

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ БАКАЛАВРИАТА**

09.03.04 Программная инженерия

(код, наименование ОПОП ВО: направление подготовки, направленность (профиль))

«Разработка программно-информационных систем»

Программно-информационная система управления книжным магазином

(название темы)

Дипломный проект

(вид ВКР: дипломная работа или дипломный проект)

Автор ВКР

(подпись, дата)

Д. Н. Рязанцев

(инициалы, фамилия)

Группа ПО-116

Руководитель ВКР

(подпись, дата)

Е. П. Кочура

(инициалы, фамилия)

Нормоконтроль

(подпись, дата)

А. А. Чаплыгин

(инициалы, фамилия)

ВКР допущена к защите:

Заведующий кафедрой

(подпись, дата)

А. В. Малышев

(инициалы, фамилия)

Курск 2025 г.

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

УТВЕРЖДАЮ:

Заведующий кафедрой

(подпись, инициалы, фамилия)

« ____ » _____ 20 ____ г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ ПО ПРОГРАММЕ БАКАЛАВРИАТА

Студента Рязанцева Д. Н., шифр 21-06-0011, группа ПО-116

1. Тема «Программно-информационная система управления книжным магазином» утверждена приказом ректора ЮЗГУ от «04» апреля 2025 г. № 1696-с.

2. Срок предоставления работы к защите «9» июня 2025 г.

3. Исходные данные для создания программной системы:

3.1. Перечень решаемых задач:

1) проанализировать функциональные требования к системе книжного магазина;

2) разработать концептуальную модель системы управления книжным магазином на основе подхода к организации онлайн-торговли;

3) спроектировать программную систему управления книжным магазином;

4) сконструировать и протестировать программную систему управления книжным магазином.

3.2. Входные данные и требуемые результаты для программы:

1) Входными данными для программной системы являются: данные из справочников, включающих информацию о книгах (название, цена, описа-

ние, количество на складе), авторах (имя, идентификатор), жанрах для классификации книг; данные о пользователях, такие как имя, адрес электронной почты, пароль и роль (клиент, сотрудник, администратор), данные о заказах, включающие список книг, их количество и сумму заказов.

2) Выходными данными для программной системы являются: сформированные заказы с деталями (книги, сумма, статус); обновленный каталог книг (добавление, редактирование, удаление); статусы заказов и их история.

4. Содержание работы (по разделам):

4.1. Введение.

4.1. Анализ предметной области.

4.2. Техническое задание: основание для разработки, цели и назначение разработки, требования к программной системе, требования к оформлению документации.

4.3. Технический проект: общая характеристика организации решения задач, обоснование выбора технологии проектирования, проектирование архитектуры программной системы, проектирование пользовательского интерфейса программной системы.

4.4. Рабочий проект: модули программной системы, тестирование программной системы, сборка компонентов программной системы.

4.5. Заключение.

4.6. Список использованных источников.

5. Перечень графического материала:

Лист 1. Сведения о ВКРБ.

Лист 2. Цель и задачи разработки.

Лист 3. Концептуальная модель сайта.

Лист 4. Еще плакат.

Руководитель ВКР

(подпись, дата)

Е. П. Кочура

(инициалы, фамилия)

Задание принял к исполнению

(подпись, дата)

Д. Н. Рязанцев

(инициалы, фамилия)

РЕФЕРАТ

Объем работы равен 85 страницам. Работа содержит 22 иллюстрации, 10 таблиц, 15 библиографических источников и 4 листа графического материала. Количество приложений – 2. Графический материал представлен в приложении А. Фрагменты исходного кода представлены в приложении Б.

Перечень ключевых слов: интернет-магазин, веб-приложение, база данных, Flask, JavaScript, HTML, CSS, PostgreSQL, аутентификация, авторизация, каталог книг, заказы, пользователи, REST API.

Объектом разработки является программно-информационная система управления книжным магазином, реализованная в виде веб-приложения с клиент-серверной архитектурой.

Целью выпускной квалификационной работы является разработка веб-приложения для автоматизации процессов управления книжным магазином, включая просмотр каталога, оформление заказов и администрирование данных.

В процессе разработки были определены основные сущности системы (книги, авторы, жанры, пользователи, заказы), спроектирована реляционная база данных PostgreSQL, реализованы серверная часть на Python с использованием Flask и клиентская часть с применением HTML, CSS и JavaScript. Система поддерживает функционал для трех ролей пользователей: покупатель, сотрудник, администратор. Разработан адаптивный пользовательский интерфейс, обеспечивающий удобство работы на различных устройствах. Проведено тестирование системы для подтверждения ее работоспособности.

ABSTRACT

The volume of work is 85 pages. The work contains 22 illustrations, 10 tables, 15 bibliographic sources and 4 sheets of graphic material. The number of applications is 2. The graphic material is presented in annex A. The layout of the site, including the connection of components, is presented in annex B.

List of keywords: online store, web application, database, Flask, JavaScript, HTML, CSS, PostgreSQL, authentication, authorization, book catalog, orders, users, REST API.

Object of development: a software-information system for managing a bookstore, implemented as a web application with a client-server architecture.

Purpose of the final qualifying work: to develop a web application for automating bookstore management processes, including browsing the catalog, placing orders, and administering data.

Development process: the main system entities (books, authors, genres, users, orders) were defined, a relational PostgreSQL database was designed, the server-side was implemented using Python with Flask, and the client-side was developed using HTML, CSS, and JavaScript. The system supports functionality for three user roles: customer, employee, and administrator. An adaptive user interface was created to ensure usability across various devices. System testing was conducted to confirm its functionality.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	10
1 Анализ предметной области	12
1.1 Понятие и ключевые аспекты интернет-торговли	12
1.1.1 Интернет-магазины как ключевая форма интернет-торговли	14
1.1.2 Интернет-торговля в современном бизнесе	15
1.1.3 Интернет-торговля для потребителя	16
1.2 История интернет-торговли	17
1.2.1 Мировая история развития интернет-торговли	18
1.2.2 Развитие интернет-торговли в России	18
1.2.3 Влияние пандемии COVID-19 на интернет-торговлю	19
1.3 Интернет-торговля книгами	20
2 Техническое задание	22
2.1 Основание для разработки	22
2.2 Цель и назначение разработки	22
2.3 Требования к программной системе	22
2.3.1 Требования к данным программной системы	22
2.3.2 Функциональные требования к программной системе	24
2.3.2.1 Вариант использования «Просмотр каталога книг»	28
2.3.2.2 Вариант использования «Поиск книг»	29
2.3.2.3 Вариант использования «Просмотр деталей книги»	29
2.3.2.4 Вариант использования «Добавление книги в корзину авторизованным пользователем»	29
2.3.2.5 Вариант использования «Управление корзиной»	30
2.3.2.6 Вариант использования «Оформление заказа»	31
2.3.2.7 Вариант использования «Просмотр истории заказов»	31
2.3.2.8 Вариант использования «Просмотр подробностей заказа»	31
2.3.2.9 Вариант использования «Просмотр истории заказов сотрудником или администратором»	32

2.3.2.10	Вариант использования «Просмотр подробностей заказа сотрудником или администратором»	32
2.3.2.11	Вариант использования «Редактирование информации о книге сотрудником или администратором»	33
2.3.2.12	Вариант использования «Удаление книги сотрудником или администратором»	33
2.3.2.13	Вариант использования «Добавление книги сотрудником или администратором»	34
2.3.2.14	Вариант использования «Регистрация пользователя»	35
2.3.2.15	Вариант использования «Авторизация пользователя»	35
2.3.3	Требования к интерфейсу	36
2.3.4	Нефункциональные требования к программной системе	36
2.3.4.1	Требования к надежности	36
2.3.4.2	Требования к программному обеспечению	37
2.3.4.3	Требования к аппаратному обеспечению	37
2.4	Требования к оформлению документации	38
3	Технический проект	39
3.1	Общая характеристика организации решения задачи	39
3.2	Обоснование выбора технологии проектирования	40
3.2.1	Язык программирования Python	40
3.2.2	Язык программирования JavaScript	41
3.2.3	Интерфейс фреймворка Flask	42
3.2.4	Графический интерфейс с использованием HTML и CSS	42
3.3	Архитектура программной системы	43
3.3.1	Компоненты системы	44
3.3.1.1	Клиентская часть	44
3.3.1.2	Серверная часть	45
3.3.1.3	База данных	46
3.4	Проект данных программной системы	46
3.4.1	Описание сущностей	47
3.5	Проектирование пользовательского интерфейса	50

4	Рабочий проект	53
4.1	Модули программной системы	53
4.1.1	Модуль app.py	53
4.1.2	Модуль app.js	57
4.1.3	Модуль index.html	64
4.1.4	Модуль styles.css	65
4.2	Системное тестирование разработанной программной системы	65
4.3	Сборка программной системы	74
	ЗАКЛЮЧЕНИЕ	76
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	76
	ПРИЛОЖЕНИЕ А Представление графического материала	79
	ПРИЛОЖЕНИЕ Б Фрагменты исходного кода программы	84
	На отдельных листах (CD-RW в прикрепленном конверте)	85
	Сведения о ВКРБ (Графический материал / Сведения о ВКРБ.png)	Лист 1
	Цель и задачи разработки (Графический материал / Цель и задачи разработки.png)	Лист 2
	Концептуальная модель сайта (Графический материал / Концептуальная модель сайта.png)	Лист 3
	Еще плакат (Графический материал / Еще плакат.png)	Лист 4

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных.

ИС – информационная система.

ИТ – информационные технологии.

ПО – программное обеспечение.

РП – рабочий проект.

СУБД – система управления базами данных.

ТЗ – техническое задание.

ТП – технический проект.

UML (Unified Modelling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения.

REST (Representational State Transfer) – архитектурный стиль для веб-сервисов.

UI (User Interface) – пользовательский интерфейс.

ВВЕДЕНИЕ

Современные информационные технологии играют ключевую роль в развитии бизнеса, особенно в сфере розничной торговли. Интернет-магазины стали важным инструментом для взаимодействия с клиентами, предоставляя удобный доступ к товарам и услугам в любое время и из любой точки мира. В книжной индустрии онлайн-торговля позволяет не только расширить ассортимент, но и автоматизировать процессы управления, упрощая работу как для покупателей, так и для сотрудников магазина. Разработка веб-приложений для таких задач требует применения современных технологий, обеспечивающих надежность, масштабируемость и удобство использования.

Создание программно-информационной системы для управления книжным магазином позволяет оптимизировать процессы обработки заказов, ведения каталога книг и взаимодействия с клиентами. Такая система предоставляет покупателям возможность просматривать книги, оформлять заказы и отслеживать их статус, а сотрудникам и администраторам — эффективно управлять ассортиментом и данными пользователей.

Цель настоящей работы – разработка веб-приложения для автоматизации управления книжным магазином, обеспечивающего удобный интерфейс для клиентов и функционал для администрирования. Для достижения цели необходимо решить *следующие задачи*:

- провести анализ предметной области интернет-торговли книгами;
- разработать концептуальную модель системы управления книжным магазином;
- спроектировать программную систему;
- реализовать и протестировать веб-приложение с использованием веб-технологий.

Структура и объем работы. Отчет состоит из введения, 4 разделов основной части, заключения, списка использованных источников, 2 приложений. Текст выпускной квалификационной работы равен 85 страницам.

Во введении сформулирована цель работы, поставлены задачи разработки, описана структура работы, приведено краткое содержание каждого из разделов.

В первом разделе на стадии описания технической характеристики предметной области приводится анализ предметной области, включая особенности интернет-торговли книгами и требования к системе.

Во втором разделе на стадии технического задания приводятся требования к разрабатываемой системе.

В третьем разделе на стадии технического проектирования представлен технический проект, включая выбор технологий, проектирование архитектуры и пользовательского интерфейса.

В четвертом разделе приводится список модулей и их методов, использованных при разработке системы, производится тестирование разработанного сайта.

В заключении излагаются основные результаты работы, полученные в ходе разработки.

В приложении А представлен графический материал. В приложении Б представлены фрагменты исходного кода.

1 Анализ предметной области

1.1 Понятие и ключевые аспекты интернет-торговли

Интернет-торговля (или электронная коммерция) – это форма коммерческой деятельности, при которой сделки купли-продажи товаров, услуг или цифровых продуктов осуществляются через интернет с использованием электронных платежных систем и цифровых коммуникационных каналов [1]. Из этого термина можно выделить две основные концепции интернет-торговли:

1. Широкий подход – интернет-торговля рассматривается как синоним электронной коммерции, охватывающей все аспекты бизнеса: маркетинг, логистику, платежи, CRM-системы и послепродажное обслуживание.

2. Узкий подход – интернет-торговля трактуется как частный случай электронной коммерции, ограниченный дистанционной продажей товаров через интернет.

В статье «Развитие электронной торговли в Российской Федерации» С.С. Корнева выделяет девять ключевых форм электронной коммерции, классифицируемых по типу взаимодействия участников сделки [2].:

1. B2B (Business-to-Business) – сделки между коммерческими организациями. Включает корпоративные электронные рынки и внутренние системы управления предприятием. На эту модель приходится наибольший объём электронных сделок.

2. B2C (Business-to-Consumer) – продажи товаров и услуг от бизнеса конечным потребителям. К такой форме относятся интернет-магазины (Ozon, Citilink) и разного рода онлайн-услуги (образование, страхование, банкинг).

3. B2G (Business-to-Government) – государство предоставляет бизнесу услуги (налоги, лицензирование, закупки) в электронном формате.

4. C2C (Consumer-to-Consumer) – торговля между частными лицами через онлайн-платформы. Яркими представителями такой интернет-торговли являются доски объявлений (Avito).

5. C2B (Consumer-to-Business) – частные лица предлагают товары/услуги компаниям. К таким, например, относятся краудсорсинговые платформы (Planeta, Kickstarter).

6. C2G (Consumer-to-Government) – частные лица участвуют в сделке с правительственной структуре, производят выплаты по счетам.

7. G2C (Government-to-Consumer) – взаимодействие государства с гражданами через цифровые сервисы, например, выплаты пособий (Госуслуги).

8. G2B (Government-to-Business) – государство предоставляет бизнесу услуги (налоги, лицензирование) в электронном формате.

9. G2G (Government-to-Government) – электронный документооборот между госучреждениями.

На данный момент доминирующими формами являются B2B и B2G [2]. Интернет-торговля обладает рядом значительных преимуществ по сравнению с традиционной розничной торговлей. Эти преимущества касаются как бизнеса, так и потребителей, а также способствуют развитию экономики в целом.

Для бизнеса можно выделить следующие основные выгоды:

1. Глобальный охват рынка: интернет-магазин доступен круглосуточно из любой точки мира, что позволяет привлекать клиентов за пределами локального рынка.

2. Снижение операционных затрат: отсутствие расходов на аренду торговых площадей, коммунальные услуги и большое количество персонала. Автоматизация процессов (учёт товаров, обработка заказов, CRM-системы) сокращает издержки.

3. Упрощение логистики и автоматизация: интеграция с курьерскими службами и маркетплейсами ускоряет доставку.

Для потребителей:

1. Удобство и экономия времени: покупки в любое время суток без необходимости посещения магазинов, доступ к широкому ассортименту товаров и сравнение цен в несколько кликов.

2. Возможность возврата и гарантийного обслуживания: законодательство о дистанционной торговле защищает права покупателей (например, право на возврат в течение 7 дней) [3], упрощённые процедуры обмена и возврата через онлайн-поддержку.

3. Доступ к скидкам и акциям: купоны, кэшбэк, программы лояльности и подписки (например, Wildberries Premium), алгоритмы персонализированных предложений на основе истории покупок.

Для экономики и общества:

- Стимулирование цифровизации: развитие ИТ-инфраструктуры, платёжных систем и логистических сервисов, создание новых рабочих мест (разработчики, маркетологи, курьерские службы).

- Поддержка малого бизнеса: низкий порог входа для стартапов (можно начать с маркетплейсов или соцсетей), доступ к инструментам продвижения (таргетированная реклама, SEO).

Интернет-торговля предлагает выгоды для всех участников рынка. Её развитие продолжает трансформировать розничную торговлю, делая её более эффективной и клиентоориентированной.

1.1.1 Интернет-магазины как ключевая форма интернет-торговли

Интернет-магазин — это виртуальная торговая площадка, функционирующая на базе интернет-технологий, где покупатели могут выбирать, заказывать и оплачивать товары или услуги онлайн. В отличие от традиционной розничной торговли, интернет-магазин не требует физического присутствия покупателя и продавца в одном месте, что значительно расширяет географию продаж и снижает издержки.

Ключевые особенности интернет-магазинов:

- электронная витрина — веб-сайт с каталогом товаров, подробными описаниями, изображениями и ценами;

- торговая система — автоматизированные процессы оформления заказа, оплаты и доставки;

– корзина покупок — функционал, позволяющий пользователю собирать выбранные товары и редактировать заказ перед оплатой.

Интернет-магазины представляют собой динамично развивающийся сегмент электронной коммерции, сочетающий технологические инновации с удобством для потребителей. Их дальнейший рост будет зависеть от адаптации к изменяющимся рыночным условиям и внедрения новых цифровых решений [4].

1.1.2 Интернет-торговля в современном бизнесе

Интернет-торговля кардинально изменила подходы к ведению бизнеса, создав новые возможности для компаний различных масштабов. Её роль в современной экономике невозможно переоценить, так как она позволяет выходить на глобальные рынки, оптимизировать издержки и выстраивать эффективные коммуникации с потребителями [5].

Одним из ключевых преимуществ интернет-торговли для бизнеса является доступ к международным рынкам без необходимости открытия физических представительств. Российские компании, такие как Wildberries и Ozon, успешно используют этот потенциал, расширяя своё присутствие за рубежом. Кроме того, онлайн-формат торговли значительно снижает операционные затраты за счёт экономии на аренде помещений, содержании персонала и других традиционных статьях расходов. Автоматизация бизнес-процессов с помощью CRM и ERP-систем дополнительно повышает эффективность управления.

Гибкость интернет-торговли позволяет компаниям быстро адаптироваться к изменениям спроса, тестировать новые продукты и масштабировать бизнес с минимальными рисками. Персонализация взаимодействия с клиентами, основанная на анализе данных, даёт возможность предлагать индивидуальные решения и повышать лояльность аудитории. Круглосуточная доступность онлайн-магазинов обеспечивает стабильный поток продаж без временных ограничений.

Влияние интернет-торговли на экономику проявляется в её растущем вкладе в ВВП. Согласно исследованиям, объём рынка электронной коммерции в России увеличился с 1,7 трлн рублей в 2017 году до 8 трлн рублей в 2020 году [6]. Этот рост сопровождается созданием новых рабочих мест в сферах логистики, ИТ и цифрового маркетинга, а также стимулированием малого бизнеса, для которого онлайн-платформы стали доступным инструментом выхода на рынок.

Однако развитие интернет-торговли сопряжено с рядом вызовов. Высокая конкуренция, особенно со стороны международных гигантов, таких как Amazon и Aliexpress, требует от компаний инновационных решений. Логистические сложности, особенно в удалённых регионах, и недостаточная развитость инфраструктуры также остаются серьёзными барьерами. Правовые риски, связанные с отсутствием единого регулирования, и угрозы кибербезопасности добавляют неопределённости в работу онлайн-бизнеса.

Перспективы развития интернет-торговли связаны с внедрением передовых технологий, таких как дополненная и виртуальная реальность для виртуальных примерок, а также голосовая коммерция. Важную роль играет государственная поддержка, включая меры по легализации отрасли, развитию цифровой инфраструктуры и созданию благоприятных налоговых условий. Таким образом, интернет-торговля продолжает трансформировать бизнес-среду, предлагая компаниям инструменты для роста и повышения конкурентоспособности. Для устойчивого развития отрасли необходимо решать существующие задачи, включая вопросы регулирования, логистики и технологий, чтобы максимально реализовать её потенциал в экономике будущего [5].

1.1.3 Интернет-торговля для потребителя

Интернет-торговля кардинально изменила потребительское поведение, предоставив покупателям новые возможности и изменив их подход к совершению покупок. Как отмечает Воскресенская О.В., онлайн-магазины стали не просто альтернативой традиционной розничной торговле, а основным каналом удовлетворения потребностей современного потребителя [7].

Онлайн-площадки обеспечивают потребителям круглосуточный доступ к товарам и услугам, позволяя совершать покупки в любое время и из любой точки мира. Это особенно важно в условиях глобализации, когда покупатели могут выбирать товары не только у локальных продавцов, но и у международных поставщиков [8].

Интернет-магазины предлагают значительно более широкий ассортимент по сравнению с физическими магазинами. Потребители могут легко сравнивать цены, читать отзывы и изучать характеристики товаров перед покупкой.

Просмотр товаров онлайн стал формой развлечения, что увеличивает частоту покупок, а изобилие аналогов учит потребителей более внимательно оценивать качество и стоимость товаров.

Интернет-торговля трансформировала потребительское поведение, сделав покупки более удобными, но и более сложными с точки зрения контроля расходов. Её дальнейшее развитие потребует баланса между удобством, прозрачностью и защитой прав покупателей.

1.2 История интернет-торговли

Интернет-торговля зародилась задолго до появления современных онлайн-магазинов. Первые шаги были связаны с электронным обменом данными между компаниями, но настоящий прорыв произошел с распространением интернета.

Сначала онлайн-продажи ограничивались простыми товарами, такими как книги и техника. Постепенно компании начали разрабатывать удобные платформы, системы оплаты и доставки, что сделало интернет-торговлю доступной для массового потребителя.

Со временем ассортимент расширился до практически любых товаров и услуг, а технологии персонализации и маркетинга позволили делать покупки в интернете еще удобнее. Сегодня интернет-торговля — это глобальная индустрия, которая продолжает развиваться, внедряя новые технологии, такие как мобильные платежи, искусственный интеллект и быструю доставку.

1.2.1 Мировая история развития интернет-торговли

Развитие интернет-торговли представляет собой многогранную историю технологического прогресса, которая коренным образом изменила мировую экономику и потребительские привычки. Ее становление можно проследить через несколько ключевых этапов.

Первые шаги электронной коммерции были сделаны еще в 1960 году, когда компании IBM и American Airlines разработали инновационную систему для бронирования авиабилетов. Этот проект стал прообразом современных онлайн-транзакций. В 1970-1980-х годах началось активное внедрение систем электронного обмена данными (EDI), что позволило автоматизировать бизнес-процессы в логистике и оптовой торговле. Важным рубежом стал 1989 год, когда интернет приобрел современные черты с появлением стандартов HTTP, что открыло новые возможности для онлайн-коммерции [9].

Настоящий прорыв произошел в 1990-х годах, когда в США были сняты ограничения на коммерческое использование интернета. Это десятилетие ознаменовалось появлением пионеров электронной коммерции: в 1994 году была создана первая платежная система CyberCash, в 1995 году Джефф Безос основал Amazon, начавший с продажи книг и превратившийся в крупнейший мировой маркетплейс. Тогда же появился eBay, революционизировавший торговлю через онлайн-аукционы. Эти платформы заложили основы современных моделей B2C и C2C [9].

1.2.2 Развитие интернет-торговли в России

В России интернет-торговля начала развиваться после распада СССР. Первые сделки в 1990-х осуществлялись по системе МОТО (заказ по телефону с последующей почтовой доставкой). На рубеже веков появились первые российские интернет-магазины, к 2020 году весомый процент продаж приходился на маркетплейсы Ozon и Wildberries.

Современный этап характеризуется доминированием маркетплейсов, объединяющих миллионы продавцов и покупателей. Особое значение при-

обрела мобильная коммерция - по данным исследований, более 60% покупок сейчас совершается через смартфоны. Российские площадки, успешно выходят на международные рынки, демонстрируя глобализацию электронной торговли [10].

1.2.3 Влияние пандемии COVID-19 на интернет-торговлю

Пандемия COVID-19 стала мощным катализатором для ускоренного развития интернет-торговли, коренным образом изменив потребительские привычки и структуру рынка. По данным исследования, в 2020 году российский рынок онлайн-торговли вырос на 57%, достигнув объема 2,7 трлн рублей, при этом наиболее значительный рост (на 172%) наблюдался в сегменте пищевых продуктов. Это было обусловлено вынужденным переходом потребителей на онлайн-шопинг из-за карантинных ограничений [11].

Ключевые изменения, вызванные пандемией:

1. Расширение аудитории:

- прирост пользователей онлайн-торговли составил 10 млн человек, включая ранее неактивную возрастную группу 55+;
- увеличилась частота покупок: 45% респондентов отметили рост повторных заказов.

2. Трансформация логистики:

- бесконтактная доставка (94% спроса);
- самовывоз из магазинов (65%);
- партнерства с курьерскими сервисами и такси для снижения нагрузки на склады.

3. Технологический прорыв:

- 65% компаний столкнулись с перебоями в ИТ-системах, что стимулировало внедрение решений для масштабирования;
- реклама в соцсетях: 20% бизнесов начали использовать ее впервые.

4. Сдвиг в товарных категориях:

- рост спроса на продукты питания и товары первой необходимости;

- снижение продаж одежды и дорогой техники из-за экономической нестабильности.

5. Долгосрочные последствия:

- онлайн-торговля стала дополнять офлайн, а не заменять его: 45% респондентов считают, что физические магазины сохраняют актуальность;
- развитие маркетплейсов как ключевого канала сбыта для производителей.

COVID-19 не только ускорил цифровизацию торговли, но и выявил ее уязвимости, заставив бизнес адаптироваться.

1.3 Интернет-торговля книгами

Интернет-торговля книгами является одним из ключевых направлений развития электронной коммерции. Зарубежный опыт, в частности успех компании Amazon.com, демонстрирует эффективность сочетания инновационных технологий, клиентоориентированного подхода и широкого ассортимента. Amazon.com достигла высоких показателей за счет персонализации сервиса, удобной навигации, системы рекомендаций и быстрой доставки, что позволило ей завоевать доверие миллионов покупателей.

В России данный сегмент также развивается, однако существуют значительные отличия от зарубежных стандартов. Крупнейшие российские книжные интернет-магазины, такие как Ozon, Labirint, Bolero и Книга.ру, пока не могут конкурировать с Amazon.com по уровню сервиса, логистики и масштабам деятельности [12]. Среди основных проблем российского рынка можно выделить:

- ограниченный ассортимент электронных книг;
- высокие тарифы на доставку, особенно в регионы;
- низкую скорость выполнения заказов;
- недоверие покупателей к онлайн-продавцам из-за некорректного указания цен и отсутствия товаров в наличии.

Для дальнейшего развития интернет-торговли книгами в России необходимо внедрение современных логистических решений, улучшение каче-

ства обслуживания клиентов и расширение ассортимента, включая цифровые форматы. Опыт Amazon.com может служить ориентиром для российских компаний, стремящихся повысить конкурентоспособность на международном рынке.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки является задание на выпускную квалификационную работу бакалавра.

Полное наименование системы: «Программно-информационная система управления книжным магазином».

2.2 Цель и назначение разработки

Разрабатываемая программно-информационная система предназначена для управления ассортиментом книг, обработки заказов, предоставления удобного интерфейса покупателям и сотрудникам.

Программа ориентирована на предпринимателей, планирующих в будущем выходить на всероссийский рынок.

Задачами данной разработки являются:

1. Создание серверной части на Flask для обработки запросов и взаимодействия с базой данных PostgreSQL.
2. Разработка клиентской части на HTML, CSS и JavaScript для отображения книг, корзины и управления заказами.
3. Реализация системы аутентификации и авторизации пользователей с разными ролями (клиент, сотрудник, администратор).
4. Интеграция механизмов поиска и пагинации для удобного просмотра каталога книг.

2.3 Требования к программной системе

2.3.1 Требования к данным программной системы

На рисунке 2.1 представлена концептуальная модель базы данных данных программной системы в виде UML-диаграммы сущность-связь.



Рисунок 2.1 – Концептуальная модель базы данных

Входными данными системы являются:

- сведения о книгах;
- параметры корзины;
- учётные данные пользователя.

Выходными данными являются:

- идентификаторы оформленных заказов;
- уведомления о статусе заказов.

Программно-информационная система должна обеспечивать хранение и обновление данных о книгах, пользователях и заказах.

2.3.2 Функциональные требования к программной системе

Разрабатываемая программно-информационная система книжного магазина предусматривает три роли пользователей: покупатели, сотрудники и администраторы. Каждой роли доступны определенные функции.

Для всех пользователей:

1. Просмотр каталога книг: отображение списка книг с возможностью сортировки и фильтрации, поиск книг по названию, автору или жанру.
2. Просмотр деталей книги: информация о названии, авторе, цене, наличии на складе и описании.
3. Авторизация и регистрация: вход в систему под своей учетной записью, создание нового аккаунта.

Для зарегистрированных покупателей:

1. Оформление заказов: подтверждение заказа с указанием итоговой суммы.
2. Просмотр истории заказов: отслеживание своих заказов и их текущего статуса.

Для сотрудников:

1. Управление книгами: добавление новых книг в каталог, редактирование информации о существующих книгах, удаление книг из каталога.
2. Управление заказами: просмотр всех заказов, изменение статуса заказов.

Для администраторов:

1. Управление пользователями: назначение ролей, просмотр списка всех пользователей.
2. Полный доступ к функциям сотрудников: все возможности сотрудников, включая управление книгами и заказами.

На рисунке 2.2 в виде диаграммы прецедентов представлены функциональные требования к системе, доступные для всех категорий пользователей.

На рисунке 2.3 представлены дополнительные функциональные требования к системе для авторизованных пользователей.

На рисунке 2.4 представлены дополнительные функциональные требования к системе для пользователей с ролью сотрудника.

На рисунке 2.5 представлены дополнительные функциональные требования к системе для пользователей с ролью администратора.



Рисунок 2.2 – Диаграмма прецедентов для неавторизованных пользователей



Рисунок 2.3 – Диаграмма прецедентов для авторизованных пользователей



Рисунок 2.4 – Диаграмма прецедентов для пользователей с ролью сотрудника



Рисунок 2.5 – Диаграмма прецедентов для пользователей с ролью администратора

2.3.2.1 Вариант использования «Просмотр каталога книг»

Заинтересованные лица и их требования: пользователь желает ознакомиться с ассортиментом книг интернет-магазина.

Предусловие: открыта главная страница сайта.

Постусловие: пользователь видит список книг.

Основной успешный сценарий:

1. Пользователь заходит на сайт.
2. Приложение загружает список книг с пагинацией (по 8 книг на страницу).

2.3.2.2 Вариант использования «Поиск книг»

Заинтересованные лица и их требования: пользователь желает найти конкретную книгу.

Предусловие: открыта главная страница сайта.

Постусловие: пользователь находит или не находит нужную ему книгу.

Основной успешный сценарий:

1. Пользователь пишет полное название книги или её часть в поле поиска.
2. Приложение отправляет запрос к API.
3. Сервер возвращает книги, название которых соответствуют поисковому запросу.
4. Пользователь может переключать страницы с результатами.

2.3.2.3 Вариант использования «Просмотр деталей книги»

Заинтересованные лица и их требования: пользователь, желает узнать подробности о книге.

Предусловие: открыта главная страница сайта.

Постусловие: пользователь видит полную информацию о книге.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку "Подробнее".
2. Приложение отправляет запрос к API для получения полной информации о книге.
3. Открывается модальное окно с деталями: автор, цена, описание, жанры и т.д.
4. Пользователь может закрыть окно или добавить книгу в корзину.

2.3.2.4 Вариант использования «Добавление книги в корзину авторизованным пользователем»

Заинтересованные лица и их требования: пользователь, который авторизован на сайте, желает добавить книгу в корзину.

Предусловие: пользователь авторизован и открыта главная страница сайта.

Постусловие: книга добавлена в корзину.

Основной успешный сценарий:

1. Пользователь нажимает кнопку "В корзину" на карточке книги.
2. Если пользователь не авторизован, появляется уведомление с предложением войти.
3. Книга добавляется в корзину (локальное хранилище и состояние UI обновляются).
4. Кнопка меняется на "В корзине" и становится неактивной.
5. Появляется возможность нажать кнопку "Оформить заказ" и отображается стоимость корзины.

2.3.2.5 Вариант использования «Управление корзиной»

Заинтересованные лица и их требования: авторизованный пользователь желает изменить свою корзину.

Предусловие: в корзине есть хотя бы один товар и открыта главная страница сайта.

Постусловие: состояние корзины обновлено.

Основной успешный сценарий:

1. Пользователь спускается в корзину.
2. Пользователь может увеличить или уменьшить количество выбранных книг.
3. Пользователь может выборочно удалить книги из корзины.
4. Обновляется общая стоимость заказа.
5. Если товара нет в достаточном количестве, отображается предупреждение.
6. При соблюдении условий становится доступна кнопка "Оформить заказ".

2.3.2.6 Вариант использования «Оформление заказа»

Заинтересованные лица и их требования: авторизованный покупатель, который желает заказать книги из корзины.

Предусловие: открыта главная страница сайта, есть книги в корзине и выбрано имеющееся на складе их количество.

Постусловие: заказ создан, корзина очищена.

Основной успешный сценарий:

1. Пользователь нажимает "Оформить заказ".
2. Приложение отправляет на сервер данные о заказе.
3. Сервер проверяет наличие товаров и создает заказ.
4. Пользователь получает уведомление об успешном оформлении с номером заказа и деталями.
5. Корзина очищается.

2.3.2.7 Вариант использования «Просмотр истории заказов»

Заинтересованные лица и их требования: авторизованный покупатель, желает посмотреть на историю заказов.

Предусловие: пользователь авторизован и открыта главная страница сайта

Постусловие: отображается список заказов.

Основной успешный сценарий:

1. Пользователь нажимает "Мои заказы".
2. Приложение загружает список заказов через API.
3. При наличии заказов для каждого отображаются номер, дата, статус, сумма.

2.3.2.8 Вариант использования «Просмотр подробностей заказа»

Заинтересованные лица и их требования: авторизованный покупатель желает подробнее изучить свой заказ.

Предусловие: пользователь авторизован, на главной странице и имеет хотя бы один заказ.

Постусловие: пользователь узнаёт подробности своего заказа.

Основной успешный сценарий:

1. Пользователь нажимает "Мои заказы".
2. Приложение загружает список заказов через API.
3. Пользователь нажимает на один из своих заказов.
4. При наличии этих книг в системе пользователь увидит сколько он заказал определённых книг и их стоимость по отдельности.

2.3.2.9 Вариант использования «Просмотр истории заказов сотрудником или администратором»

Заинтересованные лица и их требования: пользователь авторизованный на аккаунте с ролью сотрудника или администратора желает посмотреть список заказов всех пользователей.

Предусловие: пользователь авторизован, его аккаунт имеет роль сотрудника или администратора и открыта главная страница сайта.

Постусловие: пользователь видит список всех заказов.

Основной успешный сценарий:

1. Пользователь нажимает "Все заказы".
2. Приложение загружает список заказов через API.
3. При наличии заказов для каждого отображаются номер, дата, статус, сумма.

2.3.2.10 Вариант использования «Просмотр подробностей заказа сотрудником или администратором»

Заинтересованные лица и их требования: пользователь авторизованный на аккаунте с ролью сотрудника или администратора желает посмотреть подробности заказа одного из пользователей.

Предусловие: пользователь авторизован, его аккаунт имеет роль сотрудника или администратора и открыта главная страница сайта.

Постусловие: пользователь узнаёт подробности заказа одного из пользователей.

Основной успешный сценарий:

1. Пользователь нажимает "Все заказы".
2. Приложение загружает список заказов через API.
3. Пользователь нажимает на один из заказов.
4. При наличии этих книг в системе пользователь увидит сколько другой пользователь заказал определённых книг и их стоимость по отдельности.

2.3.2.11 Вариант использования «Редактирование информации о книге сотрудником или администратором»

Заинтересованные лица и их требования: пользователь авторизованный на аккаунте с ролью сотрудника или администратора желает изменить информацию об одной из книг.

Предусловие: пользователь авторизован, его аккаунт имеет роль сотрудника или администратора и открыта главная страница сайта.

Постусловие: пользователь изменил информацию о книге.

Основной успешный сценарий:

1. Пользователь нажимает "Редактировать".
2. Пользователь спускается в открывшуюся форму редактирования и вносит требуемые изменения.
3. Пользователь нажимает кнопку "Редактировать" и внесённые изменения сервер записывает в базу данных.

2.3.2.12 Вариант использования «Удаление книги сотрудником или администратором»

Заинтересованные лица и их требования: пользователь авторизованный на аккаунте с ролью сотрудника или администратора желает удалить одну из книг.

Предусловие: пользователь авторизован, его аккаунт имеет роль сотрудника или администратора и открыта главная страница сайта.

Постусловие: пользователь удаляет одну из книг.

Основной успешный сценарий:

1. Пользователь нажимает "Редактировать".
2. Пользователь спускается в открывшуюся форму редактирования и вносит и нажимает кнопку "Удалить книгу".
3. На сайте всплывает уведомление предупреждающее об удалении книги.
4. Пользователь нажимает кнопку "Да" и подтверждает удаление.
5. API отправляет запрос на сервер и удаляет книгу из базы данных.

2.3.2.13 Вариант использования «Добавление книги сотрудником или администратором»

Заинтересованные лица и их требования: пользователь авторизованный на аккаунте с ролью сотрудника или администратора желает добавить в систему новую книгу.

Предусловие: пользователь авторизован, его аккаунт имеет роль сотрудника или администратора и открыта главная страница сайта.

Постусловие: пользователь вносит в базу данных новую книгу и информацию о ней.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку "Добавить книгу" в форме панели сотрудника или администратора.
2. Приложение открывает форму для заполнения данными о книге.
3. Заполнив необходимые поля пользователь нажимает кнопку "Сохранить".
4. Приложение через API загружает книгу и информацию о ней на сервер.
5. Приложение отображает уведомление об успешном добавлении и книга появляется в общем каталоге.

2.3.2.14 Вариант использования «Регистрация пользователя»

Заинтересованные лица и их требования: неавторизованный пользователь желает зарегистрироваться на сайте.

Предусловие: авторизация не была произведена и открыта главная страница сайта.

Постусловие: пользователь создаёт аккаунт, с помощью которого, впоследствии, можно будет производить вход.

Основной успешный сценарий:

1. Пользователь нажимает кнопку "Войти".
2. В появившемся модальном окне пользователь нажимает кнопку "Регистрация".
3. Приложение отображает форму с полями для регистрации.
4. Пользователь заполняет все поля и нажимает "Регистрация"
5. Приложение отправляет данные на сервер через API и создаёт учётную запись с ролью покупателя.
6. Пользователь видит уведомление об успешной регистрации.

2.3.2.15 Вариант использования «Авторизация пользователя»

Заинтересованные лица и их требования: неавторизованный пользователь желает войти в свою учётную запись на сайте.

Предусловие: пользователь не авторизован, учётная запись уже существует и открыта главная страница сайта.

Постусловие: пользователь входит в свою учётную запись и открывает новый функционал.

Основной успешный сценарий:

1. Пользователь нажимает кнопку "Войти".
2. В появившемся модальном окне приложение отображает форму с полями для авторизации.
3. Пользователь вводит данные и нажимает "Войти".

4. Приложение отправляет запрос на сервер через API и проверяет правильность введенных данных.
5. Пользователь видит уведомление об успешной авторизации.

2.3.3 Требования к интерфейсу

Графический интерфейс реализуется с использованием современных веб-технологий HTML, CSS, JavaScript, и предоставляет пользователям интуитивно понятный и удобный интерфейс для взаимодействия с системой.

Обязательные элементы интерфейса:

1. Сетка книг на странице.
2. Шапка с названием и кнопками авторизации и просмотра заказов.
3. Модальные окна регистрации и авторизации пользователей.
4. Корзина для совершения покупок.
5. Модальное окно со списком заказов.
6. Модальные окна с подробной информацией о книгах.
7. Панели сотрудника и администратора для управления магазином.

Дополнительные требования:

1. Интерфейс должен корректно масштабироваться при изменении размеров окна.
2. Четкие контрастные цвета для текста и фона.
3. Отсутствие перегруженности элементами на странице.
4. Интерфейс должен обеспечивать простоту эксплуатации без необходимости использования сторонних инструментов.

2.3.4 Нефункциональные требования к программной системе

2.3.4.1 Требования к надежности

В процессе работы программно-информационной системы управления книжным магазином могут возникать следующие аварийные ситуации:

- потеря доступа к сети Интернет, вызванная сменой типа подключения (Wi-Fi, мобильный интернет) или отсутствием связи в определённой зоне;
- принудительная остановка выполнения приложения в браузере пользователя (например, из-за сбоя оборудования или программного обеспечения);
- сбой в работе сервера, вызванный программными или аппаратными неисправностями.

Для обеспечения надежности серверной части системы рекомендуются следующие меры:

- размещение серверных компонентов на выделенных серверах в сертифицированных дата-центрах с гарантией SLA > 99,8%;
- регулярное обновление операционной системы сервера и обеспечение работы систем резервного копирования;
- наличие источников бесперебойного питания и резервных каналов связи для предотвращения сбоев в работе системы.

2.3.4.2 Требования к программному обеспечению

Для реализации программной системы должны быть использованы следующие языки программирования:

- Python — серверная часть, веб-приложение;
- JavaScript (React) — клиентская часть веб-приложения;
- SQL — язык структурированных запросов для работы с PostgreSQL.

Для работы клиентской части требуется современный веб-браузер с поддержкой HTML5 и CSS3. Для работы серверных компонентов необходима операционная система Windows Server 2019 с установленной PostgreSQL и средой выполнения Python 3.11.

2.3.4.3 Требования к аппаратному обеспечению

Для сервера необходим центральный процессор с количеством ядер от 4 и выше с тактовой частотой не менее 2.0 ГГц. Объем оперативной па-

мента должен составлять не менее 16 ГБ. Требование к скорости интернет-соединения — 50 Мбит/с и выше.

2.4 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Программно-информационная система представляет собой современное веб-приложение, предназначенное для автоматизации управления книжным магазином. Система разработана с использованием модульной архитектуры, что позволяет легко адаптировать её к потребностям малого и среднего бизнеса, а также расширять функционал при необходимости.

В основе системы лежит серверная часть, реализованная на Flask с использованием RESTful API. Это обеспечивает быстрое и удобное взаимодействие между клиентской частью и сервером. Веб-интерфейс построен с использованием HTML, CSS и JavaScript, что делает систему доступной в любом современном браузере. База данных PostgreSQL выступает в качестве хранилища данных, а библиотека psycopg2 используется для взаимодействия с ней.

Система предоставляет удобные возможности для всех категорий пользователей. Покупатели могут просматривать каталог книг, использовать поиск, корзину для оформления заказов и отслеживать их статус. Также доступна страница книги с подробной информацией, включая описание и дополнительные характеристики.

Для сотрудников магазина предусмотрен интерфейс, позволяющий редактировать информацию о книгах, изменять их наличие, а также обновлять статусы заказов покупателей. Администраторы имеют доступ ко всему функционалу сотрудников, а также к управлению ролями пользователей. Это позволяет быстро изменять права доступа, добавлять новых сотрудников или ограничивать доступ к определённым функциям.

Система поддерживает аутентификацию и авторизацию пользователей, обеспечивая безопасное использование. Реализована возможность работы с несколькими ролями: покупатель, сотрудник и администратор, что делает её гибкой и подходящей для различных сценариев использования.

Ключевая особенность системы – её масштабируемость. Архитектура позволяет легко добавлять новые функции, такие как поддержка скидок, интеграция с платёжными системами или создание аналитических отчётов о продажах. Также предусмотрена возможность адаптации интерфейса под мобильные устройства, что обеспечивает доступность для пользователей с разных платформ.

Применение системы эффективно в книжных магазинах, где требуется централизованное управление ассортиментом и заказами. Её также можно модифицировать для других сфер розничной торговли. Перспективы развития включают добавление системы промокодов, расширение функционала аналитики и поддержку мультиязычного интерфейса, что сделает её полезным инструментом для международных пользователей.

Эта система сочетает простоту в использовании и высокую гибкость настройки, предоставляя удобные инструменты для управления книжным магазином.

3.2 Обоснование выбора технологии проектирования

Выбор технологий, языков программирования и архитектурных решений для реализации программно-информационной системы обусловлен совокупностью факторов, направленных на обеспечение высокой гибкости, надёжности и простоты сопровождения программного продукта. Используемые для создания программно-информационной системы языки и технологии отвечают современным практикам разработки, позволяют достичь высокой производительности и отказоустойчивости программы.

3.2.1 Язык программирования Python

В качестве языка программирования для серверной части выбран Python, благодаря его сочетанию выразительности, гибкости и обширной поддержки со стороны сообщества разработчиков. Python — это высокоуровневый, интерпретируемый язык, активно применяющийся как в образова-

тельных, так и в промышленных проектах. Основные причины выбора языка заключаются в следующем:

1. Простой и интуитивно понятный синтаксис значительно сокращает порог вхождения и снижает количество потенциальных ошибок при написании кода. Это особенно важно в условиях ограниченного времени на разработку и тестирование, а также при передаче проекта на сопровождение.

2. Поддержка нескольких парадигм программирования, включая объектно-ориентированную, процедурную и функциональную, делает Python универсальным инструментом. Это позволяет организовать код в соответствии с принципами модульности, инкапсуляции и повторного использования.

3. Обширная стандартная библиотека и внешняя экосистема обеспечивают доступ к готовым модулям для сериализации, построения интерфейса, анализа синтаксических деревьев, многопоточности и многого другого. Это существенно ускоряет разработку и упрощает реализацию сложных функций.

4. Кроссплатформенность языка позволяет запускать приложение на операционных системах Windows, Linux и macOS без необходимости адаптации кода под конкретную платформу. Таким образом, обеспечивается максимальная универсальность и доступность системы для пользователя.

Таким образом, Python представляет собой оптимальное решение для реализации проекта, сочетающее в себе простоту, мощь и гибкость, что делает его незаменимым инструментом в учебных и практических задачах программной инженерии.

3.2.2 Язык программирования JavaScript

JavaScript выбран для клиентской части, так как он является основным языком для динамического взаимодействия в веб-приложениях. JavaScript выполняется непосредственно в браузере, обеспечивая интерактивность без необходимости перезагрузки страницы. Благодаря JavaScript удалось реализовать следующее:

- отправку асинхронных запросов к REST API для получения данных;
- динамическое обновление интерфейса;
- управление локальным хранилищем;
- обработку событий.

JavaScript обеспечивает событийно-ориентированную модель, что позволяет обрабатывать действия пользователя в реальном времени и интегрироваться с серверной частью через REST API.

3.2.3 Интерфейс фреймворка Flask

Flask – это легковесный веб-фреймворк на основе Python, выбранный для реализации серверной части программно-информационной системы. Flask предоставляет минималистичный и гибкий интерфейс для создания RESTful API, что делает его подходящим для обработки HTTP-запросов и управления данными. Flask:

- позволяет системе обрабатывать входящие HTTP-запросы для различных эндпоинтов;
- интегрируется с PostgreSQL через библиотеку psycopg2, что упрощает обработку данных на сервере и их передачу клиенту;
- позволяет обрабатывать исключения, возникающие при выполнении запросов.

Интерфейс Flask обеспечивает маршрутизацию запросов, управление сессиями и интеграцию с внешними библиотеками, что делает его удобным для реализации REST API.

Flask выбран благодаря своей простоте и способности эффективно решать задачи, связанные с созданием REST API. Легковесная архитектура минимизирует накладные расходы, обеспечивая быструю обработку запросов, что критично для веб-приложения с большим количеством пользователей.

3.2.4 Графический интерфейс с использованием HTML и CSS

HTML и CSS используются для создания структуры и оформления пользовательского интерфейса. HTML определяет разметку страниц. CSS за-

даёт стили, такие как цвета, шрифты, расположение элементов и адаптивность для разных устройств. Использование медиазапросов обеспечивает корректное отображение на компьютерах, планшетах и смартфонах. Разделение структуры и оформления упрощает поддержку и обновление интерфейса, обеспечивая интуитивно понятное взаимодействие для пользователей.

Выбор HTML и CSS обоснован их стандартизацией, простотой и способностью создавать адаптивный и функциональный интерфейс, соответствующий требованиям книжного магазина.

3.3 Архитектура программной системы

Архитектура программно-информационной системы управления книжным магазином построена по клиент-серверной модели с использованием RESTful API для обеспечения взаимодействия между компонентами. Система состоит из трёх основных компонентов: клиентской части, серверной части и базы данных. На рисунке 3.1 представлена UML-диаграмма, иллюстрирующая архитектуру системы и взаимосвязи между её компонентами.

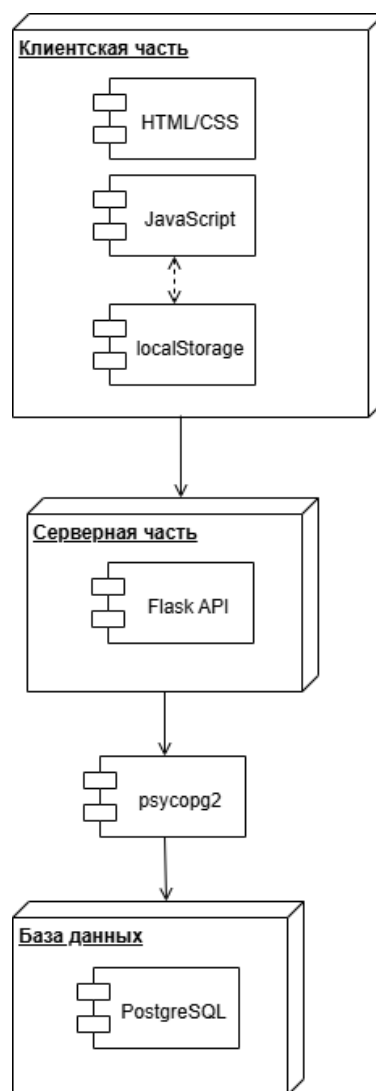


Рисунок 3.1

3.3.1 Компоненты системы

3.3.1.1 Клиентская часть

Клиентская часть реализована как одностраничное веб-приложение, работающее в браузере пользователя. Используемые технологии включают HTML для разметки, CSS для стилизации и JavaScript для динамического взаимодействия. Основные функции клиентской части:

1. Отображение пользовательского интерфейса: каталог книг, корзина, панель администратора, модальные окна для авторизации, регистрации и просмотра деталей книг.

2. Динамическое обновление данных: JavaScript использует Fetch API для асинхронных запросов к серверу, что позволяет обновлять содержимое страницы без перезагрузки.

3. Хранение временных данных: используется localStorage для сохранения содержимого корзины и данных текущего пользователя. Это обеспечивает сохранение состояния приложения между сеансами.

4. Обработка событий: JavaScript обрабатывает действия пользователя, такие как клики по кнопкам, ввод в поисковую строку или изменение количества товаров в корзине.

Клиентская часть взаимодействует с сервером исключительно через REST API, отправляя HTTP-запросы и получая JSON-ответы.

3.3.1.2 Серверная часть

Серверная часть реализована на Python с использованием фреймворка Flask. Сервер выступает в роли REST API, обеспечивая обработку запросов от клиента и взаимодействие с базой данных. Основные функции серверной части:

1. Маршрутизация запросов: Flask использует декораторы для определения эндпоинтов.

2. Обработка данных: сервер принимает JSON-данные из POST-запросов или параметры строки запроса. Данные валидируются, обрабатываются и передаются в SQL-запросы к базе данных.

3. Формирование ответов: сервер возвращает JSON-ответы, используя JSONEncoder для сериализации данных. Ответы включают данные или сообщения об ошибках с соответствующими HTTP-кодами.

4. Интеграция с базой данных: сервер использует библиотеку psycopg2 для взаимодействия с PostgreSQL. Функция устанавливает соединение с базой данных для возврата результатов SQL-запросов в виде словарей.

5. Обработка ошибок: в случае ошибок сервер откатывает транзакцию и возвращает JSON-ответ с описанием ошибки.

3.3.1.3 База данных

База данных: реализована на PostgreSQL, которая хранит данные о книгах, авторах, жанрах, пользователях, заказах и элементах заказов. Основные аспекты:

1. Структура данных: база данных включает таблицы, которые связаны через внешние ключи.
2. Доступ к данным: сервер выполняет SQL-запросы через psycopg2 для операций.

PostgreSQL обеспечивает надёжное хранение данных и поддержку сложных запросов, таких как фильтрация книг по жанрам или авторам.

3.4 Проект данных программной системы

В программной системе книжного магазина используется реляционная СУБД PostgreSQL. Для взаимодействия с базой из Python-приложения применяется библиотека psycopg2.

На клиентской стороне используется localStorage для временного хранения данных корзины и информации о пользователе, что обеспечивает сохранение состояния без синхронизации с сервером.

На рисунке 3.2 представлена модель базы данных

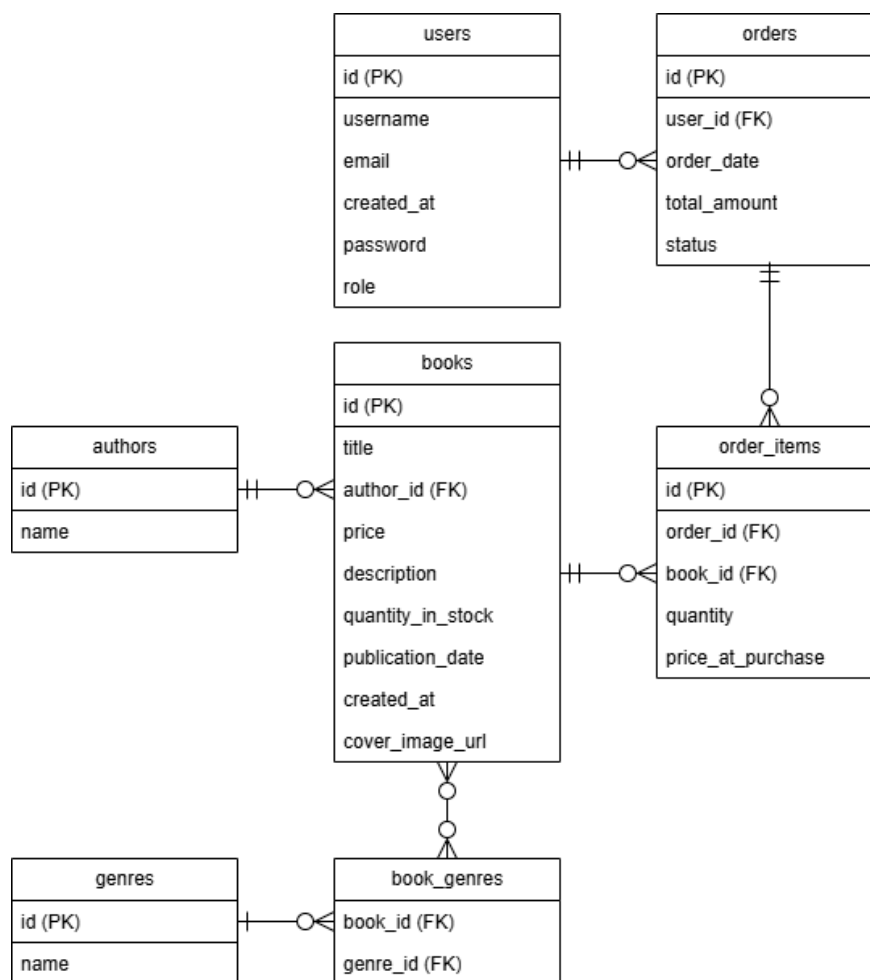


Рисунок 3.2 – Модель базы данных

3.4.1 Описание сущностей

В таблице 3.1 приведен набор полей JSON-документа и их описание для сущности «Авторы».

Таблица 3.1 – Описание сущности ”Авторы”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор автора (первичный ключ)
name	varchar(100)	Полное имя автора

В таблице 3.2 приведен набор полей JSON-документа и их описание для сущности «Жанры».

Таблица 3.2 – Описание сущности ”Жанры”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор жанра (первичный ключ)
name	varchar(50)	Название жанра (уникальное)

В таблице 3.3 приведен набор полей JSON-документа и их описание для сущности «Книги».

Таблица 3.3 – Описание сущности ”Книги”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор книги (первичный ключ)
title	varchar(200)	Название книги
author_id	integer	Внешний ключ, ссылается на таблицу авторов
genre_id	integer	Внешний ключ, ссылается на таблицу жанров
price	numeric(10,2)	Цена книги
description	text	Описание книги (может быть NULL)
quantity_in_stock	integer	Количество экземпляров на складе
sold_copies	integer	Количество проданных экземпляров
publication_date	date	Дата публикации (может быть NULL)
created_at	timestamp	Дата и время добавления книги

В таблице 3.4 приведен набор полей JSON-документа и их описание для сущности «Пользователи».

Таблица 3.4 – Описание сущности ”Пользователи”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор пользователя (первичный ключ)
username	varchar(50)	Логин пользователя (уникальный)
email	varchar(100)	Email пользователя (уникальный, может быть NULL)
password	text	Пароль пользователя
role	varchar(20)	Роль: customer, employee, admin
created_at	timestamp	Дата и время регистрации

В таблице 3.5 приведен набор полей JSON-документа и их описание для сущности «Заказы».

Таблица 3.5 – Описание сущности ”Заказы”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор заказа (первичный ключ)
user_id	integer	Внешний ключ, ссылается на таблицу пользователей
order_date	timestamp	Дата и время оформления заказа
total_amount	numeric(10,2)	Общая сумма заказа
status	varchar(20)	Статус: processing, shipped, delivered, cancelled

В таблице 3.6 приведен набор полей JSON-документа и их описание для сущности «Элементы заказа».

Таблица 3.6 – Описание сущности ”Элементы заказа”

Ключ	Тип	Описание
id	integer	Уникальный идентификатор элемента заказа (первичный ключ)
order_id	integer	Внешний ключ, ссылается на таблицу заказов
book_id	integer	Внешний ключ, ссылается на таблицу книг
quantity	integer	Количество экземпляров книги в заказе
price_at_purchase	numeric(10,2)	Цена книги на момент покупки

3.5 Проектирование пользовательского интерфейса

На основании требований к пользовательскому интерфейсу, представленных в пункте 2.3.3 технического задания, был разработан графический интерфейс программно-информационной системы управления книжным магазином. Для создания пользовательского интерфейса используется разметка, основанная на HTML и CSS.

На рисунке 3.1 представлен макет интерфейса главной страницы. Макет содержит следующие элементы:

1. Поисковая строка.
2. Карточка книги.
3. Название книги.
4. Автор книги.
5. Цена книги.
6. Количество на складе.
7. Кнопка открывающая модальное окно с подробным описанием книги.
8. Кнопка добавляющая выбранную книгу в корзину.
9. Кнопка открывающая модальное окно с авторизацией и регистрацией.

10. Пагинация.

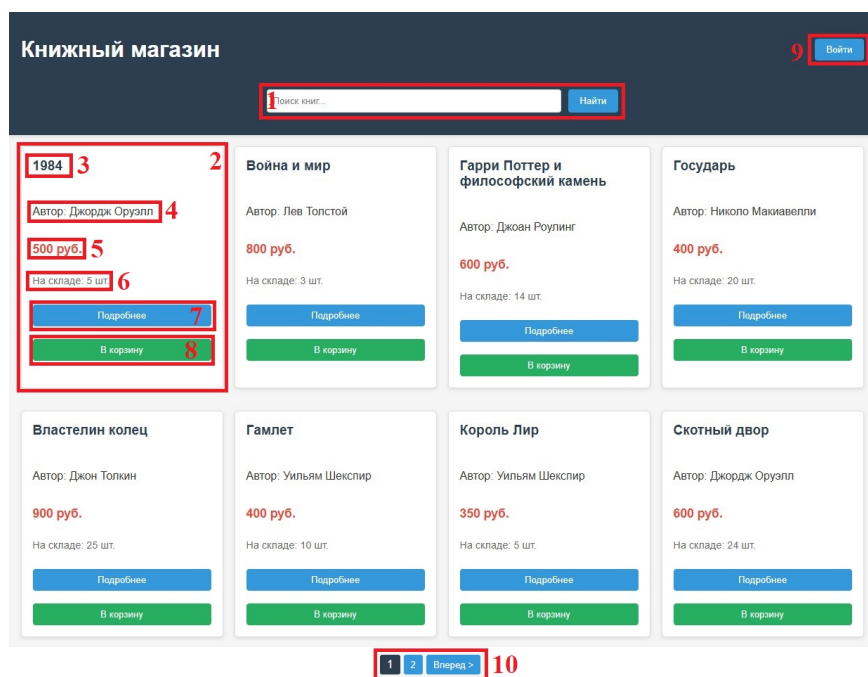


Рисунок 3.3 – Макет интерфейса главной страницы

На рисунке 3.2 представлен макет интерфейса корзины. Макет содержит следующие элементы:

1. Книга добавленная пользователем в корзину.
2. Количество и кнопки для изменения количества конкретной книги в корзине.
3. Наименование книги.
4. Суммарная стоимость выбранного количества книг одного наименования.
5. Кнопка удаления книг одного наименования из корзины.
6. Кнопка для оформления заказа.
7. Итоговая стоимость книг в корзине.

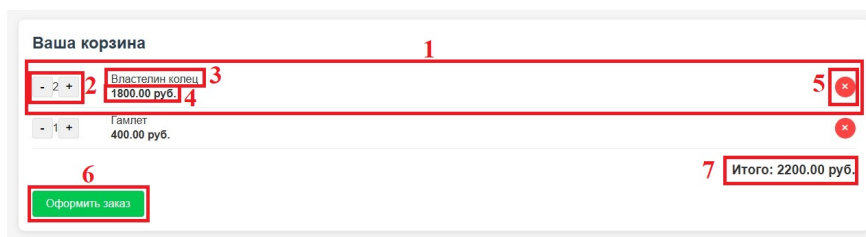


Рисунок 3.4 – Макет интерфейса корзины

На рисунке 3.3 представлен макет панели администратора. Макет содержит следующие элементы:

1. Поле ввода для имени пользователя.
2. Выпадающий список с выбором роли.
3. Кнопка применения роли для выбранного пользователя.
4. Кнопка актуализации данных списка пользователей.
5. Список всех учётных записей и их данных.
6. Кнопка для перехода к форме добавления книги.

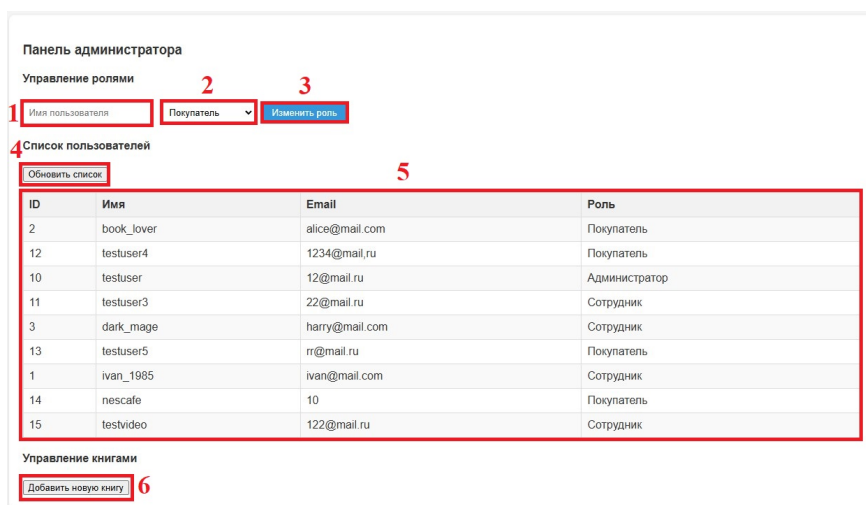


Рисунок 3.5 – Макет интерфейса панели администратора

4 Рабочий проект

4.1 Модули программной системы

4.1.1 Модуль app.py

Модуль представляет собой серверную часть приложения. Реализован с использованием фреймворка Flask. Отвечает за обработку HTTP-запросов, взаимодействие с базой данных PostgreSQL, управление заказами, книгами, авторами, жанрами, тегами и пользователями. Внутренние поля представлены в таблице 4.1.

Таблица 4.1 – Внутренние поля модуля app.py

Внутреннее поле	Тип поля	Описание поля
app	Flask	Экземпляр приложения Flask, используемый для маршрутизации и обработки запросов

Методы модуля представлены в таблице 4.2.

Таблица 4.2 – Методы модуля app.py

Название метода	Параметры метода	Возвращаемое значение	Описание метода
default	obj: Any — объект для сериализации	Сериализуемый объект (float для Decimal, str для date/datetime, или результат базового метода)	Преобразует объекты типа Decimal в float, date/datetime в строку ISO-формата, для остальных вызывает базовый метод JSONEncoder

Продолжение таблицы 4.2

Название метода	Параметры метода	Возвращаемое значение	Описание метода
get_db	Отсутствуют	psycopg2.connect — объект соединения с базой данных	Устанавливает соединение с базой данных PostgreSQL (bookstore_db)
json_response	data: Any — данные для ответа, status: int (по умолчанию 200) — HTTP-код статуса	Response — объект HTTP-ответа с MIME-типом application/json	Формирует JSON-ответ с использованием Custom JSONEncoder
create_order	Отсутствуют (данные из HTTP POST-запроса: user_id, items, total_amount)	JSON-ответ с информацией о заказе или ошибкой	Создает заказ, проверяет наличие книг на складе, обновляет склад, добавляет элементы заказа
get_user_orders	user_id: int — идентификатор пользователя	JSON-ответ со списком заказов или ошибкой	Получает список заказов пользователя с деталями о книгах, отсортированных по дате
update_order_status	order_id: int — ID заказа, JSON-данные с полем status	JSON-ответ с подтверждением или ошибкой	Обновляет статус заказа, проверяет валидность статуса

Продолжение таблицы 4.2

Название метода	Параметры метода	Возвращаемое значение	Описание метода
get_all_orders	Отсутствуют	JSON-ответ со списком всех заказов или ошибкой	Получает все заказы с информацией о пользователях и книгах
get_books	Параметры запроса: search: str — поисковый запрос, page: int — номер страницы, per_page: int — книг на странице	JSON-ответ с книгами и данными пагинации	Получает список книг с поддержкой поиска и пагинации
get_book	book_id: int — ID книги	JSON-ответ с данными книги или ошибкой	Получает детальную информацию о книге, включая жанры и автора
create_book	JSON-данные: title, author_id, price, quantity_in_stock, description, publication_date, cover_image_url, genres, tags	JSON-ответ с ID новой книги или ошибкой	Создает новую книгу и связывает её с жанрами и тегами
delete_book	book_id: int — ID книги	JSON-ответ с подтверждением или ошибкой	Удаляет книгу из базы данных

Продолжение таблицы 4.2

Название метода	Параметры метода	Возвращаемое значение	Описание метода
get_authors	Отсутствуют	JSON-ответ со списком авторов или ошибкой	Получает список всех авторов
search_authors	name: str — поисковый запрос по имени автора	JSON-ответ со списком авторов или ошибкой	Выполняет поиск авторов по имени (ограничение 10 результатов)
create_author	JSON-данные: name: str — имя автора	JSON-ответ с ID нового автора или ошибкой	Создает нового автора в базе данных
get_genres	Отсутствуют	JSON-ответ со списком жанров или ошибкой	Получает список всех жанров
get_tags	Отсутствуют	JSON-ответ со списком тегов или ошибкой	Получает список всех тегов
get_cart_books	ids: str — строка с ID книг, разделённых запятыми	JSON-ответ со списком книг или ошибкой	Получает информацию о книгах в корзине по их ID
register	JSON-данные: username: str, password: str, email: str	JSON-ответ с данными пользователя или ошибкой	Регистрирует нового пользователя с ролью customer
login	JSON-данные: username: str, password: str	JSON-ответ с данными пользователя или ошибкой	Аутентифицирует пользователя по имени и паролю

Продолжение таблицы 4.2

Название метода	Параметры метода	Возвращаемое значение	Описание метода
update_book	book_id: int, JSON-данные: title, author_id, price, quantity_in_stock, description, publication_date, cover_image_url, genres, tags	JSON-ответ с подтверждением или ошибкой	Обновляет данные книги, включая жанры и теги
update_role	JSON-данные: username: str, new_role: str	JSON-ответ с данными пользователя или ошибкой	Обновляет роль пользователя, проверяет валидность роли
get_users	Отсутствуют	JSON-ответ со списком пользователей или ошибкой	Получает список всех пользователей с их ролями

4.1.2 Модуль app.js

Модуль app.js — клиентская часть приложения, реализованная на JavaScript. Отвечает за взаимодействие с сервером через API, управление корзиной, отображение книг, заказов, управление пользователями и администраторскими функциями. Внутренние поля представлены в таблице 4.3.

Таблица 4.3 – Внутренние поля модуля app.js

Внутреннее поле	Тип поля	Описание поля
cart	Array	Локальная корзина пользователя, хранящая ID книг и их количество, сохраняется в localStorage
allBooks	Array	Список всех книг, полученных с сервера, используется для отображения и управления
cartBooks	Array	Список книг в корзине с полной информацией, полученной с сервера
cartEventInitialized	Boolean	Флаг, указывающий, инициализированы ли обработчики событий корзины
currentUser	Object или null	Объект текущего авторизованного пользователя, хранится в localStorage
currentPage	Number	Текущая страница пагинации для списка книг
booksPerPage	Number	Количество книг, отображаемых на одной странице (по умолчанию 8)
authDropdownVisible	Boolean	Флаг, указывающий, отображается ли выпадающее меню аутентификации
ordersDropdownVisible	Boolean	Флаг, указывающий, отображается ли выпадающее меню заказов

Методы модуля представлены в таблице 4.4.

Таблица 4.4 – Методы модуля app.js

Название метода	Параметры метода	Возвращаемое значение	Описание метода
toggleAuth Dropdown	Отсутствуют	Отсутствует	Переключает видимость выпадающего меню аутентификации
toggleOrders Dropdown	Отсутствуют	Отсутствует	Переключает видимость меню заказов и загружает заказы пользователя или все заказы (для админов/сотрудников)
renderOrders	orders: Array — список заказов	Отсутствует	Отрисовывает список заказов в выпадающем меню с деталями и действиями
translate OrderStatus	status: String — статус заказа	String — переведённый статус	Переводит статус заказа на русский язык
toggle OrderDetails	orderId: Number — ID заказа	Отсутствует	Переключает видимость деталей заказа
updateOrderStatus	orderId: Number — ID заказа	Отсутствует	Обновляет статус заказа через API и перерисовывает список заказов
isCartAvailable	Отсутствуют	Boolean — true, если корзина доступна	Проверяет, что все товары в корзине доступны в нужном количестве

Продолжение таблицы 4.4

Название метода	Параметры метода	Возвращаемое значение	Описание метода
saveCart	Отсутствуют	Отсутствует	Сохраняет корзину в localStorage для текущего пользователя
addToCart	bookId: Number — ID книги	Отсутствует	Добавляет книгу в корзину, обновляет интерфейс
removeFromCart	bookId: Number — ID книги	Отсутствует	Удаляет книгу из корзины, обновляет интерфейс
checkout	Отсутствуют	Отсутствует	Оформляет заказ через API, очищает корзину и обновляет интерфейс
updateCartUI	Отсутствуют	Отсутствует	Обновляет интерфейс корзины, включая товары, сумму и доступность кнопки оформления
updateBook Buttons	Отсутствуют	Отсутствует	Обновляет состояние кнопок "В корзину" для всех книг

Продолжение таблицы 4.4

Название метода	Параметры метода	Возвращаемое значение	Описание метода
setupCartEvent Handlers	Отсутствуют	Отсутствует	Устанавливает обработчики событий для кнопок управления корзиной (увеличение/уменьшение количества, удаление)
loadBooks	page: Number (по умолчанию 1) — номер страницы	Отсутствует	Загружает книги с сервера с учетом поиска и пагинации, отрисовывает их
renderBooks	books: Array — список книг, pagination: Object — данные пагинации	Отсутствует	Отрисовывает список книг и элементы пагинации
loadCart	Отсутствуют	Отсутствует	Загружает корзину из localStorage для текущего пользователя
logout	Отсутствуют	Отсутствует	Выполняет выход пользователя, очищает корзину и данные в localStorage

Продолжение таблицы 4.4

Название метода	Параметры метода	Возвращаемое значение	Описание метода
toggleAuthMode	Отсутствуют	Отсутствует	Переключает режим между входом и регистрацией в форме аутентификации
handleAuth	Отсутствуют	Отсутствует	Обрабатывает вход или регистрацию через API, обновляет UI при успехе
loadUserOrders	Отсутствуют	Promise<Array> — список заказов	Загружает заказы текущего пользователя через API
clearAuthFields	Отсутствуют	Отсутствует	Очищает поля формы аутентификации
showAdminPanel	Отсутствуют	Отсутствует	Отрисовывает панель администратора или сотрудника с функциями управления
loadAllOrders	Отсутствуют	Promise<Array> — список всех заказов	Загружает все заказы через API (для админов/сотрудников)
updateUserRole	Отсутствуют	Отсутствует	Обновляет роль пользователя через API и обновляет список пользователей

Продолжение таблицы 4.4

Название метода	Параметры метода	Возвращаемое значение	Описание метода
loadUsersList	Отсутствуют	Отсутствует	Загружает и отображает список всех пользователей
showBookModal	bookId: Number — ID книги	Отсутствует	Отображает модальное окно с подробной информацией о книге
closeModal	Отсутствуют	Отсутствует	Закрывает модальное окно, восстанавливает прокрутку страницы
showAdd BookForm	Отсутствуют	Отсутствует	Отображает форму для добавления новой книги
selectAuthor	id: Number — ID автора, name: String — имя автора	Отсутствует	Выбирает автора в форме добавления/редактирования книги
showAdd AuthorForm	Отсутствуют	Отсутствует	Отображает модальное окно для добавления нового автора
addNewAuthor	Отсутствуют	Отсутствует	Добавляет нового автора через API и обновляет форму книги
closeAuthorModal	Отсутствуют	Отсутствует	Закрывает модальное окно добавления автора

Продолжение таблицы 4.4

Название метода	Параметры метода	Возвращаемое значение	Описание метода
showEdit BookForm	bookId: Number — ID книги	Отсутствует	Отображает форму для редактирования книги с предзаполненными данными
addNewBook	Отсутствуют	Отсутствует	Добавляет новую книгу через API, обновляет список книг
updateBook	bookId: Number — ID книги	Отсутствует	Обновляет данные книги через API, обновляет список книг
deleteBook	bookId: Number — ID книги	Отсутствует	Удаляет книгу через API, обновляет список книг
translateRole	role: String — роль пользователя	String — переведённая роль	Переводит роль пользователя на русский язык

4.1.3 Модуль index.html

Данный модуль является HTML-файлом, представляющим структуру пользовательского интерфейса книжного магазина. Структура модуля:

- header – содержит логотип, панель пользователя, кнопки входа/выхода;
- search – поле поиска книг;
- books-container – контейнер для отображения списка книг;
- cart-container – контейнер для корзины;

- admin-panel – панель для администраторов/сотрудников;
- auth-dropdown – выпадающее меню для входа/регистрации.

4.1.4 Модуль styles.css

styles.css — файл стилей, определяющий внешний вид пользовательского интерфейса. Модуль задаёт стили для всех элементов интерфейса, обеспечивает адаптивность/анимации, определяет визуальные состояния.

4.2 Системное тестирование разработанной программной системы

На рисунке 4.1 изображена главная страница сайта, демонстрирующая каталог книг с пагинацией, поисковой строкой и кнопками для добавления книг в корзину.

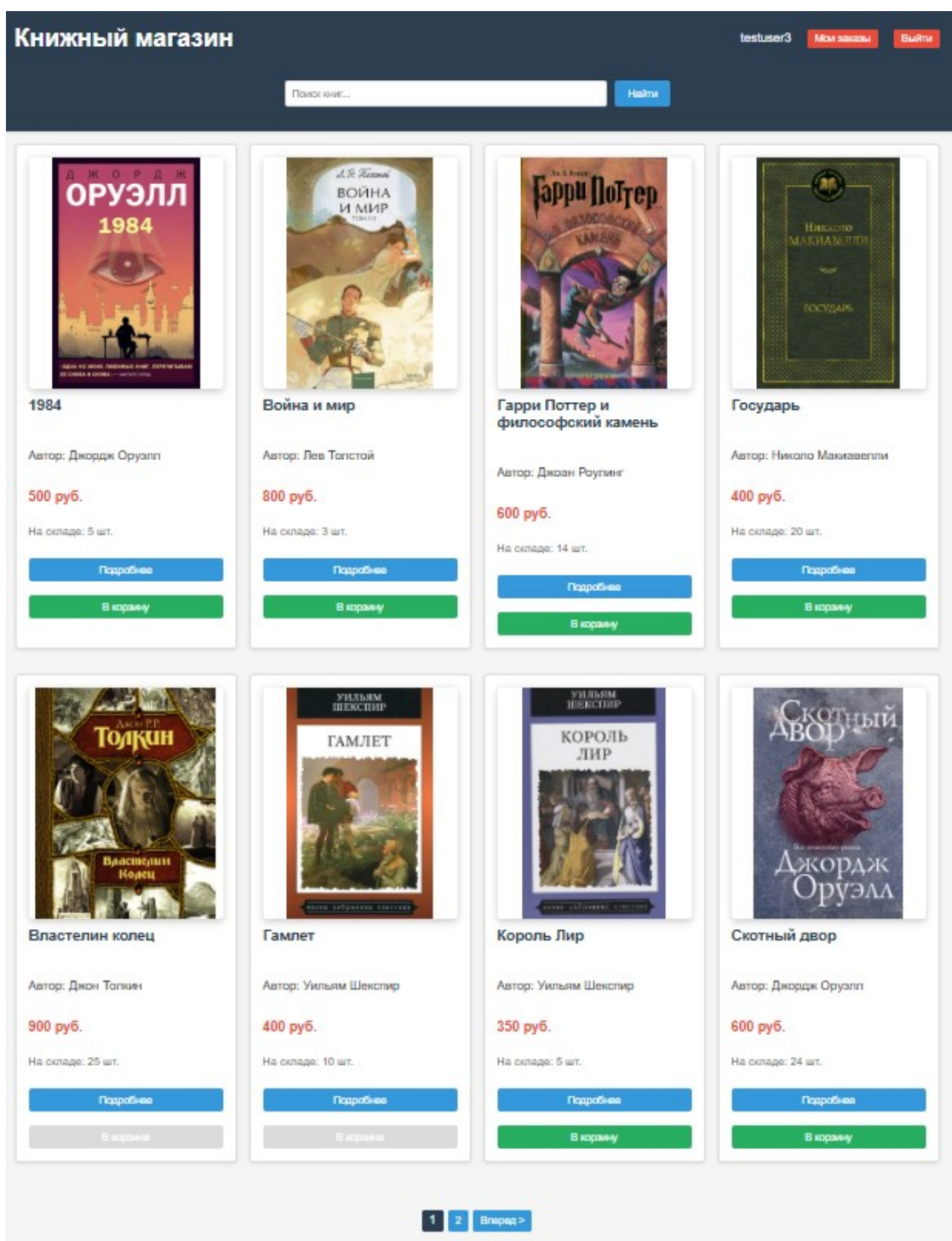


Рисунок 4.1 – Главная страница сайта

На рисунке 4.2 показано окно входа, обеспечивающее авторизацию пользователей. Модальное окно содержит поля для ввода имени пользователя и пароля, а также кнопку переключения на форму регистрации.

The image shows a login window titled "Вход" (Login). It is a white modal box with a dark blue header bar. In the top right corner of the header bar is a blue button labeled "Войти" (Login). Inside the modal box, there are two input fields: "Имя пользователя" (Username) and "Пароль" (Password). Below these fields are two buttons: a green button labeled "Войти" (Login) and an orange button labeled "Регистрация" (Registration).

Рисунок 4.2 – Окно входа

На рисунке 4.3 представлена подробная информация о книге, отображаемая в модальном окне при выборе книги из каталога. Окно включает название, автора, цену, дату публикации, жанры, описание и наличие на складе, а также кнопку добавления в корзину, что демонстрирует корректную работу функции просмотра деталей.

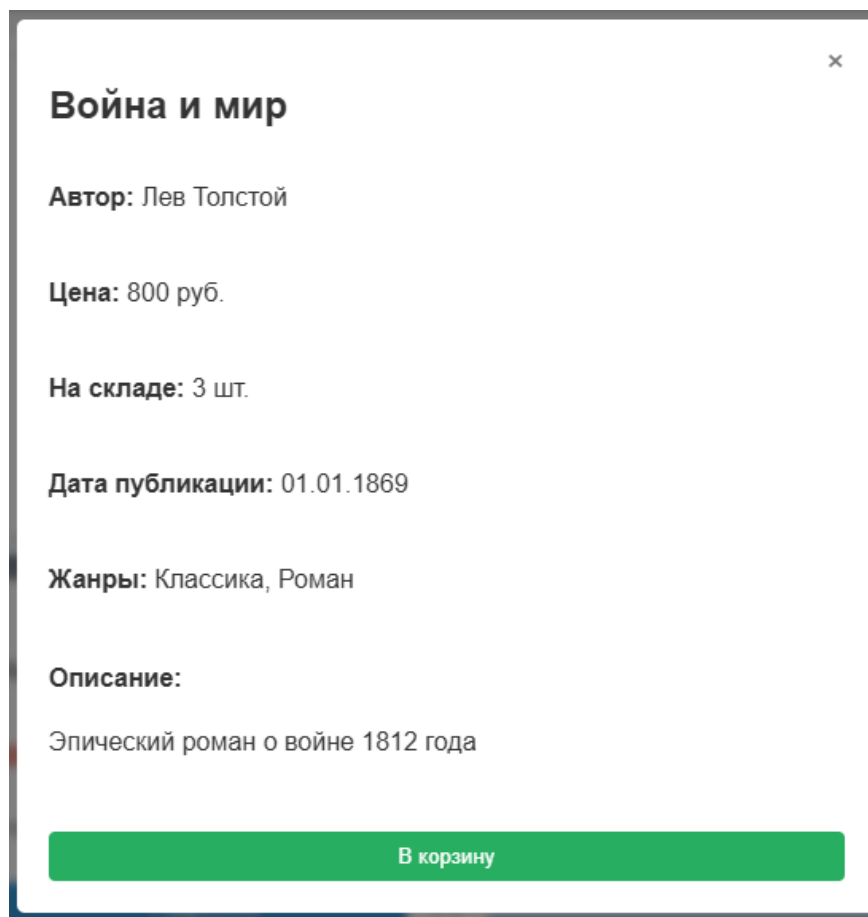


Рисунок 4.3 – Подробная информация о книге

На рисунке 4.4 продемонстрирована работа поисковой строки, позволяющая фильтровать книги по названию.

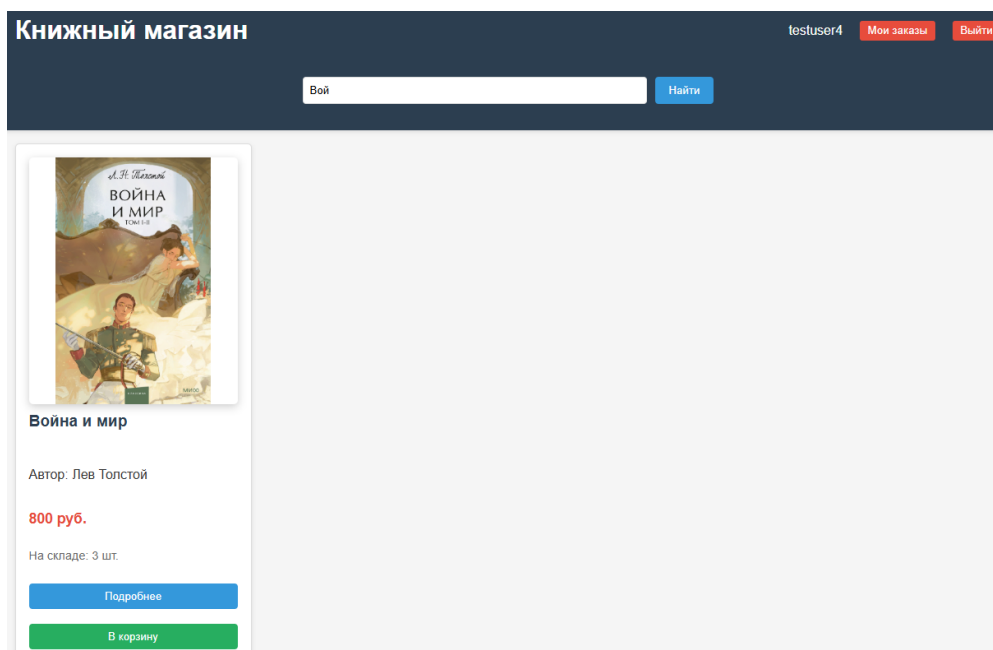


Рисунок 4.4 – Демонстрация работы поисковой строки

На рисунке 4.5 изображена корзина пользователя, содержащая выбранные книги с возможностью изменения количества и удаления позиций. Итоговая сумма обновляется автоматически, а кнопка «Оформить заказ» активна при наличии товаров.

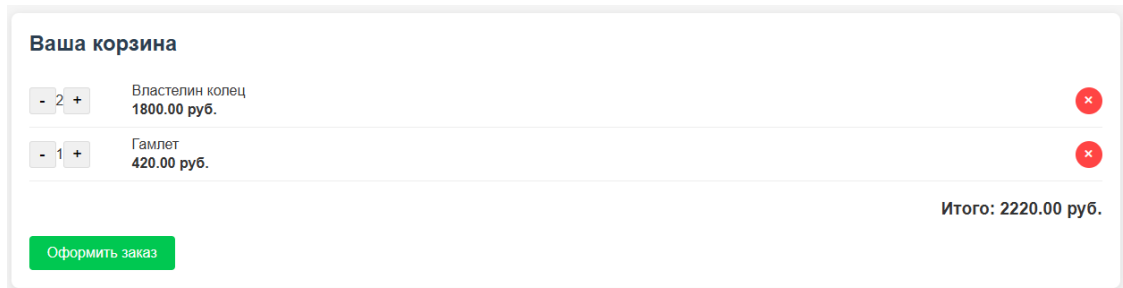


Рисунок 4.5 – Корзина пользователя

На рисунке 4.6 показано уведомление об успешном оформлении заказа с указанием номера, даты, суммы и статуса.

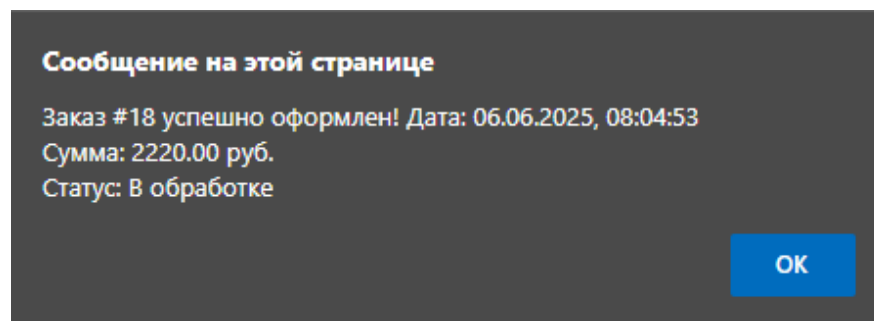


Рисунок 4.6 – Уведомление об успешном заказе

На рисунке 4.7 представлен список заказов пользователя, включающий книги, номер, дату, сумму и статус.

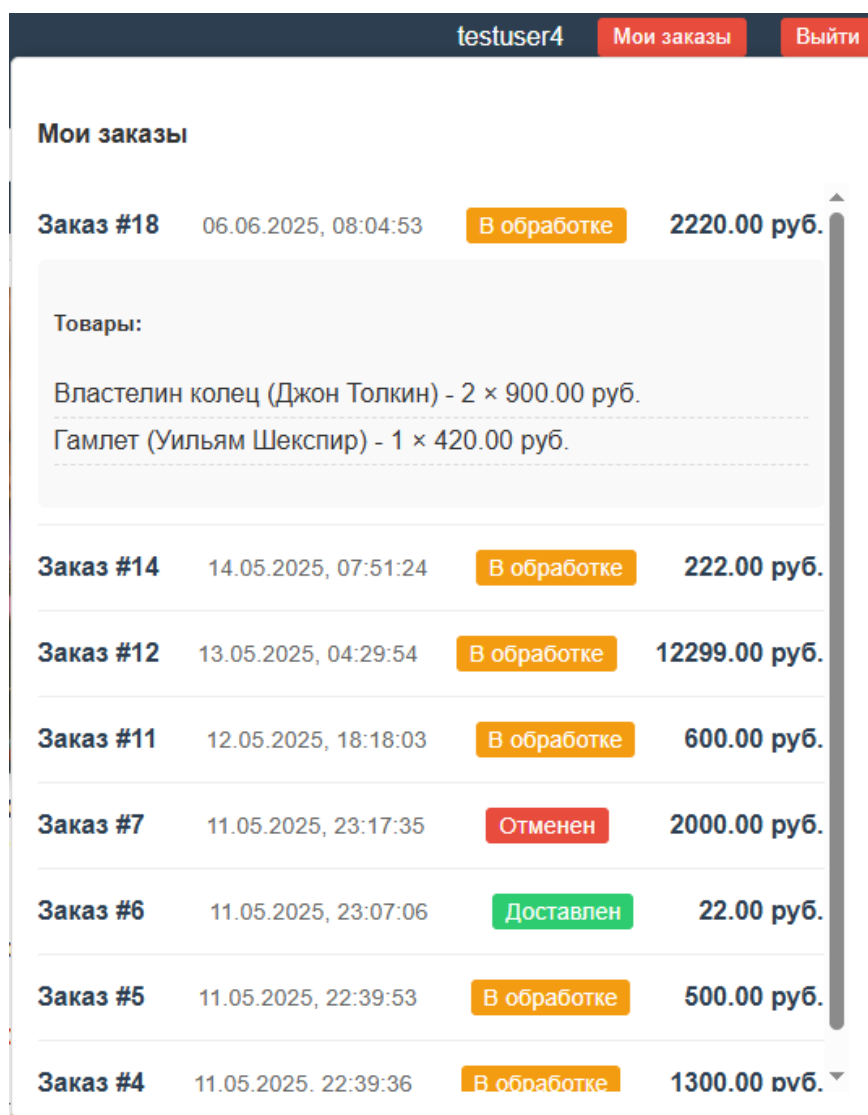


Рисунок 4.7 – Список заказов пользователя

На рисунках 4.8, 4.9, 4.10, 4.11 и 4.12 показана главная страница, окно заказов и панель для пользователей с ролью администратора или сотрудника, предоставляющая доступ к функциям управления магазином. Интерфейс включает кнопки и панели для добавления и редактирования книг, а также просмотра заказов.

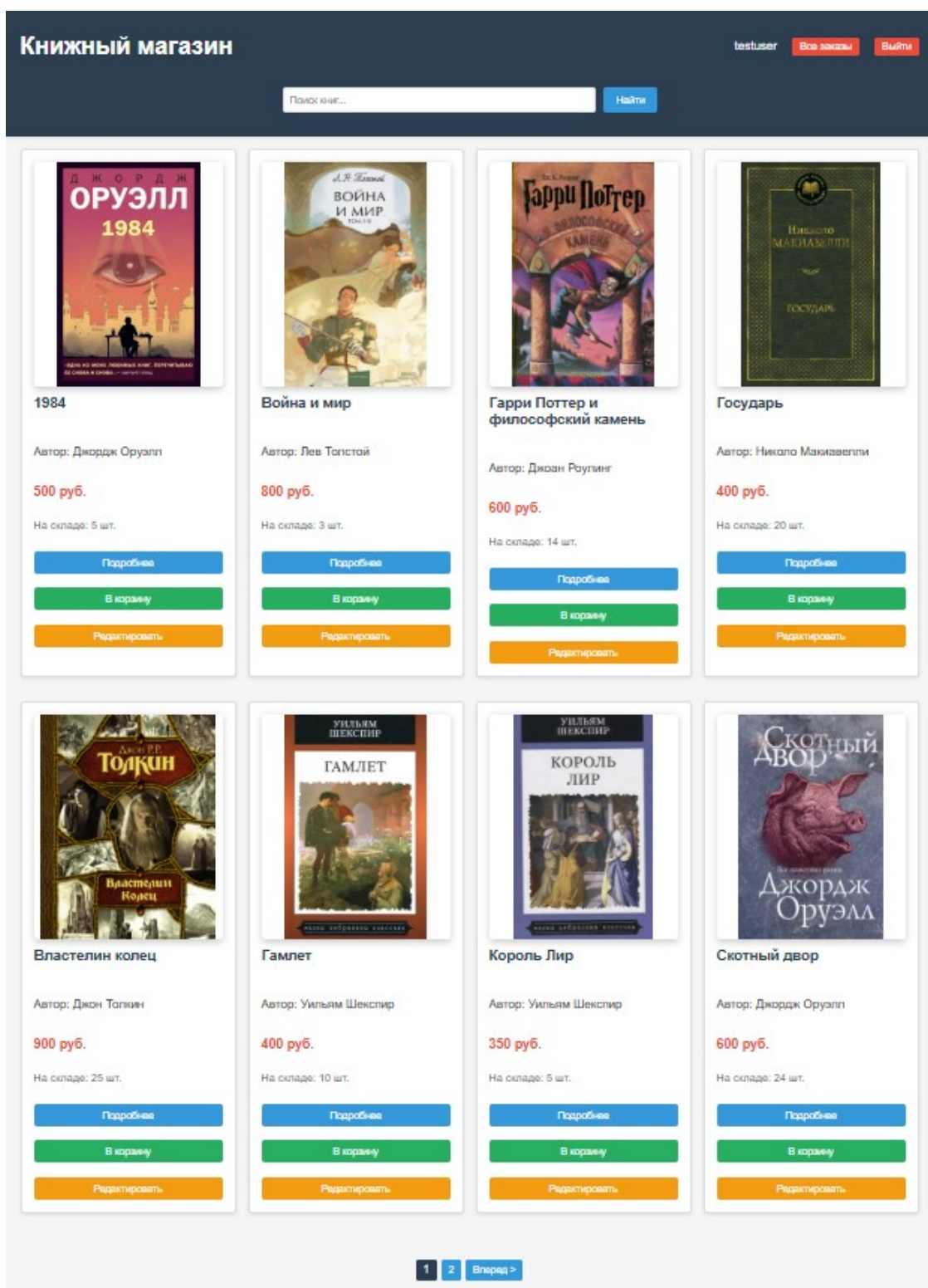


Рисунок 4.8 – Главная страница администратора/сотрудника

Панель администратора

Управление ролями

Имя пользователя: Покупатель

Список пользователей

ID	Имя	Email	Роль
2	book_lover	alice@mail.com	Покупатель
12	testuser4	1234@mail.ru	Покупатель
10	testuser	12@mail.ru	Администратор
3	dark_mage	harry@mail.com	Сотрудник
13	testuser5	tt@mail.ru	Покупатель
1	ivan_1985	ivan@mail.com	Сотрудник
14	nescafe	10	Покупатель
15	testvideo	122@mail.ru	Сотрудник
11	testuser3	22@mail.ru	Сотрудник

Управление книгами

Рисунок 4.9 – Панель администратора

Управление книгами

Добавить новую книгу

Название:

Автор:

Цена:

Количество на складе:

Описание:

Дата публикации:

URL обложки:

Жанры:

☐ Антиутопия ☐ Роман ☐ Фэнтези ☐ Классика

Рисунок 4.10 – Панель добавления новой книги

Управление книгами

[Добавить новую книгу](#)

Редактировать книгу: Сказка о рыбаке и рыбке

Название:

Сказка о рыбаке и рыбке

Автор:

Александр Пушкин

[Добавить нового автора](#)

Цена:

200

Количество на складе:

10

Описание:

Дата публикации:

01.02.1812



URL обложки:

<https://example.com/cover.jp>

Жанры:

☐ Антиутопия ☐ Роман ☐ Фэнтези ☒ Классика

[Сохранить изменения](#)

[Удалить книгу](#)

[Отмена](#)

Рисунок 4.11 – Панель редактирования существующей книги

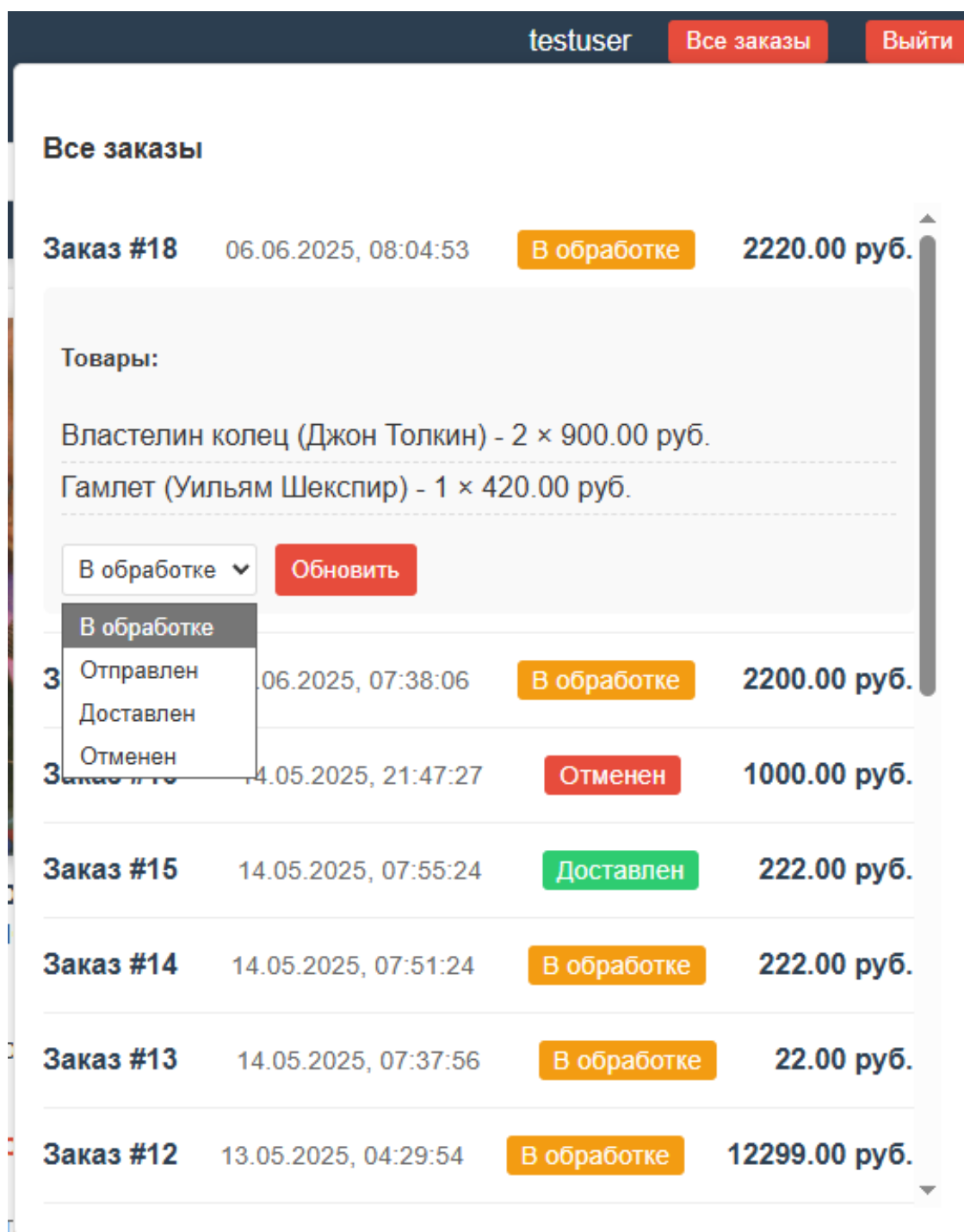


Рисунок 4.12 – Окно управления заказами всех пользователей

4.3 Сборка программной системы

Программные компоненты включают файлы исходного кода: app.py, app.js, index.html и styles.css.

Для сборки серверной части использовалась библиотека PyInstaller, которая упаковывает Python-код и зависимости в один исполняемый файл .exe. Этот файл запускается без установки Python и библиотек.

Клиентская часть не требует компиляции и работает в браузере. Файлы размещаются в статической директории сервера Flask.

Интерпретация Python-кода выполняется встроенным интерпретатором в .exe-файле. Для базы данных PostgreSQL требуется отдельная установка.

Все компоненты собраны для запуска: сервер — как .exe в Windows, клиент — через браузер.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была разработана программно-информационная система управления книжным магазином в виде веб-приложения с клиент-серверной архитектурой. Использование современных веб-технологий позволило создать гибкое, надёжное и масштабируемое решение, отвечающее требованиям автоматизации процессов книжного магазина.

Разработанное веб-приложение обеспечивает удобный доступ к каталогу книг, оформление заказов и управление данными для трёх ролей пользователей: покупатель, сотрудник и администратор. Система предоставляет адаптивный пользовательский интерфейс, обеспечивающий комфортное использование на различных устройствах, и эффективно обрабатывает запросы благодаря REST API.

Основные результаты работы:

1. Проведён анализ предметной области интернет-торговли книгами, определены ключевые сущности системы и сформулированы требования к системе.
2. Разработана концептуальная модель системы, включая модель данных и архитектуру клиент-серверного взаимодействия.
3. Спроектирована программная система. Разработана архитектура, включающая серверные и клиентские модули. REST API обеспечивает взаимодействие. Flask обрабатывает запросы и взаимодействует с PostgreSQL через `psycopg2`. Интерфейс адаптивен и поддерживает все функции.
4. Реализовано и протестировано веб-приложение. Серверная часть обеспечивает обработку запросов, клиентская часть – динамический интерфейс с адаптивным дизайном. Проведено системное тестирование, подтвердившее работоспособность функций.

Все требования, объявленные в техническом задании, были полностью реализованы, все задачи, поставленные в начале разработки проекта, были также решены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Манина А.И. Понятие и особенности правового регулирования торговли в сети Интернет // Вестник науки. – 2021. – № 1 (34), т. 3. – С. 126–130. – ISSN: 27128849. – URL: <https://www.xn——8sbempclcwd3bmt.xn—plai/article/4026>.
2. Корнева С.С. Развитие электронной торговли в Российской Федерации // Известия ОГАУ. – 2011. – № 32-1. – С. 208–211. – URL: <https://cyberleninka.ru/article/n/razvitie-elektronnoy-torgovli-v-rossiyskoy-federatsii>.
3. Статья 26.1. Дистанционный способ продажи товара // КонсультантПлюс. – URL: https://www.consultant.ru/document/cons_doc_LAW_305/1525b1a2f037db240c8e6a749619f86e53857f13.
4. Вафина Н.Х. Интернет-магазин // Мир новой экономики. – 2010. – № 2. – С. 28–32. – EDN: ZEZNAH.
5. Тагавердиева Д.С. Интернет-торговля как фактор развития бизнеса // УЭПС: экономика, политика, право. – 2021. – № 2. – С. 5–12. – URL: <https://cyberleninka.ru/article/n/internet-torgovlya-kak-faktor-razvitiya-biznesa>.
6. Новостной портал «Newsland». – URL: <https://newsland.com/news/detail/id/1007025>.
7. Воскресенская О.В. Влияние интернет-магазинов на покупательскую способность потребителей // E-Scio. – 2023. – № 2 (77). – С. 45–52. – URL: <https://cyberleninka.ru/article/n/vliyanie-internet-magazinov-na-pokupatelskuyu-sposobnost-potrebiteley>.
8. Чепелева А.Ю., Хайрхуа А.Ф. Интернет-торговля в России как основной элемент цифровой экономики // Студенческий научный форум: материалы XI Международной студенческой научной конференции, 2019. – С. 1–5. – URL: <https://scienceforum.ru/2019/article/2018010575>.
9. Косников С.Н., Королёв Д.А., Чивви Е.Н. и др. История электронной коммерции в России и за рубежом // ЕГИ. – 2022. – № 6 (44). – С. 33–40. –

URL: <https://cyberleninka.ru/article/n/istoriya-elektronnoy-kommertsii-v-rossii-i-za-rubezhom>.

10. Data Insight. Исследование DI eCommerce 2020. – URL: https://datainsight.ru/DI_eCommerce2020.

11. Хамдохова Х.Р., Кяова А.А., Шагиров А.С. и др. Влияние пандемии COVID-19 на российский рынок интернет-торговли // Авангард молодежной науки: сборник статей II Международного научно-исследовательского конкурса, Петрозаводск, 28 марта 2022 года. – Петрозаводск: МЦНП «Новая Наука», 2022. – С. 208–212. – EDN: KWHZHR.

12. Пушкарева Е.П. Книжная интернет-торговля: зарубежный опыт и российские реалии // Вестник МГУП. – 2011. – № 3. – С. 15–22. – URL: <https://cyberleninka.ru/article/n/knizhnaya-internet-torgovlya-zarubezhnyy-opyt-i-rossiyskie-realii>.

13. Мандел Т. Разработка пользовательского интерфейса. – Москва: ДМК Пресс, 2019. – 420 с. – ISBN: 9785041950606.

14. Фримен А. Практикум по программированию на JavaScript. – Москва: Вильямс, 2013. – 960 с. – ISBN: 9785845917997.

15. Буч Г., Якобсон И., Рамбо Д. Введение в UML от создателей языка. – Москва: ДМК Пресс, 2015. – 498 с. – ISBN: 9785457433793.

ПРИЛОЖЕНИЕ А

Представление графического материала

Графический материал, выполненный на отдельных листах, изображен на рисунках А.1–А.4.

ЮГО-ЗАПАДНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ	1																																
<h2 style="margin: 0;">Сведения о ВКРБ</h2> <p style="margin: 5px 0;">Минобрнауки России</p> <p style="margin: 5px 0;">Юго-Западный государственный университет</p> <p style="margin: 5px 0;">Кафедра программной инженерии</p> <h3 style="margin: 10px 0;">ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА ПО ПРОГРАММЕ БАКАЛАВРИАТА</h3> <p style="margin: 20px 0;">«Разработка web-сайта «Русатом – Аддитивные технологии» на платформе 1С – Битрикс»</p>																																	
<p>Руководитель ВКР</p> <p>к.т.н, доцент</p> <p>Малышев Александр Васильевич</p>	<p>Автор ВКР</p> <p>студентка группы ПО-81з</p> <p>Мягкая Ирина Витальевна</p>																																
<table border="1" style="border-collapse: collapse; font-size: 0.8em;"> <tr> <td colspan="4" style="text-align: center;">ВКРБ-2068443.09.03.04.23008</td> </tr> <tr> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td style="text-align: center;">Автор работы</td> <td style="text-align: center;">Мягкая И.В.</td> <td style="text-align: center;">Страницы</td> <td style="text-align: center;">Даты</td> </tr> <tr> <td style="text-align: center;">Руководитель</td> <td style="text-align: center;">Малышев А.В.</td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">Рецензент</td> <td style="text-align: center;">Мягкая И.В.</td> <td></td> <td></td> </tr> <tr> <td colspan="2" style="text-align: center;">Сведения о ВКРБ</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td colspan="2"></td> <td style="text-align: center;">Лист 1</td> <td style="text-align: center;">Листов 23</td> </tr> <tr> <td colspan="2" style="text-align: center;">Выпускная квалификационная работа бакалавра</td> <td colspan="2" style="text-align: center;">ЮЗГУ ПО-81з</td> </tr> </table>		ВКРБ-2068443.09.03.04.23008								Автор работы	Мягкая И.В.	Страницы	Даты	Руководитель	Малышев А.В.			Рецензент	Мягкая И.В.			Сведения о ВКРБ		Лист	Масштаб			Лист 1	Листов 23	Выпускная квалификационная работа бакалавра		ЮЗГУ ПО-81з	
ВКРБ-2068443.09.03.04.23008																																	
Автор работы	Мягкая И.В.	Страницы	Даты																														
Руководитель	Малышев А.В.																																
Рецензент	Мягкая И.В.																																
Сведения о ВКРБ		Лист	Масштаб																														
		Лист 1	Листов 23																														
Выпускная квалификационная работа бакалавра		ЮЗГУ ПО-81з																															

Рисунок А.1 – Сведения о ВКРБ

<h2 style="margin: 0;">Цель и задачи разработки</h2>																																																											
<p>Цель настоящей работы – разработка и внедрение web-сайта для продвижения компании ООО «Русатом – Аддитивные технологии».</p> <p>Для достижения поставленной цели необходимо решить следующие задачи:</p> <ol style="list-style-type: none"> 1. Создание информационных разделов сайта «О компании», «Продукция», «Услуги», «Рассчитать стоимость изготовления детали», «Пресс-центр», «Импортозамещение», «Контакты». 2. Реализация формы для обратной связи. 3. Реализация калькулятора расчета стоимости изготовления деталей. 4. Реализация формы заявки на изготовление деталей. 5. Создание удобного поиска по сайту. 																																																											
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td colspan="10" style="text-align: center;">ИРРБ-206843.09.03.04.23008</td> </tr> <tr> <td colspan="5"></td> <td colspan="5" style="text-align: center;">Цель и задачи разработки</td> </tr> <tr> <td colspan="5"></td> <td colspan="5" style="text-align: center;">Лист 2 из 23</td> </tr> <tr> <td colspan="5"></td> <td colspan="5" style="text-align: center;">Выпускная квалификационная работа бакалавра</td> </tr> <tr> <td colspan="5"></td> <td colspan="5" style="text-align: center;">ЮЗГУ ПО-81з</td> </tr> </table>										ИРРБ-206843.09.03.04.23008															Цель и задачи разработки										Лист 2 из 23										Выпускная квалификационная работа бакалавра										ЮЗГУ ПО-81з				
ИРРБ-206843.09.03.04.23008																																																											
					Цель и задачи разработки																																																						
					Лист 2 из 23																																																						
					Выпускная квалификационная работа бакалавра																																																						
					ЮЗГУ ПО-81з																																																						

Рисунок А.2 – Цель и задачи разработки



Рисунок А.3 – Концептуальная модель сайта



Рисунок А.4 – Еще плакат

ПРИЛОЖЕНИЕ Б

Фрагменты исходного кода программы

main.tex

TexПроект.tex

Автор ВКР

(подпись, дата)

Д. Н. Рязанцев

Руководитель ВКР

(подпись, дата)

Е. П. Кочура

Нормоконтроль

(подпись, дата)

А. А. Чаплыгин

Место для диска