

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ  
«МОСКОВСКИЙ КОЛЛЕДЖ УПРАВЛЕНИЯ, ГОСТИНИЧНОГО  
БИЗНЕСА И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ «ЦАРИЦЫНО»

ДИПЛОМНЫЙ ПРОЕКТ

Тема: Разработка системы с веб-интерфейсом для интернет-магазина художественной литературы с модулем подбора по предпочтениям

Пояснительная записка  
ДП.09.02.07.41.25.2022 01 ПЗ

Руководитель проекта: Важенина Е.Г. Дипломник: Сакилов Э.В.

Рецензент: Григорьев Е.Г.

Допустить к защите на заседании  
Государственной экзаменационной комиссии


Учебно-методический отдел отделения УИТ

« 30 » мая 2022 г. Т.Н. Михайлова

Москва, 2022

**Согласовано**

**Утверждаю**

На заседании Кафедры  
информационных технологий  
Протокол №  
от « 30 » марта 2022 г.  
Заведующий кафедрой  
 Е.Ф. Писчасова

УМО ОУИТ

« 31 » марта 2022 г.

 Т.Н. Михайлова

## **ЗАДАНИЕ НА ДИПЛОМНОЕ ПРОЕКТИРОВАНИЕ**

Студенту Сакилову Эдуарду Валерьевичу

Группы ИС4-1Д специальности 09.02.07 Информационные системы и программирование

Дата выдачи задания: « 19 » апреля 2022 г.

Срок сдачи проекта: « 27 » мая 2022 г.

### **Тема дипломного проекта:**

Разработка системы с веб-интерфейсом для интернет-магазина художественной литературы с модулем подбора по предпочтениям

## **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Исходные данные: информация о деятельности компании, целевой аудитории

Задачи:

1. Выполнить анализ предметной области
2. Разработать пользовательский интерфейс программы
3. Разработать базу данных
4. Разработать алгоритмы работы системы
5. Реализовать работу сайта с базой данных
6. Протестировать web-систему

### **Содержание пояснительной записки**

Раздел пояснительной записки	Объем, %
Введение	1%
Раздел 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	
1.1 Анализ предметной области	

1.2 Описание и выбор метода решения 1.3 Выбор СУБД 1.4 Обоснование выбора языка программирования 1.5 Выбор вспомогательных средств разработки 1.6 Техническое обеспечение 1.7 Обоснование необходимости и цели использования веб-сервисов для решения задачи 1.8 Анализ существующих разработок на рынке	36%
Раздел 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 2.1 Проектирование базы данных 2.2 Разработка физической и логической структуры сайта 2.3 Структура сайта 2.4 Создание дизайн-макета интернет-магазина 2.5 Верстка страниц 2.6 Тестирование веб-системы 2.7 Руководство пользователя	61%
Заключение	1%
Список использованной литературы	1%

### Графическая часть:

Мультимедиа презентация.


Дипломник:

 /Сакилов Э.В./

Руководитель проекта:

 /Важенина Е.Г./

Заведующий кафедрой:

 /Писчасова Е.Ф./



## Содержание

ВВЕДЕНИЕ.....	5
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	7
1.1 Анализ предметной области.....	7
1.2 Описание и выбор средств разработки.....	7
1.3 Выбор СУБД.....	8
1.4 Обоснование выбора языка программирования.....	13
1.5 Выбор вспомогательных средств разработки.....	18
1.5.1 Средства моделирования ПО.....	18
1.5.2 Средства разработки графического интерфейса.....	19
1.6 Техническое обеспечение (ТО).....	19
1.7 Обоснование необходимости и цели использования веб-сервисов для решения задач.....	20
1.8 Анализ существующих разработок на рынке.....	22
2. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	26
2.1 Проектирование баз данных.....	26
2.2 Разработка физической и логической структуры сайта.....	35
2.3 Структура сайта.....	36
2.4 Создание дизайн-макета интернет-магазина.....	37
2.5 Верстка страниц.....	45
2.5.1 Описание программных модулей.....	53
2.5.2 Логика работы программы.....	55
2.6 Тестирование веб-системы.....	56
2.6.1 Тестирование системы регистрации.....	56
2.6.2 Тестирование системы авторизации.....	57
2.7 Руководство пользователя.....	57
ЗАКЛЮЧЕНИЕ.....	65
СПИСОК ИСТОЧНИКОВ ЛИТЕРАТУРЫ.....	67
ИНТЕРНЕТ-ИСТОЧНИКИ.....	68
ПРИЛОЖЕНИЕ 1. Исходный код.....	69

					ДП.09.02.07.41.25.2022 01 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Разработка системы с веб-интерфейсом для интернет-магазина художественной литературы с модулем подбора по предпочтениям  <b>Пояснительная записка</b>	Лит.		Лист	Листов
Разраб.		Сакилов Э.В.		17.06		Д		4	69
Провер.		Важенина Е.Г.		17.06					
Реценз.		Григорьев Е.Г.		18.06					
Н. Контр.		Важенина Е.Г.		18.06					
Утверд.		Писчасова Е.Ф.		18.06					
					ГБПОУ Колледж Царицыно				

## ВВЕДЕНИЕ

В настоящее время широко применяется концепция интернет-магазинов, когда пользователю не нужно выходить из дома чтобы приобрести себе какой-либо товар. На данный момент интернет-магазины представляют собой новую, но уже устоявшуюся систему реализации товаров или услуг. По сути, эта система является более эффективной по сравнению со стандартным «физическим» магазином.

Этот способ позволяет:

- улучшить качество обслуживания клиентов, так как все действия по покупке товаров производятся в автоматическом режиме, а менеджерам необходимо только уточнить информацию и выдать распоряжения о доставке товаров;
- снизить трудоемкость и напряженность труда персонала, минимизировать ошибки в его действиях;
- улучшить качество обслуживания клиентов, так как все действия по покупке товаров производятся в автоматическом режиме, а менеджерам необходимо только уточнить информацию и выдать распоряжения о доставке товара;
- внимательный персонал, от которого можно получить помощь;
- снизить издержки, так как отпадает необходимость в аренде здания для магазина;

В данном дипломном проекте будет рассмотрена реализация веб-приложения, содержащего всю необходимую информацию и облегчающее процесс покупки книги в интернет-магазине.

Интернет-магазин – это по своей сути тот же самый магазин, но виртуальный, и у этого магазина есть множество преимуществ. Выделим тройку из них. Во-первых, это доступность. В магазин может попасть любой пользователь с доступом в интернет. Во-вторых, моментальный доступ к купленному товару. Покупателю не нужно ждать доставки или самому забирать заказ. Купленная книга будет доступна к прочтению сразу после

оплаты. В-третьих, в таком магазине нет очередей. Покупателю не нужно ждать, и он моментально может совершить покупку. В-четвертых, весь ассортимент товаров может быть организован так, как того пожелает пользователь. В целом, единственное, чего не хватает в виртуальном магазине – это наглядной оценки и подбора товара, но этот вопрос уже решается, а в дипломной работе используется узкоспециализированный магазин с книгами.

Актуальность дипломной работы состоит в том, что создаваемая система с веб-интерфейсом для книжного магазина позволяет улучшить взаимодействие владельцев книжного магазина с покупателями в соответствии с современными технологиями, позволяющими покупателям с легкостью пользоваться всеми необходимыми возможностями книжного магазина, но удаленно, то есть через сеть Интернет. В данной платформе присутствует модуль подбора по предпочтениям, что позволяет пользователю найти подходящую для него книгу.

Целью данного дипломного проекта является Разработка системы с веб-интерфейсом для интернет-магазина художественной литературы с модулем подбора по предпочтениям.

Задачи:

1. Выполнить анализ предметной области
2. Разработать пользовательский интерфейс программы
3. Разработать базу данных
4. Разработать алгоритмы работы системы
5. Реализовать работу сайта с базой данных
6. Протестировать систему с web интерфейсом
7. Написать руководство пользователя

## **1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

### **1.1 Анализ предметной области**

В качестве предметной области проекта выступает разработка интернет-магазина по продаже книг с умным подбором по предпочтениям.

Программа должна быть обеспечена возможностью выполнения следующих функций:

1. Регистрация;
2. Авторизация;
3. Отображение товаров (книг);
4. Добавление товаров (книг) в корзину;
5. Просмотр личного кабинета;
6. Просмотр корзины.

Исходными данными являются:

1. Сведения о пользователях;
2. Сведения о товарах.

Поэтому организация базы данных статическая. Также будем считать, что товара на складе бесконечное количество.

### **1.2 Описание и выбор средств разработки**

Для решения данной задачи были использованы следующие средства:

1. MySQL;
2. HTML;
3. JavaScript;
4. CSS;
5. PHP;
6. XAMPP;

### 1.3 Выбор СУБД

К основным функциям СУБД относятся:

1. Управление данными во внешней памяти (на дисках)
2. Управление данными в оперативной памяти с использованием дискового кэша
3. Журнализация изменений (сохранение истории), резервное копирование и восстановление базы данных после сбоев
4. Поддержка языков БД (язык определения данных, язык манипулирования данными)

Реляционная база данных — база данных, основанная на реляционной модели данных. В реляционной базе данных (РДБ) используется реляционная модель данных, основанная на представлении данных в виде таблиц. Для реализации задачи дипломного проекта, была выбрана реляционная модель данных, так как на данный момент самым большим объемом инструментов для реализации БД и работы с клиентскими приложениями являются именно реляционные СУБД (РСУБД).

#### MySQL

MySQL — это система управления базами данных (СУБД), распространяемая как свободное программное обеспечение (пользователи имеют право на неограниченную установку, запуск, свободное использование) Преимущества данной СУБД:

1. Открытый исходный код.
2. Простота. MySQL легко устанавливается, имеет понятный интерфейс, а разнообразие плагинов и дополнительных приложений упрощает работу с БД.
3. Функционал. Включает в себя практически весь необходимый набор инструментов.



Хотя PostgreSQL и не может похвастаться большой популярностью в отличие от MySQL, существует довольно большое число приложений облегчающих работу с PostgreSQL, несмотря на всю мощь функционала. Сейчас довольно легко установить эту СУБД используя стандартные менеджеры пакетов операционных систем.

#### Достоинства PostgreSQL

1. Открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой.
2. Большое сообщество - существует довольно большое сообщество, в котором вы запросто найдёте ответы на свои вопросы
3. Большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими.
4. Расширения - существует возможность расширения функционала за счет сохранения своих процедур.
5. Объектность – PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

#### Недостатки PostgreSQL

1. Производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL.
2. Популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество.
3. Хостинг - в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

## SQLite

Легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в эти файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором выполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.

Несколько процессов или потоков могут одновременно без каких-либо проблем читать данные из одной базы. Запись в базу можно осуществить только в том случае, если никаких других запросов в данный момент не обслуживается; в противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

В комплекте поставки идёт также функциональная клиентская часть в виде исполняемого файла `sqlite3`, с помощью которого демонстрируется

реализация функций основной библиотеки. Клиентская часть работает из командной строки, позволяет обращаться к файлу БД на основе типовых функций ОС.

Благодаря архитектуре движка возможно использовать SQLite как на встраиваемых системах, так и на выделенных машинах с гигабайтными массивами данных.

Преимущества:

1. Файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины.
  2. Используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), но основные все-таки поддерживаются.
  3. Отличная при разработке и тестировании - в процессе разработки приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки написанной на языке C. Недостатки SQLite
1. Отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа 19 пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях.
  2. Отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

Для реализации цели данного проекта была выбрана СУБД MySQL, поскольку обладает достаточным набором инструментов и документацией.

СУБД MySQL /phpMyAdmin

На данный момент вся информация, существующая в интернет-магазине, хранится на бумажных носителях, что является совершенно ненадёжным в качестве хранения и передачи информации.

Для того, чтобы создать и вести базу данных, будет использоваться СУБД MySQL и дополнительная технология phpMyAdmin. Данные, поступающие в базу данных, должны быть разделены, путем создания пользователей с различными уровнями доступа, что способствует защите поступающих данных.

PhpMyAdmin позволяет предприятию хранить всю информацию о товаре в базе данных, что поможет сократить время на передачу информации от одного отдела к другому, а также обеспечить полную работоспособность предприятия. Я использую phpMyAdmin, поскольку данная технология:

1. Проста в использовании;
2. Имеет удобный конструктор запросов с возможностью внедрения PHP функций,
3. Имеет удобный импорт/экспорт проекта и его частей.
4. В данный момент phpMyAdmin позволяет:
5. Создавать и удалять базы данных;
6. Создавать, копировать, удалять, переименовывать и изменять таблицы;
7. Удалять, править и добавлять поля;
8. Выполнять SQL-запросы, в том числе пакетные SQL-запросы;
9. Управлять ключами;
10. Загружать текстовые файлы в таблицы;
11. Экспортировать (\*) данные в форматах CSV, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel и LATEX;
12. Создавать графическую схему базы данных в формате PDF;
13. Осуществлять поиск в базе данных или в её разделах;

Также данная автоматизированная система позволит пользователю просматривать актуальную информацию о магазине, так как все заносимые данные с помощью СУБД phpMyAdmin будут доступны для просмотра любому пользователю веб-приложения (пользователи имеют доступ только к определённому участку информации). PHP (язык программирования)



## **1.4 Обоснование выбора языка программирования**

Разработка веб-приложения осуществляется с использованием определенного набора программных средств, наиболее комфортная и стабильная работа с которыми достигается при корректном выборе основного языка разработки серверной части веб-приложения.

В рамках данного проекта необходимо выбрать такой язык программирования, который бы позволил поддерживать веб-приложения на протяжении долгого времени, имеет возможности «модульной» разработки и достаточно масштабируем для работы с потенциально большим количеством одновременно подключенных клиентов.

Целью дипломного проекта являлось создание интернет-магазина с минимальным набором необходимых функций для корректной работы. Реализация данной задачи производилась в интегрированной среде разработки «Sublime text 3» с использованием языка разметки HTML. PHP – является наиболее удобным и мощным средством для написания данного проекта. В сочетании с языком разметки html, стилями css и использованием Javascript для интерактивности, можно создать удобный интерфейс между пользователем и исполняемым кодом.

Созданию информационной системы преследовало решение следующих задач:

1. Автоматизация процесса просмотра товаров;
2. Автоматизация процесса покупки товаров;
3. Экономия средств и времени во время работы;
4. Автоматизация процесса реализации товара после покупки;

JavaScript - Мультипарадигменный язык программирования. Поддерживает объективно-ориентированный, императивный и функциональный стили. Позволяет перехватывать события и выполнять различные действия. Это наиболее универсальный язык программирования, с

помощью которого можно автоматизировать задачи на страницах веб-сайтов, включать множество интерактивных функций и создавать анимации. JavaScript можно использовать во множестве сценариев разработки, включая управление браузером, редактирование содержимого в документе, не говоря уже о сценариях на стороне клиента и общения с пользователями, а также асинхронный обмен данными.

Java. Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее. Язык Java зародился как часть проекта создания передового программного обеспечения для различных бытовых приборов. Реализация проекта была начата на языке C++, но вскоре возник ряд проблем, наилучшим средством борьбы с которыми было изменение самого инструмента — языка программирования. Стало очевидным, что необходим платформу-независимый язык программирования, позволяющий создавать программы, которые не приходилось бы компилировать отдельно для каждой архитектуры и можно было бы использовать на различных процессорах под различными операционными системами. 13 Язык Java потребовался для создания интерактивных продуктов для сети Internet. Фактически, большинство архитектурных решений, принятых при создании Java, было продиктовано желанием предоставить синтаксис, сходный с C и C++. В Java используются практически идентичные соглашения для объявления переменных, передачи параметров, операторов и для управления потоком выполнением кода. В Java добавлены все хорошие черты C++.

Преимущества данного языка программирования:

1. Большое количество доступных подключаемых компонентов в системе NPM, которые позволяют решать множество конкретных задач;

2. Поддержка движка V8 — возможность компиляции языка в машинный код;
3. Обладает достаточно высокой скоростью работы и производительностью;
4. Имеет развитую инфраструктуру с большим разнообразием фреймворков и библиотек;
5. Относительно простой и рациональный;
6. Удобен при разработке интерфейсов различного программного обеспечения

К недостаткам можно отнести:

1. Низкая производительность при работе с тяжёлыми вычислительными задачами;
2. Недостаток структурированной документации в реестре NPM и как следствие сложность в использовании сторонних библиотек;
3. Нет строгой типизации, поэтому обладает достаточно вольной трактовкой;
4. Не поддерживает удалённый доступ, поэтому неприменим для сетевых приложений;

PHP — это скриптовый (сценарный) язык общего назначения. На сценарных языках пишут сценарии или скрипты — программы, которые автоматизируют некоторые задачи (которые иначе пользователь делал бы вручную). Скрипты помогают избежать ошибок, сэкономить время пользователя и изменить программу, не боясь, что всё остальное перестанет работать. В отличие от большинства языков, скриптовым языкам не нужна компиляция, и они используются в основном для небольших рутинных задач.

PHP — это серверный язык создания сценариев. Конструкции PHP, вставленные в HTML-текст, выполняются сервером при каждом посещении страницы. Результат их обработки вместе с обычным HTML-текстом передается браузеру. В настоящее время основной версией PHP является пятая. Существуют два основных конкурента PHP: Active Server Pages (ASP)

компании Microsoft и ColdFusion компании Allaire. По сравнению с ними PHP обладает рядом преимуществ, в числе которых:

1. Высокая производительность
2. Для PHP создано множество библиотек, баз данных и фреймворков.
3. PHP совместим почти со всеми современными серверами
4. PHP гибкий для использования и даёт много возможностей
5. Простая и элегантная система классов и объектов
6. Простой синтаксис
7. Простота освоения
8. Гибкость в разработке с использованием различных паттернов

проектирования

К недостаткам можно отнести:

1. Скорость. PHP не самый производительный язык;
2. Множество противоречий в стандарте языка, недостаточно строгая систематика;
3. Слишком базовые средства обеспечения безопасности

PHP-программы состоят из простого текста, поэтому набирать их можно в любом текстовом редакторе. Популярные HTML-редакторы имеют встроенную поддержку для редактирования PHP-программ. Расширение файлов PHP-программ по умолчанию в PHP4 - .php. На основании этого расширения сервер распознает файл как PHP-программу и запускает интерпретатор

HTML - язык гипертекстовой разметки страницы. представляет собой стандартизированный язык разметки документов и используется для отображения веб-страниц в браузере. Это формат разметки, с помощью которого браузер создает страницы сайтов при загрузке, формируя их из разных типов контента. HTML-код обрабатывается браузером в виде текстовых документов с расширением .htm или .html.

Преимущества данного языка разметки:



1. Его код выполняется во всех распространенных интернет-браузерах, включая Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Safari и т.д. Писать же HTML-код можно, по сути, в любом текстовом редакторе, хоть в обычном блокноте.

2. Комбинация из HTML, CSS и JavaScript позволяет создавать полноценные сайты не только с голым текстовым контентом, но еще с дизайном, скриптами и мультимедийным контентом

3. Язык имеет очень простой синтаксис, хорошо упорядоченную и последовательную разметку.

4. Официальные стандарты языка HTML разрабатываются и обновляются консорциумом World Wide Web (W3C)

5. Легко интегрируется с другими популярными языками веб-программирования, включая Node.js, PHP и т.д.

К недостаткам можно отнести:

1. HTML не является полноценным языком программирования, так как в нем нет условий, функций, переменных, операторов и других элементов, необходимых для разработки программ и приложений. Вместо этого, язык содержит только набор тегов, помогающих браузеру правильно отображать содержимое веб-страниц.

2. В основном, подходит только для разработки статических веб-страниц. Для добавления динамических элементов требуется подключение других языков веб-разработки: JavaScript или PHP.

3. Не позволяет разработчикам реализовать логику, поэтому каждую страницу сайта на HTML нужно создавать с нуля отдельно, даже если все они используют одни и те же элементы: заголовки, колонтитулы и т.д.

Языком программирования, на котором будет осуществляться функционал сайта является PHP, который создавался, развивался и

поддерживался как язык для создания Web-сайтов. Именно поэтому многие конструкции и решения в нем созданы для удобства работы в Web-среде. Также, одной из главных причин выбора конкретно этого языка является то, что PHP перенесен на все основные операционные системы. Таким образом, можно разрабатывать сайт в Windows, Mac OS X, а эксплуатировать на Linux-сервере.

Помимо этого, не потребуется платить ни за сам язык, ни за большинство сопутствующих программ (редакторы, Web-серверы, базы данных). Вложения могут потребоваться при аренде доменного имени и сервера для публикации сайта в Интернете. Данный язык способен быстро и надежно получать, а также обрабатывать данные при работе с интерфейсом веб-приложения, позволяет создать шаблоны веб-страниц.

Так как верстка сайта происходит с помощью языка разметки гиперссылок HTML, язык программирования PHP идеально подойдет для написания функционала. Картинка в браузере создается с помощью HTML, что является информативным и статичным материалом, а при помощи функций, написанных на языке программирования PHP, сайт становится удобным и гибким в управлении.

## **1.5 Выбор вспомогательных средств разработки**

### **1.5.1 Средства моделирования ПО**

В рамках дипломного проекта требуется описание моделей базы данных и структуры веб-приложения, включая самые абстрактные модели взаимодействия и непосредственно реализованные методы различных частей приложения.

Для разработки модели базы данных и ее последующей реализации был выбран PhpMyAdmin. Веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только

осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных.

Для описания и визуализации процессов взаимодействия пользователя с приложением и ее отдельных компонентов между собой выбран язык графического описания UML и соответствующий ему профессиональный редактор диаграмм Diagrams.net.

### **1.5.2 Средства разработки графического интерфейса**

Важная задача, поставленная в рамках данного проекта – универсальность и адаптивность приложения для различных устройств, для решения которой требуется создание единого лаконичного, информативного интерфейса. Достаточным набором ПО для разработки интерфейса является Adobe Photoshop для верстки основного макета и Adobe Illustrator для отрисовки графических элементов.

## **1.6 Техническое обеспечение (ТО)**

Для разработки данной автоматизированной системы необходимо хорошее техническое обеспечение, которое сможет обеспечить хорошую работу веб-приложения.

1. Серверное оборудование, чтобы установить сайт на домен;
2. Процессор Intel Core I5 для ускоренной обработки информации на сайте при регистрации пользователей, а также для передачи данных в базу данных;
3. Далее веб-приложение нуждается в оперативной памяти в 32 Гб для обработки большого количества информации и передачи ее пользователю или на сервер.

Помимо технического обеспечения также важно и программное, которое помогает работать веб-приложению на любом устройстве.

Оно включает в себя такие компоненты, как:

1. Операционная система;

## 2. Сервисные программы и т. д.

Данная автоматизированная система поддерживается на любой операционной системе:

1. Windows;
2. Mac OS;
3. Linux (и т. д.),
4. которая использует такие браузеры:
5. Google Chrome;
6. Yandex;
7. Safari;
8. Microsoft Edge;
9. Opera.

### **1.7 Обоснование необходимости и цели использования веб-сервисов для решения задачи**

В настоящее время все действия книжных магазинов проводятся вручную, что требует от его сотрудников постоянного и внимательного контроля за этапами пополнения и продажи товаров. Также работникам магазина необходимо находиться всегда за кассой, чтобы обслужить покупателей, которые, в свою очередь, должны покупать товар в самом магазине, а не в интернет-магазине. В отдел склада товаров, поступает информация о спросе и наличии данного товара. Вследствие чего возникает проблема при обновлении склада товаров, из-за чего покупатель не может купить книгу.

На данный момент времени для того, чтобы узнать есть в наличии товар сотрудникам необходимо вести документацию в бумажном виде. Данная схема обмена информации выглядит следующим образом. Товар поступает на склад, далее сотрудники обновляют список товаров. Чтобы сократить цепочку



действий, состоявшую из множества процессов, необходимо разработать специальную автоматизированную систему. При этом разработанный веб-сайт поможет покупателям покупать книги онлайн, что значительно сокращает большое количество времени.

Для наглядной демонстрации обмена информации в интернет-магазине ниже представлена диаграмма Гейна Сарсона (Рис. 1.1 и 1.2):

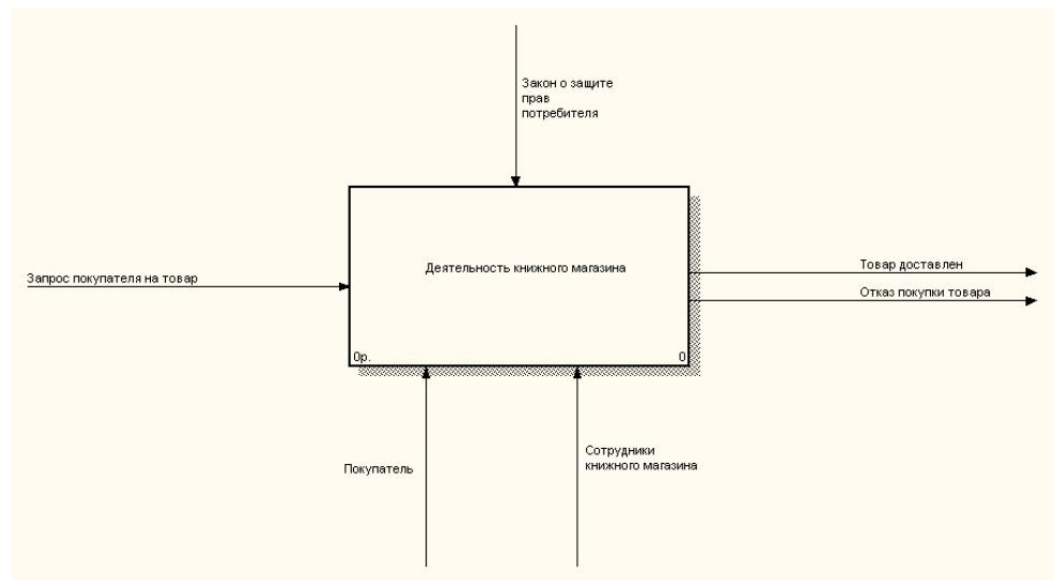


Рис 1.1 Диаграмма потоков данных интернет-магазина по Гейну Сарсону

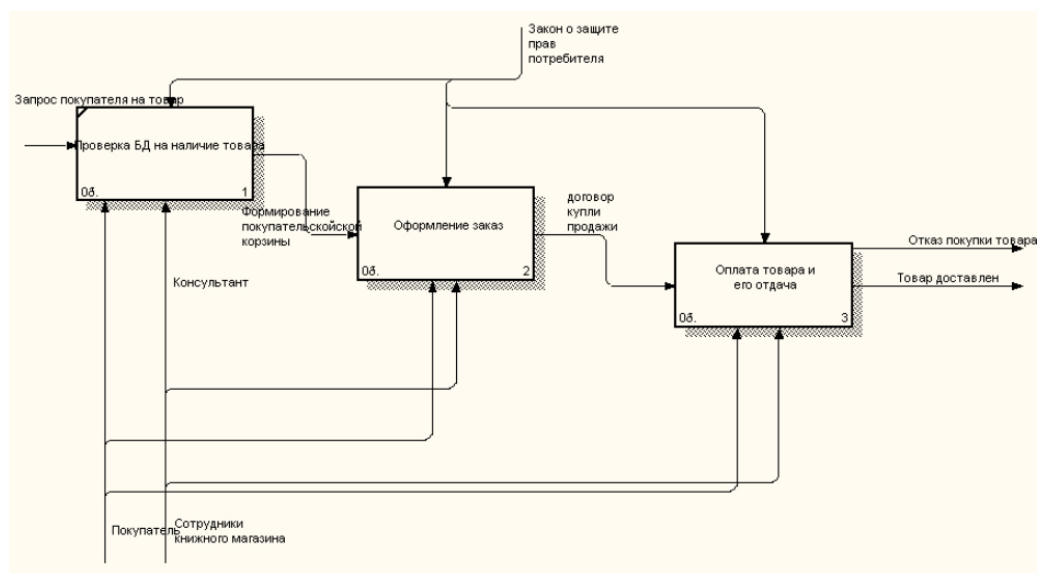


Рис 1.2 Диаграмма потоков данных интернет-магазина по Гейну Сарсону

Далее представлен список недостатков, вызванных текущей системой обмена и обработки информации:

1. Высокая трудоемкость обработки информации;
2. Сложные односторонние отношения между предприятием и клиентом, из-за отсутствия возможности просмотра и получения информации о товарах на складе;
3. Необходимость постоянного учета склада в бумажном виде со стороны персонала магазина;
4. Потеря актуальности из-за длительности процессов;
5. Низкая производительность труда в производственной сфере;
6. Низкая оперативность, снижающая качество управления объектом.

### **1.8 Анализ существующих разработок на рынке**

Стоит отметить, что данная технология уже существует на рынке информационных технологий. Данная автоматизированная система будет иметь возможность покупки книги из ассортимента схожую с аналогичными интернет-магазинами. Были рассмотрены такие системы как “Читай-город”, “Лабиринт”, “Book.ru - Книги России” (Рис. 1.2), (Рис. 1.3), (Рис. 1.4) и многие другие сайты, чьи основные принципы работы завязаны на сдачу номеров и обновлении ассортимента. Читай-город представляет собой полнофункциональную Торговую Интернет-Систему, в которой интегрированы все функции обычной торговой системы (ТС) и Интернет-торговли. Покупатель в Интернет-магазине “Читай-город” может воспользоваться обычным веб-браузером для просмотра каталога товаров и услуг. Помимо описания товара, каталог может включать изображение товара. Покупателю предоставляется информация о цене, индивидуальных скидках, наличии товара на складе. Помимо основного каталога Покупатель видит предложения по новым товарам и товарам повышенного спроса, также он может воспользоваться системой поиска необходимых товаров.

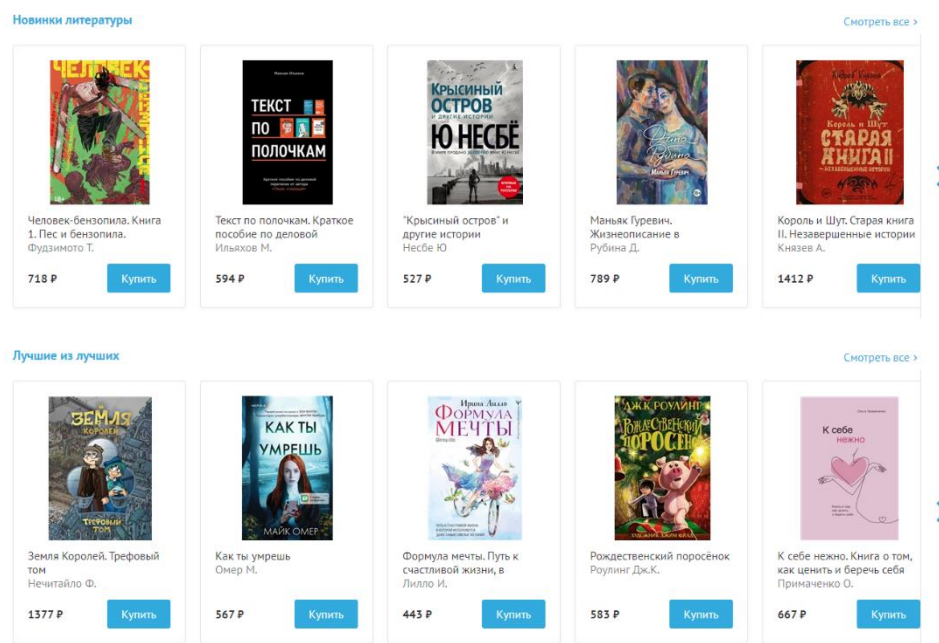


Рис. 1.2 Схема каталога товаров интернет-магазина “Читай-город”

“Лабиринт” - российский интернет-магазин. Магазин предлагает книги, музыкальную и видеопroduкцию, электронику и цифровую технику, программные продукты, товары для детей и их родителей, всё для дома, подарки и ювелирные изделия, а также эксклюзивные антикварные издания. Лабиринт предоставляет разнообразные схемы обслуживания клиентов, в том числе предусмотрена возможность "разовой покупки" - данные о Покупателе не заносятся в базу данных постоянных клиентов. Постоянные клиенты имеют возможность видеть содержимое не только текущей корзины, но и корзин, набранных ранее (даже если они не были оплачены), и могут воспользоваться "старой" корзиной как шаблоном для формирования новой. Покупатель может изменять количество отобранного товара, удалять товар из корзины, выбирать способ доставки.

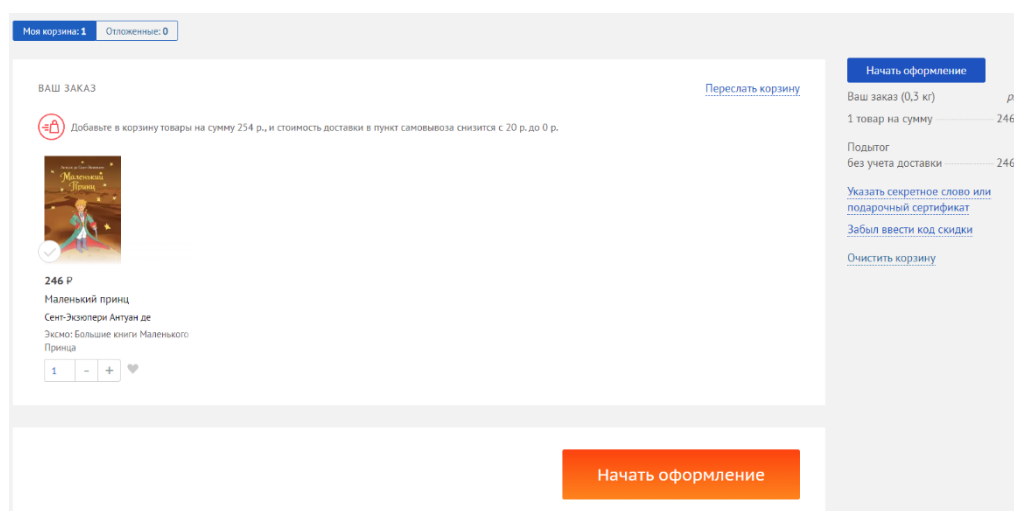


Рис. 1.3 Схема покупки товара интернет-магазина “Лабиринт”

Старейший книжный интернет-магазин в России. Начал свою работу 30 августа 1996 года по адресу symbol.ru, позже переехав на books.ru. К настоящему моменту в ассортименте магазина не только бумажные, но и электронные книги, а также электроника, софт, музыка и видео. Является официальным интернет-магазином издательства «Символ-Плюс». Главная страница сайта представлена в соответствии с рисунком 1.4

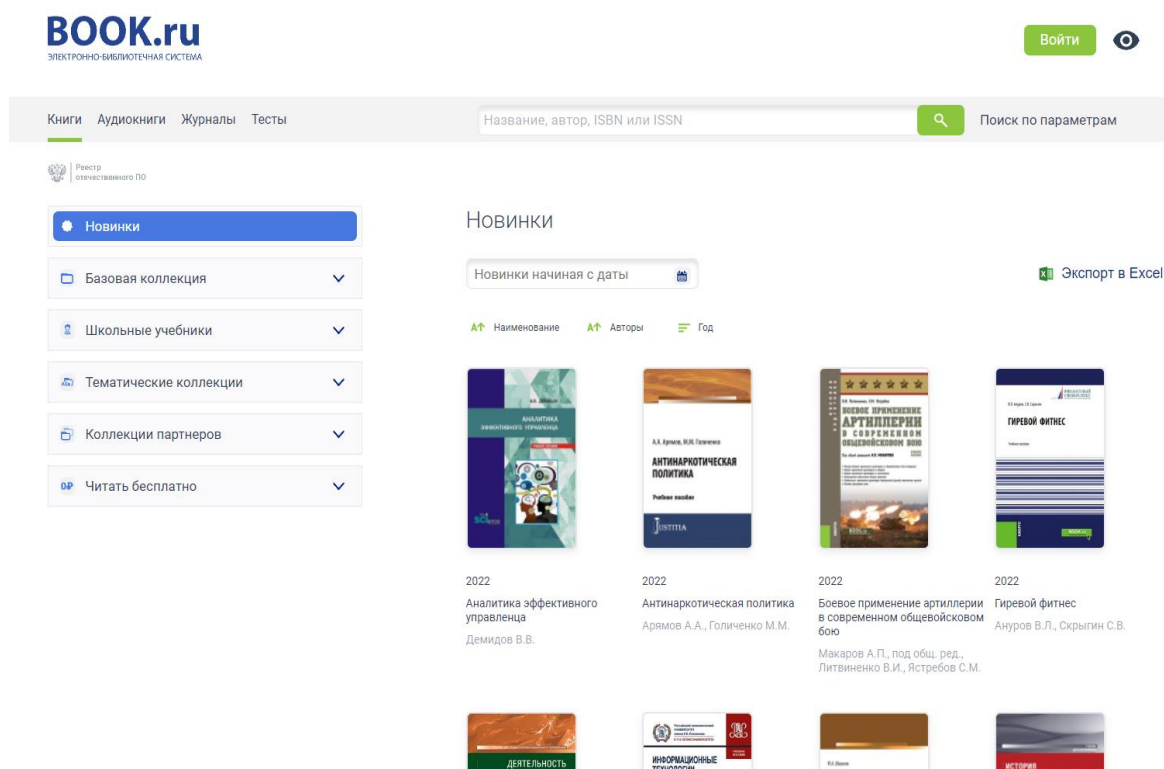


Рис. 1.4 Каталог товаров интернет-магазина “Book.ru”



Ozon.ru (OZON.py) - российский интернет-магазин. Магазин предлагает книги, музыкальную и видеопродукцию, электронику и цифровую технику, программные продукты и мультимедийные игры, товары для детей и их родителей, всё для дома, косметику, парфюмерию, подарки и ювелирные изделия, а также эксклюзивные антикварные издания. С 2009 года был запущен магазин путешествий OZON.travel, а в 2011 открыт новый раздел «Подбор книг».

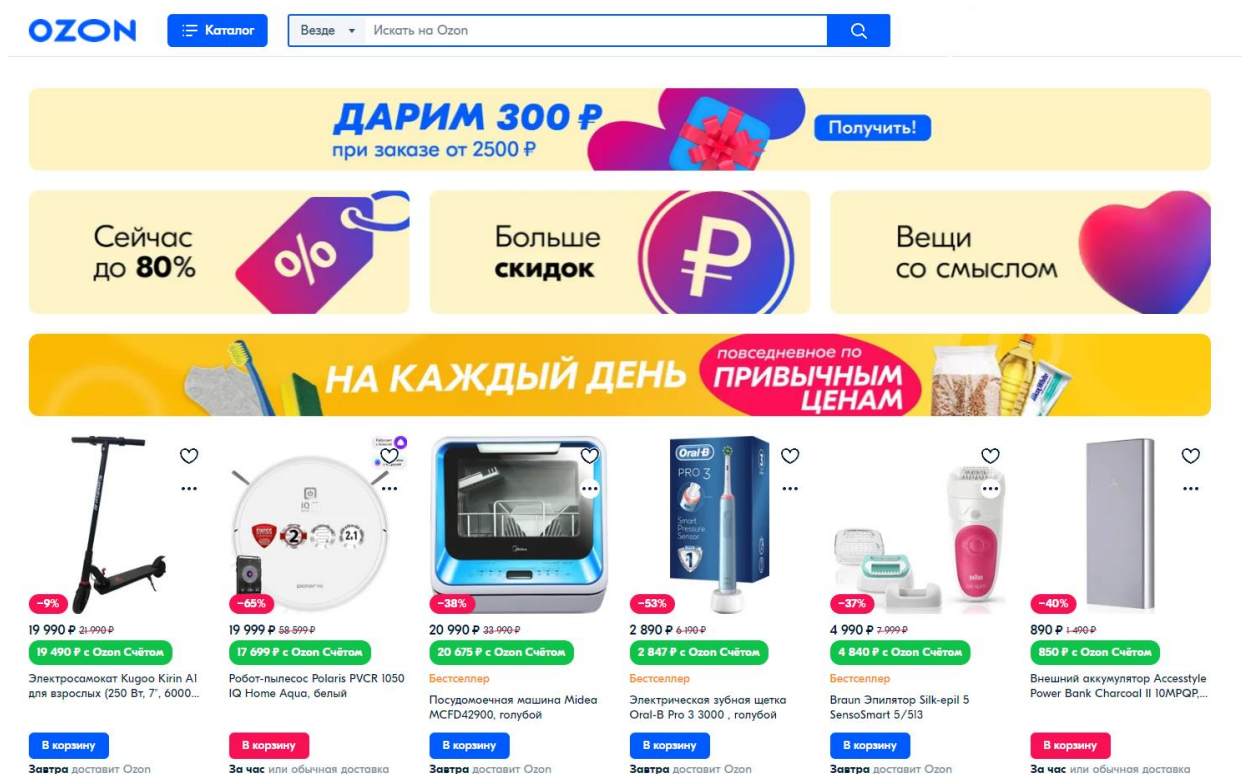


Рис 1.5 Главная страница интернет-магазина "Ozon.ru"

Преимущества современных решений:

1. Стабильность;
2. Простота использования со стороны пользователей.

Недостатки современных решений:

1. Низкая производительность;
2. Сложность использования со стороны персонала.

## 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1 Проектирование баз данных

ER-модель (Entity-relationship model или Entity-relationship diagram) – это семантическая модель данных, которая предназначена для упрощения процесса проектирования базы данных. ER-модель определяет понятия «сущность», «связь» и «атрибут».

ER-модель – это только концептуальный уровень моделирования. ER-модель не содержит деталей реализации. Для той же самой ER-модели детали ее реализации могут отличаться.

Сущность — это собирательное понятие, некоторая абстракция реально существующего объекта, процесса, явления или некоторого представления об объекте, информацию о котором требуется хранить в базе данных

База данных является одной из главных частей разрабатываемого веб-приложения. Главными целями проектирования базы данных является:

1. Обеспечение защиты и целостности данных;
2. Интерпретация и выполнение запросов;
3. Управление пользователем и распределение прав доступа;
4. Управление механизмом работы транзакции (операция или блок операции, которые совершаются вместе).

Грамотное распределение и формирование корректных данных и их связей требует создания разных сущностей данных, предназначенных для хранения определенного блока информации.

Описание сущностей, которые будут реализованы при создании базы данных:

Таблица 1. Описание сущностей

Номер сущности	Имя сущности	Определение	Описание (когда появляется)

			новый экземпляр?)
E1	Пользователь [client]	Пользователи, чьи данные хранятся на сайте.	Новый экземпляр сущности появляется после прохождения процедуры регистрации.
E2	Книги [books]	Доступные товары, содержащиеся в каталоге веб- приложения.	Новый экземпляр сущности появляется при появлении нового товара в магазине.
E3	Тест [test]	Хранится информация о пройденном тесте	Новый экземпляр сущности появляется при создании пройденного теста.
E4	Категория [category]	Обобщенная группа товаров.	Новый экземпляр сущности появляется при создании новой категории.
E5	Результат[result]	Результат пройденного пользователем теста.	Новый экземпляр сущности появляется при создании нового

			типа категории склада магазина
--	--	--	-----------------------------------

Концептуальная модель также отражает связи между сущностями.

Отношения между сущностями характеризуются глаголом, который используется при их взаимодействии. Количественный характер участия экземпляров сущностей задается типом связи (мощностью связи). Возможны следующие типы: «один к одному» (1:1), «один ко многим» (1:M), «многие к одному» (M:1), «многие ко многим» (M:M).

Связи описываются следующим образом:

1. 1:1 — Этот тип связи подразумевает, что одной записи главной таблицы соответствует только одна запись в дочерней таблице;
2. 1:M — Этот тип связи подразумевает, что одной записи главной таблицы соответствует несколько записей в дочерней таблице;
3. M:M — Это означает, что нескольким экземплярам родительской сущности может соответствовать несколько экземпляров дочерней сущности.

Для корректной работы базы данных между всеми существующими в ней сущностями необходимо установить соотношение между сущностями(связи). Данная процедура позволит создавать структуру данных и избавиться от необходимости постоянного учета и добавления одинаковых данных в одну и ту же таблицу.

Матрица связей вышеупомянутых сущностей базы данных интернет-магазина (таблица 2.):

Таблица 2. Матрица связи.

	Е1. Пользователь [client]	Е2. Книги[ books]	Е3. Тест [test]	Е4. Категория [category]	Е5. Результат [result]
Е1. Пользователь [client]					R1
Е2. Книги [books]					R2
Е3. Тест [test]		X			
Е4. Категория [category]		R4	R3		
Е5. Результат [result]		X		X	

Таблица описания связей, указанных в таблице 1 (таблица 3.):

Таблица 3. Описание связей

Номер связи	Родительская сущность	Дочерняя сущность	Имя связи	Тип связи	Мощность	Описание связи
R1	E1. Клиент [client]	E5. Результа т [result]	Имеет/И меется	Опре д.иде нт	P	Один и тот же человек может купить один и тот же товар несколько раз
R2	E2. Книги [books]	E5. Результа т [result]	Заноситс я/ Включае т в себя	Опре д.иде нт	P	Один и тот- же товар могут покупать разные люди

R3	E4. Категория [category]	E3. Тест[test]	Произво дит/ Произво дится	Опре д.неи ден.о бяз.	Z	Обусловлен тем что, на одну категорию может приходитьс я несколько товаров, и одну категорию обслуживае т несколько работников
R4	E4. Категория [category]	E2. Книги [books]	Определ яет/ Определ яется	Опре д.неи ден.о бяз	Z	Категория включает в себя множество разных товаров интернет- магазина

Таким, образом, описанные сущности и связи формируют концептуальную модель, которую можно представить в виде диаграммы

На основе разработанной концептуальной модели была реализована база данных в СУБД MySQL. Разрабатываемая платформа предполагается для использования на различных устройствах и рассчитана на одновременную работу множества пользователей, при разработке БД, поэтому качество разработки БД при учете таких требований определяется достаточной

нормализацией хранения данных при наиболее оптимизированном доступе к ним.

Ниже описаны определенные в проектируемой базе данных таблицы и типы данных их атрибутов. Указан первичный ключ таблицы.

Этап определения типов данных необходим, чтобы оптимизировать занимаемое базой данных место на хранилище, и также чтобы ускорить запросы к базе данных, исключить ситуации связанные с нарушением целостности данных.

Для удобства дальнейшей разработки, все таблицы и их атрибуты имеют англоязычное название. Для всех сущностей в качестве первичного ключа использовался искусственный атрибут – личный номер (код). Значения атрибута – целые положительные числа (порядковый номер).

Атрибут – это свойство экземпляра сущности. Атрибуты, как правило, называются также, как и домены.

Домен определяет тип и смысл данных, а также накладывает ограничения на сравнение значений.

Описание атрибутов уровня ключей представлено в таблице 4.

Таблица 4. Описание атрибутов уровня ключей

Имя сущности/Номер сущности	Имя атрибута	Имя домена	Наличие нулевого значения	Примечание
Клиент[client]/E1	ID_client	Номер	Нет	PK
	ID_book	Номер	Нет	FK
	ID_result	Номер	Нет	FK
	name	Номер	Нет	FK
Книги[books]/E2	ID_book	Номер	Нет	PK
	ID_category	Номер	Нет	PK
	ID_client	Номер	Нет	FK
	name	Номер	Нет	FK



Тест[test]/E3	ID_test	Номер	Нет	PK
	name	Номер	Нет	FK
Категория[category]/E4	ID_category	Номер	Нет	PK
	name	Номер	Нет	FK
Результат[result]/E5	ID_result	Номер	Нет	PK
	ID_book	Номер	Нет	PK
	ID_client	Номер	Нет	FK

### Концептуальная схема уровня ключей

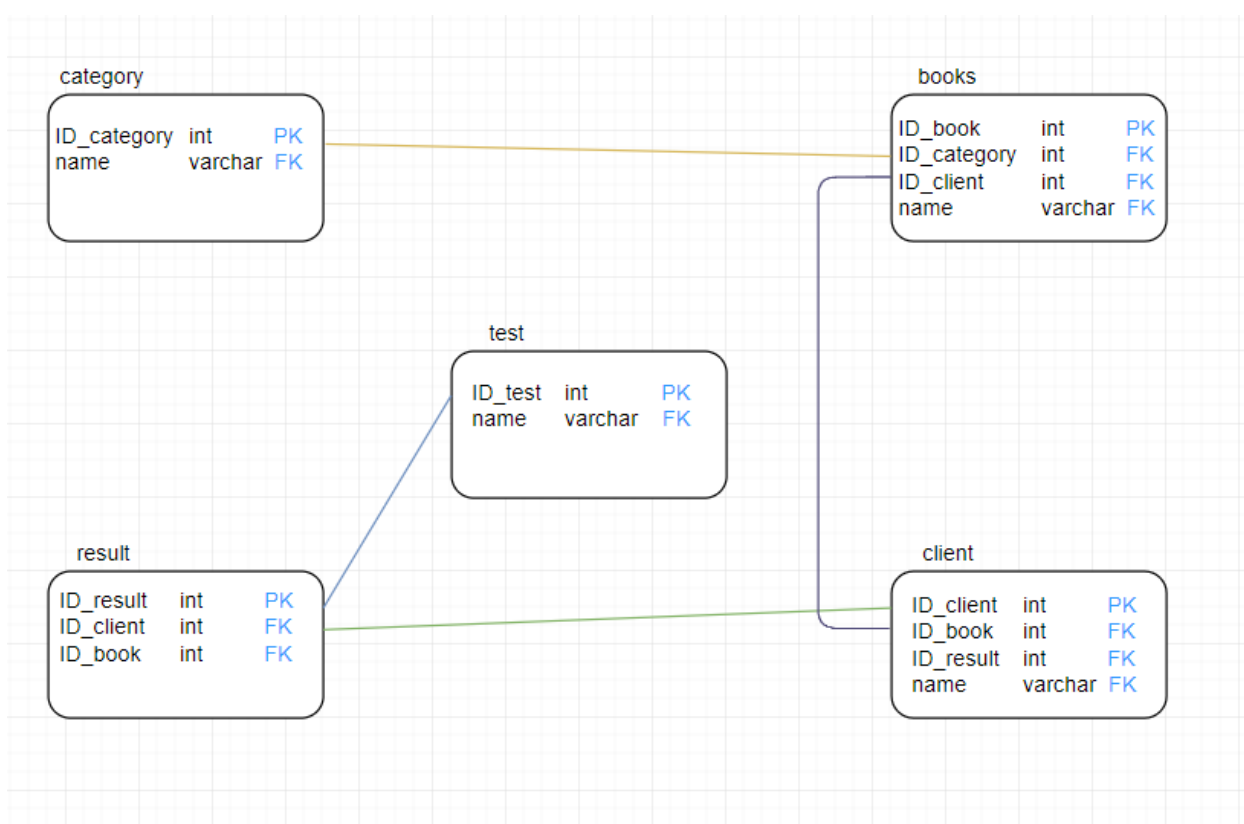


Рис. 2.2 Диаграмма уровня ключей базы данных интернет-магазина

На диаграмме уровней ключей показывается миграция ключевых атрибутов и реализуются все неопределенные связи и дополнительные сущности множества процессов. Данная диаграмма представлена на рисунке (2.2). Построение полно-атрибутной диаграммы

На полно-атрибутной диаграмме отображаются все атрибуты, в том числе и ключевые. В таблице (5) описаны все атрибуты сущностей.

Таблица 5. Описание всех атрибутов сущностей

Имя сущности/Номер сущности	Имя атрибута	Имя домена	Наличие нулевого значения	Примечание
Клиент/Е1	ID_client	Номер	Нет	РК
	ID_result	Номер	Нет	FK
	ID_book	Номер	Нет	Определяет номер книги закреплённой за пользователем
	name	Строка	Нет	Определяет имя пользователя
Книги/Е2	ID_book	Номер	Нет	РК
	ID_category	Номер	Нет	РК
	ID_client	Номер	Нет	Номер пользователя
	name	Текст	Есть	Название книги
Тест/Е3	ID_test	Номер	Нет	РК
	name	Строка	Нет	Название вопроса
Категория/Е4	ID_category	Номер	Нет	РК
	name	Строка	Нет	Название категории
Результат/Е5	ID_result	Номер	Нет	РК
	ID_book	Строка	Нет	РК
	ID_client	Текст	Нет	FK

## 2.2 Разработка физической и логической структуры сайта

Работа над сайтом предполагает определить логическую и физическую структуру для лучшего понимания сайта.

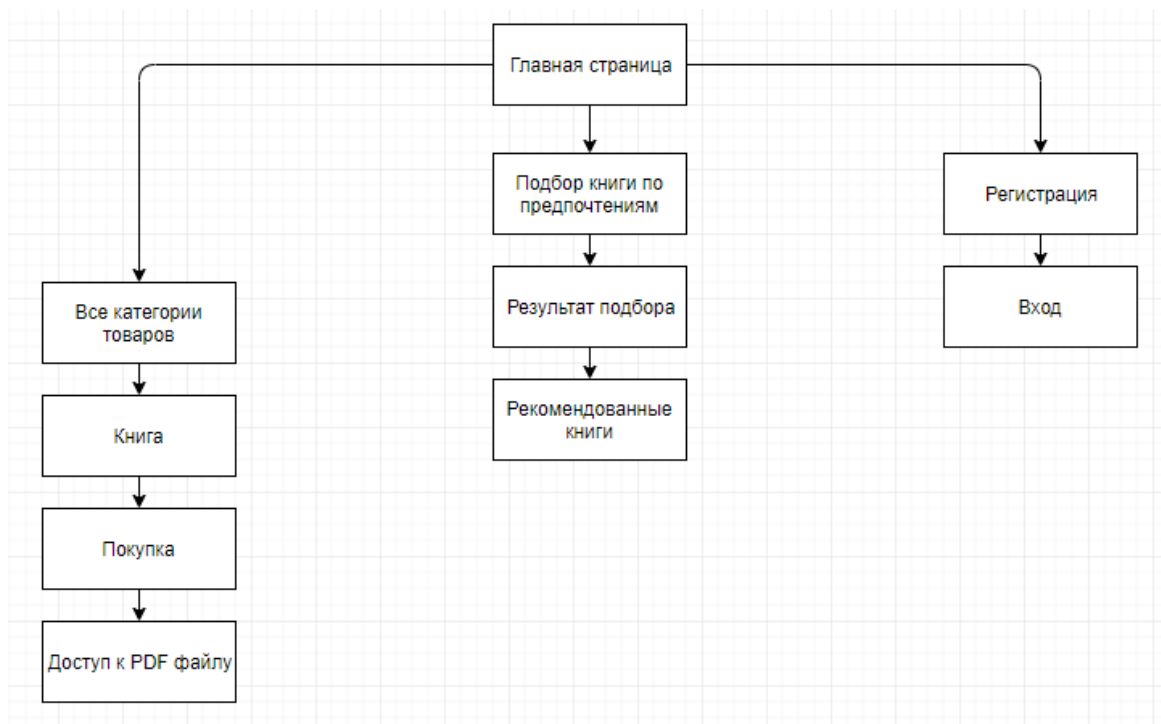


Рис. 2.4 Логическая модель веб-приложения

Логическая структура сайта – это совокупность всех страниц на сайте, расположенных с учётом иерархии, то есть взаимосвязь страниц, в которых прослеживается их принадлежность к разделам, категориям и подкатегориям (Рис. 2. 4.). А физическая структура подразумевает алгоритм размещения физических файлов по директориям папок и файлов, в которых опубликован сайт (Рис. 2. 5.).

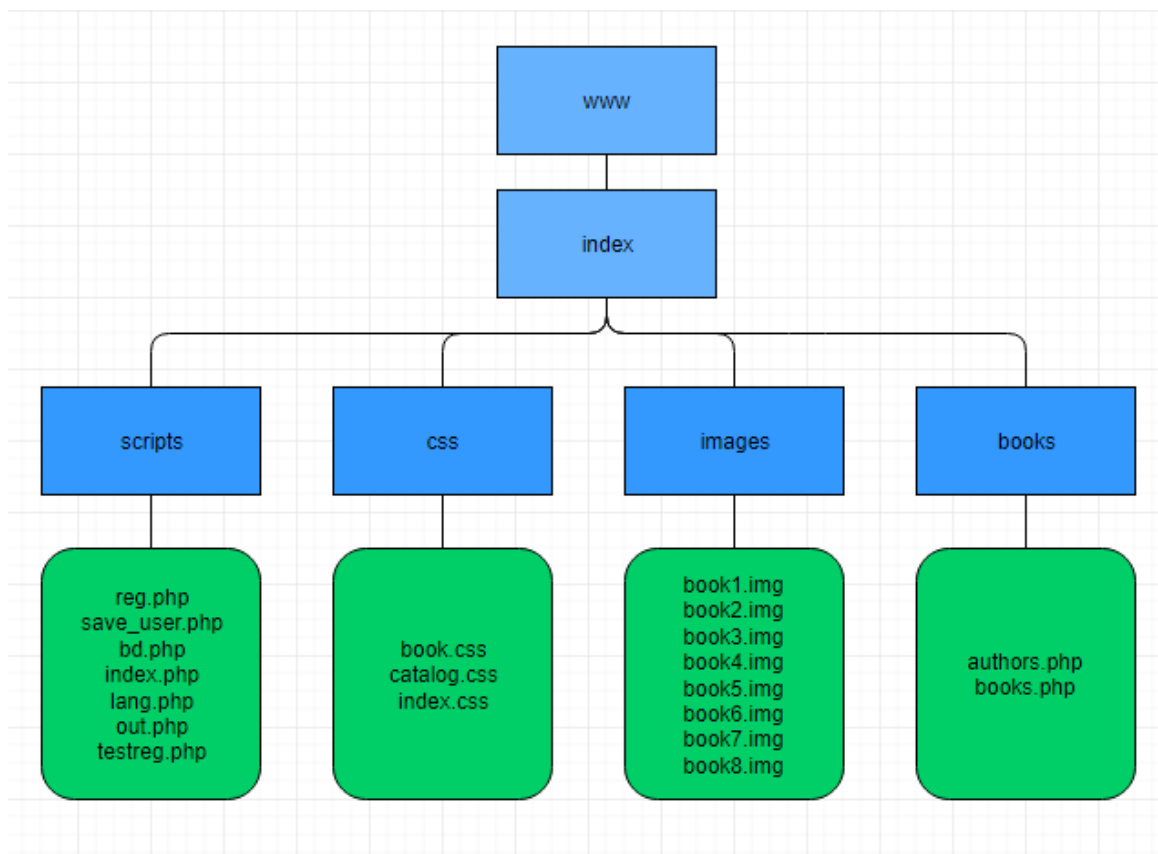


Рис. 2.5 Физическая модель веб-приложения

## 2.3 Структура сайта

Для полного представления о сайте и лучшего его понимания в графическом виде, ниже представлена визуальная карта сайта. (Рис. 2.6)

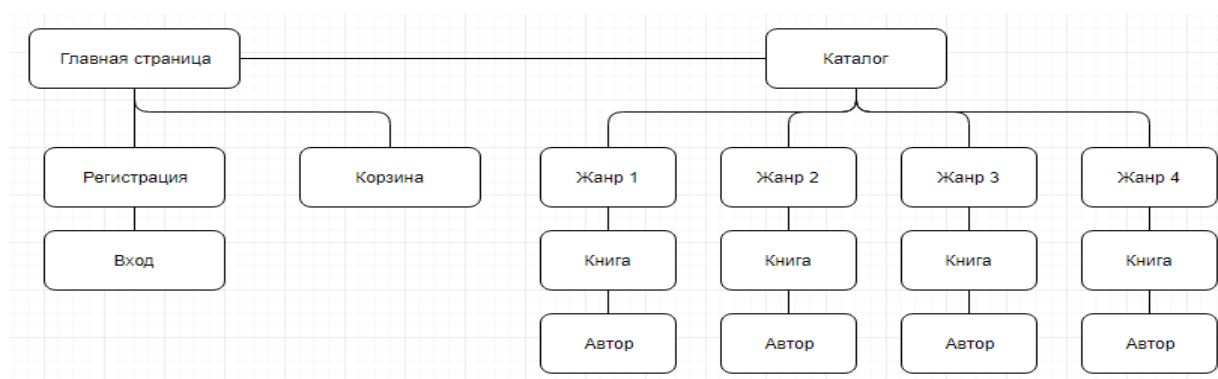


Рис. 2.6 Карта интернет-магазина

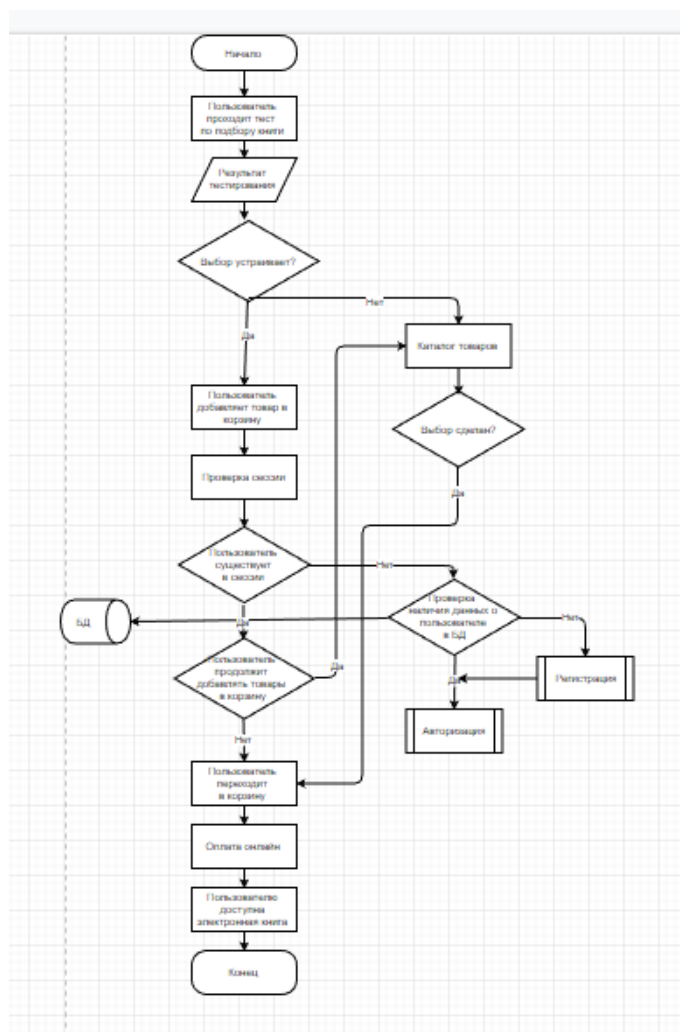


Рис. 2.6.1 Блок-Схема

## 2.4 Создание дизайн-макета интернет-магазина

Графическая схема в виде основных блочных элементов страницы называется макетом сайта. Он необходим для наиболее точного и наглядного представления о внешней статической картинке сайта. Ниже представлен макет главной страницы сайта (Рис. 2.7.)



Рис. 2.7 Дизайн-макет главной страницы интернет-магазина

Для того, чтобы иметь общее представление о разрабатываемой автоматизированной системе, ниже представлен список страниц, которые это веб-приложение будет содержать.

#### Главная страница

Главная страница сайта должна содержать графическую часть, навигационное меню сайта, а также область контента для того, чтобы посетитель сайта с первой страницы мог получить вводную информацию о продукции компании, а также ознакомиться с последними новостями. Контактная область первой страницы должна делиться на следующие разделы:

1. вступительная статья о компании со ссылкой «подробнее», ведущей на раздел «О компании»;

2. краткая контактная информация - наименование, телефон и e-mail компании;

Главная страница сайта:

1. Авторизация на сайте:

2. Регистрация на сайте:

3. вверху страницы отображаются облегченная навигационная панель, которая обеспечивает переход к основным пунктам меню сайта

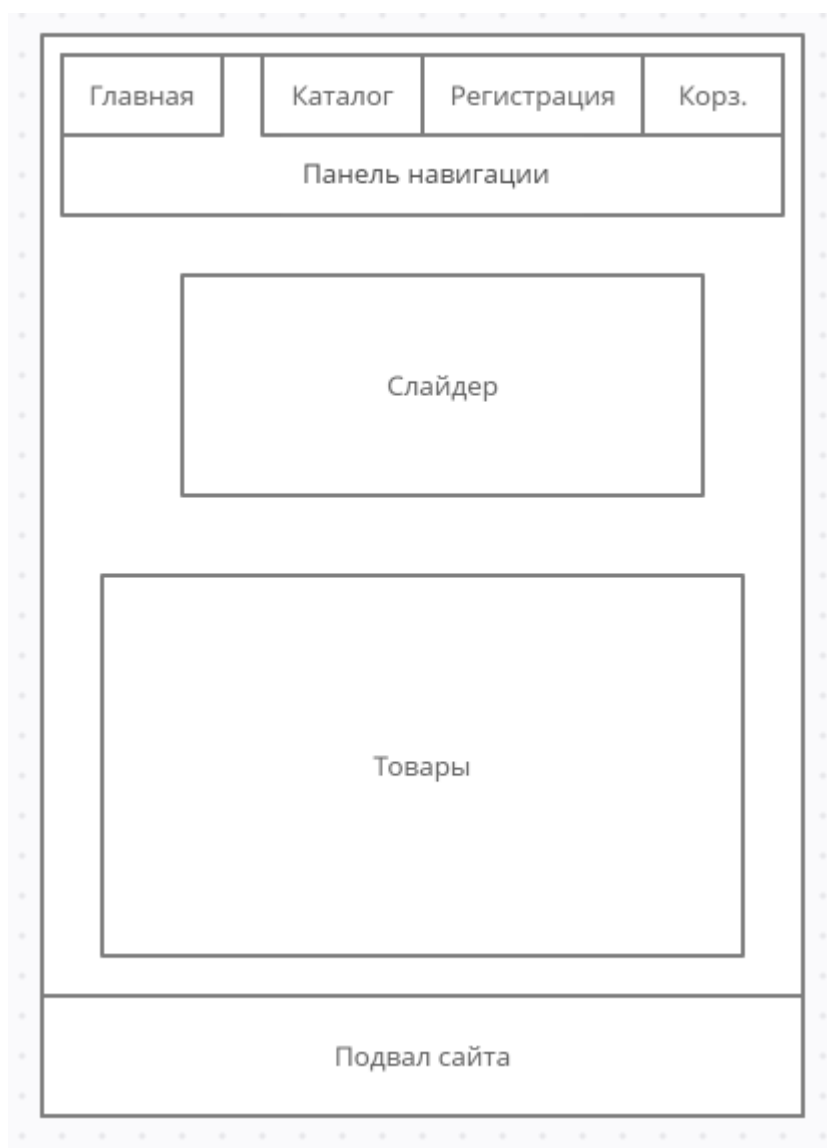


Рис. 2.8 Макет главной страницы сайта

Страница «Регистрация»

Данная страница является доступной только авторизованным пользователем сайта. Содержит в себе информацию о пользователе.

Страница «Вход»

Страница, предназначенная для авторизации пользователей.

Страница «Регистрация»

Страница, предназначенная для регистрации пользователя.

Страница «Корзина»

Данная страница доступна только авторизованным пользователям. Предназначена для покупки товара.

Описание интерфейса (состав Web-страниц и используемых на них графических и мультимедийных файлов)

Далее содержится список и описание всех элементов, содержащихся на каждой странице веб-приложения.

Главная страница:

Общий вид страницы:

1. Header:
2. Логотип;
3. Панель навигации:
4. Регистрация;
5. Главная;
6. О компании;

Основное:

1. Ассортимент (шаблон);
2. Название категории товара;
3. Товар (шаблон);
4. Изображение;
5. Описание.

Общий вид страницы:

Header:

1. Логотип;



## 2. Панель навигации:

Главная страница:

1. О компании;
2. Каталог;
3. Регистрация;

Основное:

1. Оглавление страницы;
2. Блок с товарами;
3. Название;
4. Картинка;
5. Информация;

Footer:

1. О нас.
2. Товары
3. Адреса магазинов
4. Помощь
5. Контакты

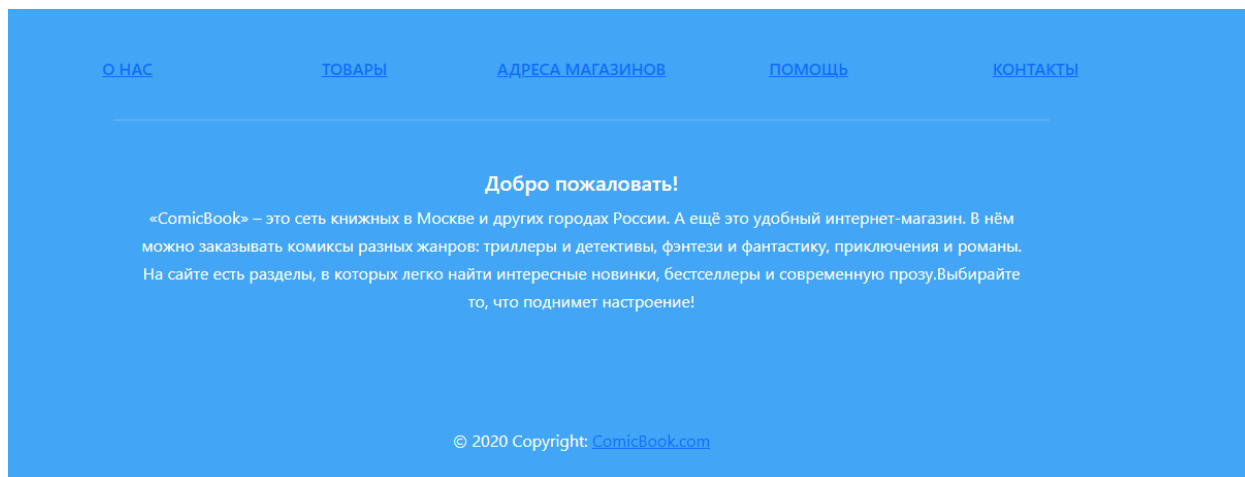


Рис. 2.9 Макет раздела «Подвал сайта»

Страница «Подвал сайта» представлена на рис. 2.9

Страница «Главная»

Общий вид страницы:

Header:

- Логотип;

Панель навигации:

- Главная страница;
  1. Логотип сайта;
  2. Каталог;
  3. Регистрация;

Основное:

1. Оглавление страницы;
2. Блок с товарами;
3. Название;
4. Картинка;
5. Информация;

Footer:

1. Контактная информация.
2. Товары
3. Адреса магазинов
4. Помощь

Общий вид страницы:

Header:

1. Логотип;

Панель навигации:

1. Главная страница;
2. О компании;
3. Контакты;
4. Регистрация;

Основное:

1. Оглавление страницы;
2. Блок с товарами;
3. Название;
4. Картинка;

## 5. Информация;

Раздел, доступен для зарегистрированных пользователей.

В данном разделе авторизованному посетителю доступны информация о пользователе портала, либо свои личные данные. Редактирование раздела любого пользователя доступно членам группы «Администраторы». Изменение информации данного раздела производится путём заполнения данных формы, состоящей из полей:

- Фамилия\* - текстовое поле
- Имя\* - текстовое поле
- E-mail адрес - текстовое поле
- Поля доступные для изменения зависят от уровня доступа:
- Пользователи группы «Администраторы»
- все поля
- Пользователь - собственник информации
- E-mail адрес - текстовое поле
- Псевдоним - текстовое поле
- О себе - текстовое поле
- Остальные авторизованные пользователи не имеют прав на

редактирование.

- Страница «Личный кабинет»

Общий вид страницы:

Header:

1. Логотип;
2. Панель навигации:
3. Главная страница;
4. Каталог;
5. Регистрация

- Основное:

Вход:

1. Логин;
2. Пароль;
3. Войти.

Регистрация:

1. Логин;
2. Пароль.

Блок информации о покупке товара (шаблон):

1. Категория товара;
2. Цена товара;
3. Информация о товаре.

Footer:

1. Контактная информация.
2. Товары
3. Адреса магазинов
4. Помощь
5. Контакты

Основное:

Форма регистрации:

1. Логин;
2. Пароль;
3. Кнопка регистрации.

Footer:

1. Кнопка возвращения на главную страницу.
2. Авторизация
3. Пользователи могут авторизоваться на любой странице портала с помощью специальной формы авторизации. Форма содержит:
  1. Текстовое поле для ввода логина пользователя
  2. Кнопку отправки формы
  3. Данные для доступа (авторизации):
    4. Пароль

5. строка, содержащая от 8 символов, состоящая из A-z, 0-9.
6. Оформление заказа
7. Страница «Покупка товара/Спасибо за покупку!»

Общий вид страницы:

Header:

1. Логотип;

Основное:

1. GIF-изображение;
2. Кнопка перехода в профиль.

Footer:

1. Кнопка возвращения на главную страницу.

## 2.5 Верстка страниц

Для создания статической визуальной части используется язык вёрстки HTML с использованием каскадных таблиц CSS. Для упрощения и сокращения процесса верстки страниц используется определенный алгоритм. К примеру, разработка шапки страницы(header), разработка подвала страницы/footer), разработка основной зоны контента(main). Header и footer сайта могут быть аналогичными на всех страницах. Именно поэтому используются шаблоны (Рис. 2.5). В приложении 1 представлены код верстки шапки и подвала сайта.

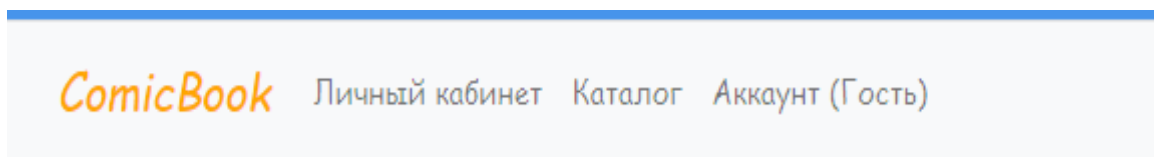


Рис. 2.5 Листинг HTML-код шапки интернет-магазина

Для дизайна стилей каждого элемента на странице используется таблица каскадных стилей CSS. Для характеристики каждого элемента были использованы классы и селекторы, которые помогают настроить всё необходимое: размер, цвет, обводка, стиль и т. д. (Рис. 2.6).

```

.button {
  background-color: #00bfff;
  border-radius: 4px;
  color: white;
  padding: 15px 40px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 14px;
  margin: 4px 2px;
  cursor: pointer;
}

body{
  width: 100%;
  max-width: 100%;
  background: url(https://вебджем.рф/800/600/https://krot.info/uploads/posts/2021-01/1610159696_29-p-nezhnii-pastelnii-fon-40.jpg) no-repeat;
  -moz-background-size: 100%; /* Firefox 3.6+ */
  -webkit-background-size: 100%; /* Safari 3.1+ и Chrome 4.0+ */
  -o-background-size: 100%; /* Opera 9.6+ */
  background-size: 100%; /* Современные браузеры */
  margin: 0; /* Убираем отступы */
  height: 100%; /* Высота страницы */
  background-size: cover; /* Фон занимает всю доступную площадь */
}

.footer,
.push {
  height: 200px;
}
.footer {
  background: #42A5F5;
  color: white;
  line-height: 50px;
  padding: 0 20px;
}

nav {
  font-family: 'Kaushan Script', cursive;
  font-size: 1em;
}

img{
  max-width: 100%;
}

.thumb img {
  border: 2px solid #55c5e9; /* Рамка вокруг фотографии */
  padding: 2px; /* Расстояние от картинки до рамки */
  background: #666; /* Цвет фона */
  margin-right: 5px; /* Отступ справа */
  margin-bottom: 5px; /* Отступ снизу */
}

.sim-slider {
  max-width: 1000px;
  min-width: 1000px;
  margin: 20px auto;
}

```

Рис. 2.6 Используемые классы и селекторы в CSS

Теперь по созданной структуре необходимо сверстать главную страницу с помощью фреймворка Bootstrap Font Awesome и jQuery, а также добавить кнопку для прохождения авторизации, которая перенаправит пользователя на форму авторизации. В данном случае этими кнопками являются “Авторизоваться” и “Зарегистрироваться”. Изначально необходимо создать базу html страницы, а если точнее, то необходимо заполнить тэг head информацией о сайте, а также сделать ссылки на подключенные фреймворки. Пример готового тэга head изображен на рисунке 2.7

```

<!doctype html>
<?php session_start(); ?>
<html lang="ru" dir="ltr">
<head>
  <!-- Обязательные метатеги -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4F7kqJByhZMI3AhiU" crossorigin="anonymous">

  <!-- fancyBox CSS -->
  <link href="/examples/vendors/fancybox/jquery.fancybox.min.css" rel="stylesheet">
  <link rel="stylesheet" type="text/css" href="css\home.css">

  <link rel="stylesheet" type = "text/css" href="styles/search.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
  integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
  crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
  integrity="sha384-U02eT0CpHqD5JQ6h7Jty5KVphtPhzWj9WO1cL1HTMga3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
  integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEeFFnJGzIxFOsf4x0xIM+807jRM" crossorigin="anonymous"></script>
  <title>Главная</title>
</head>

```

Рис. 2.7 Структура кода в тэге head

Теперь необходимо разобрать что к чему относится:

1. meta charset – установка кодировки, которая в дальнейшем позволит правильно отображать используемые символы.
2. meta viewport – используется для корректного отображения на мобильных телефонах.
3. meta description – добавляет описание к сайту, которое видно в поисковике.
4. link – ссылка на подключаемый предмет. В данном примере можно видеть подключение ярлыка-картинки для сайта, подключение css файлов, подключение шрифтов.
5. script – добавляет скрипт, функции которого можно будет использовать на самой странице. При этом лучше всего добавлять скрипты в самый конец файла, чтобы страница быстрее загружалась у пользователей, но иногда приходится использовать js скрипт в начале html-страницы.
6. title – установка названия сайта, которое будет отображаться во вкладке.
7. div – Элемент <div> является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого.

Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут class или id с именем селектора.

8. ul – устанавливает маркированный список. Каждый элемент списка должен начинаться с тега <li>. Если к тегу <ul> применяется таблица стилей, то элементы <li> наследуют эти свойства.
9. li – определяет отдельный элемент списка. Внешний тег <ul> или <ol> устанавливает тип списка - маркированный или нумерованный.
10. p – Определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.
11. a – Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.
12. h – HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег <h1> представляет собой наиболее важный заголовок первого уровня, а тег <h6> служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги <h1>,...,<h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них



другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

После заполнения тэга `head` нужно перейти к написанию самой страницы. Для написания html страницы будут использоваться классы из фреймворка Bootstrap, которые существенно упростят задачу для написания страницы, а также адаптируют сайт под телефон. Для создания странички будет использоваться много Bootstrap классов, но не на все из них стоит обратить внимание:

- `container` – этот класс необходим для обертывания содержимого сайта и размещения сетки.
- `row` – это обертки для столбцов. Каждый столбец имеет горизонтальное заполнение для управления пространством между ними.
- `col` – используется для размещения блока в созданной сетке. Всего 12 клеток.

Все остальные классы не заслуживают внимания и отвечают за цвет блока, цвет текста в блоке и так далее. Для создания шапки сайта потребуются логотип, а также небольшой блок, в котором будет информация о сайте. Код готовой шапки сайта представлен на рис. 2.8.

```
<html lang="ru" dir="ltr">
<head>
  <!-- Обязательные метатеги -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- fancyBox CSS -->
  <link href="/examples/vendors/fancybox/jquery.fancybox.min.css" rel="stylesheet">
  <link rel="stylesheet" type="text/css" href="css/main.css">

  <link rel="stylesheet" type="text/css" href="css/main.css">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/fancybox/3.5.7/jquery.fancybox.min.js"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
  <title>Главная</title>
</head>
```

Рис. 2.8 Структура кода шапки страницы

Теперь необходимо дописать оставшуюся часть страницы, а точнее – контекстную часть и футер страницы. Код изображен на рисунке 2.9

```
<body>
  <!-- Дополнительный JavaScript; выберите один из дв...

  <!-- Вариант 1: Bootstrap в связке с Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap...

  <!-- Вариант 2: Bootstrap JS отдельно от Popper
  <script src="https://cdn.jsdelivr.net/npm/@popperjs...
  <script src="https://cdn.jsdelivr.net/npm/bootstrap...
  -->

  <h1 align="center">Лучшие комиксы</h1>
  <div class="sim-slider" align="center">
    <ul class="sim-slider-list">
      <li></div>
    <div class="sim-slider-arrow-right"></div>
    <div class="sim-slider-dots"></div>
  </div>
  <h2 align="center">Новинки</h2>

  <p class="thumb" align="center">
    

  <h2 align="center">Лучшие из лучших</h2>

  <div class="container">
    <div class="row row-cols-4">
      <div class="col">
        <a data-fancybox="gallery" href="https://images-r...
          Dr. Stone Том 12</p>
        <p align="center" class="lead"><strong>876</strong></p>
        <div align="left">
          <a href="home\drstonehome.html" class="butt...
        </div>
      </div>
      <div class="col">
        <a data-fancybox="gallery" href="https://s1.livef...
```

Рис. 2.10 Код тэга body и footer

Далее необходимо написать форму авторизации для пользователей. Пример кода и визуальное отображение в системе представлен на рисунках 2.11 и 2.12

```

<?php
// вся процедура работает на сессиях. Именно в ней хранятся данные пользователя, пока он находится на сайте. Очень важно запустить их в самом
session_start();
?>
<html>
<head>
<title>Главная страница</title>
</head>
<body>
<h2>Главная страница</h2>
<form action="testreg.php" method="post">

<!--**** testreg.php - это адрес обработчика. То есть, после нажатия на кнопку "Войти", данные из полей отправятся на страничку testreg.php мет
<p>
<label>Ваш логин:<br></label>
<input name="login" type="text" size="15" maxlength="15">
</p>

<!--**** В текстовое поле (name="login" type="text") пользователь вводит свой логин ****> -->

<p>
<label>Ваш пароль:<br></label>
<input name="password" type="password" size="15" maxlength="15">
</p>

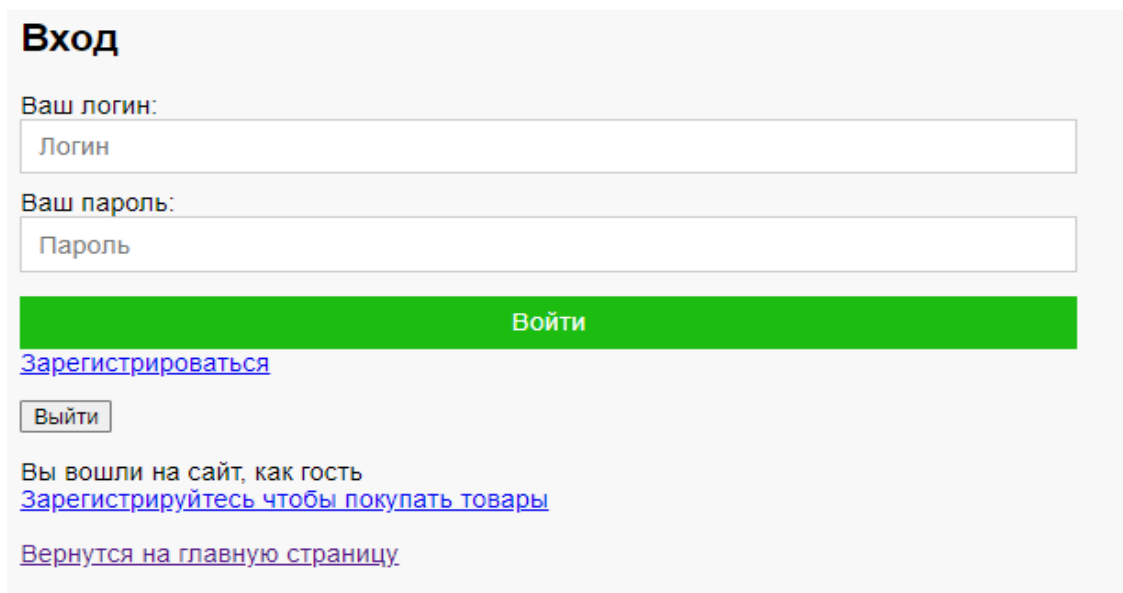
<!--**** В поле для паролей (name="password" type="password") пользователь вводит свой пароль ****> -->

<p>
<input type="submit" name="submit" value="Войти">
<!--**** Кнопочка (type="submit") отправляет данные на страничку testreg.php ****> -->
<!--**** ссылка на регистрацию, ведь как-то же должны гости туда попадать ****> -->
<a href="reg.php">Зарегистрироваться</a>
</p>
</form>
<form class="" action="out.php" method="post">
| <input type="submit" name="submit" value="Выйти">
</form>
<?php
// Проверяем, пусты ли переменные логина и id пользователя
if (empty($_SESSION['login']) or empty($_SESSION['id']))
{
// Если пусты, то мы не выводим ссылку
echo "Вы вошли на сайт, как гость<br><a href='#>Эта ссылка доступна только зарегистрированным пользователям</a>";
}
else
{
// Если не пусты, то мы выводим ссылку
echo "Вы вошли на сайт, как ".$_SESSION['login'].<br><a href='../index.php'>Эта ссылка доступна только зарегистрированным пользователям</a>";
}

echo "<br> <br> <a href='../index.php'>Вернутся на главную страницу</a>";
?>
</body>
</html>

```

Рис. 2.11 Код формы авторизации



**Вход**

Ваш логин:

Ваш пароль:

**Войти**

[Зарегистрироваться](#)

**Выйти**

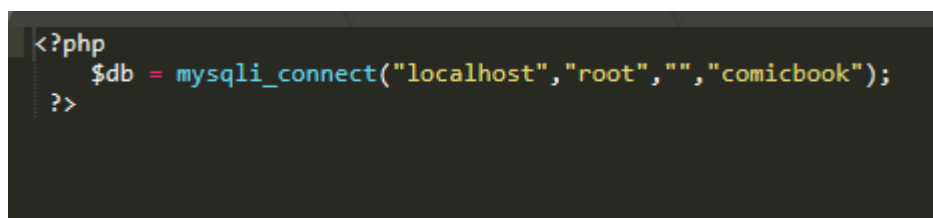
Вы вошли на сайт, как гость

[Зарегистрируйтесь чтобы покупать товары](#)

[Вернуться на главную страницу](#)

Рис. 2.12 Форма авторизации

Все остальные страницы необходимо создать по подобной технологии. С помощью фреймворка Bootstrap это не доставит неудобств. Для верстки страниц необходимо подготовить PHP код для взаимодействия с базой данных. Данный код позволит в будущем легко обращаться к базе данных, а также упростит работу с базой данных. Для этого создадим PHP файл для подключения к Базе данных и назовем его bd.php. Результат существующего кода изображен на рисунке 2.13



```
<?php
$db = mysqli_connect("localhost","root","","comicbook");
?>
```

Рис. 2.14 Структура кода файла bd.php

Для ссылок на другие страницы были использованы псевдоклассы, с помощью которых можно изменять вид ссылки при нажатии или наведении на неё (Рис.2.15).

```

.sim-slider {
    max-width: 500px;
    min-width: 320px;
    margin: 20px auto;
    padding: 30px 50px;
    border: 1px solid #ccd;
    background-color: white;
}

/* General styles */
.sim-slider {
    position: relative;
}

.sim-slider-list {
    margin: 0;
    padding: 0;
    list-style-type: none;
}

.sim-slider-element {
    width: 100%;
    transition: opacity 1s ease-in;
    opacity: 0;
    position: absolute;
    z-index: 2;
    left: 0;
    top: 0;
    display: block;
}

/* Navigation item styles */
div.sim-slider-arrow-left,
div.sim-slider-arrow-right {
    width: 22px;
    height: 40px;
}

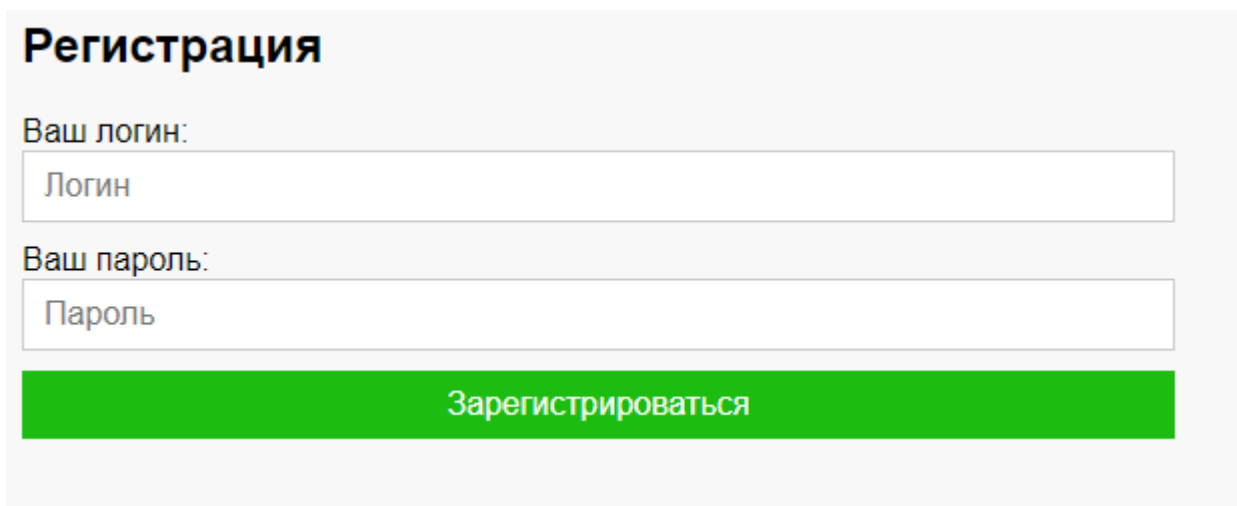
```

Рис. 2.15 Используемые псевдоклассы в CSS

### 2.5.1 Описание программных модулей

Для реализации многих функций интернет-магазина используется модульное программирование. Ниже представлен реализованные программные модули веб-приложения:

1. Модуль «Регистрация»



The registration form is titled "Регистрация" in bold black text. It contains two input fields: "Ваш логин:" with a placeholder "Логин" and "Ваш пароль:" with a placeholder "Пароль". Below these fields is a large green button with the text "Зарегистрироваться" in white.

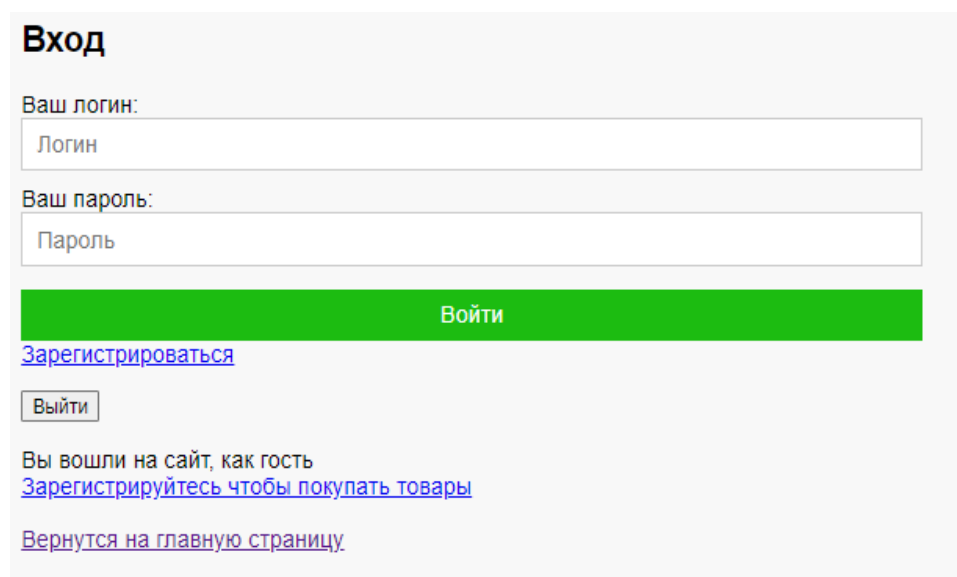
Рис. 2.16 Страница регистрации

Модуль, соединенный с базой данных, выполняющий запись новых пользователей в базу данных для того, чтобы они могли покупать книги в интернет-магазине.

При переходе на страницу регистрации будет представлена форма для заполнения данных.

#### 1. Модуль «Авторизация»

Модуль, соединенный с базой данных, выполняющий проверку данных, введенных пользователем для входа в личный кабинет интернет-магазина.

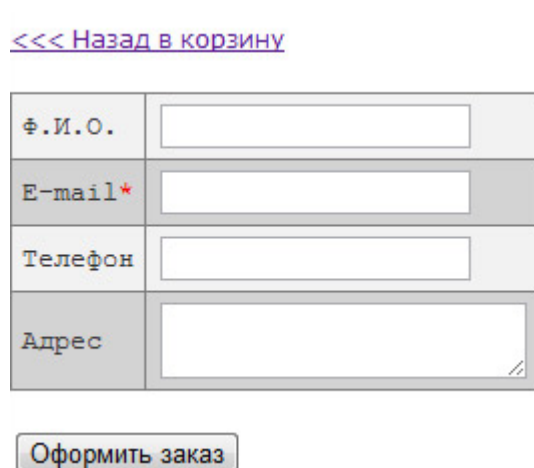


The authorization form is titled "Вход" in bold black text. It contains two input fields: "Ваш логин:" with a placeholder "Логин" and "Ваш пароль:" with a placeholder "Пароль". Below these fields is a large green button with the text "Войти" in white. Below the button is a blue link "Зарегистрироваться". Below the link is a small button with the text "Выйти". Below the "Выйти" button is the text "Вы вошли на сайт, как гость" followed by a blue link "Зарегистрируйтесь чтобы покупать товары". At the bottom is a blue link "Вернуться на главную страницу".

Рис. 2.17 Страница авторизации

#### 1. Модуль «Покупка товара»

Модуль, соединенный с базой данных, выполняющий покупку книги, выбранных пользователем.



<<< Назад в корзину

Ф.И.О.	<input type="text"/>
E-mail*	<input type="text"/>
Телефон	<input type="text"/>
Адрес	<input type="text"/>

Оформить заказ

Рис. 3.7 Страница покупки

### 2.5.2 Логика работы программы

Изначально создаётся главная страница сайта, а затем добавляются все разделы, категории и подкатегории в соответствии с картой сайта. Так как на этапе верстки были созданы статические страницы сайта, следующим этапом является подключение базы данных с помощью функций, которые представлены в приложении 1 «Листинг функций, используемых в интернет-магазине».

Поскольку веб-приложение взаимосвязано с базой данных, большинство функций имеет SQL-запросы, с помощью которых все данные, вводимые пользователем, обрабатываются, сохраняются и обновляются в базе данных, также как параллельно с этим из базы данных экспортируются данные в интернет-магазине. В SQL-запросах используются основные функции, такие как SELECT, INSERT INTO, UPDATE, DELETE, INNER JOIN (Рис. 3.5)

```
$result_query_select = $mysqli->query("SELECT * FROM `users` WHERE email = '". $email.'" AND password = '". $password.'"");  
if(!$result_query_select){  
    $_SESSION["error_messages"] .= "<p class='message_error' >Ошибка запроса на выборке пользователя из БД</p>";  
    header("HTTP/1.1 301 Moved Permanently");  
    header("Location: " . $address_site . "/form_auth.php");  
    exit();  
}
```

Рис. 2.18 Использование SQL-запросов

## 2.6 Тестирование веб-системы

Для проверки функциональности системы были внесены тестовые данные в таблицы:

1. Пользователь [client]
2. Тест [test]

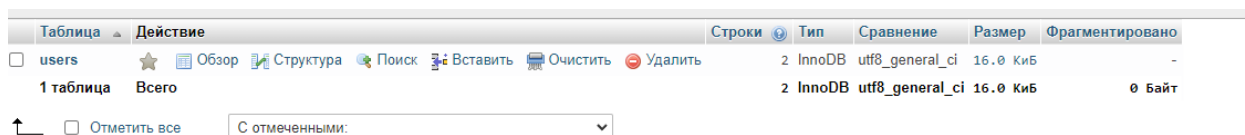
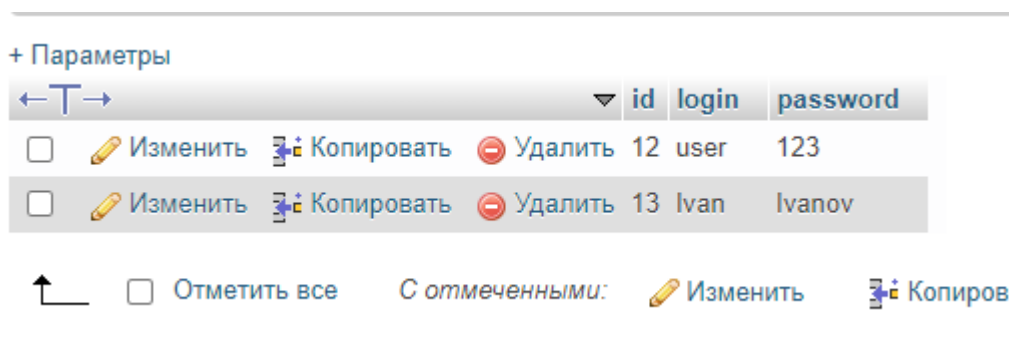


Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
users	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16.0 Киб	-
1 таблица	Всего	2	InnoDB	utf8_general_ci	16.0 Киб	0 Байт

Рис. 3.8 Тестовые данные

Тестовые данные представлены на рисунке 3.8 и 3.9



+ Параметры			
	id	login	password
<input type="checkbox"/> Изменить	12	user	123
<input type="checkbox"/> Изменить	13	Ivan	Ivanov

Рис. 2.19

### 2.6.1 Тестирование системы регистрации

При вводе правильного логина и пароля система перенаправляет на страницу личного кабинета пользователя. В ином случае, будет высвечено окно, говорящее об ошибке (Рис. 2.20)

Вы ввели не всю информацию, вернитесь назад и заполните все поля!

Рис. 2.20 Окно ошибки ввода пустого поля

Если пользователь некорректно введет пароль, появится сообщение об ошибке с уточнением использования необходимых символов (Рис.2.21)

Вы ввели не всю информацию, вернитесь назад и заполните все поля!

Рис. 2.21 Окно ошибки ввода при некорректном Логине\Пароле



Если Логин\Пароль введены некорректно, появляется сообщение об этой ошибке (Рис. 2.22).

### **2.6.2 Тестирование системы авторизации**

Извините, введённый вами login или пароль неверный.  
[Вернуться на главную страницу](#)

Рис. 2.22 Окно ошибки ввода некорректного пароля

После ввода корректных Логина и Пароля пользователь будет перенаправлен на главную страницу. В ином случае, будет высвечено окно, говорящее об ошибке (Рис. 2.22).

## **2.7 Руководство пользователя**

Данное руководство описывает возможные сценарии взаимодействия с веб-платформой. Приветственная страница платформы предоставляет возможность регистрации аккаунта и формирование запроса к администраторам платформы.

### **Главная страница**

На главной странице представлены наиболее популярные книги весь каталог существующей продукции в данном интернет-магазине. Также находятся несколько функций:

1. «Каталог»
2. Поиск по каталогу
3. «Корзина»
4. «Аккаунт (Зарегистрироваться или Войти)»
5. Тест

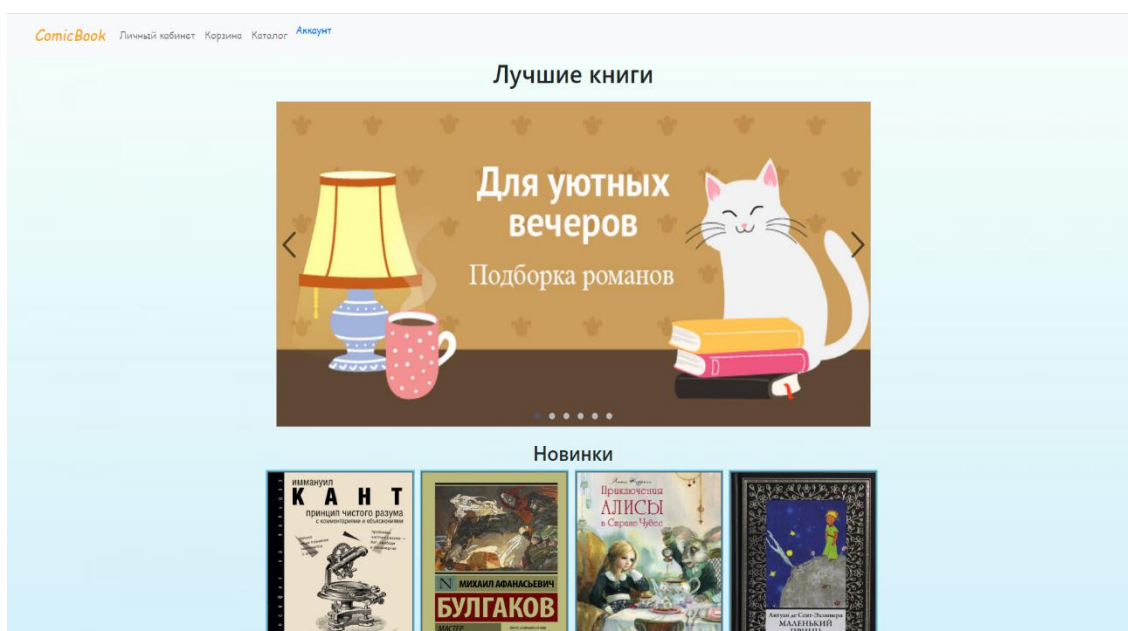


Рис. 2.23. Приветственная страница платформы.

## Аккаунт

При переходе на данную вкладку будет представлена форма для входа. Необходимо знать логин и пароль. После заполнения идет перенаправление на вкладку «Главная страница».

После авторизации пользователю будет доступна новая функция:

### 1. Корзина

## Вход

Ваш логин:

Ваш пароль:

**Войти**

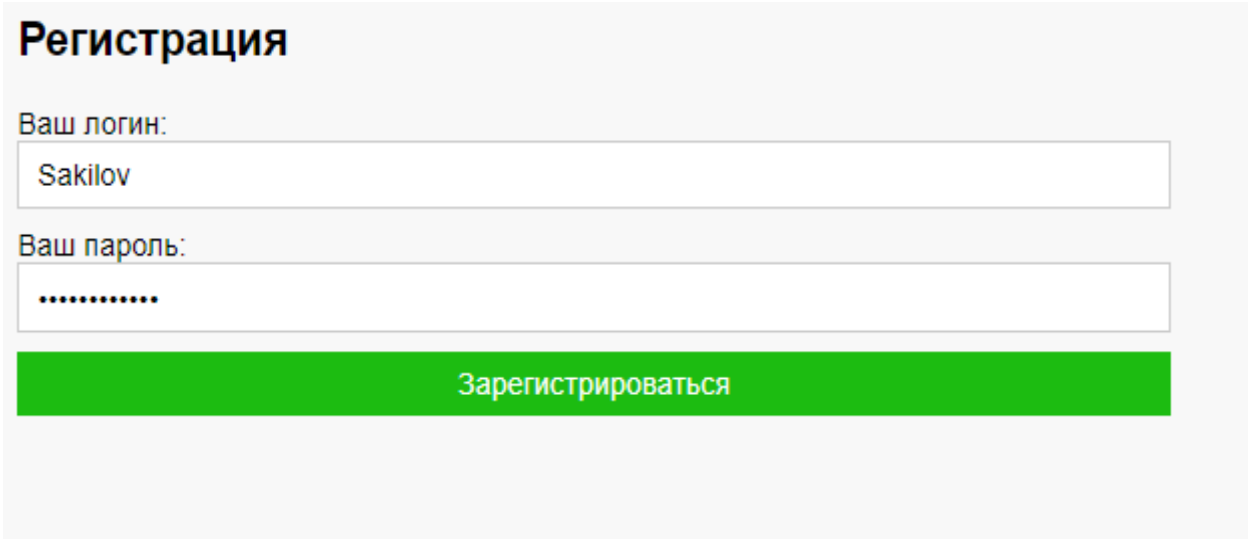
[Зарегистрироваться](#)

Вы вошли на сайт, как гость  
[Зарегистрируйтесь чтобы покупать товары](#)  
[Вернуться на главную страницу](#)

Рис 2.24 Вход в личный кабинет пользователя

## Зарегистрироваться

При переходе на данную вкладку будет представлена регистрационная форма, которую необходимо заполнить и отправить на сервер. После заполнения идет перенаправления на вкладку «войти».



The image shows a registration form titled "Регистрация". It contains two input fields: "Ваш логин:" with the text "Sakilov" and "Ваш пароль:" with masked characters ".....". Below these fields is a prominent green button with the text "Зарегистрироваться".

Рис. 2.25 Создание новой учетной записи.

Доступ к платформе также можно получить, зайдя на главную страницу. На данной странице предлагаются различные сценарии использования платформы и доступны интерактивные кнопки.

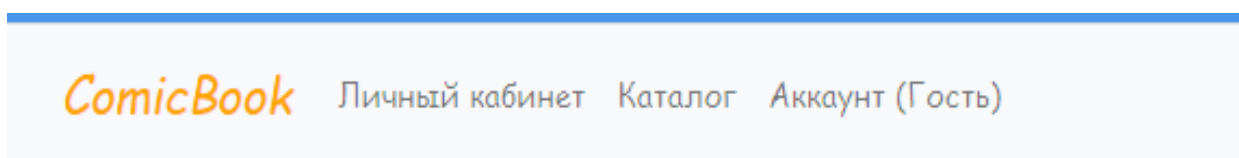


Рис. 2.26 Интерактивные кнопки сайта

## Подбор книги по предпочтениям

Чтобы пользователь мог подобрать подходящую для него книгу. Он должен пройти маленький тест.

### Подбор книги по предпочтениям

На что вы обращаете внимание при выборе книги?

- ☐ На стиль обложки
- ☐ На имя автора
- ☐ На описание
- ☐ На цену

Какие черты характера вы цените в себе больше всего?

- ☒ Доброту и искренность
- ☐ Внимательность и организованность
- ☐ Активность и целеустремленность
- ☐ Любознательность и любопытство

Какие книги вы предпочитаете?

- ☐ Электронные
- ☐ Бумажные
- ☐ Аудиокниги
- ☐ Серии книг
- ☐ На один вечер

Какой жанр для вас предпочтителен?

- ☒ Классическая художественная литература
- ☐ Предпочитаю читать о искусстве
- ☐ Саморазвитие
- ☐ Нон фикшн

В каком времени должны происходить события в книге?

- ☐ В XX веке
- ☐ В современности
- ☐ Не имеет значения
- ☐ В будущем

Какой по объёму должна быть книга?

- ☒ Небольшой
- ☐ Средней
- ☐ Большой
- ☐ Не имеет значения

[Узнать результат](#)

Рис. 2.27 Тест интернет-магазина

## Результат теста

После прохождения пользователю высветится окно с результатом.

**Вы прошли тест!**

Ваш результат: "Шерлок Холмс" Конан Дойл. Наверное, самый умный и захватывающий детектив. Читая книгу, вы почувствуете дух викторианской Англии, где происходят гениальные расследования самого выдающегося литературного сыщика. Так же узнаете много новых открытий и запутанных историй, которые завоевывают внимание с первых строк книги, получившая множество наград и всемирного признания.

**Похожие товары**



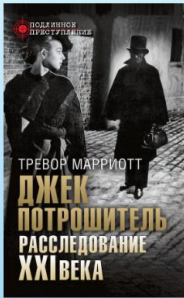
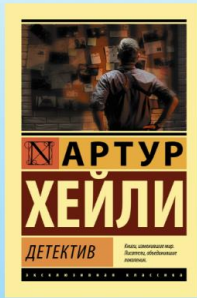
 <p>Ошибка резидента</p> <p>851 Р</p> <p><a href="#">Купить</a></p>	 <p>Молчание ягнят</p> <p>980 Р</p> <p><a href="#">Купить</a></p>	 <p>Джек Потрошитель. Расследование XXI века</p> <p>1024 Р</p> <p><a href="#">Купить</a></p>	 <p>Артур Хейли</p> <p>999 Р</p> <p><a href="#">Купить</a></p>
--	--	--	---

Рис 2.28 Результат теста

## Жанры

Используя верхнее меню или нижнюю панель доступа в случае использования мобильной версии приложения, можно получить доступ к остальным функциональным страницам платформы. Раздел «Каталог» позволяет ознакомиться с произведениями других авторов.

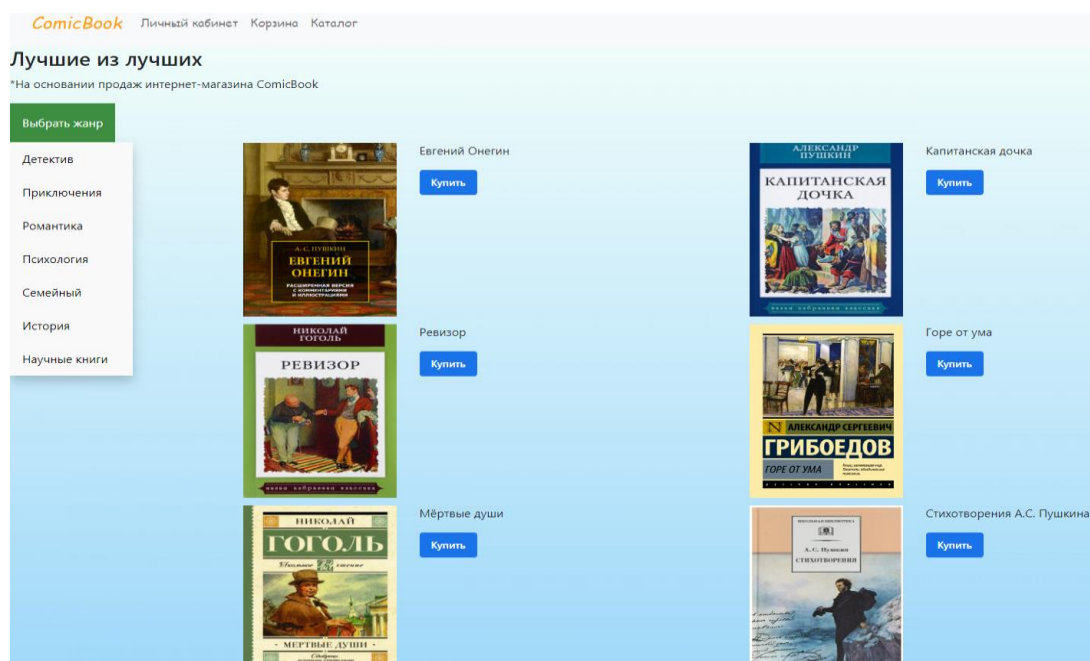


Рис 2.29 Выбор жанра

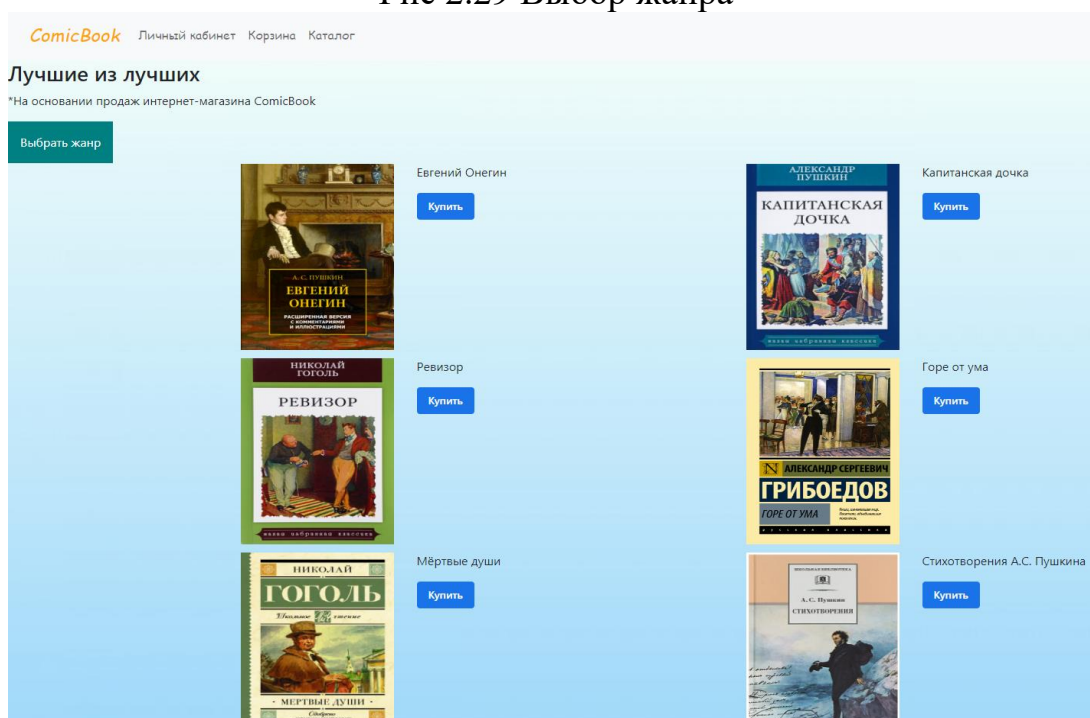


Рис 2.30 Отображение только выбранных книг

## Подробнее

При нажатии на данную кнопку производится перенаправление на отдельную страницу, где указана полная информация по выбранной книге.

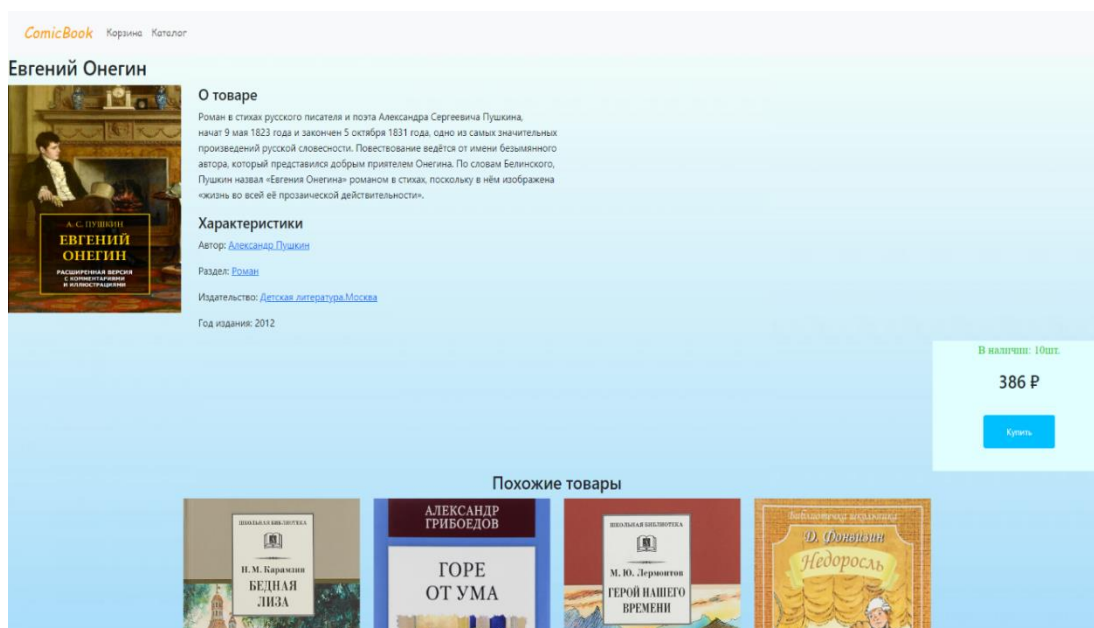


Рис. 2.31 Отображение книги с полным описанием

## Похожие товары

Под книгой отображаются похожие товары по интересу и жанру. Пользователь всегда может найти интересную для него книгу.

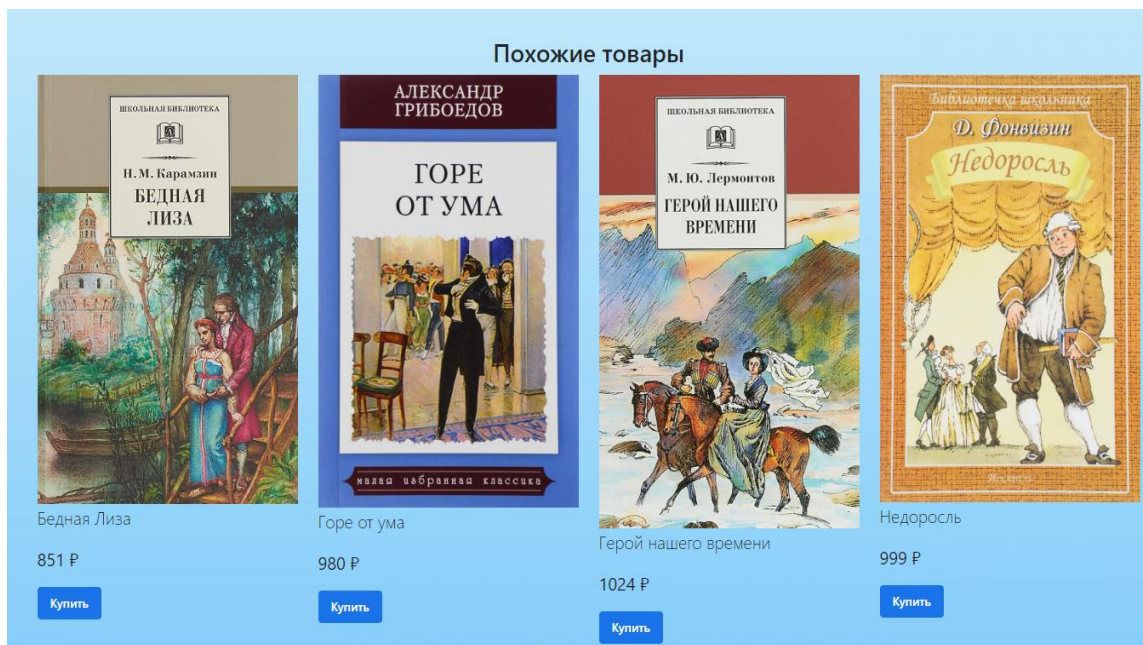


Рис. 2.32 Похожие товары



## Страница автора

Пользователь может выбрать автора и посмотреть все его работы.

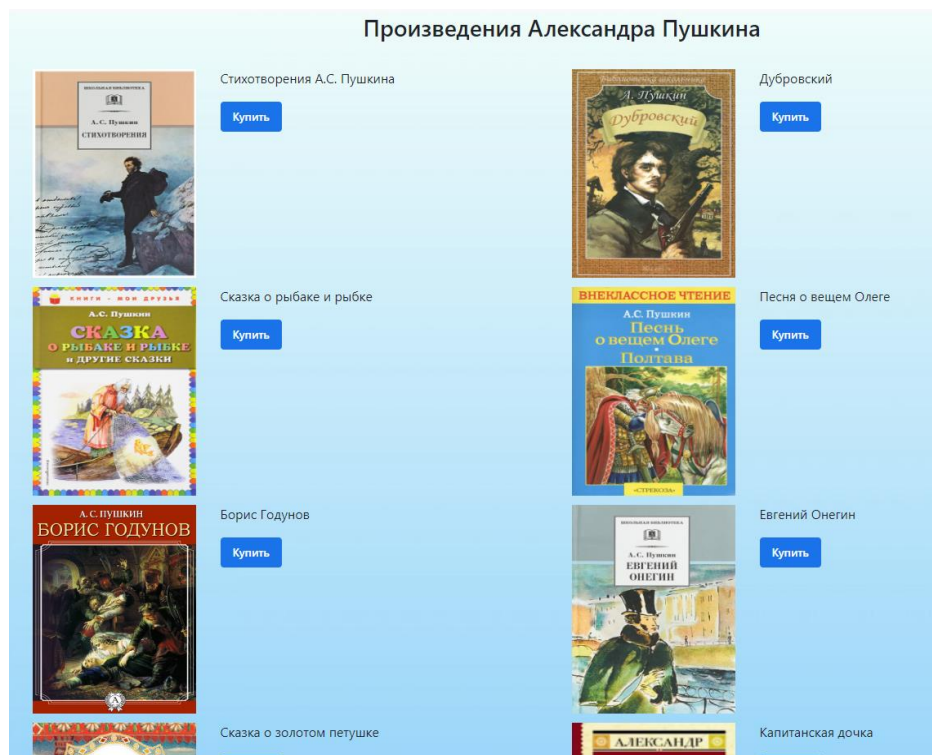


Рис. 2.33 Произведения автора

## Корзина

При переходе на данную вкладку будут представлены товары (книги) из корзины авторизованного пользователя. Также внизу отображается итоговая цена по всем товарам (книгам), которые пользователь добавил в корзину.

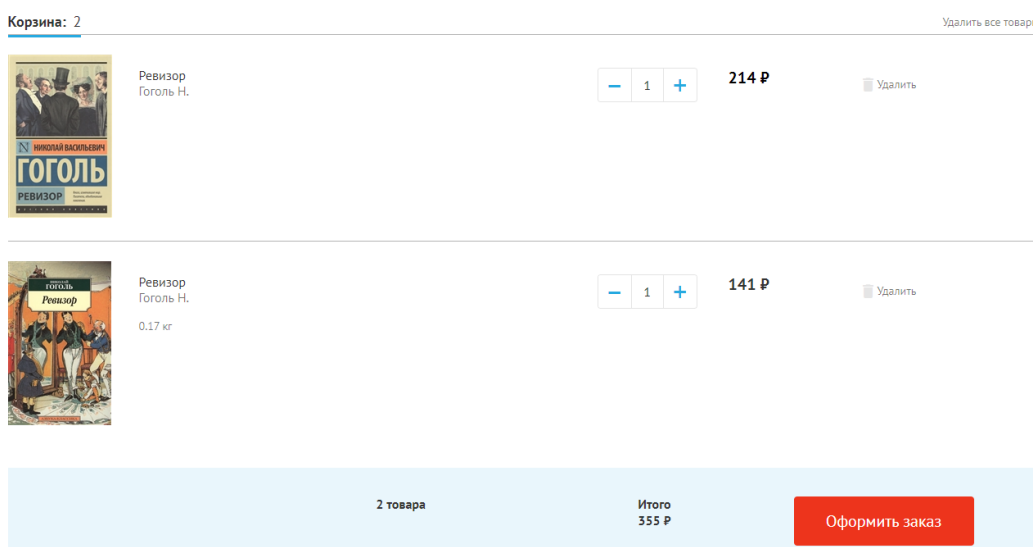


Рис. 2.34 Корзина с добавленной книгой

## Книга

После покупки пользователю приходит уведомление о покупке товара. Пользователь должен нажать на кнопку “Начать читать”, чтобы приступить к чтению книги.

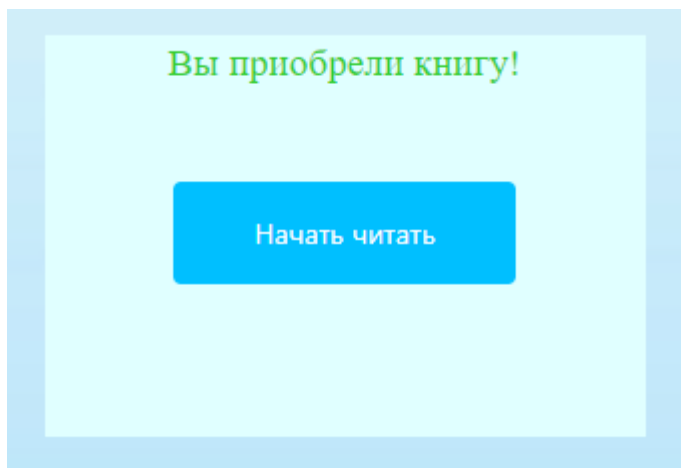


Рис. 2.35 Готовый продукт

## Электронная книга

Пользователь должен нажать на кнопку “Начать читать” и у него откроется электронная книга в формате pdf.

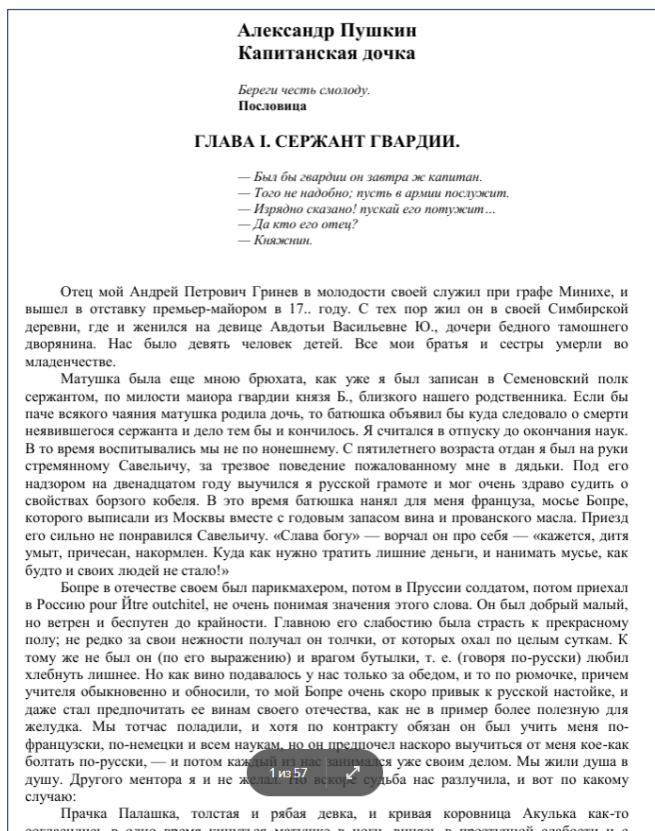


Рис. 2.36 Электронная книга



## ЗАКЛЮЧЕНИЕ

Целью данного дипломного проекта являлась разработка веб-системы интернет-магазина книг с подбором по предпочтениям.

В данной дипломной работе были рассмотрены роль службы покупки товара, описали типы систем электронной покупки, рассмотрели существующие системы электронной покупки на российском рынке и их актуальные проблемы. В связи с этим можно сделать следующие выводы:

Электронный способ покупки товара открывает широкие возможности для покупателей. Существует 3 типа систем электронного покупки товара: альтернативная система покупки, глобальная система покупки и централизованная система покупки;

На данный момент терминалы электронных систем покупки установлены 600 000 в магазинах. Сегодня крупных игроков на российском рынке предоставления услуг подключения к системам покупки не так много и лидером среди них на данный момент является Нота-Бена. В России доля электронного покупки товара занимает малую доля по сравнению с другими странами.

Помимо прочего, описаны функциональные требования, которым должна удовлетворять программа. Для визуального описания работы, разработан алгоритм в виде структурных схем. Были выбраны средства реализации – для интерфейса пользователя и хранилища базы данных. Была произведена отладка, а также автономное и комплексное тестирование разработанной программы. В итоге был получен конечный продукт, для которого составлена подробная инструкция пользователя, описывающая основной функционал программного изделия. Программа является мобильной (в рамках серверов) и, при необходимости, может быть модернизирована.

При разработке были использованы следующие программные средства:

1. Веб-сервер NGINX с менеджером процессов PHP-FPM;

2. PHP как основной язык разработки движка и описания представлений страниц;
3. В качестве СУБД выбран дистрибутив MySQL, к нему надстройка PhpMyAdmin и средство моделирования и разработки БД MySQL Workbench;
4. Также для формирования запросов к обработчикам и динамической загрузки данных была использована библиотека JQuery.

Для разработки универсального адаптивного графического интерфейса веб-платформы были использованы дополнительные средства:

1. Adobe Illustrator для разработки основных графических элементов;
2. Adobe Photoshop для макетирования элементов веб-платформы.

С использованием средств проектирования и разработки были решены следующие задачи:

1. Разработан пользовательский интерфейс программы
2. Разработана база данных интернет-магазина
3. Разработаны алгоритмы работы сайта
4. Реализована работа сайта с базой данных
5. Протестирован web-сайт

Разработанная платформа доступна для использования на любом устройстве, подключенном к сети Интернет с установленным браузером, поддерживающем технологии HTML5, CSS3 и динамические запросы.

Исходя из вышесказанного, цели и задачи, поставленные в рамках дипломного проекта можно считать выполненными.

## СПИСОК ИСТОЧНИКОВ ЛИТЕРАТУРЫ

### Основные источники:

1. ГОСТ 19781. Программное обеспечение систем обработки данных, термины и определения.
2. ГОСТ 15971. Системы обработки данных
3. ГОСТ 20806. Качество программных средств
4. Базы данных: Учебный курс. Глушаков С.В. 2019
5. Видеокурсы по разработке веб-приложений с использованием Flask на языке Python, М. Гринберг, 2020г
6. Выразительный JavaScript, М. Haverbeke, 2021г.
7. Дакетт Джон. HTML и CSS. Разработка и дизайн веб-сайтов. Москва, 2019. 480 с.
8. Дакетт Джон. Javascript и jQuery. Интерактивная веб-разработка. Москва, 2019. 640 с.
9. Изучаем НТМ 5, Б. Лоудсон, Р. Шарп, 2019г.
10. Морето Сильвио. Bootstrap в примерах. Москва, 2020. 314 с.
11. Разработка веб-приложений с использованием Flask на языке Python, М.Гринберг, 2018г.
12. Системы баз данных. Грег Риккарди, 2019г.
13. Фримен Э., Робсон Э. Изучаем программирование на JavaScript Питер, 2020. – 640 с.
14. Шелли Пауэрс. Изучаем PHP. Переходим на сторону сервера. Питер, 2019. 304 с.

15. Этан Браун. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. Москва, 2019. – 368 с.

16. CSS 3 и Web 2.0. Разработки современных Web-сайтов. В. Дронов, 2020г

### **Интернет-ресурсы:**

1. Интернет-магазин «Читай-город». Изображение системы покупки  
<https://www.chitai-gorod.ru/>
2. <https://www.cyberforum.ru/>
3. <https://www.php.net/>
4. <https://qna.habr.com/>
5. <https://html5css.ru/>
6. <http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/>
7. <http://www.lissyara.su/>
8. <http://progopedia.ru/>
9. <http://htmlweb.ru/>
10. <http://flask.pocoo.org/>
11. <https://habrahabr.ru/post/246699/>

## ПРИЛОЖЕНИЕ 1. Исходный код

### Листинг. Код верстки шапки сайта

```
<!doctype html>
<?php session_start(); ?>
<html lang="ru" dir="ltr">
<head>
    <!-- Обязательные метатеги -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU"
crossorigin="anonymous">
    <!-- fancyBox CSS -->
    <link href="/examples/vendors/fancybox/jquery.fancybox.min.css" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="css\home.css">
    <title>Главная</title>
    <script type="text/javascript">
    $(function() {
        $(".search_button").click(function() {
            // получаем то, что написал пользователь
            var searchString = $("#search_box").val();
            // формируем строку запроса
            var data = 'search=' + searchString;
            // если searchString не пустая
            if(searchString) {
                // делаем ajax запрос
                $.ajax({
                    type: "POST",
                    url: "scripts/search.php",
                    data: data,
                    beforeSend: function(html) { // запустится до вызова запроса
                        $("#results").html("");
                        $("#searchresults").show();
                        $(".word").html(searchString);
                    },
                    success: function(html){ // запустится после получения результатов
                        $("#results").show();
                        $("#results").append(html);
                    }
                });
            }
            return false;
        });
    });
});
```

```

</script>
</head>

```

## Листинг. Главная страница (HTML, CSS, PHP)

### HTML, PHP:

```

<?php
    $db = mysqli_connect("localhost","root","","comicbook");
?>

<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
            <a class="nav-link" href="index.php" style="color: #ffa500; font-size: x-large; font-style: italic;">ComicBook</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" href="login.html">Личный кабинет</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="Category\Catalog.php">Каталог</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="scripts/index.php">
                            Аккаунт
                        <?php
                            if (empty($_SESSION['login']) or empty($_SESSION['id']))
                            {
                                echo "(Гость)";
                            }
                            else
                            {
                                echo "($_SESSION['login'])";
                            }
                        <?>
                        </a>
                    </li>
                </ul>
            </div>
            <p>Корзина: <a id="clicks">0</a></p>
        </div>
    </nav>

    <!-- Дополнительный JavaScript; выберите один из двух! -->
    <!-- Вариант 1: Bootstrap в связке с Popper -->

```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
/bQdsTh/da6pk11MST/rWKFjNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ" crossorigin="anonymous"></script>

<!-- Вариант 2: Bootstrap JS отдельно от Popper
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js" integrity="sha384-
W8fXfP3gkOKtndU4JGtKdVxbO53Wy8SZCQHczT5FMiiqmQfUpWbYdTil/SxwZgAN" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.min.js" integrity="sha384-
skAcpIdS7UcVUC05LJ9Dxay8AXcDYfBJqt1CJ85S/CFujBsIzCIv+I9liuYLaMQ/" crossorigin="anonymous"></script>
-->

<h1 align="center">Лучшие комиксы</h1>
<div class="sim-slider" align="center">
<ul class="sim-slider-list">
<li></li> <!--700x437 -->
<li class="sim-slider-element"></li>
<li class="sim-slider-element"></li>
<li class="sim-slider-element"></li>
<li class="sim-slider-element"></li>
<li class="sim-slider-element"></li>
<li class="sim-slider-element"></li>
</ul>
<div class="sim-slider-arrow-left"></div>
<div class="sim-slider-arrow-right"></div>
<div class="sim-slider-dots"></div>
</div>
<h2 align="center">Новинки</h2>

<p class="thumb" align="center">




</p>

<h2 align="center">Лучшие из лучших</h2>

<div class="container">
<div class="row row-cols-4">
<div class="col">
<a data-fancybox="gallery" href="home\drstonehome.php">


```

```

        </a>
        <p class="lead">Dr. Stone Том 12</p>
        <p align="center" class="lead"><strong>876 Р</strong></p>
        <div align="left">
            <a href="home\drstonehome.php" class="button">Купить</a>
        </div>
    </div>
    <div class="col">
        <a
                                data-fancybox="gallery"
href="https://s1.livelib.ru/boocover/1003487718/o/41f0/Hiroyuki_Takei__Shaman_King_Vol._20_Epilogue.jpeg">
            
            </a>
            <p class="lead">Shaman king Том 20</p>
            <p align="center" class="lead"><strong>930 Р</strong></p>
            <div align="left">
                <a href="home\shamanking.php" class="button" onclick="alert('Товар добавлен в корзину!')"
value="Кнопка">Купить</a>
            </div>
        </div>
    <div class="col">
        <a data-fancybox="gallery" href="home\berserkhome.php">
            
            </a>
            <p class="lead">Берсерк Том 15</p>
            <p align="center" class="lead"><strong>840 Р</strong></p>
            <div align="left">
                <a href="home\berserkhome.php" class="button">Купить</a>
            </div>
        </div>
    </div>
    <div class="col">
        <a data-fancybox="gallery" href="home\hunterhome.php">
            
            </a>
            <p class="lead">HunterxHunter Том 17</p>
            <p align="center" class="lead"><strong>590 Р</strong></p>
            <div align="left">
                <a href="home\hunterhome.php" class="button">Купить</a>
            </div>
        </div>
    </div>
</div>

```



```

<br>
<h2 align="center">10 Захватывающих детективов</h2>

<div class="container">
  <div class="row row-cols-4">
    <div class="col">
      <a data-fancybox="gallery" href="home\machihome.php">
        
      </a>
      <p class="lead">Город в котором меня нет</p>
      <p align="center" class="lead"><strong>480 Р</strong></p>
      <div align="left">
        <a href="home\machihome.php" class="button">Купить</a>
      </div>
    </div>
    <div class="col">
      <a data-fancybox="gallery" href="home\detectivehome.php">
        
      </a>
      <p class="lead">Богатый детектив</p>
      <p align="center" class="lead"><strong>510 Р</strong></p>
      <div align="left">
        <a href="home\detectivehome.php" class="button">Купить</a>
      </div>
    </div>
    <div class="col">
      <a data-fancybox="gallery" href="home\kenomohome.php">
        
      </a>
      <p class="lead">Инцидент кеномо том 1</p>
      <p align="center" class="lead"><strong>950 Р</strong></p>
      <div align="left">
        <a href="home\kenomohome.php" class="button">Купить</a>
      </div>
    </div>
    <div class="col">
      <a data-fancybox="gallery" href="home\firehome.php">
        
      </a>
    </div>
  </div>
</div>

```

```

</a>
<p class="lead">Пламенная бригада<br>пожарных том 1</p>
<p align="center" class="lead"><strong>715 Р</strong></p>
<div align="left">
<a href="home\firehome.php" class="button">Купить</a>
</div>

</div>
</div>
<br>
<h2 align="center">Скидки</h2>

<div class="container">
<div class="row row-cols-4">
<div class="col">
<a data-fancybox="gallery" href="home\cloverhome.php">

</a>

<p class="lead">Чёрный клевер</p>
<p align="center" class="lead" style="text-decoration: line-through;"><strong>1024 Р</strong></p>
<p align="center" class="lead" style="color: orange;"><strong>632 Р</strong></p>
<div align="left">
<a href="home\cloverhome.php" class="button">Купить</a>
</div>
</div>
<div class="col">
<a data-fancybox="gallery" href="Category\С.Исида\ТГ14.php">

</a>

<p class="lead">Токийский гуль том 14</p>
<p align="center" class="lead" style="text-decoration: line-through;"><strong>802 Р</strong></p>
<p align="center" class="lead" style="color: orange;"><strong>574 Р</strong></p>
<div align="left">
<a href="Category\С.Исида\ТГ14.php" class="button">Купить</a>
</div>
</div>
<div class="col">
<a data-fancybox="gallery" href="home\soumahome.php">

</a>

<p class="lead">В поисках божественного рецепта Том 32</p>

```

```

        <p align="center" class="lead" style="text-decoration: line-through;"><strong>960 P</strong></p>
        <p align="center" class="lead" style="color: orange;"><strong>600 P</strong></p>
        <div align="left">
            <a href="home\soumahome.php" class="button">Купить</a>
        </div>

    </div>

    <div class="col">
        <a data-fancybox="gallery" href="home\lovehome.php">
            
            </a>
            <p class="lead">Притворная любовь Том 1</p>
            <p align="center" class="lead" style="text-decoration: line-through;"><strong>1100 P</strong></p>
            <p align="center" class="lead" style="color: orange;"><strong>830 P</strong></p>
            <div align="left">
                <a href="home\lovehome.php" class="button">Купить</a>
            </div>

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<br>
<br>
<br>
<script type="text/javascript">
    var clicks = 0;
    function onClick() {
        clicks += 1;
        document.getElementById("clicks").innerHTML = clicks;
    };

</script>
<script src="script.js">

</script>
<!-- Footer -->
<footer class="page-footer font-small indigo">

<!-- Footer Links -->
<div class="container">

<!-- Grid row-->
<div class="row text-center d-flex justify-content-center pt-5 mb-3">

```

```

<!-- Grid column -->
<div class="col-md-2 mb-3">
  <h6 class="text-uppercase font-weight-bold">
    <a href="#">О нас</a>
  </h6>
</div>
<!-- Grid column -->

<!-- Grid column -->
<div class="col-md-2 mb-3">
  <h6 class="text-uppercase font-weight-bold">
    <a href="#">Товары</a>
  </h6>
</div>
<!-- Grid column -->

<!-- Grid column -->
<div class="col-md-2 mb-3">
  <h6 class="text-uppercase font-weight-bold">
    <a href="#">Адреса магазинов</a>
  </h6>
</div>
<!-- Grid column -->

<!-- Grid column -->
<div class="col-md-2 mb-3">
  <h6 class="text-uppercase font-weight-bold">
    <a href="#">Помощь</a>
  </h6>
</div>
<!-- Grid column -->

<!-- Grid column -->
<div class="col-md-2 mb-3">
  <h6 class="text-uppercase font-weight-bold">
    <a href="#">Контакты</a>
  </h6>
</div>
<!-- Grid column -->

</div>
<!-- Grid row-->
<hr class="rgba-white-light" style="margin: 0 15%;">

<!-- Grid row-->

```

```
<div class="row d-flex text-center justify-content-center mb-md-0 mb-4">
```

```
<!-- Grid column -->
```

```
<div class="col-md-8 col-12 mt-5">
```

```
<h5 align="center">Добро пожаловать!</h5>
```

```
<p style="line-height: 1.7rem">«ComicBook» – это сеть книжных в Москве и других городах России.
```

А ещё это удобный интернет-магазин. В нём можно заказывать комиксы разных жанров: триллеры и детективы, фэнтези и фантастику, приключения и романы.

На сайте есть разделы, в которых легко найти интересные новинки, бестселлеры и современную прозу. Выбирайте то, что поднимет настроение!</p>

```
</div>
```

```
<!-- Grid column -->
```

```
</div>
```

```
<!-- Grid row-->
```

```
<hr class="clearfix d-md-none rgba-white-light" style="margin: 10% 15% 5%;">
```

```
<!-- Grid row-->
```

```
<div class="row pb-3">
```

```
<!-- Grid column -->
```

```
<div class="col-md-12">
```

```
<div class="mb-5 flex-center">
```

```
<!-- Facebook -->
```

```
<a class="fb-ic">
```

```
<i class="fab fa-facebook-f fa-lg white-text mr-4"> </i>
```

```
</a>
```

```
<!-- Twitter -->
```

```
<a class="tw-ic">
```

```
<i class="fab fa-twitter fa-lg white-text mr-4"> </i>
```

```
</a>
```

```
<!-- Google +-->
```

```
<a class="gplus-ic">
```

```
<i class="fab fa-google-plus-g fa-lg white-text mr-4"> </i>
```

```
</a>
```

```
<!--Linkedin -->
```

```
<a class="li-ic">
```

```
<i class="fab fa-linkedin-in fa-lg white-text mr-4"> </i>
```

```
</a>
```

```
<!--Instagram-->
```

```
<a class="ins-ic">
```

```
<i class="fab fa-instagram fa-lg white-text mr-4"> </i>
```

```
</a>
```

```
<!--Pinterest-->
```

```

        <a class="pin-ic">
          <i class="fab fa-pinterest fa-lg white-text"> </i>
        </a>

      </div>

    </div>

    <!-- Grid column -->

  </div>

  <!-- Grid row-->

</div>

<!-- Footer Links -->

<!-- Copyright -->
<div class="footer-copyright text-center py-3">© 2020 Copyright:
  <a href="https://mdbootstrap.com/">ComicBook.com</a>
</div>
<!-- Copyright -->

</footer>
<!-- Footer -->

```

CSS:

```
.button {
background-color: #00bfff;
border-radius: 4px;
color: white;
padding: 15px 40px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 14px;
margin: 4px 2px;
cursor: pointer;
}
```

```
body{
width: 100%;
max-width: 100%;
background: url(https://вебджем.рф/800/600/https/krot.info/uploads/posts/2021-01/1610159696_29-p-nezhnii-pastelnii-fon-40.jpg) no-repeat;
-moz-background-size: 100%; /* Firefox 3.6+ */
-webkit-background-size: 100%; /* Safari 3.1+ и Chrome 4.0+ */
-o-background-size: 100%; /* Опера 9.6+ */
background-size: 100%; /* Современные браузеры */
margin: 0; /* Убираем отступы */
height: 100%; /* Высота страницы */
background-size: cover; /* Фон занимает всю доступную площадь */
}
```

```
.footer,
.push {
height: 200px;
}
footer {
```

```

background: #42A5F5;
color: white;
line-height: 50px;
padding: 0 20px;
}

nav {
font-family: 'Kaushan Script', cursive;
font-size: 1em;
}

img{
max-width: 100%;
}

.thumb img {
border: 2px solid #55c5e9; /* Рамка вокруг фотографии */
padding: 2px; /* Расстояние от картинки до рамки */
background: #666; /* Цвет фона */
margin-right: 5px; /* Отступ справа */
margin-bottom: 5px; /* Отступ снизу */
}

.sim-slider {
max-width: 1000px;
min-width: 1000px;
margin: 20px auto;

border: 1px solid #ccd;
background-color: white;
}

/* General styles */
.sim-slider {

```



```
position: relative;
}
```

```
.sim-slider-list {
margin: 0;
padding: 0;
list-style-type: none;
position: relative;
}
```

```
.sim-slider-element {
width: 100%;
transition: opacity 1s ease-in;
opacity: 0;
position: absolute;
z-index: 2;
left: 0;
top: 0;
display: block;
}
```

```
/* Navigation item styles */
div.sim-slider-arrow-left,
div.sim-slider-arrow-right {
width: 22px;
height: 40px;
position: absolute;
cursor: pointer;
opacity: 0.6;
z-index: 4;
}
```

```
div.sim-slider-arrow-left {
left: 10px;
top: 40%;
```

```
display: block;
background: url("http://pvbk.spb.ru/inc/slider/sim-files/sim-arrow-left.png") no-repeat;
}
```

```
div.sim-slider-arrow-right {
    right: 10px;
    top: 40%;
    display: block;
    background: url("http://pvbk.spb.ru/inc/slider/sim-files/sim-arrow-right.png") no-repeat;
}
```

```
div.sim-slider-arrow-left:hover {
    opacity: 1.0;
}
```

```
div.sim-slider-arrow-right:hover {
    opacity: 1.0;
}
```

```
div.sim-slider-dots {
    width: 100%;
    height: auto;
    position: absolute;
    left: 0;
    bottom: 0;
    z-index: 3;
    text-align: center;
}
```

```
span.sim-dot {
    width: 10px;
    height: 10px;
    margin: 5px 7px;
    padding: 0;
    display: inline-block;
```

```
background-color: #BBB;
border-radius: 5px;
cursor: pointer;
}
```

## Листинг. Регистрация (HTML, CSS, PHP)

### HTML:

```
<html>
<head>
<title>Регистрация</title>
</head>
<body>

<h2>Регистрация</h2>
<form action="save_user.php" method="post">
<!--**** save_user.php - это адрес обработчика. То есть, после нажатия на кнопку "Зарегистрироваться",
данные из полей отправятся на страничку save_user.php методом "post" **** -->

<label>Ваш логин:</label>
<input name="login" type="text" class="input" placeholder="Логин">

<!--**** В текстовое поле (name="login" type="text") пользователь вводит свой логин **** -->

<label>Ваш пароль:</label>
<input name="password" type="password" class="input" placeholder="Пароль">

<!--**** В поле для паролей (name="password" type="password") пользователь вводит свой пароль **** -->

<button class="btn" type="submit" name="submit">Зарегистрироваться</button>
<!--**** Кнопочка (type="submit") отправляет данные на страничку save_user.php **** -->
</form>
```

### CSS:

```
body {
font-family: Arial, sans-serif;
font-size: 16px;
color: #000;
background-color: #f8f8f8;
max-width: 80%;
width: 30%;
}

* {
box-sizing: border-box;
}
```

```

.form {
    max-width: 320px;
    padding: 15px;
    margin: 20px auto;
    background-color: #fff;
}

.input {
    display: block;
    width: 100%;
    padding: 8px 10px;
    margin-bottom: 10px;

    border: 1px solid #ccc;

    font-family: inherit;
    font-size: 16px;
}

.btn {
    display: block;
    width: 100%;
    padding: 8px 10px;

    border: 0;
    background-color: #1cbc11;
    cursor: pointer;

    font-family: inherit;
    font-size: 16px;
    color: #fff;
}

.btn:hover {
    background-color: #14a20a;
}

```

## PHP:

```

<?php
    if (empty($_SESSION['login']) or empty($_SESSION['id']))
    {
        echo "Вы вошли на сайт, как гость<br><a href='#>Зарегистрируйтесь чтобы покупать товары</a>";
    }
    else
    {

```

```
        echo "Вы вошли на сайт, как ".$_SESSION['login']. "<br><a href='../index.php'>Зарегистрируйтесь чтобы  
покупать товары</a>";  
    }  
  
    echo "<br> <br> <a href='../index.php'>Вернутся на главную страницу</a>";  
    ?>
```

## Листинг. Авторизация (HTML, CSS, PHP)

### HTML:

```
<html>
  <head>
    <title>Главная страница</title>
  </head>
  <body>
    <h2>Главная страница</h2>
    <form action="testreg.php" method="post">

      <!--**** testreg.php - это адрес обработчика. То есть, после нажатия на кнопку "Войти", данные из полей
      отправятся на страничку testreg.php методом "post" ****>

      <label>Ваш логин:</label>
      <input name="login" type="text" class="input" placeholder="Логин">

      <!--**** В текстовое поле (name="login" type="text") пользователь вводит свой логин ****>

      <label>Ваш пароль:</label>
      <input name="password" type="password" class="input" placeholder="Пароль">

      <!--**** В поле для паролей (name="password" type="password") пользователь вводит свой пароль ****>

      <p>
        <button type="submit" name="submit" class="btn">Войти</button>
        <!--**** Кнопочка (type="submit") отправляет данные на страничку testreg.php ****>
        <!--**** ссылка на регистрацию, ведь как-то же должны гости туда попадать ****>
        <a href="reg.php">Зарегистрироваться</a>
      </p>
    </form>
    <form class="" action="out.php" method="post">
      <input type="submit" name="submit" value="Выйти">
    </form>
```

### CSS:

```
body {
  font-family: Arial, sans-serif;
  font-size: 16px;
  color: #000;
  background-color: #f8f8f8;
  max-width: 80%;
  width: 30%;
}
* {
  box-sizing: border-box;
}
```

```

.form {
    max-width: 320px;
    padding: 15px;
    margin: 20px auto;
    background-color: #fff;
}

.input {
    display: block;
    width: 100%;
    padding: 8px 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    font-family: inherit;
    font-size: 16px;
}

.btn {
    display: block;
    width: 100%;
    padding: 8px 10px;
    border: 0;
    background-color: #1cbc11;
    cursor: pointer;
    font-family: inherit;
    font-size: 16px;
    color: #fff;
}

.btn:hover {
    background-color: #14a20a;
}

```

## PHP:

```

<?php
    // Проверяем, пусты ли переменные логина и id пользователя
    if (empty($_SESSION['login']) or empty($_SESSION['id']))
    {
        // Если пусты, то мы не выводим ссылку
        echo "Вы вошли на сайт, как гость<br><a href='#>Зарегистрируйтесь чтобы покупать товары</a>";
    }
    else
    {

        // Если не пусты, то мы выводим ссылку
        echo "Вы вошли на сайт, как ".$_SESSION['login']. "<br><a href='../index.php'>Зарегистрируйтесь чтобы
покупать товары</a>";
    }

```

```
echo "<br> <br> <a href='../index.php'>Вернутся на главную страницу</a>";  
?>
```

## Листинг. Добавление в корзину (HTML, PHP)

### HTML:

```
<?php  
    session_start();  
    require("includes/connection.php");  
    if(isset($_GET['page'])){  
  
        $pages=array("products", "cart");  
  
        if(in_array($_GET['page'], $pages)) {  
  
            $_page=$_GET['page'];  
  
        }else{  
  
            $_page="products";  
  
        }  
  
    }else{  
  
        $_page="products";  
  
    }  
  
?>  
<!DOCTYPE html>  
<head>  
  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <link rel="stylesheet" href="css/reset.css" />  
    <link rel="stylesheet" href="css/style.css" />  
  
    <title>Shopping Cart</title>  
  
</head>  
  
<body>  
  
    <div id="container">
```



```

        <div id="main">
            <?php require($_page.".php"); ?>

        </div><!--end of main-->

        <div id="sidebar">

        </div><!--end of sidebar-->

    </div><!--end container-->

</body>
</html>

```

## PHP:

```

<?php
if(isset($_GET['action']) && $_GET['action']=="add"){
    $id=intval($_GET['id']);
    if(isset($_SESSION['cart'][$id])){
        $_SESSION['cart'][$id]['quantity']++;
    }else{
        $sql_s="SELECT * FROM products
        WHERE id_product={ $id}";
        $query_s=mysql_query($sql_s);
        if(mysql_num_rows($query_s)!=0){
            $row_s=mysql_fetch_array($query_s);

            $_SESSION['cart'][$row_s['id_product']]=array(
                "quantity" => 1,
                "price" => $row_s['price']
            );
        }else{
            $message="This product id it's invalid!";
        }
    }
}

?>
<h1>Product List</h1>
<?php
if(isset($message)){
    echo "<h2>$message</h2>";
}
?>
<table>
    <tr>
        <th>Name</th>
        <th>Description</th>
        <th>Price</th>
        <th>Action</th>

```

```

</tr>

<?php
    $sql="SELECT * FROM products ORDER BY name ASC";
    $query=mysql_query($sql);

    while ($row=mysql_fetch_array($query)) {

?>
        <tr>
            <td><?php echo $row['name'] ?></td>
            <td><?php echo $row['description'] ?></td>
            <td><?php echo $row['price'] ?></td>
            <td><a href="index.php?page=products&action=add&id=<?php echo $row['id_product'] ?>">Add to cart</a></td>
        </tr>
    <?php
        }

?>

</table>
<?php

if(isset($_POST['submit'])){

    foreach($_POST['quantity'] as $key => $val) {
        if($val==0) {
            unset($_SESSION['cart'][$key]);
        }else{
            $_SESSION['cart'][$key]['quantity']=$val;
        }
    }

}

?>

<h1>View cart</h1>
<a href="index.php?page=products">Go back to the products page.</a>
<form method="post" action="index.php?page=cart">

<table>

    <tr>
        <th>Name</th>
        <th>Quantity</th>
        <th>Price</th>
        <th>Items Price</th>
    </tr>

    <?php

        $sql="SELECT * FROM products WHERE id_product IN (";

        foreach($_SESSION['cart'] as $id => $value) {
            $sql.=$id.", ";
        }

        $sql=substr($sql, 0, -1).") ORDER BY name ASC";
        $query=mysql_query($sql);
        $totalprice=0;
        while($row=mysql_fetch_array($query)){
            $subtotal=$_SESSION['cart'][$row['id_product']]['quantity']*$row['price'];
            $totalprice+=$subtotal;
        }

        <tr>
            <td><?php echo $row['name'] ?></td>

```

```

        <td><input type="text" name="quantity[<?php echo $row['id_product'] ?>]" size="5" value="<?php echo
$_SESSION['cart'][$row['id_product']]['quantity'] ?>" /></td>
        <td><?php echo $row['price'] ?>$</td>
        <td><?php echo $_SESSION['cart'][$row['id_product']]['quantity']*$row['price'] ?>$</td>
    </tr>
</?php
    }
?>

    <tr>
        <td colspan="4">Total Price: <?php echo $totalprice ?></td>
    </tr>

</table>
<br />
<button type="submit" name="submit">Update Cart</button>
</form>
<br />
<p>To remove an item set its quantity to 0. </p>

```

## CSS:

```

body {
    font-family: Verdana;
    font-size: 12px;
    color: #444;
}

a {color: #48577D; text-decoration: none;}

a:hover {text-decoration: underline;}

h1, h2 {margin-bottom: 15px}

h1 {font-size: 18px;}
h2 {font-size: 16px}

#container {
    width: 700px;
    margin: 150px auto;
    background-color: #eee;
    padding: 15px;
    overflow: hidden;
}

#main {
    width: 490px;
    float: left;
}

#main table {
    width: 480px;
}

#main table th {
    padding: 10px;
    background-color: #48577D;
    color: #fff;
    text-align: left;
}

#main table td {

```

```
padding: 5px;
}

#main table tr {
background-color: #d3dcf2;
}

#sidebar {
width: 200px;
float: left;
}
```

