

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

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка и администрирование системы управления контентом интернет магазина»

Обучающийся

В.В. Родичев

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н. Н.В. Хрипунов

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

Консультант

С.А. Гудкова

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

Тольятти 2024

Аннотация

Тема бакалаврской работы – «Разработка и администрирование системы управления контентом интернет магазина».

Работа выполнена студентом Тольяттинского государственного университета, института математики, физики и информационных технологий, группы Моб-2002а, Родичевым Владимиром Владимировичем.

В современном информационном обществе интернет-магазины становятся неотъемлемой частью коммерческой деятельности, предоставляя потребителям возможность совершать покупки онлайн в любое удобное время. Развитие электронной коммерции и увеличение конкуренции на рынке делают необходимым постоянное совершенствование процессов управления и предоставляемого контента.

Объектом исследования является процесс разработки и администрирования системы управления контентом интернет-магазина.

Предметом исследования является функциональность, структура и методы администрирования системы управления контентом, а также их влияние на эффективность работы интернет-магазина.

Целью данной работы является разработка и исследование системы управления контентом для интернет-магазина с целью повышения его эффективности и конкурентоспособности, а также анализ методов администрирования данной системы.

Основной задачей работы является изучение и анализ современных подходов к разработке и администрированию систем управления контентом, а также определение требований к функциональности системы на основе особенностей интернет-магазина и потребностей пользователей. Кроме того, проектирование и разработка собственной системы управления контентом, ориентированной на специфику онлайн-платформы, станет важным этапом в работе над проектом.

В первом разделе были рассмотрены функциональные и технологические возможности контент-систем.

Во втором разделе мы исследовали и анализировали различные программные инструменты, необходимые для реализации системы. Этот процесс включал в себя оценку функциональности, производительности.

Третий раздел работы был посвящен проектированию базы данных и сервера разметок а также тестированию функциональности готового приложения.

Бакалаврская работа состоит из введения, трех разделов, заключения и списка используемой литературы.

В ней присутствуют 16 рисунков, 10 таблиц. Список используемой литературы состоит из 32 источников. Общий объем выпускной квалификационной работы составляет 50 страниц.

Abstract

The theme of the bachelor's thesis is ‘Development and administration of the content management system of the Internet shop’.

The work was done by a student of Togliatti State University, Institute of Mathematics, Physics and Information Technology, group Mob-2002a, Rodichev Vladimir Vladimirovich Rodichev.

In the modern information society online shops are becoming an integral part of commercial activity, providing consumers with the opportunity to shop online at any convenient time. The development of e-commerce and increased competition in the market make it necessary to constantly improve the management processes and the content provided.

The object of the study is the process of development and administration of the content management system of an online shop.

The subject of the study is the functionality, structure and methods of administration of the content management system, as well as their impact on the efficiency of the online shop.

The purpose of this paper is to develop and research a content management system for an online shop in order to improve its efficiency and competitiveness, as well as to analyse the methods of administration of this system.

In the first section we considered the functional and technological capabilities of content systems.

In the second section, we investigated and analysed the different software tools required to implement the system. This process included evaluation of functionality, performance.

The third section of the paper was devoted to designing the database and markup server and testing the functionality of the finished application.

Содержание

Введение.....	6
1. Анализ предметной области архитектура и среда реализации проектируемой среды	8
1.1 Виды сайтов и основные понятия CMS.....	8
1.2 Классификация web-приложений.....	11
1.3 Аналитический обзор решений и требований предъявляемых к системам управления контентом	13
1.4 Выбор архитектуры системы управления контентом	14
2. Архитектура и среда реализации проектируемой системы	19
2.1 Выбор программных средств.....	19
2.2 Структура системы управления контентом	20
3 Реализация и тестирование системы	27
3.1 Проектирование базы данных и сервера разметок.....	27
3.2 Разработка модульной структуры приложения	35
3.3 Тестирование функциональности	37
Заключение	46
Список используемой литературы и используемых источников	47

Введение

С тем, как интернет изначально начал своё существование как средство обмена информацией между научными центрами, прозрачность и простота его технологий были не в приоритете. Но в последние годы этот огромный мир технологий начал обретать новые формы, становясь более доступным для всех.

Системы управления контентом (CMS) стали основным строительным блоком этого нового интернета. Теперь сайты, созданные на их основе, стали не просто визитной карточкой, а мощным инструментом, управляемым контент-менеджерами и специалистами по PR. Технические аспекты управления ресурсом ушли на второй план, что позволяет существенно снизить затраты на поддержку сайта и оперативно вносить изменения в его структуру. Этот подход изменил представление о сайтах - они перестали быть статическими и начали жить своей жизнью, активно реагируя на изменения в окружающем мире.

Теперь пользователь видит лишь вершушку айсберга, а подводная часть – система управления, остаётся скрытой. Для пользователей интернета это чёрный ящик, но разработчики знают: основа современных сайтов - это системы управления контентом.

Современная система управления контентом превосходит видимую часть веб-ресурса не только в объеме, но и в сложности разработки. Она обеспечивает возможность редактирования информации, выполнение различных административных задач, сбор и анализ разнообразной статистики о посетителях и многое другое.

В современном информационном обществе интернет-магазины становятся неотъемлемой частью коммерческой деятельности, предоставляя потребителям возможность совершать покупки онлайн в любое удобное время. Развитие электронной коммерции и увеличение конкуренции на

рынке делают необходимым постоянное совершенствование процессов управления и предоставляемого контента.

Актуальность данного исследования обусловлена стремительным развитием интернет-торговли и повышением требований к функциональности и эффективности интернет-магазинов. Разработка и администрирование системы управления контентом играют ключевую роль в обеспечении конкурентоспособности интернет-магазина, улучшении пользовательского опыта и повышении его эффективности.

Объектом исследования является процесс разработки и администрирования системы управления контентом интернет-магазина.

Предметом исследования является функциональность, структура и методы администрирования системы управления контентом, а также их влияние на эффективность работы интернет-магазина.

Данное исследование нацелено на создание инновационной системы управления контентом для интернет-магазина с целью оптимизации его работы и повышения конкурентоспособности.

Основной задачей работы является изучение и анализ современных подходов к разработке и администрированию систем управления контентом, а также определение требований к функциональности системы на основе особенностей интернет-магазина и потребностей пользователей. Кроме того, проектирование и разработка собственной системы управления контентом, ориентированной на специфику онлайн-платформы, станет важным этапом в работе над проектом.

1. Анализ предметной области архитектура и среда реализации проектируемой среды

1.1 Виды сайтов и основные понятия CMS

«В основе новой эпохи интернета лежат системы управления контентом (CMS), которые превращают сайты в мощные бизнес-инструменты, управляемые легко и прямо контент-менеджерами и пиар-специалистами. Это исключает необходимость постоянной технической поддержки, что снижает общую стоимость и обеспечивает быструю реакцию на изменения в структуре сайта» [1].

«Поэтому разработка сайта – это только первый шаг в его жизненном цикле. Хороший веб-сайт не статичный монумент, а живой организм, способный быстро адаптироваться к изменениям во внешней среде. Вся информация на сайте должна обновляться с необходимой оперативностью, соответствуя требованиям бизнеса» [3].

«Наилучшим решением для владельцев сайтов, с точки зрения оперативности, является наличие постоянного веб-мастера. Только такой специалист сможет быстро и качественно вносить изменения. Однако этот вариант может быть не самым экономичным, особенно когда требуется поддержка сайта без постоянного потока новых разработок, если это не большой портал или интернет-магазин» [29].

«Альтернативный подход заключается в использовании системы управления содержимым, которая позволяет управлять сайтом без необходимости обращаться к сторонним веб-мастерам или заключать договоры на поддержку» [2].

Вместо обращения к веб-мастерам, что не только требует тщательного описания всех изменений, но и подразумевает дополнительные расходы, стоит рассмотреть альтернативу – заключение договора на поддержку сайта. Однако, такой подход не лишен недостатков: платежи за поддержку

продолжаются даже тогда, когда изменения не требуются. Именно здесь на сцену выходит CMS – система управления содержимым сайта, позволяющая даже неопытным пользователям оперативно и самостоятельно вносить изменения, минуя сложности языка разметки и программирования.

«Такой подход дает возможность быть независимым от внешних специалистов и свободно управлять содержимым своего сайта. Дополнительно, можно возложить ответственность за обслуживание сайта на секретаря или другого сотрудника, который сможет эффективно осуществлять данные задачи» [4].

Давайте рассмотрим привилегии, которые системы управления контентом (CMS) предоставляют специалистам в области веб-разработки и администрирования веб-сайтов.

«CMS (Content Management System) решает эту проблему. После ее настройки в соответствии с дизайном сайта все последующие изменения будут автоматически соответствовать оформлению. При этом редактирование HTML-кода будет требоваться только в случае действительно сложных дизайнерских задач» [5].

Внедрение систем управления контентом (CMS) открывает перед разработчиками и заказчиками целый спектр преимуществ [6].

Для разработчиков это:

- повышение стандартов: CMS обеспечивает единые инструменты разработки, способствуя созданию продуктов высокого качества;
- оптимизация процессов: Благодаря модульной структуре и функциональности CMS, разработчики могут сконцентрироваться на ключевых аспектах проекта, сокращая время на разработку основных функций;
- максимизация прибыли: Использование CMS позволяет разработчикам сконцентрироваться на более прибыльных задачах, в то время как управление контентом автоматизировано, снижая затраты на рутинные операции.

Для заказчиков:

- гибкость в редактировании: CMS обеспечивает возможность быстрого и удобного редактирования контента без необходимости обращения к разработчику, позволяя быстро адаптировать информацию под текущие потребности рынка;
- свобода выбора в управлении: Заказчики имеют возможность вести управление содержимым своего веб-ресурса независимо, обходя необходимость постоянной связи с разработчиком, что способствует увеличению гибкости и эффективности в процессе управления веб-сайтом;
- экономия времени и ресурсов: Использование CMS сокращает время и деньги, затрачиваемые заказчиком, так как он может оперативно вносить изменения и управлять контентом без дополнительных расходов на обучение или ожидание ответа от разработчика.

Таким образом, использование систем администрирования контента при разработке и обслуживании веб-сайтов принесет ощутимые преимущества как разработчикам, так и заказчикам. Это улучшит эффективность работы, повысит качество и оперативность изменений на сайте, а также способствует более эффективному взаимодействию между всеми участниками процесса.

Давайте рассмотрим преимущества, которые системы управления контентом (CMS) предоставляют специалистам в области веб-разработки и веб-мастерам. При управлении десятком веб-сайтов, внесение изменений в каждый из них с учетом их уникального дизайна может представлять собой чрезвычайно сложную задачу. Более того, необходимо учитывать все особенности стилизации каждого веб-ресурса.

«Система управления контентом (Content Management System - CMS) представляет собой инновационную платформу, разработанную для эффективного контроля и организации содержимого и структуры интернет-ресурсов в режиме онлайн. Одним из главных преимуществ CMS является

возможность для любого пользователя, даже без специальных навыков в области информационных технологий, самостоятельно управлять содержимым своего веб-сайта, без необходимости привлекать технический персонал» [7].

«По своей сути, веб-сайт – это определенная территория на сервере, представляющая собой целостную часть, посвященную определенной тематике или концепции. Безусловно, большинство сайтов содержат множество подразделов, каждый из которых может быть дополнительно разделен на более мелкие компоненты. Однако все эти составляющие объединяет общая идея, основное содержание и общий стиль выполнения. Относительно их характеристик, сайты могут условно разделяться на два основных типа: статические и динамические» [8].

«В 90-х годах, доминирующей системой управления контентом была следующая: для создания веб-сайтов использовались набор инструментов, включающих web-сервер, предоставляемый хостинг-провайдером, редакторы HTML-документов (от простых Notepad до мощного Macromedia DreamWeaver) и FTP-клиенты (например, CuteFTP, gFTP, FAR или TotalCommander)» [9]. Структура сайта формировалась путем создания ссылок между страницами и, в некоторых случаях, добавления меню для навигации. Эта методика получила название "статическая", поскольку содержимое каждой страницы сохранялось в статических файлах.

1.2 Классификация web-приложений

На рисунке 1 представлена категоризация онлайн-приложений, востребованных коммерческими предприятиями в своей деятельности:

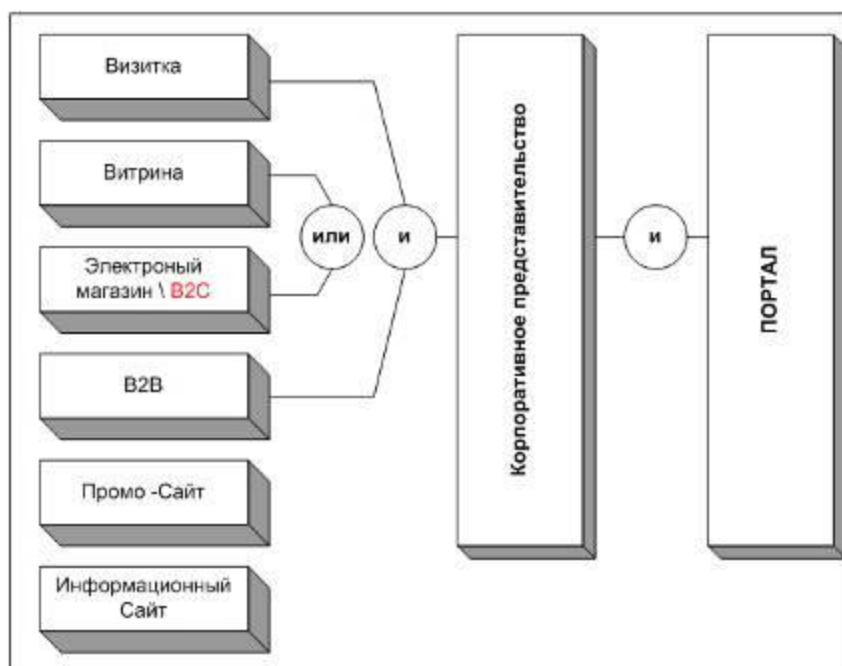


Рисунок 1 – Классификация web-приложений

- представительство: веб-платформы данного типа обычно служат для распространения информации о компании, предоставляя контактные данные, ценовую политику, местоположение и прочее. Они создают набор страниц, возможно, с формой для обратной связи;
- виртуальный магазин: помимо базовых функций представительства, здесь имеется каталог товаров или услуг. Посетители могут ознакомиться с подробными описаниями, сопровождаемыми фотографиями, схемами и презентациями. Часто встречается информация о ценах. Заказы могут совершаться через электронную почту или альтернативные средства связи и обрабатываются сотрудниками;
- электронная торговая площадка (B2C): подобные ресурсы предлагают продукцию конечному потребителю, включая традиционные магазины, аукционы и другие торговые платформы.
- взаимодействие бизнеса с бизнесом (B2B): это приложения, предназначенные для взаимодействия с партнерами и дилерами,

обычно снабжены инструментами для платежей, бирж и партнерских программ;

- промо-площадка: с использованием языков программирования, таких как JavaScript и ActiveX, создаются динамичные и впечатляющие страницы, призванные привлечь внимание посетителей;
- информационный ресурс: эти веб-сайты предоставляют обширную информацию в сфере деятельности предприятия и поддерживают взаимодействие между посетителями;
- корпоративный портал: самые развитые приложения, объединяющие функционал представительства, магазина, B2B-платформы и других систем управления, применяемые в крупных компаниях с разветвленной инфраструктурой;
- мультипортал: сложные веб-приложения, объединяющие функционал нескольких типов ресурсов и интегрирующиеся с различными системами управления и сетями.

В зависимости от контекста и требований проекта, можно использовать и другие критерии классификации так, как их гораздо больше.

1.3 Аналитический обзор решений и требований предъявляемых к системам управления контентом

«Пока еще нет четкой классификации систем управления контентом, потому что рынок таких систем относительно молод и разработчики предлагают разные продукты. Классификацию делает еще сложнее то, что нужно учитывать множество параметров, которые различают одну систему от другой. Поэтому любую классификацию можно рассматривать как условную, а не абсолютно точную» [10].

Для полной оценки систем управления контентом учитываются следующие ключевые аспекты:

- пользовательский опыт: разнообразие функций и модулей, доступность и простота использования для конечного пользователя являются ключевыми факторами. Чем больше функциональных возможностей предлагает система и чем проще она в освоении, тем выше оценивается уровень пользовательского сервиса;
- используемые технологии: оценивается применение передовых технологий, способствующих увеличению производительности, пропускной способности и надежности системы управления контентом. Применение современных технологий в хранении данных и обработке запросов признается как показатель технологичности системы;
- совместимость: оценивается способность системы функционировать на различных аппаратных и программных платформах, а также совместимость с различными программными решениями, включая системы управления базами данных. Важным является также возможность подключения дополнительных модулей для расширения функциональности;
- масштабируемость: учитывается способность системы расти и масштабироваться в зависимости от потребностей пользователя. Гибкость и возможность расширения функциональности и мощности системы считаются важными аспектами, особенно для организаций, планирующих увеличение контента и потоков пользователей.

Подходящие системы управления контентом оцениваются по каждому из этих критериев, чтобы помочь пользователям выбрать продукт, отвечающий их требованиям и предпочтениям.

1.4 Выбор архитектуры системы управления контентом

Разрабатываемая система управления контентом (CMS) должна эффективно работать в Интернете для веб-сайта. Поскольку это онлайн-

приложение, оно должно работать со многими пользователями одновременно. Код приложения должен находиться на веб-сервере с необходимым программным обеспечением, таким как система управления базами данных (СУБД) и языковой компилятор. В базе данных СУБД хранятся сведения о фильмах, пользователях и т. д.

Поскольку база данных расположена удаленно от клиента, используется архитектура системы «клиент-сервер». Это разделяет приложение на два компонента: клиент (видимый интерфейс в браузере пользователя) и сервер (удаленный компьютер, на котором размещено системное программное обеспечение). Клиент отправляет запросы на сервер через интернет-протоколы, которые обрабатывают их и возвращают ответы, отображаемые на мониторе клиента.

Основное преимущество клиент-серверной архитектуры заключается в том, что на стороне клиента не требуется дополнительного программного обеспечения, за исключением интернет-браузера. Это позволяет программным кодам выполняться на стороне сервера, уменьшая необходимость в сторонних программах на клиентском компьютере. Следовательно, система может охватить более широкую базу пользователей благодаря более низким требованиям к клиентскому компьютеру по сравнению с системами с более высокими требованиями.

На рисунке 2 показано, как клиентские запросы обрабатываются сервером через интерпретатор языка программирования в типичном сценарии представления.

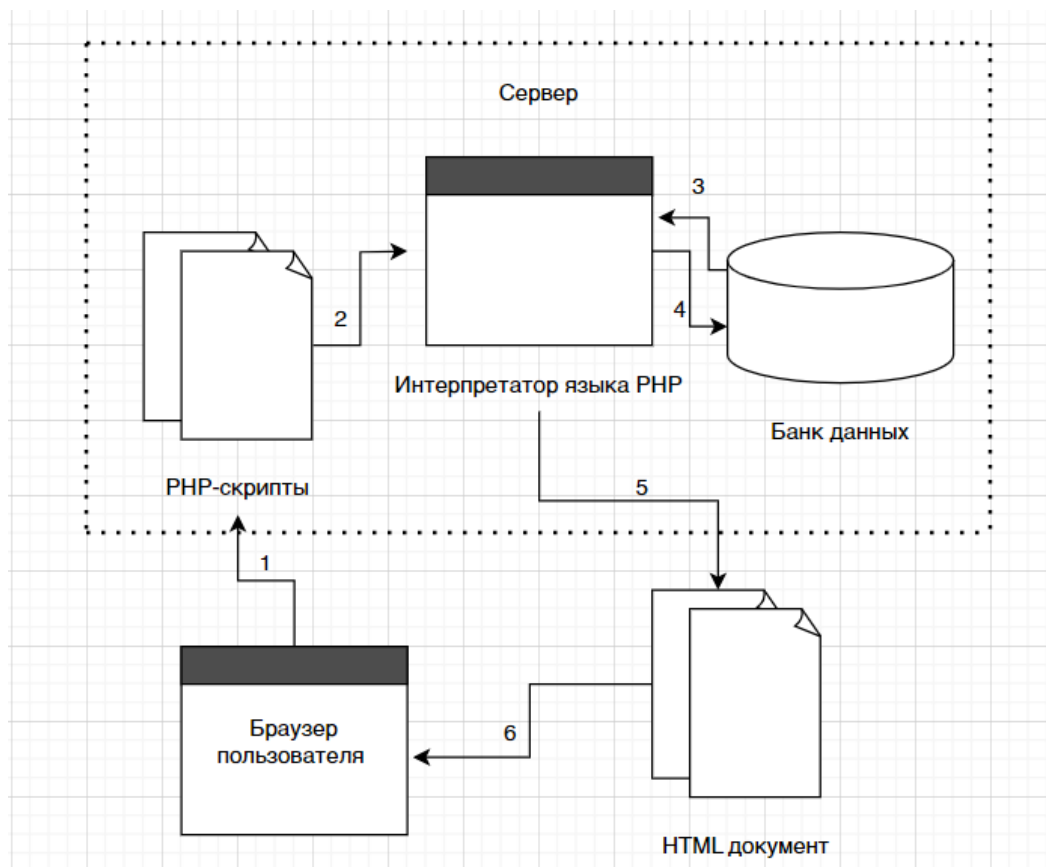


Рисунок 2 – Схема перемещения потоков информации

Клиент инициирует запрос к веб-серверу, который затем пересылает запрос на обработку интерпретатору языка PHP. Интерпретатор взаимодействует с базой данных по мере необходимости.

После завершения обработки интерпретатор отправляет ответ на сервер, который впоследствии доставляет клиенту окончательную HTML-страницу.

Система управления контентом представляет собой сайт просмотра товаров интернет-магазина. Поэтому при выборе инструментов разработки систем следует отдавать предпочтение технологиям, подходящим для веб-программирования.

Поскольку система постоянно хранит данные о продуктах, клиентах и т.д., использование базы данных имеет важное значение для бесперебойной работы приложений. Разрабатываемое веб-приложение требует наличия веб-

сервера для интерпретации программных кодов и выполнения запросов к базе данных. В целом, разработка системы включает в себя веб-сервер, СУБД и язык программирования для интернет-приложений.

Выбор веб-сервера – простая задача. Для пользователей Windows главными претендентами являются «Internet Information Server (IIS)» и Apache. Хотя IIS легко интегрируется с Windows, он ограничен семейством Windows Server и требует опыта настройки. Apache, сервер с открытым исходным кодом, пользуется популярностью во всем мире благодаря своей кроссплатформенной совместимости, обширным библиотекам и удобной конфигурации.

Несмотря на необходимость перезапуска служб после внесения изменений, для данного конкретного проекта выбрана версия веб-сервера Apache 1.3.41.

На рисунке 3 представлено взаимодействие выбранных технологий.

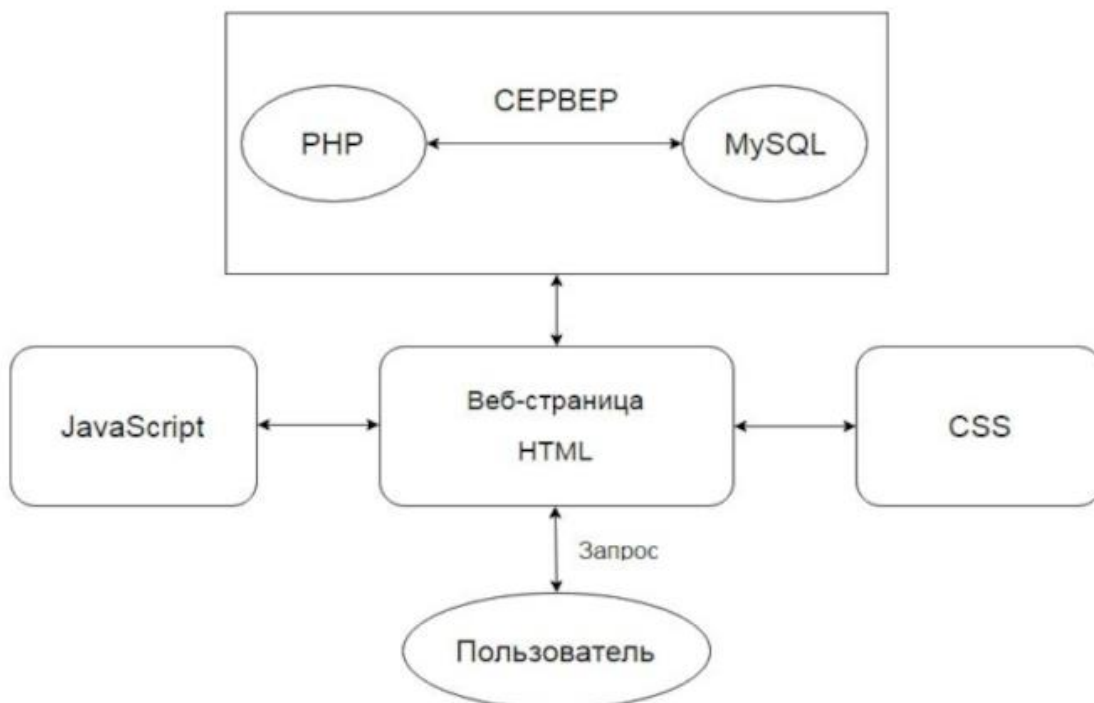


Рисунок 3 – Взаимодействие технологии

Вывод по первому разделу:

В первом разделе бакалаврской работы были рассмотрены функциональные и технологические возможности CMS. В данном разделе были определены основные ключевые функции.

В рамках анализа предметной области была проведена оценка архитектуры и среды реализации проектируемой среды. Были рассмотрены различные виды сайтов и основные понятия систем управления контентом (CMS). Также была проведена классификация веб-приложений и аналитический обзор решений, а также требований, предъявляемых к системам управления контентом. На основе проведенного анализа был сделан выбор архитектуры системы управления контентом.

Анализ предметной области позволил получить полное представление о существующих подходах и технологиях, используемых в разработке веб-приложений и систем управления контентом. Это позволило определить требования и цели проектируемой среды, а также выбрать наиболее подходящую архитектуру для системы управления контентом.

В ходе рассмотрения уже существующих решений было принято решение на разработку собственной системы управления контентом интернет-магазина, так как ни одно из решений не подошло.

2. Архитектура и среда реализации проектируемой системы

2.1 Выбор программных средств

Разрабатываемая нами система управления, должна эффективно обрабатывать текстовую информацию и обеспечивать добавление, изменение и удаление записей в базе данных. Далее мы перейдем к рассмотрению нескольких языков программирования, которые специализируются на создании подобных веб-приложений.

«Одними из самых популярных и востребованных языков программирования для веб-приложений являются Perl и PHP. Они оба предназначены специально для веба. Однако, при выборе между ними важно помнить, что Perl сильнее ориентирован на обработку текста, включая множество функций и мощные инструменты для работы с регулярными выражениями» [11].

Ключевой особенностью языка PHP является его практичность, предоставляющая программисту средства для быстрого и эффективного решения задач. Этот практичный характер PHP обусловлен пятью важными качествами: традиционностью, простотой, эффективностью, безопасностью и гибкостью.

PHP привлекателен не только своими техническими характеристиками, но и тем, что он доступен бесплатно и имеет открытый исходный код (Open Source). Все перечисленные положительные качества языка PHP свидетельствуют о том, что данный язык оптимально подходит для решения поставленных задач. Поэтому, PHP версии 8.2 является лучшим выбором для разработки системы.

Разрабатываемая система требует для своей работы наличия базы данных. Поскольку приложение и база данных будут размещены на веб-сервере, требуется провести сопоставительный обзор различных СУБД,

совместимых с языком программирования PHP, для определения наиболее подходящего варианта.

«Для web-сервера можно установить различные СУБД, такие как MySQL, SQLite, Firebird и Oracle. Из-за того, что MySQL бесплатен, широко распространен и прост в использовании, для разрабатываемой системы мы выберем именно MySQL» [12].

«MySQL представляет собой быструю, легко настраиваемую и простую в использовании и администрировании СУБД. Она существует в различных версиях для Unix и Windows, а приложения, работающие с MySQL, могут быть написаны на различных языках программирования. MySQL активно применяется совместно с веб-серверами для создания динамических веб-сайтов на основе баз данных» [13].

При разработке сайта, использовался объектно-ориентированный скриптовый язык программирования JavaScript. Он был выбран из-за своей оперативности, что позволяет обрабатывать скрипты непосредственно в браузере без обращения к серверу. JavaScript использовался для проверки форм перед их отправкой на сервер, создания подсказок, предупреждений и других элементов, а также для обработки различных событий, таких как onClick, onLoad, onFocus и т.д.

Мы представили список всех необходимых и используемых в работе технических и программных средств и инструментов для реализации поставленной задачи: языки программирования PHP и JavaScript, язык вёрстки HTML/CSS, среда разработки Sublime Text 2, а также серверные технологии: PHP 5.5, веб-сервер Apache 1.3.41, СУБД MySQL 5.5 и MySQL Workbench 6.3.

2.2 Структура системы управления контентом

Использование CMS при разработке и поддержке сайта дает множество преимуществ. И они обеспечивают возможность создания и редактирования

контента сайта интернет-магазина а также, удобное управление и контроль доступа. Даже без специальных навыков в IT можно публиковать информацию в нужном формате. Тем не менее, современные веб-платформы не всегда могут удовлетворить разнообразные запросы пользователей в отношении необходимой им информации.

В современном мире интернет-торговли, где конкуренция нарастает с каждым днем, важно пересмотреть стратегию взаимодействия с посетителями сайта. Каждый пользователь имеет уникальные интересы, и именно поэтому персонализация контента становится ключевым фактором.

Если информация на сайте не соответствует ожиданиям или сильно отличается от запросов пользователя, это может привести к уменьшению активности или даже к уходу посетителя. Создание персонализированного контента может показаться сложной задачей, однако современные технологии сбора и анализа данных делают это осуществимым. Группировка пользователей по интересам и индивидуальная настройка сайта для каждой группы - вот путь к эффективному взаимодействию с аудиторией.

В связи с ограничениями типовых структур CMS, неспособных к эффективной персонализации контента, требуется модификация модуля управления контентом и базы данных для адаптации к новым потребностям. Кроме того, необходимо интегрировать Базу знаний, где будут храниться правила формирования персонализированного контента можно увидеть на рисунке 4.

Новая система предлагает следующие функциональные возможности:

- автоматическое создание контента для сайта, включая динамические материалы для страниц и настраиваемые блоки с учетом индивидуальных предпочтений;
- централизованное управление контентом, обеспечивающее версионный контроль документов и файлов, безопасное хранение, доступ и интеграцию с другими системами;

- классификацию пользовательских профилей по их уникальным характеристикам, что позволяет создавать целевые группы;
- адаптивную публикацию и отображение контента, которое учитывает индивидуальные потребности различных групп пользователей и соответствует общему визуальному стилю сайта.



Рисунок 4 – Структура системы управления контентом

В архитектуре системы присутствуют следующие важнейшие составляющие:

- модуль навигации: Он формирует структуру сайта, устанавливает взаимосвязи и иерархию страниц, а также определяет отображение меню;
- модуль контента: Здесь хранится информация о содержании структурных элементов страницы, обеспечивая необходимый контент;

- модуль управления контентом: Этот модуль осуществляет гиперсегментацию пользовательских профилей и управляет процессом подбора контента для страниц;
- CSS технология: применяется для формирования стиля элементов веб-страниц через язык разметки гипертекста;
- дизайн-шаблон: определяет внешний вид всех элементов страниц сайта;
- файловая система: управляет загружаемыми файлами с сервера, включая ценовые листы, каталоги моделей и изображения товаров;
- модуль аутентификации: обеспечивает управление процессами регистрации и входа пользователей в систему;
- база данных: содержит различные параметры, включая информацию о дизайне, заголовки, метаданные, настройки структуры и модули, контент, учетные записи и данные пользователей;
- база знаний: включает правила для выбора контентных элементов на страницах для различных пользовательских групп.

В дополнение к основным функциям типичных систем управления контентом, наша новая платформа осуществляет кластеризацию профилей пользователей для их группировки в соответствии с общими характеристиками. Каждая сформированная группа получает персонализированный контент при посещении веб-страницы. Для эффективного анализа данных о пользователях необходимо хранить информацию о них в централизованной базе данных системы, включая данные о новых пользователях. При внедрении модуля персонализации, который отвечает за управление контентом с использованием новой методологии, информация о пользователях передается в соответствующий модуль для проведения анализа.

«Варианты материалов также хранятся в базе данных. Администраторы также создают и сохраняют правила, по которым будет осуществляться подбор контента для пользователей различных групп. Все эти правила

сохраняются в базе знаний, и при обращении к ней конкретное правило передается в модуль управления контентом. Этот модуль выбирает необходимый контент из базы данных и формирует страницу для пользователя.

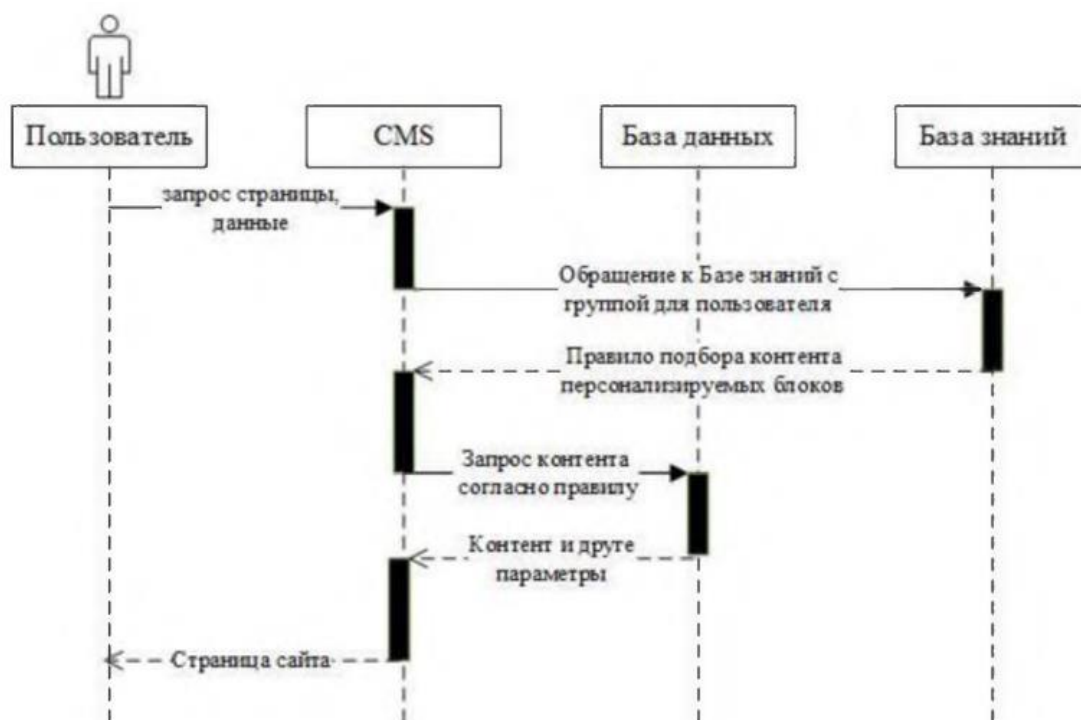


Рисунок 5 – Диаграмма последовательности

Процесс взаимодействия компонентов новой системы в рамках персонализированного управления контентом отображается на рисунке 5 с использованием диаграммы последовательности в нотации UML» [14].

- «При обращении пользователя к странице сайта, запрос попадает в систему управления контентом сайта, сопровождаемый данными о пользователе (например, источник перехода на сайт, поисковый запрос пользователя, его географическое расположение, Cookies, история просмотров страниц в текущем сеансе работы браузера и другие данные);

- на основе определенного кластера система управления контентом обращается к Базе знаний для запроса информации;
- в ответ на запрос системы поступает правило подбора содержимого персонализируемых блоков страницы сайта;
- система направляет запрос контента в Базу данных;
- система получает не только необходимый контент, но и все параметры, включая дизайн, структуру и настройки модулей. На основе этой информации из Базы данных, шаблона страницы и настроек дизайна происходит сборка страницы;
- затем система управления контентом сайта передает собранную страницу, специально настроенную для пользователя» [15].

Далее нами была разработана функциональная модель на рисунке 6.

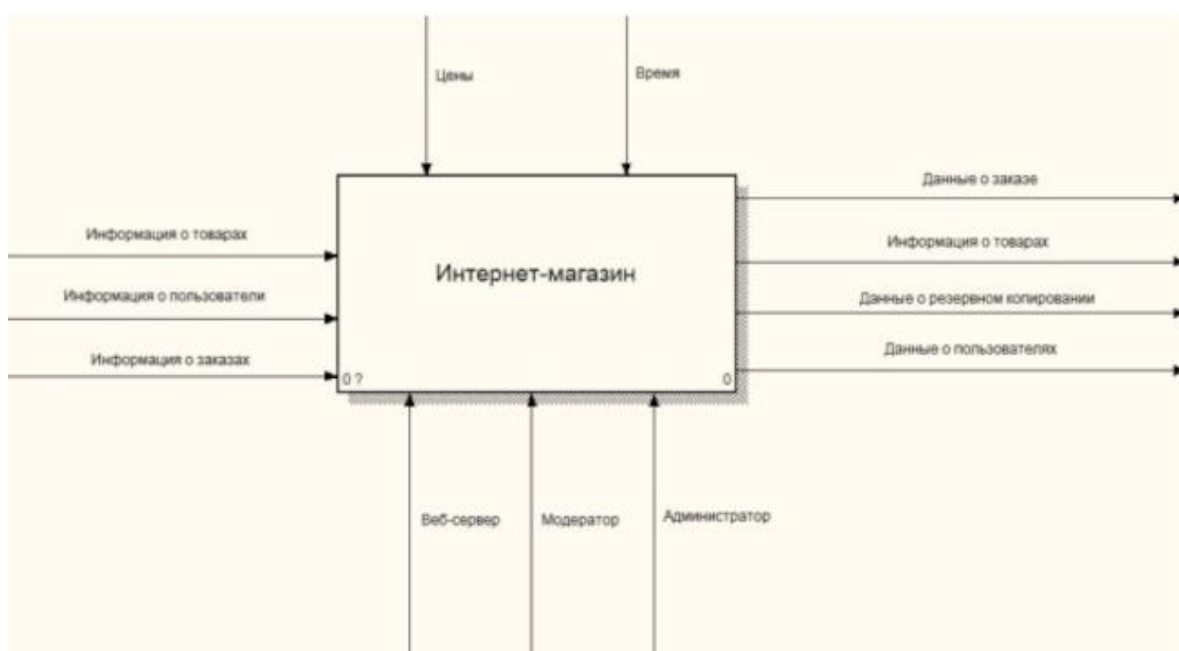


Рисунок 6 – Функциональная модель

Функциональная модель системы включает в себя следующие компоненты:

- управление каталогом товаров: возможность добавления, удаления и редактирования товаров, категорий и подкатегорий;

- управление описаниями товаров: функционал для изменения текстовых и графических описаний товаров;
- управление акциями и скидками: возможность создания и редактирования акций, а также управление скидками на товары;
- управление заказами: модуль для просмотра, обработки и управления заказами от покупателей;
- управление пользователями: возможность администрирования пользовательских аккаунтов и прав доступа.

Вывод по второму разделу

В ходе проектирования системы управления контентом были рассмотрены ключевые аспекты ее архитектуры и среды реализации. Выбор программных средств для разработки системы основывался на анализе требований к функциональности, масштабируемости и интеграции с существующими системами. Были выбраны наиболее подходящие инструменты и технологии, обеспечивающие эффективное решение поставленных задач.

Во втором разделе наш основной фокус был направлен на выбор программных инструментов, определение структуры системы управления контентом, проектирование БД и сервера разметок, а также разработку модульной структуры приложения.

Мы исследовали и анализировали различные программные инструменты, необходимые для реализации нашего проекта. Этот процесс включал в себя оценку функциональности, производительности.

3 Реализация и тестирование системы

3.1 Проектирование базы данных и сервера разметок

«Основа работы создаваемой CMS заключается в использовании базы данных, где будет храниться информация, вводимая пользователями (контент страниц). Для этого необходимо определить группы и параметры данных, структурировать их в реляционные таблицы, наладить связи между ними, а также определить первичные ключи, индексы и провести нормализацию структуры» [16].

Завершением процесса проектирования станет схема базы данных и стандартный SQL-скрипт для ее создания. На данном этапе разработки используются шесть таблиц.

В таблице 1, Admins хранятся учетные записи пользователей и их пароли.

Таблица 1 – Admins

Поле	Тип	Ноль	Комментарии
id	int(3)	Нет	Уникальный индекс пользователя
login	varchar(255)	Нет	Имя пользователя
PASSWORD	varchar(255)	Нет	Пароль пользователя

Таблица для хранения информации о администраторах системы управления контентом интернет-магазина. Здесь могут быть храниться данные о пользователях, имеющих доступ к административной панели, такие как их идентификаторы, имена, адреса электронной почты, пароли, роли и привилегии.

Contents предназначена для хранения текстового контента, данные представлены в таблице 2.

Таблица 2 – Mails

Поле	Тип	Ноль	Комментарии
id	bigint(20)	Нет	Уникальный идентификатор
mail	varchar(255)	Нет	Имя электронного адреса

Эта таблица может использоваться для хранения данных о переписке, например, для обратной связи с клиентами или для управления рассылками. Она может содержать информацию о сообщениях, отправителях, получателях, дате отправки, статусе доставки и другие метаданные.

В таблице 3 – News хранится информация о последних событиях.

Таблица 3 – News

Поле	Тип	Ноль	Комментарии
id	int(3)	Нет	Уникальный идентификатор
title	varchar(255)	Нет	Имя новости
preview	text	Нет	Предпросмотр, который отображается в «ленте новостей»
content	text	Нет	Содержание
image	varchar(50)	Нет	Изображение, которое выводится в ленту новостей
news_date	varchar(8)	Нет	Число в формате (Ymd), когда контент была опубликована.

Таблица для хранения новостей или обновлений интернет-магазина. Здесь могут содержаться заголовки, тексты новостей, дата публикации, авторы и другая связанная информация.

Gb (от англ. guest book) – хранит информацию о посетителях и их отзывы, таблица 4.

Эта таблица предназначена для хранения отзывов и комментариев посетителей интернет-магазина. В ней могут быть поля для имени пользователя, текста отзыва, даты написания, оценки и других соответствующих данных.

Таблица 4 – Gb

Поле	Тип	Ноль	Комментарии
id	bigint(20)	Нет	Уникальный идентификатор
author	varchar(255)	Нет	Имя автора
mail	varchar(255)	Нет	Адрес электронной почты
content	text	Нет	Содержание
dt	varchar(8)	Нет	Дата
active	enum('yes', 'no')	Нет	Индикатор доступности статьи для просмотра, влияющий на видимость её для посетителей сайта. (При редактировании статьи она может временно скрываться от пользователей)

В таблице 5 Photo_album, предназначена для организации изображений в альбомах.

Таблица 5 – photo_album

Поле	Тип	Ноль	Комментарии
id	int(11)	Нет	Уникальный идентификатор фотоальбома (раздела)
name	varchar(255)	Нет	Название фотоальбома
order_id	int(11)	Нет	Порядковый индекс определяет порядок отображения названий фотоальбомов на странице
Поле	Тип	Ноль	Комментарии
id	int(11)	Нет	Уникальный идентификатор фотоальбома (раздела)
name	varchar(255)	Нет	Название фотоальбома

Таблица для хранения информации об альбомах с фотографиями. Здесь могут содержаться данные о названии альбома, описании, дате создания и другие метаданные.

В таблице Photo_image представлены данные для хранения и управления графическими файлами.

Таблица 6 – photo_image

Поле	Тип	Ноль	Комментарии
id	int(11)	Нет	Уникальный идентификатор
name	varchar(255)	Нет	Название фотоальбома
order_id	int(11)	Нет	Порядковый индекс определяет порядок отображения названий фотоальбомов на странице.
Поле	Тип	Ноль	Комментарии
id	int(11)	Нет	Уникальный идентификатор фотоальбома (раздела)
name	varchar(255)	Нет	Название фотоальбома

Эта таблица может использоваться для хранения самих изображений, которые относятся к альбомам в таблице photo_album. В ней могут храниться пути к файлам изображений, описания, метки и другая связанная информация.

Благодаря таблицам, которые мы разработали, составим логическую модель данных, которая определяет структуру базы данных на рисунке 7.

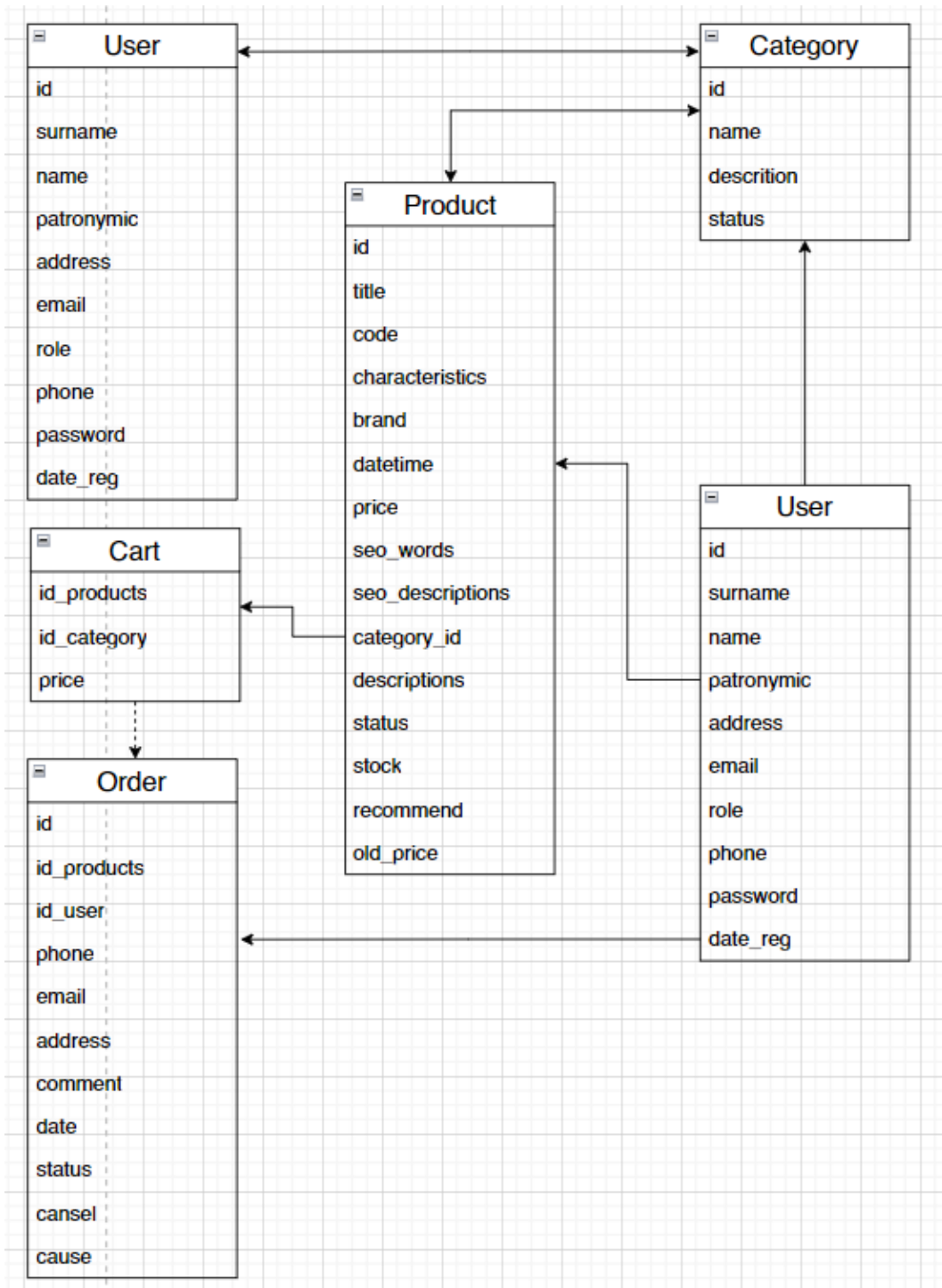


Рисунок 7 – Логическая модель данных

Физическая реализация данных описывает конкретное воплощение логической модели в форме таблиц базы данных, включая определение типов данных и ограничений (рисунок 8).

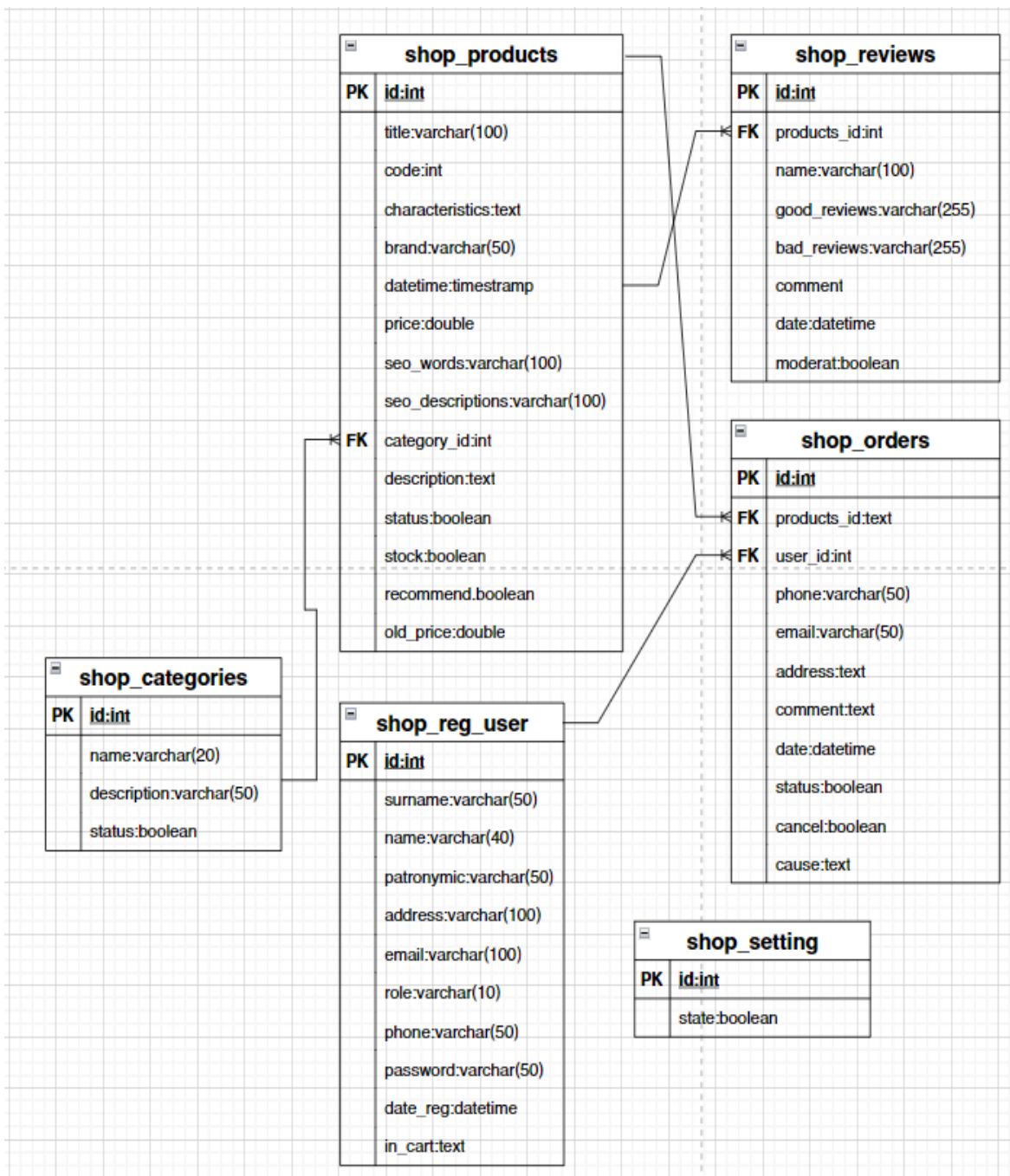


Рисунок 8 – Физическая модель данных

«Основная цель системы управления контентом (CMS) заключается в эффективной обработке контента. Поэтому первый компонент базы данных должен обеспечивать его надежное хранение. Информационный контент сайта обычно структурирован и разделен на тематические категории, следовательно, база данных содержимого сайта включает как минимум две таблицы: одну для самого контента и другую для категорий, к которым он

относится. Рекурсивная модель категорий характеризуется параметрами, описанными в таблице» [28].

При выборе типов данных для полей `Id` и `Part` учитывается, что количество категорий обычно значительно меньше, чем количество контента, такого как статьи, изображения, новости и т.д. Для небольших веб-сайтов обычно достаточно зарезервировать 999 категорий или подкатегорий, поэтому для обеих колонок используется тип `INT` [27], [30].

Поле `Title` ограничено максимальной длиной в 255 символов, учитывая изменчивость длины названий статей. Использование переменной длины поля (тип `VARCHAR`, а не `CHAR`) позволяет избежать хранения лишней информации, фиксируя длину поля в зависимости от максимально возможного или достаточного значения для большинства записей [31].

«В таблице `Admins` хранятся учетные записи пользователей, необходимые для аутентификации. При вводе пользователем пароля в форму аутентификации он сравнивается с хэшированным паролем из базы данных, обеспечивая элементарную защиту от несанкционированного доступа к системе администрирования» [32]. В планах - добавить поле категории в базу данных, чтобы управлять доступом пользователей, таких как администраторы, редакторы и другие.

В базе данных `Photo_image` для оптимизации производительности используется подход, при котором изображения хранятся физически в папке `"Files"`, а в самой базе хранятся лишь пути к ним. Это позволяет эффективно кешировать изображения на HTTP-сервере. Важно учитывать, что в большинстве операционных систем длина пути ограничена 255 символами.

При обращении к основному файлу `index.php`, расположенному в корневой папке, происходит автоматическое подключение файла `head.inc` через строку `"include_once('head.inc');"`. Файл `head.inc` выполняет функцию сборки содержимого заголовка и включает в себя подключение файла настроек `conf.inc`, библиотеки классов `lib.inc`, а также определение необходимых шаблонов. Файл `conf.inc`, в свою очередь, определяет URL-

пути к основным компонентам системы, включая пути к библиотеке классов, шаблонам и системе управления.

«Кроме того, он определяет параметры подключения к базе данных, такие как:

- имя сервера;
- имя базы данных;
- пароль пользователя базы данных» [24].

Ниже на рисунке 9 представлена диаграмма классов.

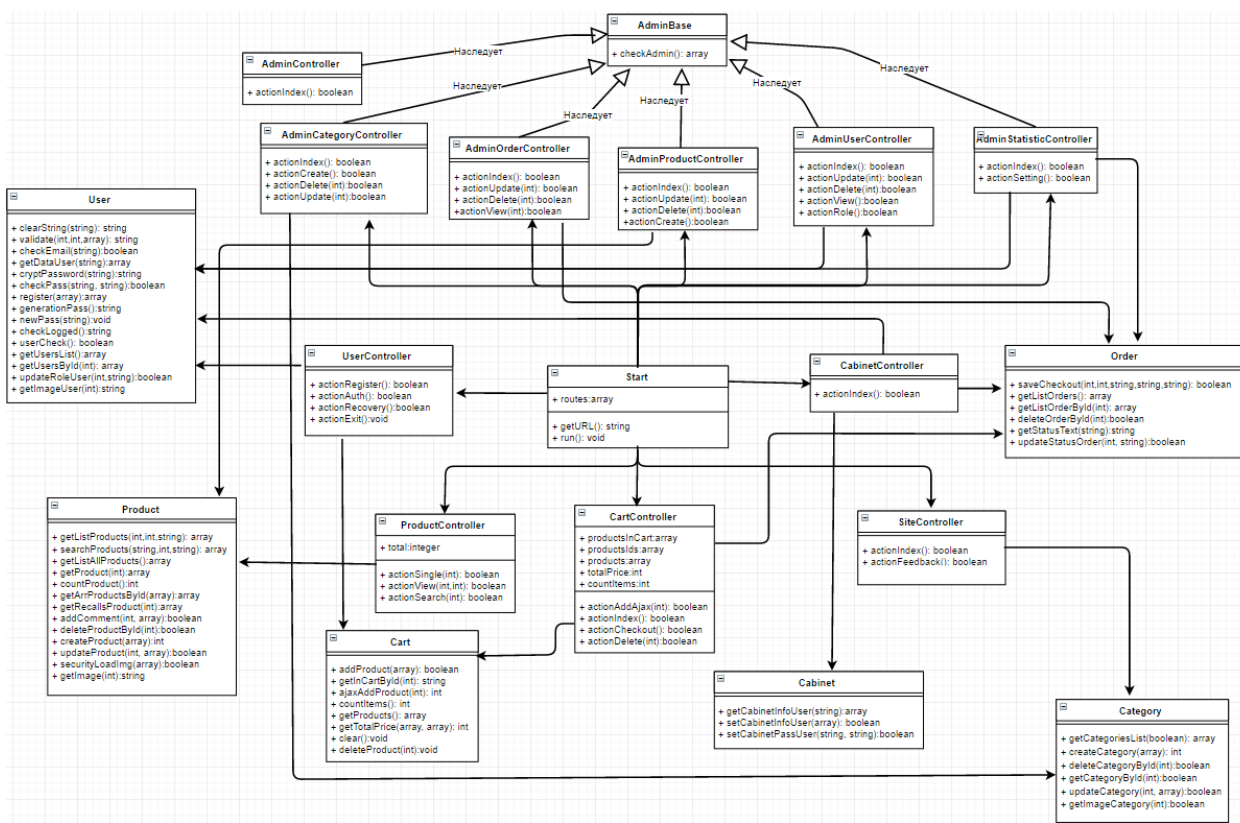


Рисунок 9 – Диаграмма классов

«Файл библиотеки классов содержит инструкции для включения файлов классов, предопределенные константы для всей библиотеки и необходимые функции для ее работы» [17].

3.2 Разработка модульной структуры приложения

«errorHandler.class - этот класс предназначен для обработки ошибок в приложении. В зависимости от типа и идентификационного номера ошибки, он генерирует соответствующее сообщение для вывода пользователю» [22].

Класс validator.class представляет собой утилиту для проверки данных на соответствие определенному формату, используя регулярные выражения. В данной CMS-системе поддерживаются три основных типа данных: новости, статьи и изображения.

«mysqlHandler.class - этот класс предназначен для взаимодействия с базой данных MySQL. Он обеспечивает управление параметрами подключения к базе данных, а также осуществляет операции выборки, сортировки, добавления, удаления пользователей, таблиц и записей» [23].

Класс mailHandler.class представляет широкие возможности для работы с электронной почтой, включая многоуровневую проверку введенных пользователем адресов электронной почты (в том числе проверку на их существование), добавление вложений к сообщениям, установку кодировки для писем и множество других полезных функций.

«templateEngine.class - это модуль для работы с шаблонами, который обеспечивает управление файлами шаблонов. Он определяет соответствие между типами данных и соответствующими шаблонами, а затем передает результаты работы templateProcessor.class» [18].

«templateProcessor.class - отвечает за обработку контента и вставку его в соответствующий шаблон, создавая тело HTML-документа. Шаблоны представляют собой небольшие фрагменты HTML с синтаксическими конструкциями, понятными шаблонизатору. Этот подход позволяет быстро изменять дизайн страниц, сокращая время в сравнении с ручной версткой» [23].

«Шаблоны играют роль расширения программного кода, автоматизируя процесс написания кода и обеспечивая структурное

разделение проекта на рабочие группы. Это особенно важно при увеличении объемов проекта и числа участников, а также при усложнении его архитектуры, не только на этапе разработки, но и в ходе последующего сопровождения» [19].

Давайте иллюстрируем эту ситуацию на конкретном примере. Представьте себе команду разработчиков, где одна часть занимается веб-дизайном, а другая – программированием. Веб-дизайнеры стремятся создать привлекательный и удобный интерфейс сайта, в то время как программисты фокусируются на функциональности и возможностях веб-приложения. Применение шаблонов значительно упрощает этот процесс, позволяя каждой группе сосредоточиться на своих задачах без лишних сложностей.

В мире PHP существует несколько подходов к созданию шаблонов:

- внедрение HTML в PHP-код;
- включение файлов в страницу.

Хотя первый метод проще в реализации, он также ограничивает вашу свободу действий. Основная проблема заключается в том, что PHP-код смешивается с HTML-компонентами, что влечет за собой ряд проблем, включая сложности при одновременном доступе к странице и ее изменениям, а также повышенный риск ошибок при просмотре и редактировании.

«В ряде ситуаций второй подход оказывается предпочтительнее первого. Однако, хотя схема "заголовок - основная часть - колонтитул" отлично подходит для небольших сайтов с простой структурой, с ростом масштабов и сложности проекта эти ограничения становятся более заметными» [20], [26]. Для решения этих проблем была разработана новая схема применения шаблонов, более сложная, но гораздо более гибкая. «Она предполагает разделение двух основных компонентов веб-приложения: дизайна и программирования. Это позволяет проводить параллельную разработку без постоянной координации на протяжении всего процесса» [21]. Кроме того, данный подход обеспечивает возможность внесения изменений в

один компонент без влияния на функциональность другого. В приложении А показан основной макет, разработанный в соответствии с этой идеей.

«{Заголовок}, {Дата новости}, {Краткое описание}, {Содержание}, заключенные в фигурные скобки ({}), трактуются как соответствующие переменные, которые заменяются соответствующими значениями. Дизайнер имеет свободу построения страницы, вставляя эти ключевые строки в нужные места документа. Однако перед началом работы программисты и дизайнеры должны согласовать имена переменных» [25].

«Каждый метод класса выполняет определенную задачу в обработке шаблона, разделяя процесс на четыре стадии: регистрацию файлов, переменных, обработку файлов и вывод обработанных файлов в браузере» [22].

3.3 Тестирование функциональности

Для проверки корректного выполнения программой функций, заложенных в техническом задании, были составлены тестовые наборы, показанные в таблицах следующих таблиц.

В таблице 7 были рассмотрены тестовые наборы для проверки функциональности авторизации

Таблица 7 – Тестовые наборы для проверки функциональности авторизации

№ Теста	Действие	Ожидаемый результат	Отметка о выполнении
1	Ввод данных для входа Логин: admin Пароль: admin (ввод корректных данных)	Успешная авторизация и переход на страницу администратора	проверено
2	Ввод данных для входа Логин: anonim Пароль: admin (ввод некорректных данных – неверный логин)	Появление сообщения о неправильно введенном логине или пароле	проверено
3	Ввод данных для входа Логин: admin Пароль: password (ввод некорректных данных – неверный пароль)	Появление сообщения о неправильно введенном логине или пароле	проверено

Таблица содержит данные, необходимые для тестирования процесса аутентификации пользователей в системе. В ней представлены сценарии входа в систему с разными учетными записями

В таблице 8 мы рассмотрели тестовые наборы для проверки администрирования каталога и товаров

Таблица 8 – Тестовые наборы для проверки администрирования каталога и товаров

№ Теста	Действие	Ожидаемый результат	Отметка о выполнении
1	Выбор "Раздел каталога"	Отображение меню редактирования категорий и товаров	проверено
2	Выбор "Добавить категорию"	Появление формы для ввода данных о новой категории	проверено
3	Заполнение формы и нажатие "Сохранить"	Создание новой категории	проверено
4	Выбор "Редактировать"	Открытие формы для изменения данных о категории	проверено
5	Выбор "Добавить товар"	Открытие формы для ввода данных о новом товаре	проверено
6	Заполнение формы и нажатие "Сохранить"	Добавление нового товара в выбранную категорию	проверено
7	Выбор "Удалить товар"	Удаление выбранного товара	проверено
8	Выбор названия товара	Отображение формы для редактирования данных	проверено
9	Выбор "Специальные предложения"	Перемещение товара в раздел "Специальные предложения"	проверено

Содержит данные, необходимые для тестирования процессов администрирования каталога и товаров в интернет-магазине. В нее входят данные о категориях товаров, описания товаров, цены, изображения, а также информация о наличии товаров на складе и другие связанные данные.

В таблице 9 мы рассмотрели тестовые наборы для проверки администрирования каталога и товаров

Таблица 9 – Тестовые наборы для проверки функциональности поиска

№ теста	Действие	Ожидаемый результат	Отметка о выполнении
1	Поиск товара по запросу	Отображение товаров, в названии или описании которых содержится указанный запрос	проверено
2	Поиск товара по запросу, который не существует	Отображение сообщения о том, что ничего не найдено	проверено

В таблице содержатся данные, необходимые для тестирования функциональности поиска в системе. В ней представлены различные запросы поиска, включая запросы с разными ключевыми словами, фильтрами и сортировкой, а также данные о ожидаемых результатах для каждого запроса.

В таблице 10 были рассмотрены тестовые наборы для проверки администрирования каталога и товаров.

Таблица 10 – Тестовые наборы для проверки функциональности закупки

№ теста	Действие	Ожидаемый результат	Отметка о выполнении
1	Избрание определенной категории:	Отображение продуктов из выбранной категории и предоставление описания категории	проверено
2	Выбор конкретного товара	Представление информации о выбранном товаре	проверено
3	Нажатие на кнопку "Добавить в корзину"	Отображение содержимого корзины	проверено
4	Установка количества товаров на 20 и нажатие кнопки "Обновить"	Умножение суммы заказа на 20	проверено

Продолжение таблицы 10

№ теста	Действие	Ожидаемый результат	Отметка о выполнении
5	Установка количества товаров на 0 или меньше и нажатие кнопки "Обновить"	Очистка корзины	проверено
6	Подтверждение заказа	Появление формы для ввода контактной информации	проверено
7	Заполнение всех обязательных полей	Добавление информации о заказе и покупателе в базу данных	проверено
8	Неполное заполнение обязательных полей	Появление сообщения о незаполненных полях	проверено

Таблица имеет данные, необходимые для тестирования функциональности процесса закупки товаров в интернет-магазине. В нее входят данные о заказах, клиентах, адресах доставки, способах оплаты и другие связанные данные.

Представленные в таблицах тесты позволили проверить программу и убедиться в корректном выполнении функционала.

Проверим разработанную программу и выполним оформление заказа, на рисунке 10 показан некорректный ввод данных при регистрации.

Регистрация

Пример: Иванов Фамилия может состоять только из букв

Иванов ' OR `id`=1"

Пример: Сергей Имя может состоять только из букв

Сергей ' OR `id`=1"

Пример: Андреевич

Андреевич

Пример: ivanov@yandex.ru Домашний e-mail уже используется

art@ya.tu

Пример: г. Уфа, ул. Роз, д.4, кв. 1 Адрес должен быть коротким

кв. 3

Пример: +7 (903) 459-13-94

+7 (980) 285-98-39

Придумайте пароль Пароль должен не соответствовать никаким правилам

llm_ЯЯЯ_8E0#96z

- убедитесь, что не включена кнопка Caps-Lock;
- пароль должен быть не менее 8 символов в длину;
- допустимые символы в пароле: [a-zA-Z], [0-9], [#_!?];
- доG и doG являются разными паролями;

Рисунок 10 – Ввод некорректных данных

Далее введем корректные данные, чтобы успешно завершить регистрацию и попасть в личный кабинет пользователя, рисунок 11, 12.

Пример: Сергей

Сергей

Пример: Андреевич

Андреевич

Пример: ivanov@yandex.ru

ivanov@ya.ru

Пример: г. Уфа, ул. Роз, д.4, кв. 1

г. Муром, ул. Свердлова, д.38, кв. 17

Пример: +7 (903) 459-13-94

+7 (904) 857-63-23

Придумайте пароль

i8pF9_YX3

- убедитесь, что не включена кнопка Caps-Lock;
- пароль должен быть не менее 8 символов в длину;
- допустимые символы в пароле: [a-zA-Z], [0-9], [#_!?];
- доG и doG являются разными паролями;

Надежный

Введите защитный код с картинки:

 skr8

Рисунок 11 – Ввод корректных данных

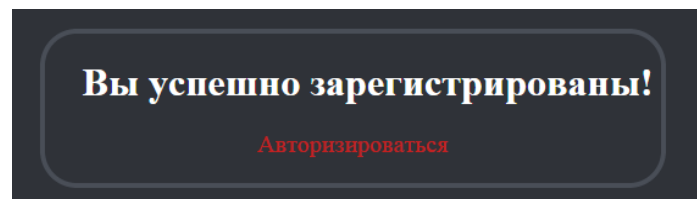


Рисунок 12 – Успешная регистрация

После того как мы авторизовались по средствам ввода корректных данных мы попадаем в личный кабинет, который изображен на рисунке 13.

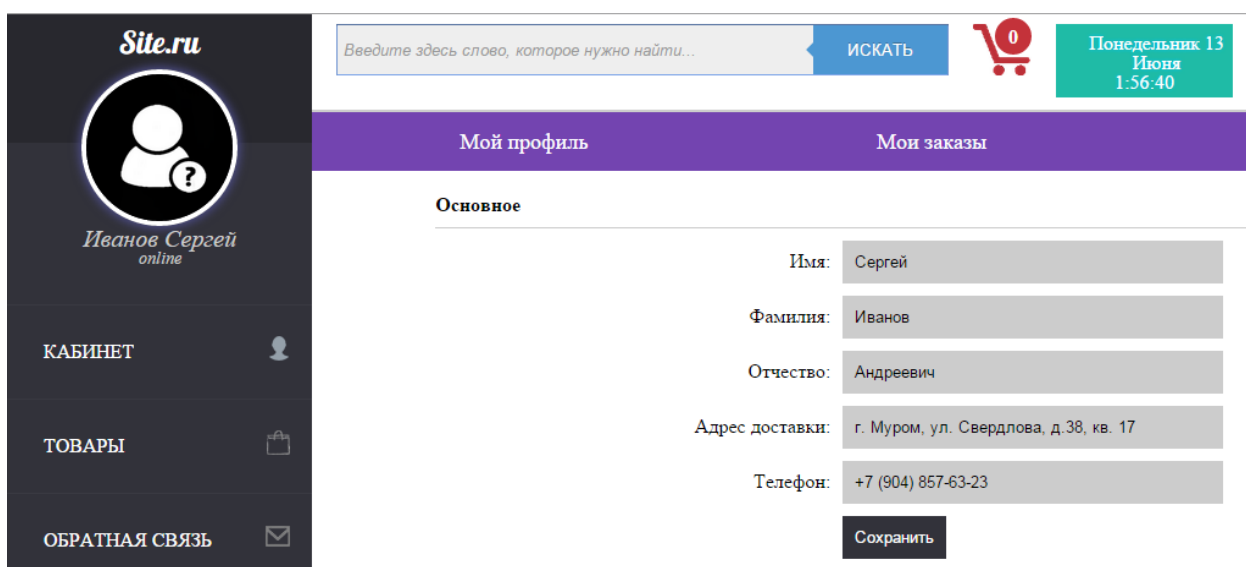


Рисунок 13 – Кабинет пользователя

После успешной авторизации мы переходим в корзину для того чтобы оформить заказ, на рисунке 14 изображена корзина пользователя.

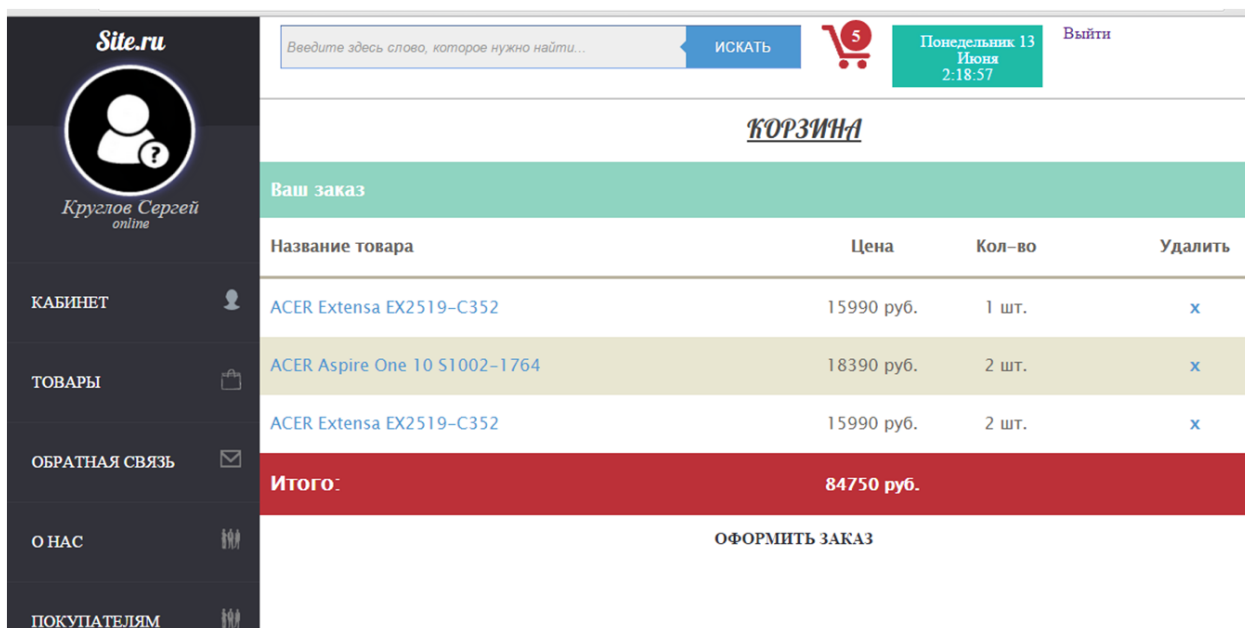


Рисунок 14 – Корзина пользователя

Далее нажимаем на кнопку оформить заказ и переходим на страницу для оплаты товаров на рисунке 15, которые были у нас в корзине.

Введи... ИСКАТЬ

Понедельник 13
Июня
2:21:05

Выйти

ОФОРМЛЕНИЕ ЗАКАЗА

Выбрано товаров: **5**, на сумму **84750** руб.

Имя:

Сергей

Телефон:

+7 (904) 857-63-23

Адрес доставки:

г. Муром, ул. Свердлова, д.38, кв. 17

Комментарий:

Оформить

Рисунок 15 – Оформление заказа

После того как мы оплатили покупки нас возвращает в пустую корзину (рисунок 16).

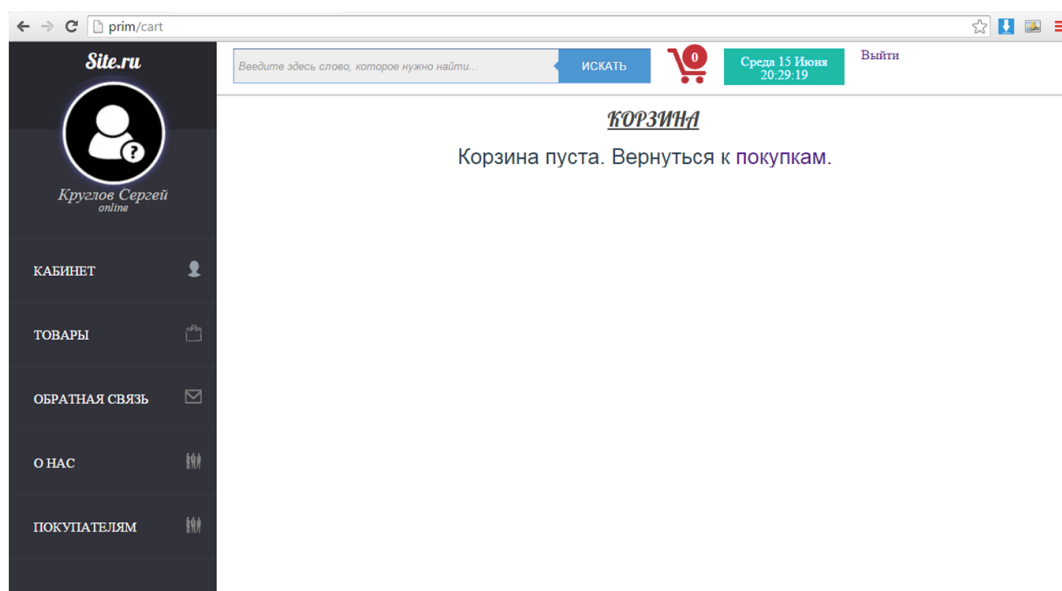


Рисунок 16 – Пустая корзина

Вывод по третьему разделу

Раздел, посвященный разработке и администрированию системы управления контентом интернет-магазина, включает в себя три ключевых аспекта: проектирование базы данных и сервера разметок, разработку модульной структуры приложения и тестирование функциональности готового приложения.

Проектирование базы данных и сервера разметок представляет собой фундаментальный этап разработки, определяющий эффективность и надежность системы. На основе проведенного анализа была спроектирована база данных, обеспечивающая эффективное хранение и обработку информации о товарах, заказах и пользователях, а также разработан сервер разметок для динамической генерации контента.

Разработка модульной структуры приложения позволяет обеспечить гибкость и масштабируемость системы, упрощает ее поддержку и

расширение. Правильно спроектированная структура позволяет легко вносить изменения и добавлять новый функционал, не нарушая работу других компонентов.

Тестирование функциональности готового приложения необходимо для обеспечения его качества и корректной работы. Проведение различных видов тестирования позволяет выявить и устранить ошибки и недочеты, обеспечивая высокую степень надежности и удовлетворение потребностей пользователей.

Общий вывод по данному разделу заключается в том, что правильное проектирование, разработка и тестирование ключевых компонентов системы управления контентом интернет-магазина являются важными шагами для обеспечения ее эффективной работы, удобства использования и конкурентоспособности на рынке.

Заключение

В ходе выполнения данной бакалаврской работы была проведена разработка и администрирование системы управления контентом для интернет-магазина. Процесс разработки включал в себя анализ и исследование различных аспектов, начиная от изучения видов сайтов и основных понятий CMS, и заканчивая тестированием функциональности готового приложения.

Первые разделы работы были посвящены теоретическим основам, таким как классификация веб-приложений, аналитический обзор решений и требований к системам управления контентом. Этот этап позволил определить ключевые аспекты и критерии выбора архитектуры и программных средств. Мы исследовали и анализировали различные программные инструменты, необходимые для реализации нашего проекта. Этот процесс включал в себя оценку функциональности, производительности. Выбор архитектуры системы управления контентом был сделан на основе анализа требований проекта и с учетом его масштабируемости и гибкости.

Последующие разделы работы были посвящены прямой разработке системы, включая проектирование базы данных, разработку модульной структуры приложения и ее тестирование.

В процессе нашей работы мы достигли следующих результатов:

- проанализировали требования к функциональности системы управления контентом для интернет-магазина;
- спроектировали и разработали систему управления контентом, адаптированную под специфику интернет-магазина.

Процесс разработки и администрирования системы управления контентом интернет-магазина позволил погрузиться во все аспекты создания и поддержки подобных систем. Полученный опыт и знания являются ценным вкладом и будут полезны в будущей профессиональной деятельности.

Список используемой литературы и используемых источников

1. Алексунин В.А., Родигина В.В. Электронная коммерция и маркетинг в Интернет. – Учебное пособие. – М.: «Дашков и К0», 2012.
2. Беляев В. Зарубежные информационные и аналитические ресурсы в сети Интернет // Рынок ценных бумаг. – 2009. – №7.
3. Блох, Д. Java. Эффективное программирование / Д. Блох. — М.: Лори, 2014. - 310 с.
4. Бурдинский А. Перспективы развития рынка финансовых Интернет-услуг/ <http://www.osp.ru/ecom/2010/03/024.htm>.
5. Бурмистров А. Золотая лихорадка русского Интернета, или Реально ли получить инвестиции в сетевые проекты/ <http://www.osp.ru/ecom/2010/034.htm>.
6. Вартамян М. SET – не догма! // Мир Интернет. – 2009. – №2.-С. 64 – 68.
7. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес. — Питер, 2016. - 366 с.
8. Гершман А.Б., Киселева А.В. (2014). Исследование и реализация алгоритмов интеллектуального анализа сетевого трафика. Материалы 22-й Международной конференции IEEE по компьютерному анализу, стр. 295-301.
9. Дейтел, П. Android для программистов. Создаем приложения / П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. — М.: Питер, 2013. - 560 с.
10. Завалеев В. Курс ЦИТ «Internet-технологии в проектах с пластиковыми карточками», Центр Информационных Технологий, 2012.
11. Завалеев В. Что такое электронная коммерция?, Центр Информационных Технологий, 2011.
12. Киселев Ю.Н. Электронная коммерция: практическое руководство. – СПб: ООО «ДиаСофтЮП», 2011.

13. Крылов И. Теория и практика рекламы в России, «Центр», 1996
14. Л. Веллинг. Разработка WEB-приложений с помощью PHP и MySQL. 2-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2009.- 800с
15. Малевский П., Чернышев А., Дубина А. Системы электронной коммерции для операторов связи // ПаРаДис. 2009. №1
16. Малышева Н. Виртуальные покупки Электронная версия еженедельника «Аргументы и факты» (<http://www.aif.ru>)
17. Маршавин Р.А., Ляпунов С.И. Глобальный бизнес и информационные технологии, Учебное пособие. М.: 2012 г.
18. Нельзина О.Г. Актуальность формализованного сравнения отечественных систем электронной коммерции // RELGA, №2 (147), /www.relga.ru
19. Нельзина О.Г. Системы электронного документооборота на предприятии: проблемы внедрения и использования // RELGA, №11 (133), 04.06.11 /www.relga.ru
20. Нельзина О.Г. Характеристики, проблемы и перспективы развития систем электронной коммерции в современной России // RELGA, №23 (145), 25.12.11 /www.relga.ru
21. Орлов, Л. Как создать электронный магазин в Интернет. – М.: Альянс–пресс, 2011.
22. Орлов, Л. Как создать электронный магазин в Интернет. Все об организации Интернет–магазина. – М.: Новый издательский дом, 2011.
23. Павлов А.С. Сравнительный анализ потребительского качества программных продуктов для электронной торговли // Диссертация на соиск. степени канд. экон. наук. – Ростов-на-Дону, 2012.
24. Ухин Д. Оценка перспектив электронной коммерции (в мире) // [<http://www.m4u.webprovider.com/articles/ecperspecives.htm>]
25. Хубаев Г.Н. Маркетинг информационных продуктов и услуг. – Ростов – на – Дону. Изд-во РГЭУ «РИНХ», 2012.

26. Хубаев Г.Н. Сравнение сложных программных систем по критерию функциональной полноты // Программные продукты и системы (SOFTWARE and SYSTEMS). – №2/2011.
27. Griffiths, D. Head First Android Development / D. Griffiths — O'Reilly Media, Inc., 2015. - 698 с.
28. Hellman, E. Android Programming: Pushing the Limits / E. Hellman — Wiley, 2014. – 416 с.
29. Hodson, R. Android Programming Succinctly / R. Hodson. — Syncfusion Inc., 2014. – 113 с.
30. Pilgrim, P. Digital Java EE 7 Web Application Development / P. Pilgrim — Packt Publishing, 2015. – 486 с.
31. Smyth, N. Android 4.2 App Development Essentials / N. Smyth. — Techotopia, 2013. – 478 с.
32. Stark, J. Building Android Apps with HTML, CSS, and JavaScript, 2nd Edition / J. Stark, B. Jepson. — O'Reilly Media, Inc., 2012. – 176 с.