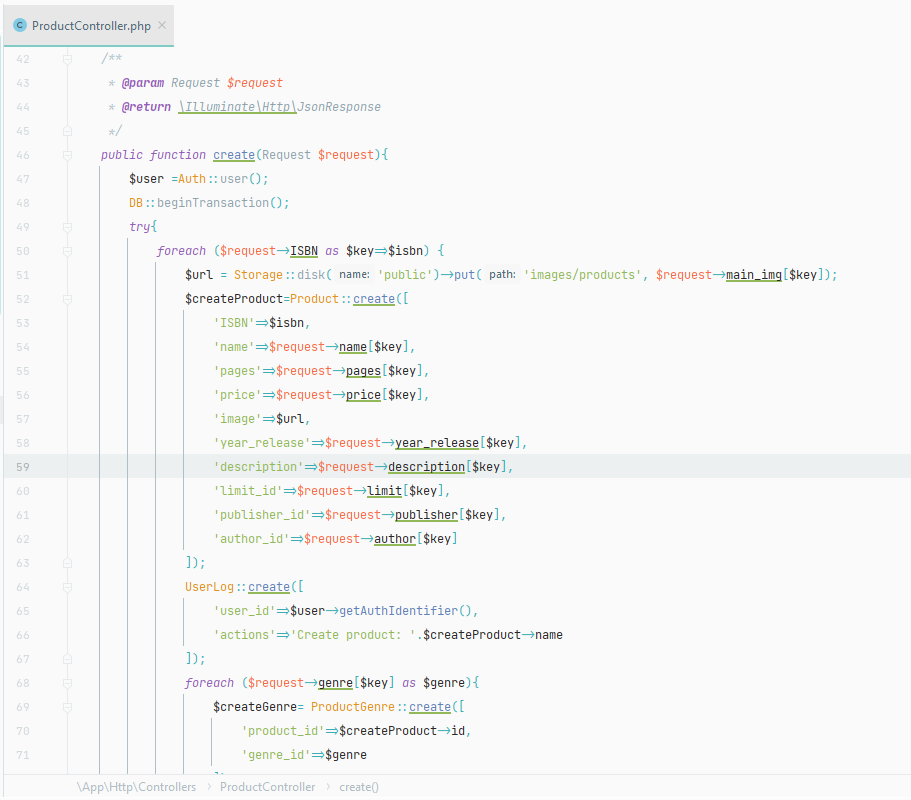


Рисунок 19 – Фрагмент контроллера ProductController

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 20 изображен фрагмент кода этого представления.



Рисунок 20 – 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 21).



Рисунок 21 – Фрагмент функции фильтров

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

1. Внедрение и сопровождение программного обеспечения

Внедрение приложения будет происходить на основе заказа от какого-либо предприятия на соответствующее действие. В процессе внедрения будут предоставлены: приложение и вся необходимая программная документация.

Сопровождение приложения будет происходить по договоренности разработчиков и предприятия.

1. Документирование программного обеспечения
   1. Руководство по установке программного обеспечения

Руководство по установке приложения написано для последующих пользователей для облегчения взаимодействия с приложением.

Установка приложения «Интернет-магазин книг» производится путем копирования репозитория приложения на персональный компьютер или сервер.

Для установки необходимо проделать следующие шаги:

1. Вставьте компакт-диск приложения в дисковод или flash-накопитель, содержащий приложение.
2. Скопируйте репозиторий по необходимому пути.

Нажмите правой кнопкой мыши по папке Bookmarket и выберите «Копировать» (или левой кнопкой мыши и сочетание клавиш Ctrl+C). После чего перейдите в папку, где будет хранится приложение (в примере Program Files) и в свободно месте нажмите правую кнопку мыши и выберите «Вставить» (или сочетание клавиш Ctrl+V). Дождитесь окончания копирования.

Установка приложения завершена успешна.

* 1. 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 22.

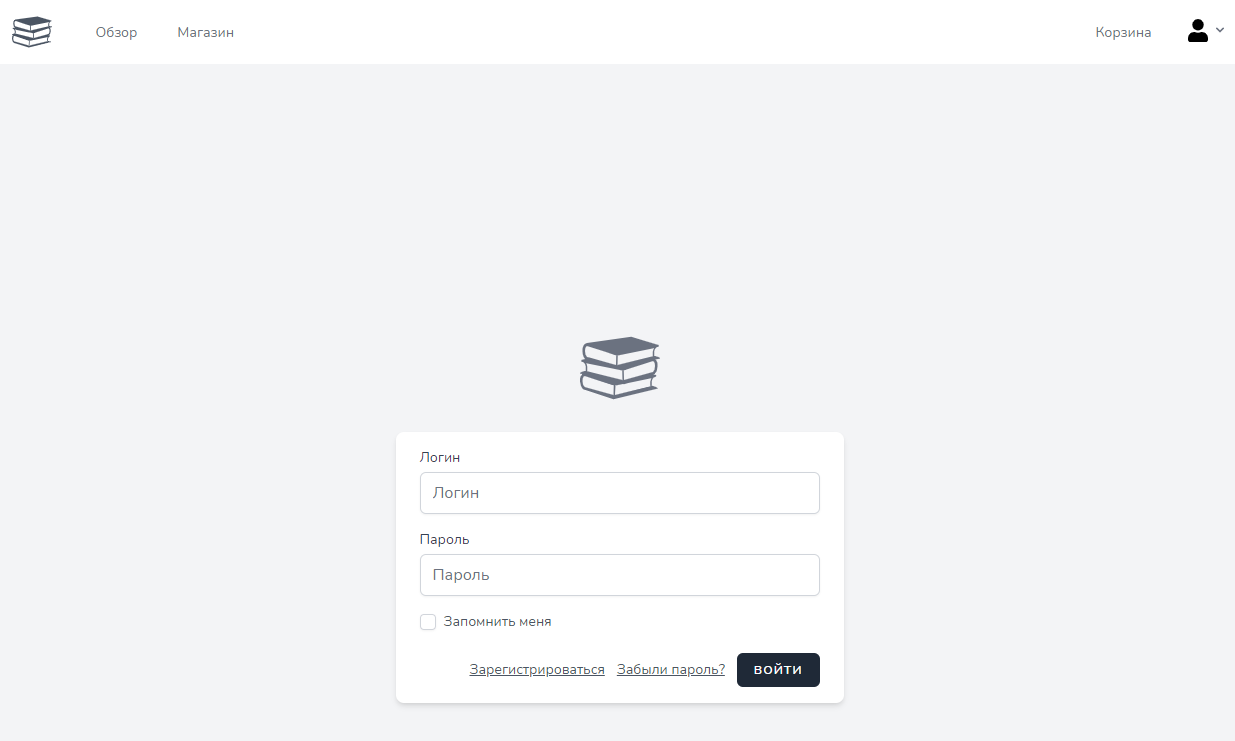


Рисунок 22 – Страница входа

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 2. 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 1 – 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 рисунке23.

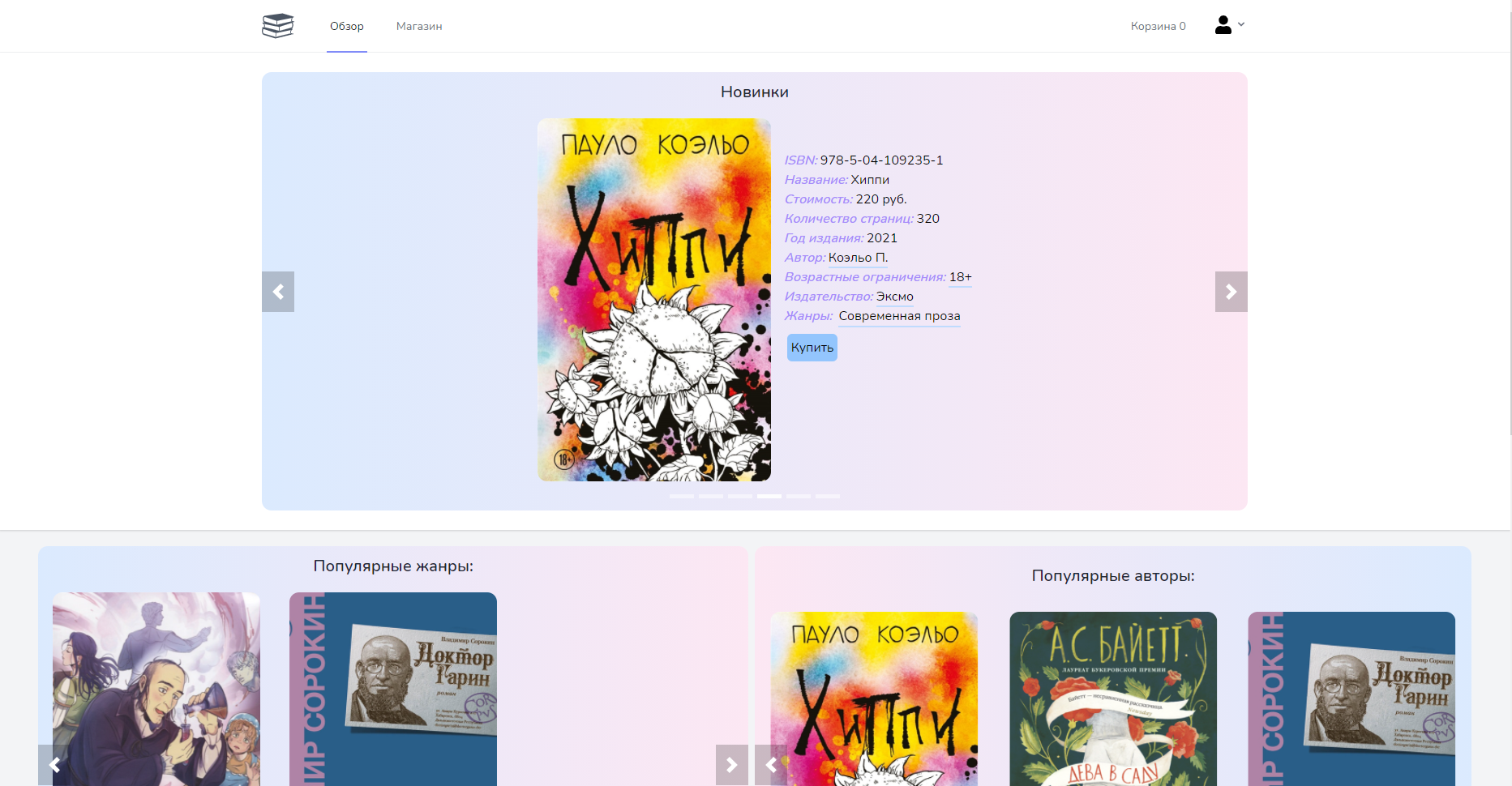


Рисунок 23 – Фрагмент раздела «Обзор»

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 24.

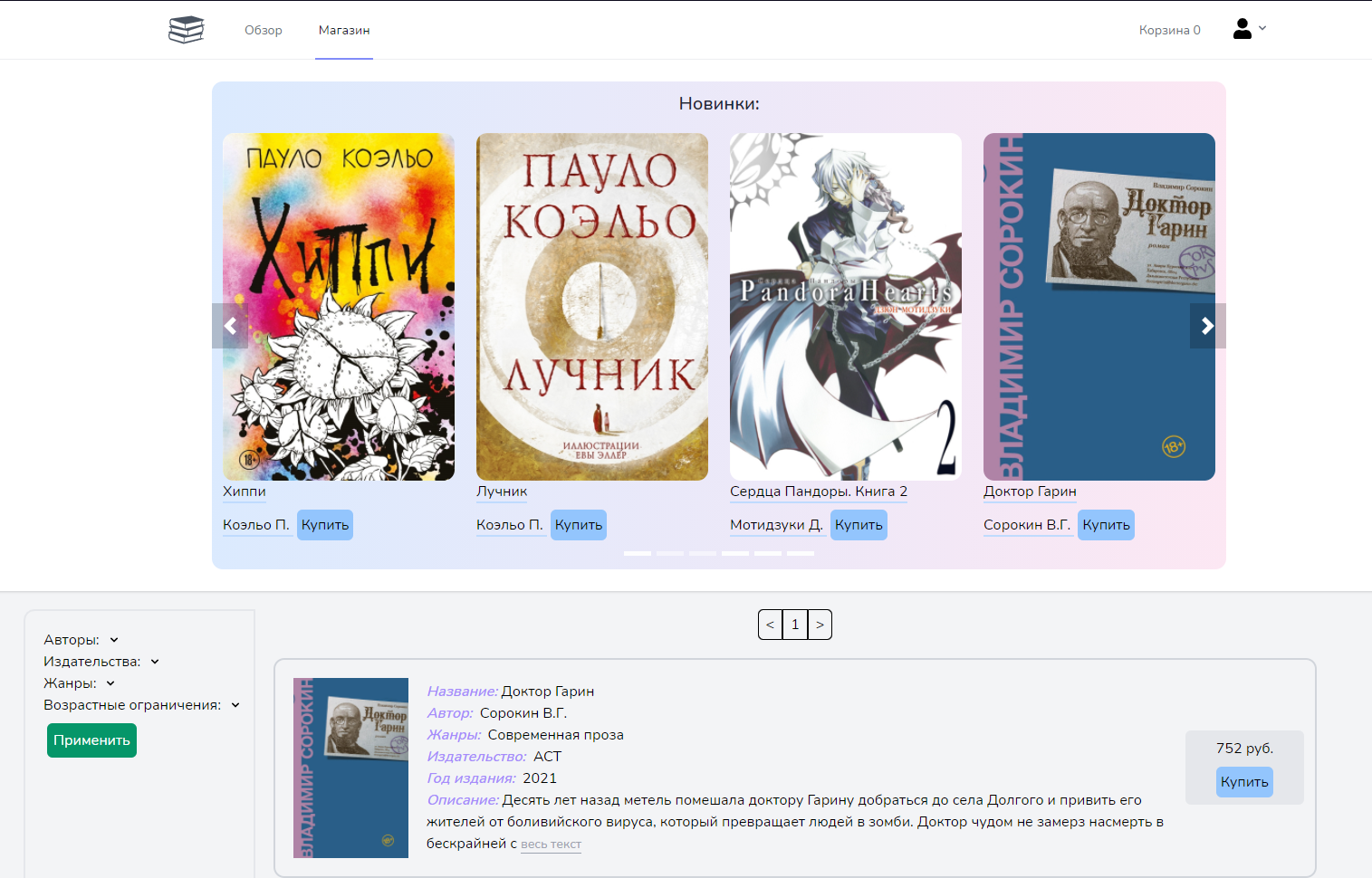


Рисунок 24 – Раздел «Магазин»

Кнопка «Применить» позволяет применить к странице выбранные пользователем фильтры.

* + 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 25.

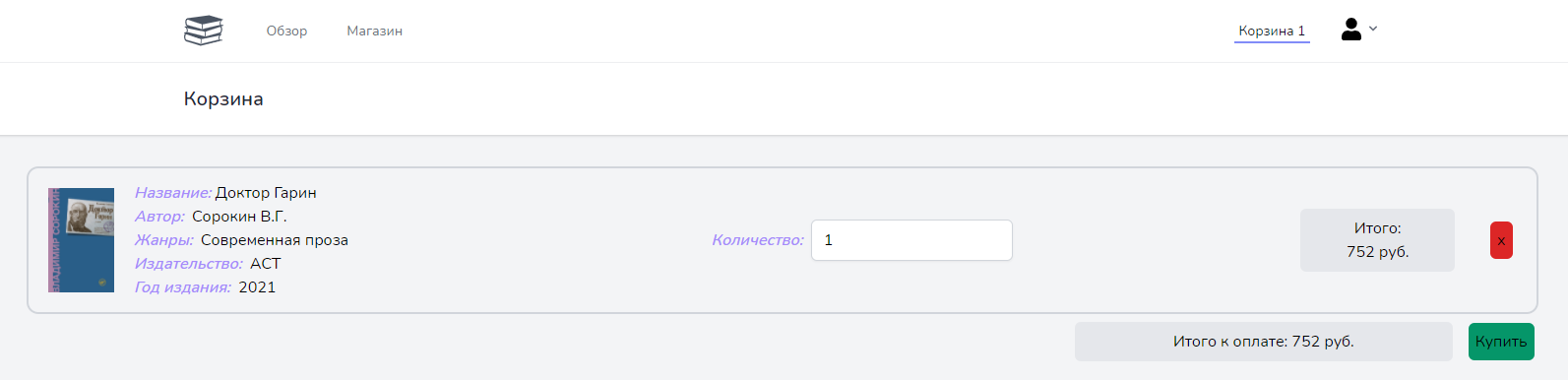


Рисунок 25 –Раздел «Корзина»

Нажав на кнопки «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 «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 пользователя 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 26.

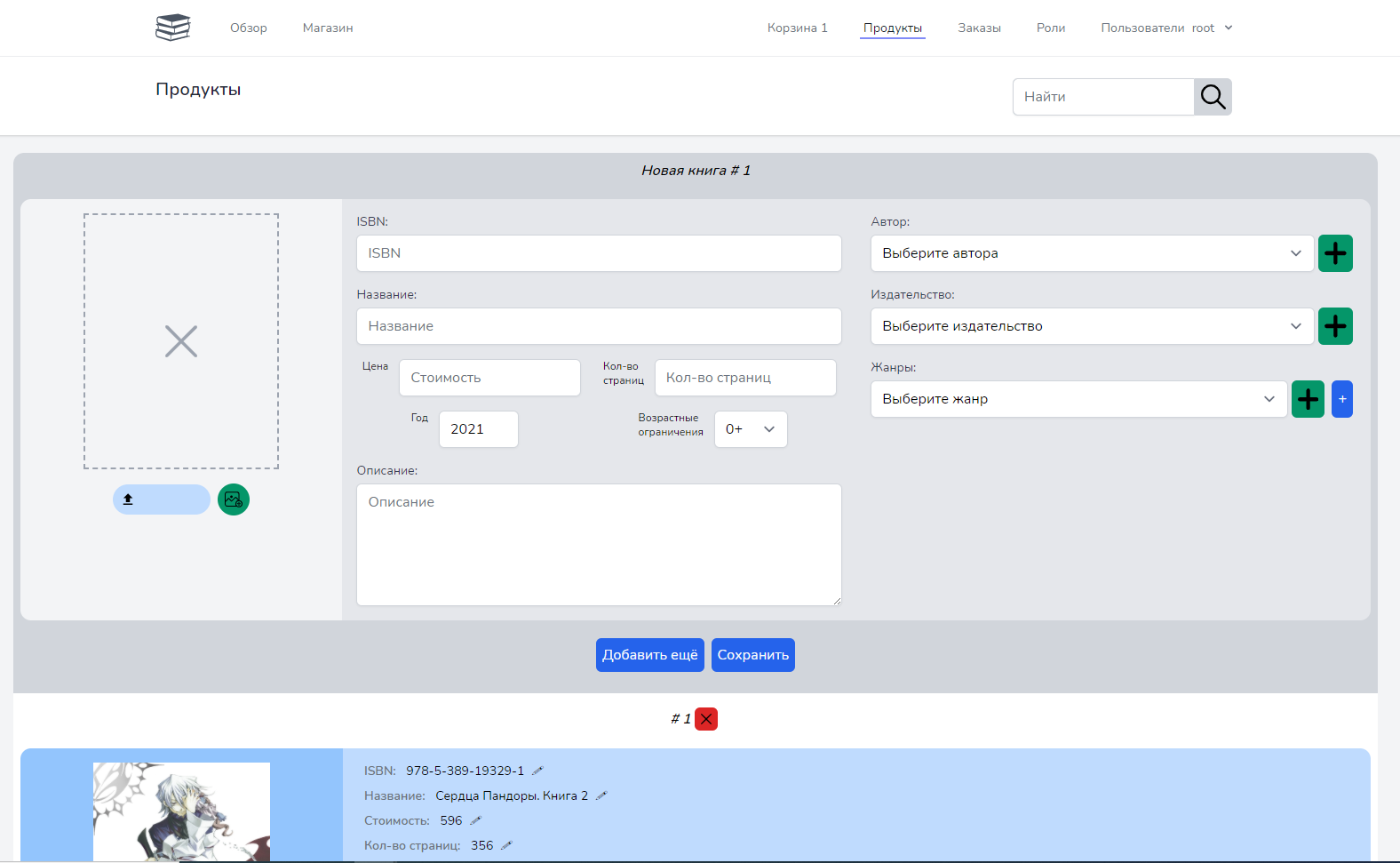


Рисунок 26 – Раздел «Продукты»

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 27).

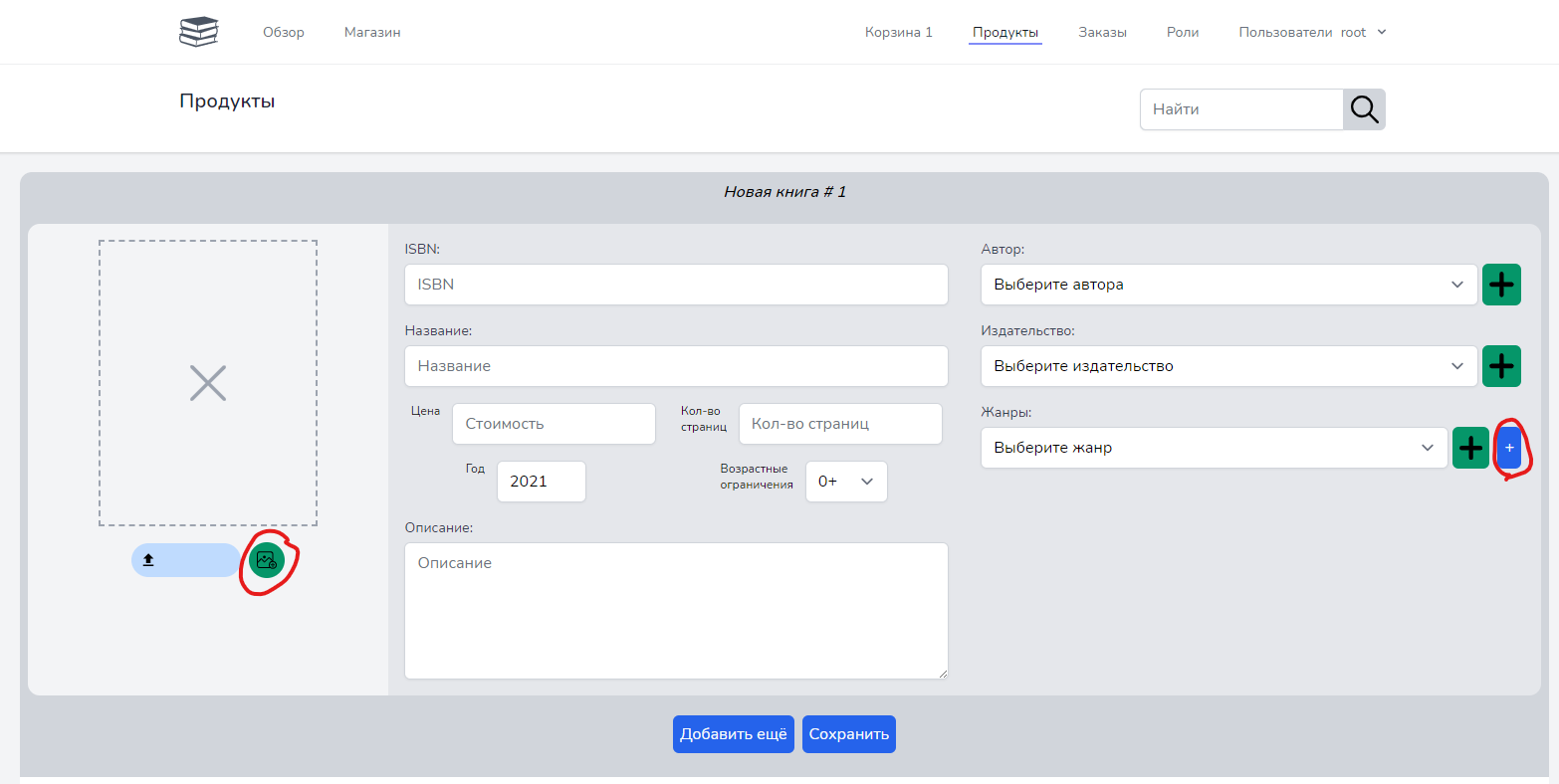


Рисунок 27 – Раздел «Продукты»: Кнопки нескольких жанров и доп. Изображений

Также на странице присутствует поиск, который позволит найти необходимый товар намного быстрее. Поле поиска ищет по всем атрибутам товара.

* + 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 28.

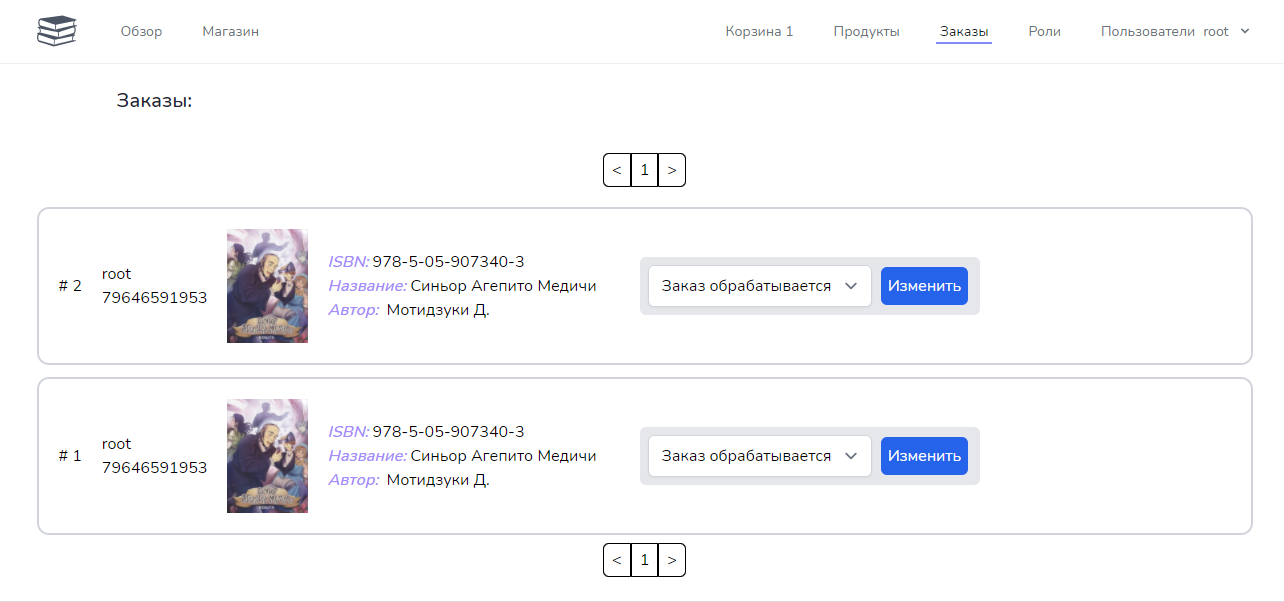


Рисунок 28 – Раздел «Заказы»

* + 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 29.

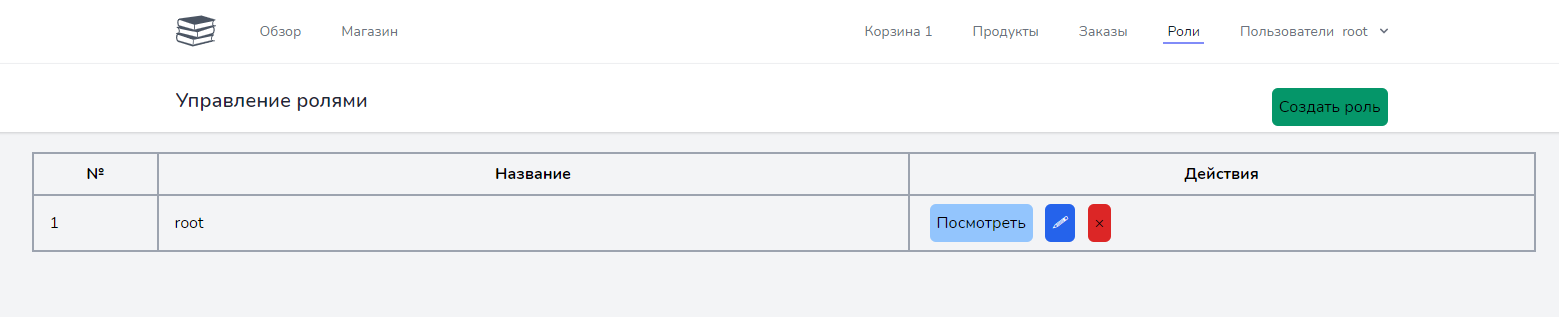


Рисунок 29 – Раздел «Роли»

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 30.

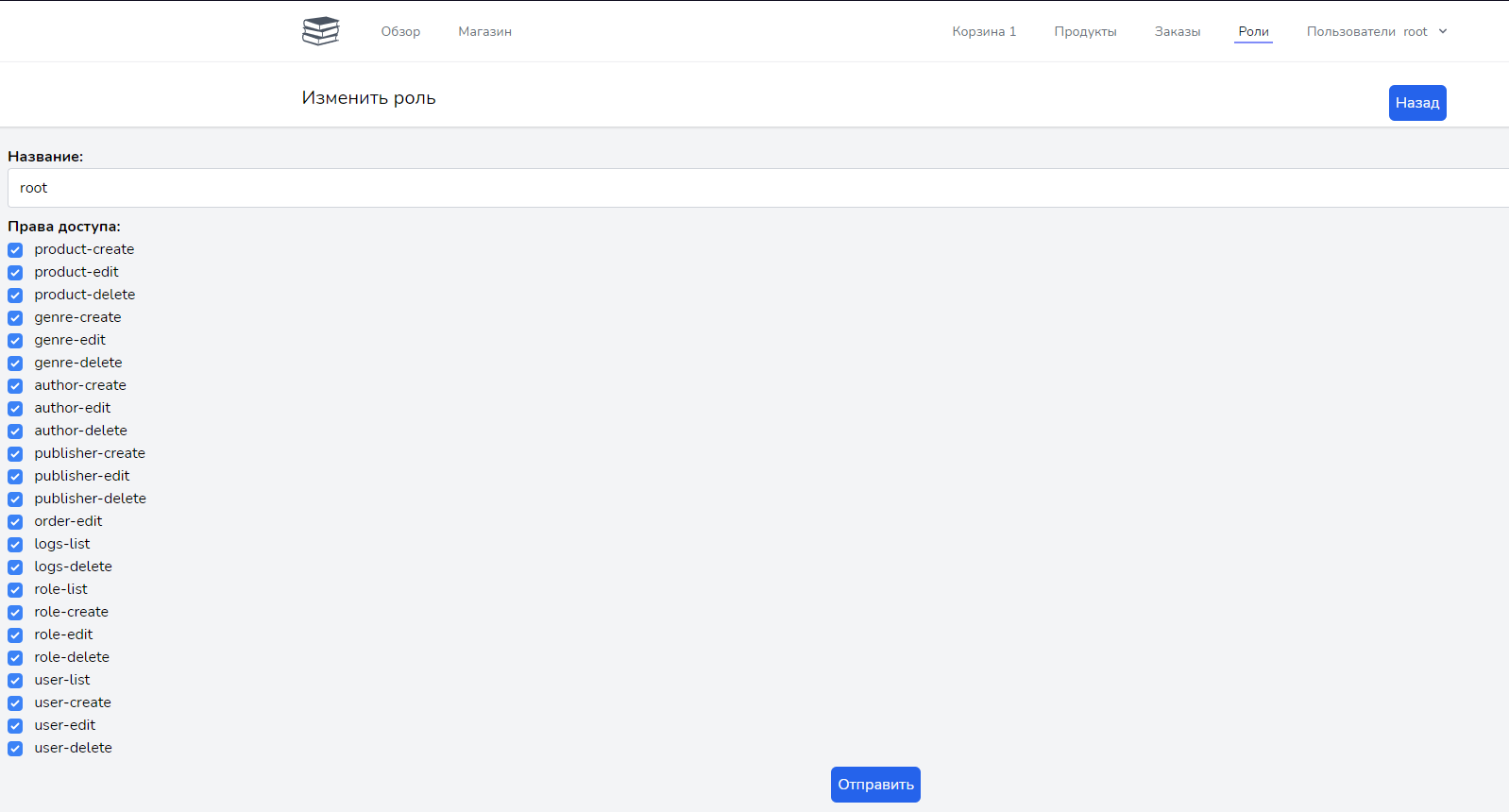


Рисунок 30 – Страница редактирования раздела «Роли»

* + 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. Все действия производятся аналогично разделу «Роли». Внешний вид продемонстрирован на рисунке 31.

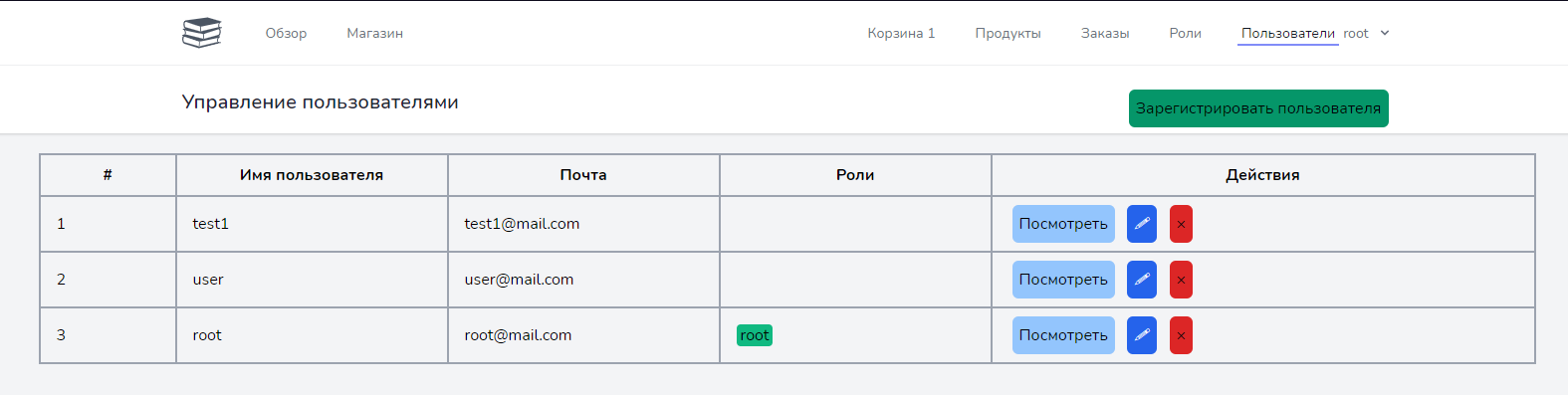


Рисунок 31 – Раздел «Пользователи»

* 1. Руководство системного администратора программного обеспечения

Приложение «Интернет-магазин книг» - это клиент-серверное приложение, которому необходимо иметь веб-сервер.

Дальше будет описываться как настроить веб-сервер при помощи программного обеспечения Xampp.

Из предоставленного вам носителя с приложением «Интернет-магазин книг» скопируйте всё в корень диска (пример D:/).

Если название диска D, то производить какие-либо настройки не нужно. В противном случае, необходимо перейти к файлу с настройкой веб-сервера (пример пути до файла: D:\xampp\apache\conf\extra\httpd-vhosts.conf).

В этом файле в строках 46,49,59 и 60 измените диск D букву своего диска (рисунок 32).



Рисунок 32 – Настройка веб-сервера

После чего запустите xampp\_start.exe папке xampp (рисунок 33). В открывшемся окне запустите apache и mysql (рисунок 34).

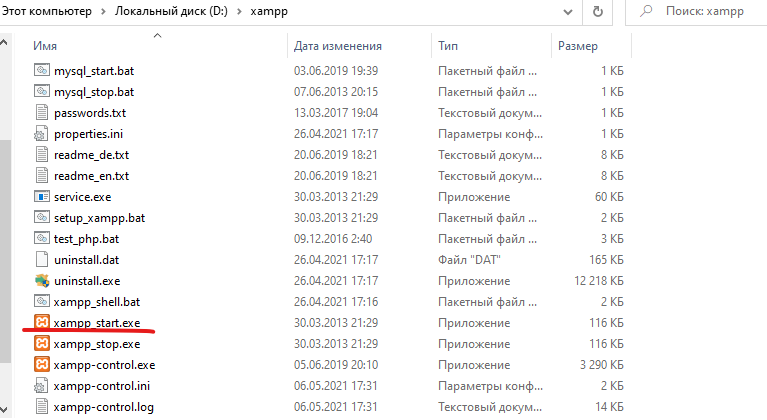


Рисунок 33 – Запуск xampp

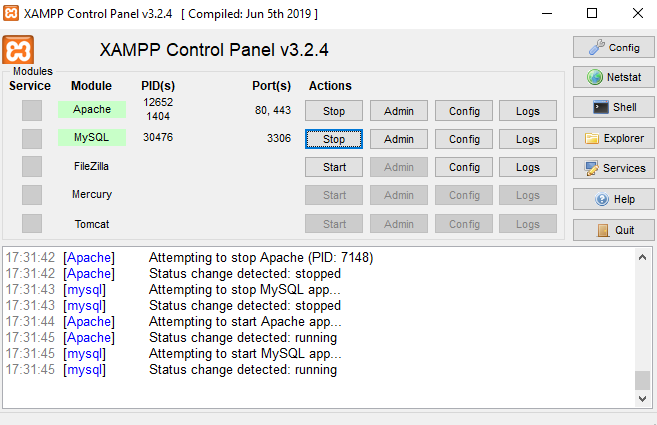


Рисунок 34 – Запуск веб-сервера apache и сервера mysql

После чего в браузере по адресу http://bookmarket запустите приложение «Интернет-магазин книг»

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 автосервисом.

Средняя стоимость аналогов проводится по формуле 6.1:

(6.1)

,

где Ца – среднее стоимость аналогов,

n – количество рассмотренных аналогов,

Зi – стоимость аналога i.

Стоимости рассмотренных аналогов демонстрируется в таблице 3.

Таблица 3 – Стоимость рассмотренных аналогов

|  |  |
| --- | --- |
| Аналог | Стоимость, руб. |
| Chitai-gorod.ru | 0 |
| Books.ru | 0 |
| Labirint | 0 |

Средняя стоимость рассмотренных аналогов равна 0 рублей.

Данное приложение разрабатывается для предприятия, связанного с оказанием услуг продажи книг.

* 1. Расчет затрат на разработку программного продукта

Расчет полных затрат на разработку проектного решения (КРПР) осуществляется по формуле 6.2:

(6.2)

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 4). 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 6.3:

(6.3)

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

Таблица 4 - 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 |

Продолжение таблицы 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 6.4):

(6.4)

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 6.5:

,

(6.5)

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 6.6:

Е=W×t×T,

(6.6)

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 6.7:

(6.7)

где – первоначальная стоимость ПК, руб;

– процент амортизации в год.

При разработке программного продукта использовались следующие свободно распространяемые специальные продукты: Xampp, Bootstrap, PhpStorm, Laravel.

Затраты на хозяйственно-организационные нужды приводятся в таблице 5 и вычисляются по формуле 6.8:

,

(6.8)

где Цτ – цена τ-го товара, руб.;

Кτ – количество τ-го товара.

Таблица 5 - 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 6).

Таблица 6 – 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 | 564,77 |
| Затраты на амортизацию ПК | А | 1250 |
| Затраты на хозяйственно-операционные нужды |  | 1480 |
| Итого затрат на разработку |  | 21170,97 |

* 1. Расчет затрат на внедрение программного продукта

Внедрение приложения в работу предприятия не требует дополнительных затрат, так как все внедряемые программные продукты свободно распространяются.

* 1. Основные выводы

Исходя из расчетов стоимости разработки, можно прийти к выводу, что приложение обойдется предприятию (без учета оплаты труда) в размере 7090,97 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. Bootstrap [Электронный ресурс] – Режим доступа: http://getbootstrap.ru – Bootstrap Самый популярный в мире front-end фреймворк. (Дата обращения: 12.05.21).
2. CyberForum [Электронный ресурс] – Режим доступа: http://www.cyberforum.ru – (Дата обращения: 08.05.21).
3. Draw.io это [Электронный ресурс] – Режим доступа: https://bizzapps.ru/p/draw-io/ – Документация (Дата обращения: 11.05.2021).
4. Laravel по-русски [Электронный ресурс] – Режим доступа: <https://laravel.ru/docs/v5/queries> – Конструктор запросов (Дата обращения: 06.05.21).
5. MySQL Workbench [Электронный ресурс] – Режим доступа: https://dev.mysql.com/doc/workbench/en/ – Документация по MySQL (Дата обращения: 09.05.2021).
6. PHP [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/PHP – PHP (Дата обращения: 10.05.21).
7. PHP.NET [Электронный ресурс] – Режим доступа: https://www.php.net/ – Документация по PHP (Дата обращения: 18.04.2021).
8. PHP-MyAdmin.ru [Электронный ресурс] – Режим доступа: https://php-myadmin.ru/doc/ability.html – Руководство. Возможности PhpMyAdmin (Дата обращения: 03.05.21).
9. PHPStorm [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/PHPStorm – PHPStorm (Дата обращения: 11.05.21).
10. SQL. Полное руководство. Грофф Дж. Р., Вайнберг П.Н., Оппелъ Э. Дж. 2015 – 959 с.
11. Tailwindcss [Электронный ресурс] – Режим доступа: https://tailwindcss.com – (Дата обращения: 14.05.21).
12. Базы данных [Электронный ресурс] – Режим доступа: https://life-prog.ru – Документация (Дата обращения: 23.04.2021).
13. Передача значений переменных из JavaScript в PHP и наоборот [Электронный ресурс] – Режим доступа: https://webformyself.com/peredacha-znachenij-peremennyx-iz-javascript-v-php-i-naoborot/ – Документация (Дата обращения: 29.04.2021).
14. Что такое PHP? [Электронный ресурс] – Режим доступа: https://www.php.net/manual/ru/intro-whatis.php – Что такое PHP? (Дата обращения: 01.05.21).
15. Электронный учебник: JavaScript [Электронный ресурс] – Режим доступа: https://learn.javascript.ru/manuals-specifications – Справочники и спецификации (Дата обращения: 02.05.2021).

# 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');