Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.А. Виденин
подпись инициалы, фамилия
« 23 » июня 2016г.

ДИПЛОМНЫЙ ПРОЕКТ

230102.65 Автоматизированные системы обработки информации и управления
код и наименование специальности

Программный комплекс для автоматизированного наполнения Web-контента
тема

Пояснительная записка

Руководитель _____ Л.С. Троценко
подпись, дата инициалы, фамилия

Выпускник _____ П.С. Ахметшин
подпись, дата инициалы, фамилия

Нормоконтролер _____ Л.С. Троценко
подпись, дата инициалы, фамилия

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программный комплекс для автоматизированного наполнения Web-контента» содержит 56 страниц текстового документа, 1 приложение, 22 иллюстрации, 7 таблиц, 5 источников литературы.

ТЕЛЕКОММУНИКАЦИОННЫЕ СЕТИ, ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ, СЕТЕВОЙ РЕСУРС, АНАЛИЗ, ШАБЛОН, СИСТЕМА УПРАВЛЕНИЯ СОДЕРЖИМЫМ

Объектом исследования являются электронные ресурсы в телекоммуникационных сетях и глобальной сети Интернет.

Предметом исследования является система управления содержимым web-ресурса.

Цели и задачи исследования:

Разработка автоматизированной системы управления содержимым, позволяющей создавать и поддерживать информационный сетевой ресурс.

Задачи:

- изучение типов web-ресурсов и способов представления информации;
- изучение технологий, используемых для создания web-ресурсов;
- проектирование и создание автоматизированной системы управления содержимым.

Актуальность работы:

В результате проделанной работы были спроектированы шаблон представления информации и автоматизированная система, его реализующая. Разработанная система позволяет создавать web-ресурсы, а так же имеет весь необходимый функционал для их настройки и наполнения.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 5
1 Характеристика объекта разработки 7
1.1 Основные понятия web-ресурса 7
1.2 Системы управления содержимым 9
1.2.1 Назначение 9
1.2.2 Классификация 10
1.3 Сравнительный анализ существующих систем 12
1.3.1 1С-Битрикс 12
1.3.2 WordPress 13
1.3.3 Drupal 14
1.4 Описание предполагаемой модели функционирования 15
2 Проектирование системы управления содержимым 17
2.1 Архитектура 17
2.1.1 Модель клиент-сервер 17
2.1.2 Протокол HTTP 18
2.1.3 Серверная часть 19
2.1.4 Клиентская часть 20
2.1.5 Структура хранения данных 21
2.2 Выбор средств реализации 25
2.2.1 Web-сервер Apache 25
2.2.2 Язык разметки HTML 26
2.2.3 Язык программирования PHP 27
2.2.4 СУБД MySQL 28
2.2.5 Платформа 29
3 Описание разработанной системы управления содержимым 31
3.1 Структура системы 31
3.2 Описание формируемого web-ресурса 32
3.2.1 Файловая структура 32
3.2.2 Структура шаблона 32
3.3 Описание модулей системы 34
3.3.1 Модуль учетных записей 34
3.3.2 Модуль мастера шаблонов 37
3.3.3 Модуль контрольной панели 42

ЗАКЛЮЧЕНИЕ 55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 56
ПРИЛОЖЕНИЕ А Техническое задание 57

ВВЕДЕНИЕ

Телекоммуникационные сети прочно вошли в жизнь человека со второй половины XX века. Изначально эти сети, такие как радио и, позже, телевидение, носили односторонний характер. То есть их пользователь мог лишь получать предлагаемые материалы без возможности публиковать свои. Организация же источника передачи данных требовала больших материальных затрат.

Однако с развитием техники и технологий возможности телекоммуникационных сетей существенно расширились. В частности, развитие вычислительной техники позволило объединять в сети множество участников, где каждый мог быть как источником информации, так и ее потребителем.

Одной из крупнейших телекоммуникационных сетей современности является глобальная сеть Интернет. Изначально проектировавшаяся как военная система коммуникаций, сегодня сеть Интернет прочно проникла в жизнь всего общества, объединив страны и континенты, за что получила второе название «глобальная паутина» (англ. «World Wide Web»). И, если изначально сеть была чем-то вроде песочницы для технически подкованных энтузиастов и ученых, то сегодня ее ресурсы доступны любому желающему независимо от уровня подготовки.

Общедоступность Интернета в свою очередь породила целую сферу услуг, направленную на размещение информации в сети. В настоящее время в ней размещено огромное количество информационных ресурсов различной направленности. Для того чтобы развивать ресурс, наполнять его содержимым и управлять им в настоящее время используются специальные системы управления содержимым, без которых поддержка любого сетевого ресурса становится весьма неудобной. Эти автоматизированные информационные системы берут на себя всю специфику обслуживания сетевого ресурса, избавляя пользователя от необходимости разбираться во всех тонкостях организации размещения информации в сети.

Целью данной работы является изучение и разработка программных средств, позволяющих организовать информационный сетевой ресурс в сети Интернет и схожих по архитектуре локальных сетях.

Основные задачи работы:

- проведение анализа существующих web-ресурсов;
- изучение программных и аппаратных средств создания web-ресурсов;
- изучение автоматизированных средств управления содержимым web-ресурсов;
- проектирование и реализация автоматизированной системы, позволяющей обеспечивать размещение информации в сети Интернет.

1 Характеристика объекта разработки

1.1 Основные понятия web-ресурса

Web-ресурс (Интернет-ресурс, Интернет-сайт) – это информационная система, использующая web-технологии на уровне представления и передачи данных, предназначенная для оказания публичных информационных услуг в сети Интернет. Подразумевается, что web-ресурс имеет постоянный адрес (англ. Uniform Resource Locator, URL) во всемирной сети Интернет, позволяющий идентифицировать ресурс, а также осуществлять доступ к нему.

Для организации web-ресурса необходим специальный серверный программно-аппаратный комплекс, подключенный к сети Интернет. Он может быть организован пользователем самостоятельно или же взят в аренду у хостинг-провайдера – компании, предоставляющей услуги размещения и поддержки сайтов.

Интернет-сайты в общем случае состоят из набора специально отформатированных документов, зачастую называемых страницами. Каждая страница сайта так же имеет собственный URL адрес, по которому можно обратиться к конкретной странице сайта.

Основное содержимое сайта, без учета оформления, навигации и прочего, принято называть контентом (англ. Content, «содержимое»). Именно контент ресурса представляет основной интерес для пользователей.

В настоящее время существует огромное количество web-ресурсов самой различной направленности. Если попытаться классифицировать существующие ресурсы, то целесообразно делать это по типу решаемых задач и объему контента.

Таким образом, можно выделить следующие основные категории ресурсов:

- представительства;
- информационные ресурсы;
- сервисы.

Интернет-представительства, как следует из названия, служат для представления информации о фирме, продукте или конкретном человеке в сети. Типичные представители данной категории сайтов – это сайты-визитки человека или фирмы, домашние страницы программных продуктов, просто небольшие информативные сайты. Сайты-представительства как правило обладают минимальным набором страниц (нередко обходятся одной единственной) и функций, доступных пользователю.

Информационные ресурсы – это сайты, содержащие большой объем информации по какой-либо определенной тематике. Эти ресурсы представляют собой большие web-справочники с четким разбиением информации на категории и темы, а так же предоставляющие функции поиска. Нередко такие ресурсы предоставляют возможность своим пользователям общаться в рамках тематики сайта.

Web-сервисы предназначены для выполнения каких-либо задач или предоставления услуг в рамках сети Интернет. Типичные и старейшие примеры подобных сервисов, знакомые всем, это поисковые сер висы и электронная почта. Однако, помимо них, существуют так же доски объявлений, форумы, облачные хранилища данных, видеохостинги, социальные сети и многие другие сервисы. Регистрация в сервисах, связанных с размещением и публикацией данных почти всегда осуществляется с приемом пользовательского соглашения, регулирующего права и обязанности сторон (владельца сервиса и пользователя).

В общем случае любой web-ресурс представляет из себя набор страниц. С точки зрения пользователя, наиболее удобным было бы, если бы все страницы имели одинаковое оформление, расположение структурных элементов и навигацию. Соблюдение этого простого правила избавит от необходимости полностью заново изучать каждую страницу ресурса при посещении.

Поэтому при составлении страниц как правило используется единый шаблон оформления. Под шаблоном в данном случае понимается некий бланк

страницы, заполняемый содержимым в определенных местах и содержащий статичные элементы оформления, присутствующие на всех страницах сайта.

Типовые элементы шаблона страницы:

- верхний информационный блок;
- нижний информационный блок;
- меню навигации;
- область содержимого.

При этом от страницы к странице меняется только содержание страницы в то время как остальные элементы остаются неизменны.

1.2 Системы управления содержимым

1.2.1 Назначение

Для реализации применения шаблонов при представлении информации используются специальные автоматизированные системы управления содержимым.

Система управления содержимым (англ. Content Management System, CMS) – это информационная система, используемая для обеспечения и организации процесса создания, редактирования и управления содержимым сайта.

Конечно, можно сверстать несколько страниц и без использования каких-либо систем. Например, с использованием текстового редактора Microsoft Word и его функции сохранения документа в виде web-страницы. Однако в этом случае даже с небольшим сайтом-визиткой из пяти страниц могут возникнуть трудности, если потребуется, например, изменить название одного из пунктов навигации. В этом случае пришлось бы вручную повторно править информацию на каждой из существующих страниц и снова повторно размещать их в Интернете, что очень трудозатратно.

Ресурсы же с большим количеством контента, с ролевыми моделями и web-сервисы однозначно не могут обойтись без системы управления контентом.

Таким образом, система управления содержимым является неотъемлемой частью большинства web-ресурсов, существенно упрощающих администрирование, наполнение и поддержку неспециалистами в области программирования и web-технологий

1.2.2 Классификация

Единой и принятой классификации систем управления содержимым не существует, однако их можно условно рассмотреть по некоторым критериям, у каждого из которых есть как плюсы, так и минусы.

Прежде всего, по уровню доступности системы бывают:

- платные;
- бесплатные.

За платные системы приходится платить реальные деньги, порой немалые. Взамен пользователь получает непосредственную техническую поддержку разработчика системы. К тому же разработчик платных систем заинтересован в постоянном развитии своей системы и оперативном устранении ошибок, потому что от этого зависит его непосредственный доход с продаж.

Бесплатные системы не требуют покупки лицензии, однако на техническую поддержку можно будет рассчитывать только в том случае, если ее разработчик идет на контакт, либо если система уже успела стать популярной и собрать достаточно большое сообщество.

При этом следует понимать, что зачастую бесплатная система ничуть не хуже платной. Общедоступность бесплатной системы привлекает к ней новых пользователей и разработчиков, которые стремятся доработать и сделать ее лучше. Доля сайтов в российском сегменте Интернета, использующих бесплатные системы управления содержимым, составляет более 80% на 2015 год.

По классам решаемых задач системы делятся:

- специализированные;
- универсальные.

Специализированные системы оптимизированы под решение какой-либо конкретной задачи. Например, система организации форумов или Интернет-магазин. Создать сайт-визитку, используя такую систему, будет проблематично, однако именно с решением своей конкретной задачи такая система отлично справится.

Универсальные системы пытаются предоставить пользователю сразу весь спектр функционала. Зачастую, это достигается за счет модульной архитектуры системы, когда недостающий функционал добавляется в системы в виде отдельных расширений.

По своей архитектуре системы могут быть:

- монолитные;
- модульные.

Монолитные системы представляют собой цельный, законченный продукт. Внести какие-либо изменения в него можно только доработав основные алгоритмы системы. Недостатком таких систем является ограниченность функционала первоначальной задумкой автора.

Модульные системы могут дополняться расширениями, зачастую написанными сторонними разработчиками. Это позволяет постоянно дорабатывать систему, предоставляя пользователям недостающий функционал. Недостатком такого подхода является отсутствие контроля со стороны основного разработчика. Если устанавливаемое расширение содержит ошибку или уязвимость, это может привести к некорректной работе всей системы или же ее взлому злоумышленником.

1.3 Сравнительный анализ существующих систем

1.3.1 1С-Битрикс

Среди платных систем бесспорным лидером является 1С-Битрикс. На 2015 год эта система занимала более 60% российского рынка систем управления содержимым, в то время как ее ближайший конкурент не смог набрать и 10%.

1С-Битрикс выпускается в нескольких редакциях, функционал которых пропорционален цене. Версии бывают от простой редакции для организации несложного сайта-визитки до редакции уровня web-портала с множеством сервисов.

Данная платная система, как и прочие подобные ей, обладают перед бесплатными системами существенным преимуществом в виде технической поддержки со стороны непосредственно разработчика. Если у пользователя возникнет вопрос, то он всегда может обратиться в службу поддержки компании-разработчика и получить консультацию. Помимо этого, данная система обеспечивает высокую интеграцию с прочими продуктами фирмы «1С».

Минусом же, не дающим подобной системе получить большую часть рынка web-ресурсов, является непосредственно ее платность, пропорциональная требуемому функционалу. К тому же в рамках одной купленной лицензии существует возможность создать не больше двух независимых сайтов. Если потребуется больше количество, то необходимо будет приобретать отдельные лицензии.

К примеру, самый дешевый пакет «Старт» стоит 1990 рублей. Он позволяет создать один сайт. Чтобы на базе этого пакета создать второй независимый сайт, потребуется докупать лицензии на каждый дополнительный за 1620 рублей, либо купить неограниченную лицензию за 48600 рублей. При этом чем выше цена базового пакета, тем выше стоимость дополнительных лицензий.

Поскольку в настоящее время существует большое количество различных бесплатных систем, дублирующих функции платных, то пользователь логично выбирает не переплачивать.

1.3.2 WordPress

WordPress на сегодняшний день является одной из самых популярных систем управления содержимым. По данным на 2015 год, среди бесплатных систем, WordPress выбрали почти 39% пользователей.

Основные преимущества данной системы следующие:

- бесплатное использование;
- большой набор шаблонов, расширений и дополнений;
- интуитивно-понятный интерфейс.

Бесплатность системы и ее простота в использовании привлекли к ней большое количество пользователей и независимых разработчиков, собрав вокруг нее достаточно обширное сообщество. Оно постоянно дорабатывает систему, создает расширения и шаблоны для нее. Большинство из них так же, как и сама система, находятся в свободном доступе.

Однако WordPress имеет и свои недостатки:

- весомая нагрузка на сервер, как следствие уменьшение скорости работы системы;
- ограниченный набор базовых функций;
- большое количество некачественных дополнений и шаблонов;
- принцип «одна система – один сайт».

В своем базовом исполнении WordPress прежде всего система для организации блогов. Поэтому, если управляемый ею web-ресурс блогом не является, то возникает необходимость подбирать соответствующие темы оформления и модули расширений. Они в свою очередь могут содержать ошибки и уязвимости, либо быть просто некачественными.

Существенным недостатком системы является ее требовательность к ресурсам. Поскольку механизм кэширования страниц изначально не предусмотрен, то большое количество пользователей ресурса могут создать существенную нагрузку на сервер.

Еще одним существенным недостатком является невозможность создания двух и более совершенно независимых ресурсов в рамках одной установленной системы, что не позволяет использовать ее в качестве корпоративного средства публикации информации.

1.3.3 Drupal

Система Drupal обладает всеми возможностями, необходимыми для управления содержимым сайта. Базовая комплектация содержит все основные инструменты, которые могут потребоваться для полноценного функционирования сайта.

Как и большинство современных универсальных систем управления, Drupal имеет модульную архитектуру и может расширяться в плане функционала по мере необходимости.

Однако система Drupal предназначена, в первую очередь, для решения сложных задач. Из этого вытекают не только преимущества, но и недостатки. Drupal требует наличие специальных технических знаний, и новичку не так просто начать пользоваться этой системой, в отличие, например, от WordPress.

Из всего вышесказанного выделяются следующие минусы системы:

- весомая нагрузка на сервер, как следствие уменьшение скорости работы системы;
- высокая сложность для неподготовленного пользователя;
- нецелесообразность использования в простых проектах.

Систему Drupal, не рекомендуется использовать для простых сайтов, так как в противном случае такая система только будет только создавать излишние

сложности. Так же стоит заметить, что подобная система потребляет немало ресурсов, что не может не сказаться на скорости загрузки страниц.

Таким образом, система Drupal является хорошим решением для создания сложной и функциональной системы, однако для большинства пользователей, которым нужно просто разместить свой контент в сети, она является не самым удачным выбором.

1.4 Описание предполагаемой модели функционирования

Проектируемая система представляет собой систему управления содержимым, позволяющую пользователям создавать и поддерживать web-ресурс в сети Интернет или в корпоративной сети.

Система обеспечивает раздельную работу нескольких пользователей. Для этих целей предусмотрена функция регистрации учетных записей. При этом каждой учетной записи пользователя соответствует уникальное имя (логин) и пароль, используемые в дальнейшем для авторизации в системе. Каждой учетной записи ставится в соответствие отдельный web-ресурс со своим собственным URL на основе логина.

Система обеспечивает хранение и извлечение пользовательских данных и данных самой системы. Эти данные необходимы для прохождения процедуры авторизации в системе и управлении содержимым web-ресурсов. Для этих целей используется централизованная база данных, содержащая всю необходимую для работы системы информацию.

Система предоставляет пользователям необходимый и достаточный набор средств для автоматизированного создания web-ресурса, его настройки, создания и редактирования содержимого ресурса. К данному набору функций системы относится:

- добавление, удаление, редактирование страниц ресурса;
- добавление, удаление, редактирование меню ресурса;
- изменение шаблона оформления ресурса;

- загрузка, удаление файлов;
- публикация контента.

Система обладает интуитивно-понятным графическим интерфейсом с общепринятыми описаниями и логикой функционирования.

Система обеспечивает автоматическое размещение контента в сети Интернет или корпоративной сети с возможностью доступа к нему другим пользователям сети.

Для снижения нагрузки на сервер и исключения возможных перебоев в работе, система использует модель кеширования данных. Она заключается в том, что в момент обращения к web-ресурсу, запрашиваемая страница уже заранее сформирована и готова к отправке.

Таким образом, можно выделить следующий комплекс основных задач, автоматизацию которых обеспечивает проектируемая система:

- регистрация и авторизация пользователей;
- систематизированное хранение данных;
- взаимодействие посредством графического интерфейса;
- формирование и публикация web-страниц ресурса.

2 Проектирование системы управления содержимым

2.1 Архитектура

2.1.1 Модель клиент-сервер

Поскольку система ориентирована на неограниченное количество пользователей, работающих с хранимой информацией, то целесообразно использовать модель клиент-сервер.

Данная модель подразумевает, что существует аппаратно-программный комплекс, предоставляющий некоторый набор сервисов. Для доступа к сервисам, клиенту необходимо осуществить подключение к серверу с использованием определенного протокола, поддерживаемого как серверным приложением, так и клиентским.

Основное приложение располагается на сервере, поэтому каждый клиент всегда работает с его актуальной версией. Это избавляет от необходимости следить за новыми версиями, так как все нововведения и обновления сразу становятся доступны всем пользователям после обновления приложения на сервере. Так же поскольку интерфейс пользователя предоставляется серверным приложением, то исчезает необходимость постоянно обновлять клиентское приложение, если до обновления серверного приложения все работало успешно.

Все необходимая информация так же располагается на сервере в базе данных и клиент избавлен от необходимости самостоятельно хранить и размещать ее. Пользователь всегда работает с самыми последними данными.

На рисунке 1 представлена общая схема работы проектируемой системы с клиент-серверной архитектурой.

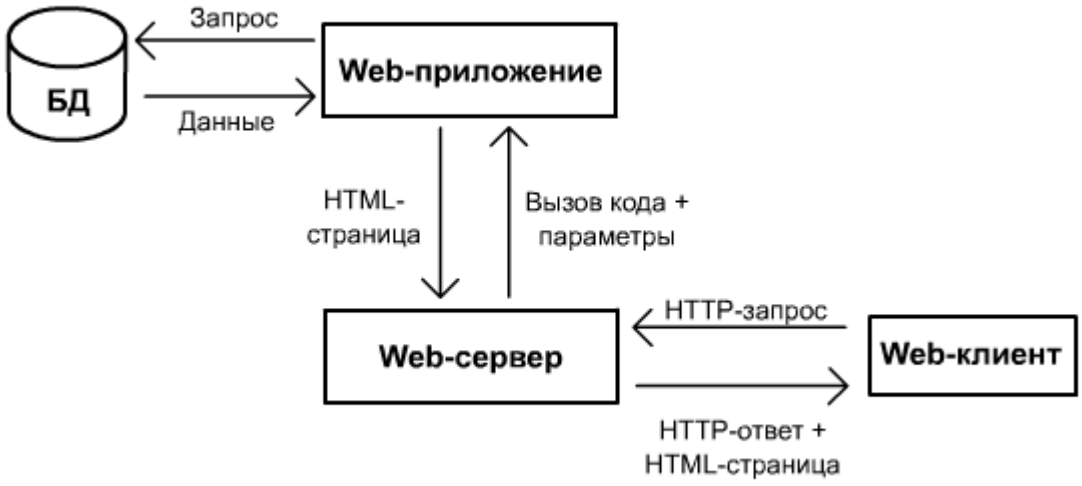

Рисунок 1 – Архитектура проектируемой системы

Общее взаимодействие компонентов осуществляется следующим образом. Клиент посылает web-серверу запрос по протоколу HTTP. Сервер принимает запрос, обрабатывает его и вызывает соответствующий код web-приложения, передавая ему необходимые параметры, извлеченные из запроса пользователя. Приложение анализирует полученные данные и в соответствии с ними вызывает соответствующие подпрограммы, которые осуществляют запрос к базе данных. После обработки запроса база данных возвращает приложению запрошенные данные и приложение формирует HTML-страницу. После чего итоговая страница передается web-сервером клиенту.

2.1.2 Протокол HTTP

Для реализации модели клиент-сервер и организации взаимодействия пользователя с проектируемой системой по сети Интернет или в корпоративной ЛВС используется протокол HTTP.

HTTP (англ. HyperText Transfer Protocol – «протокол передачи гипертекста») – протокол прикладного уровня передачи данных (изначально – в виде гипертекстовых документов в формате HTML, в настоящий момент используется

для передачи произвольных данных). Работа по нему осуществляется в режиме «запрос-ответ». То есть клиент инициирует соединение с сервером и посылает запрос. Сервер в свою очередь получает запрос, обрабатывает его и возвращает клиенту результат.

Обычно, передача данных по протоколу HTTP осуществляется через TCP/IP-соединения. При этом web-сервер обычно использует TCP-порт 80 (и, если порт не указан явно, то обычно клиент по умолчанию использует именно 80-й порт для открываемых соединений), хотя можно использовать и любой другой.

Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления информации по различным параметрам, таким как формат, кодировка, язык. Благодаря этому клиент и сервер могут обмениваться двоичными данными, хотя сам протокол является текстовым.

2.1.3 Серверная часть

Для обеспечения доступа клиентов к проектируемой системе по сети, осуществления взаимодействия и предоставления информации используется web-сервер.

Web-сервер – сервер, принимающий HTTP-запросы от клиентов и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом или другими данными. Web-сервером называют как программное обеспечение, выполняющее функции web-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает.

В общем случае основные функции web-сервера следующие:

- организация ожидания входящего соединения;
- передача данных средствами протокола HTTP;
- вызов серверных приложений;
- ведение журнала обращений пользователей к ресурсам;
- аутентификация и авторизация пользователей;

- поддержка динамически генерируемых страниц (списки файлов, страницы ошибок и т.п.).

Для обработки запросов, формирования нужных HTML-документов и работы с данными используется web-приложение, разработанное на языке программирования высокого уровня. Данное приложение тесно взаимодействует с web-сервером, принимая от него пользовательские запросы и параметры и формируя в ответ HTML-документ, который web-сервер отсылает обратно клиенту в ответ на его запрос.

Web-приложение выполняет следующие задачи:

- логическая обработка HTTP-запроса, выделение параметров;
- контроль учетных записей и разделение доступа;
- извлечение, добавление, удаление, обновление данных в базе данных;
- формирование HTML-страницы в соответствии с запросом.

2.1.4 Клиентская часть

В качестве клиентского приложения, реализующего взаимодействие с проектируемой системой посредством протокола HTTP, используется браузер. Браузер (от англ. browser, «обозреватель») – программное обеспечение для просмотра web-страниц, содержания web-документов, взаимодействия с web-приложениями. Помимо поддержки приема и передачи данных по протоколу HTTP, современные браузеры также используются для непосредственного просмотра содержания файлов многих графических форматов, воспроизведения мультимедиа, различных текстовых форматов и других задач, связанных с работой в сети Интернет.

В настоящее время подавляющее большинство браузеров относятся к свободно распространяемому бесплатному программному обеспечению. Многие распространенные операционные системы в своем составе уже имеют браузеры как одну из самых необходимых прикладных программ.

2.1.5 Структура хранения данных

В качестве основного способа хранения данных используется база данных, состоящая из 5 таблиц.

- таблица «Учетные записи»;
- таблица «Страницы»;
- таблица «Шаблоны»;
- таблица «Блоки»;
- таблица «Меню»;
- таблица «Файлы».

Таблица «Учетные записи» предназначена для хранения данных пользовательских учетных записей. Таблица содержит логин, используемый для авторизации в системе, а так же как часть имени web-ресурса. Так же таблица содержит хешированный, а не исходный, пароль. Это нужно для повышения безопасности, потому что даже в случае утечки данных, исходный пароль пользователя раскрыт не будет.

Полная структура таблицы «Учетные записи» и описание ее полей представлены в таблице 1.

Таблица 1 – Структура таблицы «Учетные записи»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
login	строка	Логин пользователя
passhash	строка	Пароль пользователя
status	число	Флаг состояния наличия сайта
site_title	строка	Название сайта в системе

Таблица «Страницы» предназначена для хранения страниц сайтов. Страницы – это основное содержимое сайта. Каждая страница имеет свой заголовок и текст. Так же каждой странице автоматически присваивается порядковый номер, который одновременно служит именем HTML-страницы. Страница с номером 1 считается главной (отображаемой по умолчанию).

Структура таблицы «Страницы», описание ее полей, и их назначение представлены в таблице 2.

Таблица 2 – Структура таблицы «Страницы»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
owner_login	строка	Логин владельца страницы
title	строка	Заголовок страницы
text	текст	Содержимое страницы
inner_id	число	Порядковый номер страницы
last_update	число	Время последнего обновления

Таблица «Шаблоны» содержит в себе данные шаблона оформления каждого сайта в системе, его цветовую палитру, структуру и размеры элементов, составляющих шаблон.

Структура таблицы «Шаблоны», описание ее полей, и их назначение представлены в таблице 3.

Таблица 3 – Структура таблицы «Шаблоны»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
owner_login	строка	Логин владельца страницы
tpl_blocks	число	Флаг наличия блоков
tpl_menu	число	Флаг положения меню
clr_text	строка	Цвет текста
clr_link	строка	Цвет ссылок
clr_block_bg	строка	Цвет фона элементов
clr_block_border	строка	Цвет границ элементов
clr_bg	строка	Цвет фона сайта
size_font	число	Размер шрифта
size_menu	число	Ширина меню
size_block_top	число	Высота верхнего блока
size_block_bottom	число	Высота нижнего блока
tpl_source	текст	Код сформированного шаблона

Таблица «Блоки» содержит в себе такие структурные элементы как верхний и нижний блоки. Эти блоки являются частью шаблона сайта и отображаются на каждой без исключения странице сайта.

Структура таблицы «Блоки», описание ее полей, и их назначение представлены в таблице 4.

Таблица 4 – Структура таблицы «Блоки»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
owner_login	строка	Логин владельца страницы
text	текст	Содержимое страницы
flags	число	Флаг, определяющий тип блока

Таблица «Меню» содержит в себе данные пунктов меню. Меню – это основное средство навигации по ресурсу. Каждый пункт меню имеет свой отображаемый на сайте заголовок и гипертекстовую ссылку. Так же каждый пункт имеет свой порядковый номер, который определяет порядок вывода пунктов меню на сайте.

Структура таблицы «Меню», описание ее полей, и их назначение представлены в таблице 5.

Таблица 5 – Структура таблицы «Блоки»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
owner_login	строка	Логин владельца страницы
title	строка	Название пункта
link	строка	Адрес страницы
inner_id	число	Внутренний порядковый номер

Таблица «Файлы» содержит в себе информацию о загруженных пользователем файлах.

Структура таблицы «Файлы», описание ее полей и их назначение представлены в таблице 6.

Таблица 6 – Структура таблицы «Файлы»

Название поля	Тип	Назначение
id	число	Первичный ключ таблицы
owner_login	строка	Логин владельца страницы
filename	строка	Имя загруженного файла
filelink	строка	Ссылка на файл
inner_id	число	Внутренний порядковый номер

Так видно, все таблицы содержат поле «id», которое выступает в роли первичного ключа. Так же все таблицы (кроме таблицы «Учетные записи») содержат поле «owner_login», которое является внешним ключом, указывающим на соответствующую запись в таблице «Учетные записи» и являющимся признаком, отделяющим данные одного пользователя от данных другого в рамках одной таблицы.

2.2 Выбор средств реализации

2.2.1 Web-сервер Apache

Согласно спроектированной архитектуре, проектируемая система является серверным web-приложением. То есть для ее выполнения необходимо прежде всего серверное программное обеспечение, обеспечивающее работу с системой посредством сети Интернет.

Таким серверным программным обеспечением выступает популярный веб-сервер Apache. Разработан он был в 1995 году, а начиная с 1996 и до настоящего времени Apache является самым популярным веб-сервером в Интернете.

Apache имеет систему модульных расширений, что позволяет существенно расширить функционал сервера. Модули разрабатываются сторонними разработчиками. В настоящее время созданные модули, реализующие поддержку

языков программирования, таких как PHP, Perl, Python и других. Это дает возможность размещать на сервере не только набор статичных веб-страниц, но и программные модули, которые будут исполнены на стороне сервера при обращении к ним.

Использование Apache позволяет проектируемой системе:

- работать в качестве сервера с возможностью доступа через Интернет;
- выполнять программные модули системы на стороне сервера.

2.2.2 Язык разметки HTML

Для выдачи информации в удобном для пользователя виде система в своей работе использует специальный язык разметки HTML (от англ. HyperText Markup Language – «язык гипертекстовой разметки»). Он широко используется в сети Интернет для оформления сайтов.

Принцип разметки документа с использованием языка HTML заключается в том, что составитель документа при помощи семантических элементов языка, называемых «тегами», составляет несложный, но лаконично оформленный документ.

Язык разметки HTML позволяет:

- определять шрифт, размер, цвет и кодировку для текста;
- вставлять в документ мультимедиа данные, такие как графические, видео и аудио файлы;
- использовать гиперссылки как для перехода на другие документы, так и для навигации в рамках одного документа;
- представлять данные в табличном виде и задавать размеры и положение для каждого элемента документа.

Для интерпретации и корректного отображения HTML-документа в его отформатированном виде используется специальное программное обеспечение, называемое браузером (англ. browser – «обозреватель»).

В настоящее время язык обрел много новых мультимедиа возможностей и структурных элементов, направленных на еще большее упрощение верстки и реализацию типовых задач при оформлении веб-ресурсов.

Используя возможности языка разметки HTML, проектируемая система может решить две задачи:

- предоставление пользователю графического интерфейса для удобной работы с системой;
- создание выходных HTML-документов, оформленных в соответствии с выбранным шаблоном.

2.2.3 Язык программирования PHP

В качестве средства для создания веб-приложения, обеспечивающего работу связанную работу всей разработанной системы, был выбран сценарный язык общего назначения PHP.

PHP, называемый так же препроцессором гипертекста, является одним популярных средств, применяемых для создания серверной части динамических веб-сайтов, к которым, по сути, и относится проектируемая система. При помощи данного языка были созданы, к примеру, такие популярные веб-ресурсы, как социальная сеть Facebook и сетевая энциклопедия Wikipedia.

Поскольку это сценарный язык, то программный код, написанный на нем, выполняется непосредственно специальной программой – интерпретатором. В настоящее время, подавляющее большинство компаний, предоставляющих услуги веб-хостинга, имеют поддержку этого языка и следят за актуальными его обновлениями. Поэтому запуск системы на практически любом хостинге не вызывает никаких проблем.

Популярность PHP обусловлена большим и постоянно пополняемым набором средств разработки веб-приложений, среди которых:

- поддержка большого количества популярных СУДБ (MySQL, Oracle, Microsoft SQL Server);

- удобное взаимодействие с веб-сервером (примем и формирование заголовков страниц, получение и изменение параметров);
- работа с сессиями, что позволяет легко организовать многопользовательскую систему с разграничением прав;
- работа с файлами (запись, чтение, загрузка средствами HTTP, формирование архивов);
- работа со строками, как простая (поиск, замена, слияние), так и с использованием сложных регулярных выражений.

Помимо этого, PHP весьма лоялен к разработчику и дает ему некоторые послабления, среди которых необязательность предварительного объявления переменных и отсутствие необходимости жесткой типизации данных. Поэтому работа с PHP не требует каких-либо громоздких средств разработки.

Таким образом, язык программирования PHP является удачным решением для создания проектируемой системы, поскольку предоставляет возможности для реализации всех необходимых функций:

- поддержка СУБД для удобного хранения и извлечения данных;
- работа со строками и веб-сервером для генерации демонстрируемых пользователю веб-страниц;
- организация сессий, обеспечивающая одновременную работу нескольких пользователей с системой.

2.2.4 СУБД MySQL

Для выполнения функций хранения и извлечения данных была выбрана популярная СУБД MySQL. Так же, как и язык PHP, данная СУБД предоставляется на большом количестве хостингов.

В плане архитектуры, MySQL обычно выступает в роли сервера, к которому непосредственно подключаются клиенты под своими учетными записями удаленно или локально. При этом существует возможность задать каждой учетной записи права на различные операции с данными.

Работа с данными в MySQL осуществляется при помощи структурированного языка запросов SQL. Этот язык предназначен для описания, изменения и извлечения данных, хранимых в базах данных.

Основным объектом хранения является таблица с записями, поэтому почти все основные SQL-запросы – это операции над таблицами, такие как добавление, изменение, просмотр и удаление. Поскольку запросы подразумевают групповые операции, то SQL использует систему задания условий, однозначно определяющую целевой набор записей по ряду каких-либо признаков. К примеру, чтобы изменить какую-либо конкретную запись, в условии указывается первичный ключ, уникальный для каждой записи в изменяемой таблице. Помимо этого можно использовать вычисления, сравнения и ряд других вспомогательных функций.

Основным преимуществом работы с SQL является то, что она не зависит от конкретной СУБД. Пользователь просто отправляет запрос, а СУБД уже сама реализует его обработку. Это дает относительную свободу выбора хостинга и инструментов отладки, поскольку все современные базы данных корректно реализуют подавляющее большинство основных функций языка SQL.

Таким образом, СУБД MySQL предоставляет все необходимые функции для работы с хранимыми данными проектируемой системы, а именно:

- функции хранения, добавления, изменения и удаления данных;
- поддержка языка запросов SQL;
- интеграция с выбранным языком программирования.

2.2.5 Платформа

Все выбранные для реализации программные продукты являются кроссплатформенными, то есть имеют реализации под различные операционные системы, включая наиболее популярные, такие как:

- Linux;
- Mac OS;

- Microsoft Windows.

Таким образом, условия для корректной работы проектируемой системы могут быть созданы на различных платформах в зависимости от необходимости, предпочтений пользователя или политики компании, предоставляющей услуги хостинга.

3 Описание разработанной системы управления содержимым

3.1 Структура системы

Разработанная система представляет собой совокупность трех модулей:

- модуль учетных записей;
- модуль мастера шаблонов;
- модуль контрольной панели.

В зависимости от состояния учетной записи, для пользователя вызывается тот или иной модуль. В общем случае система проверяет следующие параметры для каждой пользовательской сессии:

- прохождение авторизации;
- наличие созданного ресурса.

В таблице 7 знаком «+» помечена доступность того или иного модуля в зависимости от параметров сессии.

Таблица 7 – Доступность модулей системы

	Модуль учетных записей	Модуль мастера шаблонов	Модуль контрольной панели
Пользователь не авторизирован	+	-	-
Авторизирован, нет ресурса	+	+	-
Авторизирован, есть ресурс	+	-	+

3.2 Описание формируемого web-ресурса

3.2.1 Файловая структура

При создании ресурса на сервере создается дочерняя директория относительно корневой, имя которой соответствует логину пользователя. Эта директория является хранилищем всех документов и файлов пользователя. Чтобы обратиться к конкретному ресурсу, нужно использовать URL вида «http://имя_сервера/логин_пользователя».

Все страницы ресурса имеют имя вида «N.html», где N – порядковый номер документа, автоматически присвоенный в момент создания страницы. Документ с порядковым номером 1 является документом по умолчанию для ресурса, его содержимое так же дублируется в файл «index.html».

Таким образом, все ресурсы системы можно представить как набор независимых каталогов, содержащих HTML документы, относящиеся к конкретному ресурсу.

3.2.2 Структура шаблона

Для представления информации система использует шаблоны. Наполнением шаблона служит пользовательский контент, взятый из базы данных.

Шаблон состоит из трех частей:

- меню;
- блоки информации;
- основная часть.

Меню – это специальный блок, помогающий посетителю ресурса с навигацией. Он содержит список из ссылок и выводится на каждой странице ресурса. В системе используется три варианта расположения меню:

- меню слева;
- меню сверху;

- без меню.

Блоки информации – аналогично блоку меню, так же присутствуют на каждой странице ресурса, однако могут содержать произвольную информацию. В системе используется четыре варианта расположения блоков информации:

- верхний и нижний блоки;
- только верхний блок;
- только нижний блок;
- без блоков.

Основная часть – это основное содержание страницы. Она всегда присутствует в шаблоне и содержит определенную для этой страницы информацию.

Общая структурная схема шаблона приведена на рисунке 2.

Рисунок 2 – Структура шаблона страницы ресурса

В случае есть для меню и блоков информации отображение не предусмотрено, то страница ресурса будет состоять из одной основной части, содержащей непосредственно контент.

3.3 Описание модулей системы

3.3.1 Модуль учетных записей

Модуль учетных записей предназначен для осуществления регистрации и авторизации пользователей. Данный модуль выполняет следующие функции:

- регистрация пользователя;
- авторизация пользователя;
- вывод списка ресурсов.

Регистрация пользователя подразумевает создание персональной учетной записи для дальнейшей работы с системой. Для регистрации используются имя учетной записи (логин) и пароль. Логин впоследствии становится частью адреса URL создаваемого пользователем ресурса. Поэтому в качестве логина могут использоваться сочетания букв латинского алфавита от «A» до «Z» и символ «_». Длина имени учетной записи может составлять от 3 до 32 символов. В качестве пароля учетной записи может использоваться любое сочетание букв, цифр и символов любой длины. После успешного прохождения регистрации, учетная запись пользователя сохраняется в базе данных системы в таблице «Учетные записи».

Внешний вид формы регистрации представлен на рисунке 3. Форма регистрации состоит из поля ввода будущего логина, поля ввода пароля, поля повторного ввода пароля и кнопки «Регистрация».

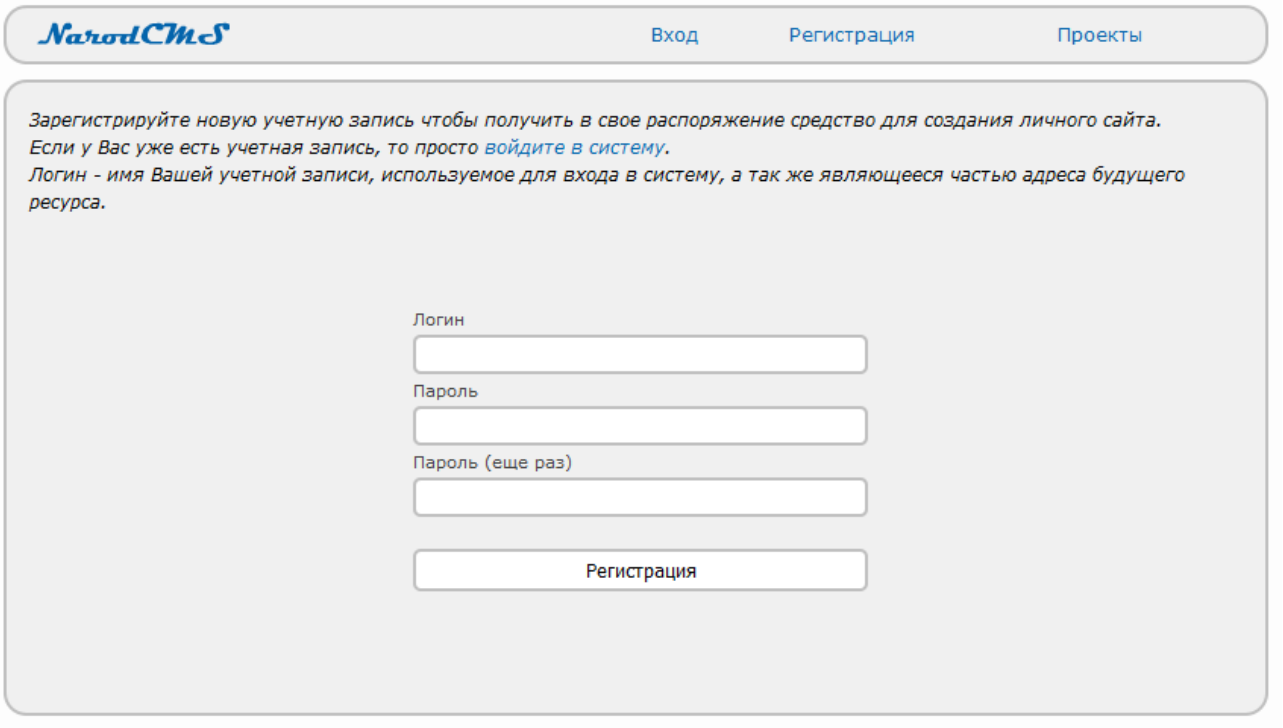

Рисунок 3 – Регистрация учетной записи

В случае если введенные данные неверны (логин уже используется, логин содержит недопустимые символы, введенные пароли не совпадают), пользователю выдается всплывающее сообщение, сообщающее об ошибке. Если пользователь ввел данные корректно, ему выдается сообщение об успешно завершенной регистрации.

Авторизация пользователя подразумевает сопоставление введенных пользователем логина и пароля с данными, имеющимися в базе данных системы. В случае успешного прохождения авторизации, система создает для пользователя рабочую сессию и предоставляет доступ к функционалу.

Внешний вид формы авторизации представлен на рисунке 4. Форма содержит поле ввода логина, поле ввода пароля и кнопку «Вход».

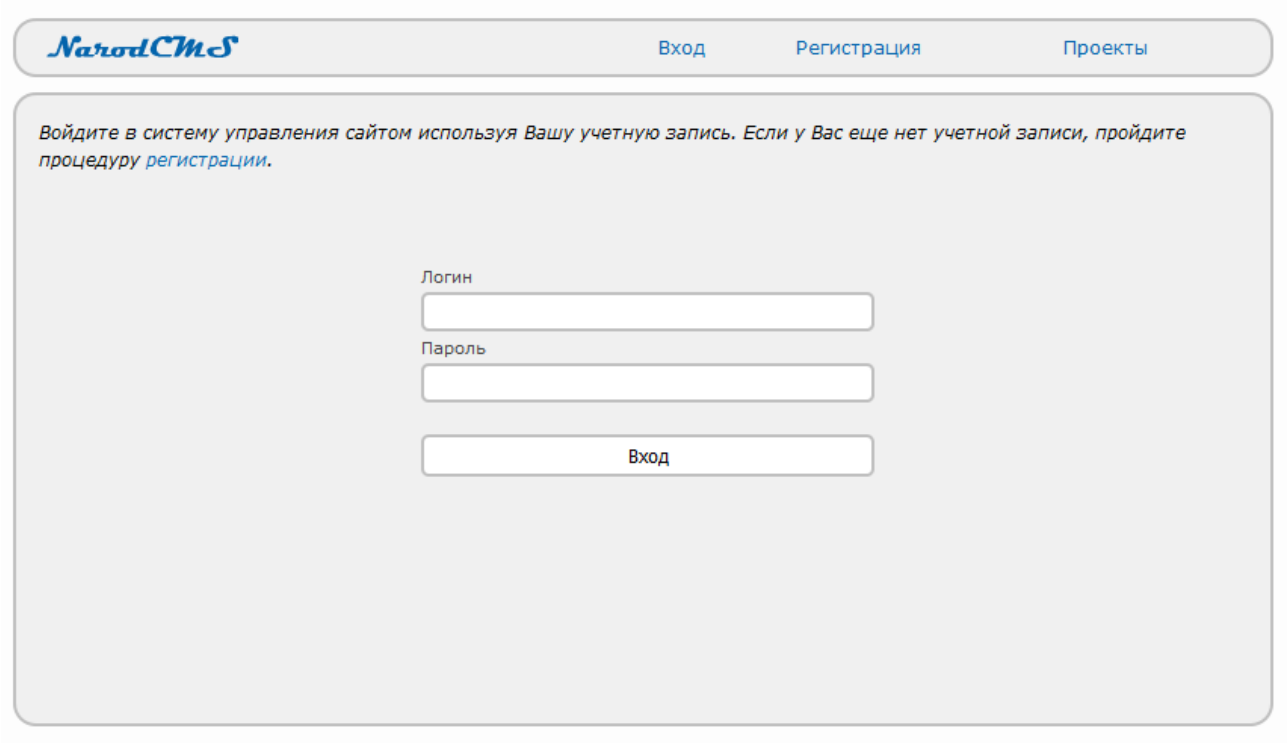

Рисунок 4 – Авторизация пользователя

В случае некорректно введенных данных при попытке входа пользователю выдается сообщение «Неверный логин либо пароль» и ему предлагается ввести данные повторно. В случае успешной авторизации вызывается либо модуль мастера шаблонов либо модуль контрольной панели, в зависимости от того, создан ли уже для этой учетной записи ресурс или еще нет.

Вывод списка ресурсов служит для составления списка имеющихся в системе ресурсов пользователей и вывод их списка в виде группы ссылок с возможностью перехода.

Внешний вид раздела, содержащего список созданных ресурсов, представлен на рисунке 5.

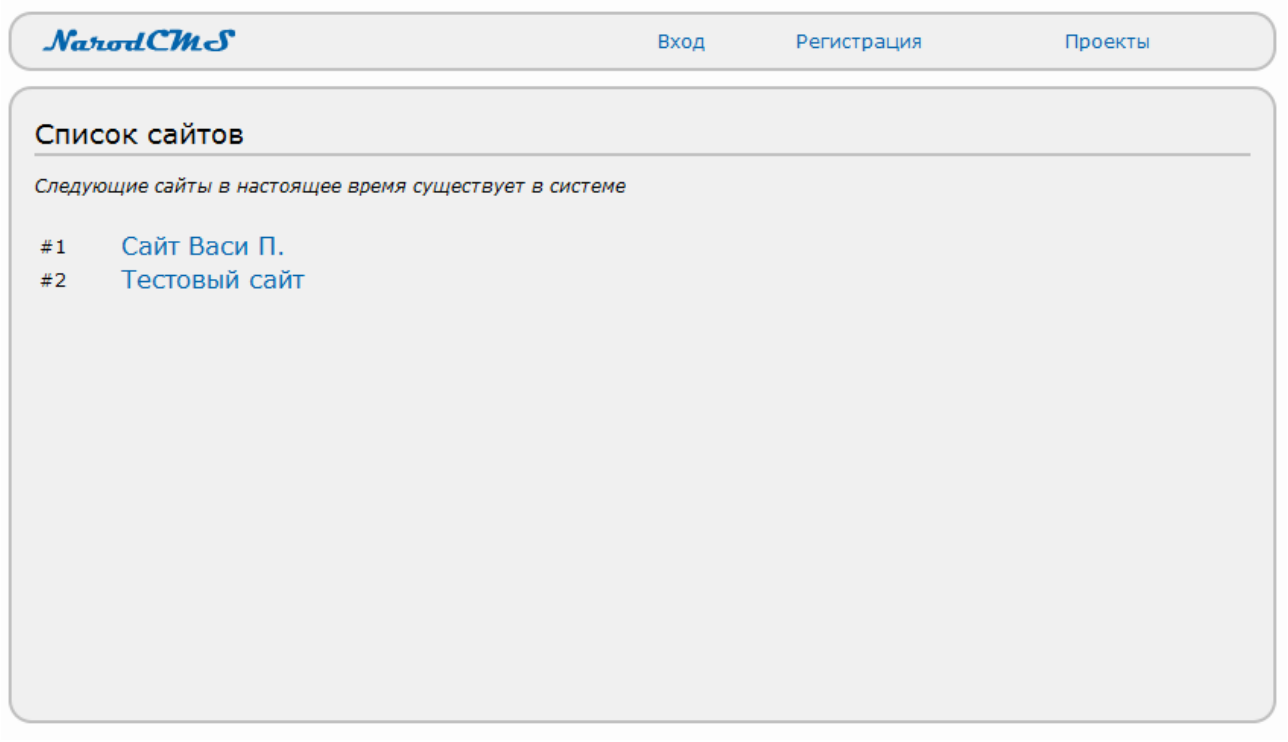

Рисунок 5 – Список созданных ресурсов

3.3.2 Модуль мастера шаблонов

Модуль мастера шаблонов предназначен для создания шаблона представления информации в диалоговом режиме. Данный модуль подразумевает создание шаблона ресурса в пять шагов:

- шаг 1 «Название и страницы»;
- шаг 2 «Блоки»;
- шаг 3 «Меню»;
- шаг 4 «Цветовая схема»;
- шаг 5 «Подтверждение».

Шаг 1 предназначен для задания названия будущего ресурса и выбора количества создаваемых страниц. Название ресурса будет использоваться как на страницах самого ресурса, так и при выводе списка всех существующих ресурсов. Количество создаваемых страниц задает количество автоматически создаваемых страниц ресурса и автоматическое размещение ссылок на них в пунктах меню.

Внешний вид мастера шаблонов на шаге 1 представлен на рисунке 6. Пользователю предлагается заполнить поле «Название сайта», выбрать количество страниц из выпадающего списка и нажать кнопку «Далее» для перехода к следующему шагу.

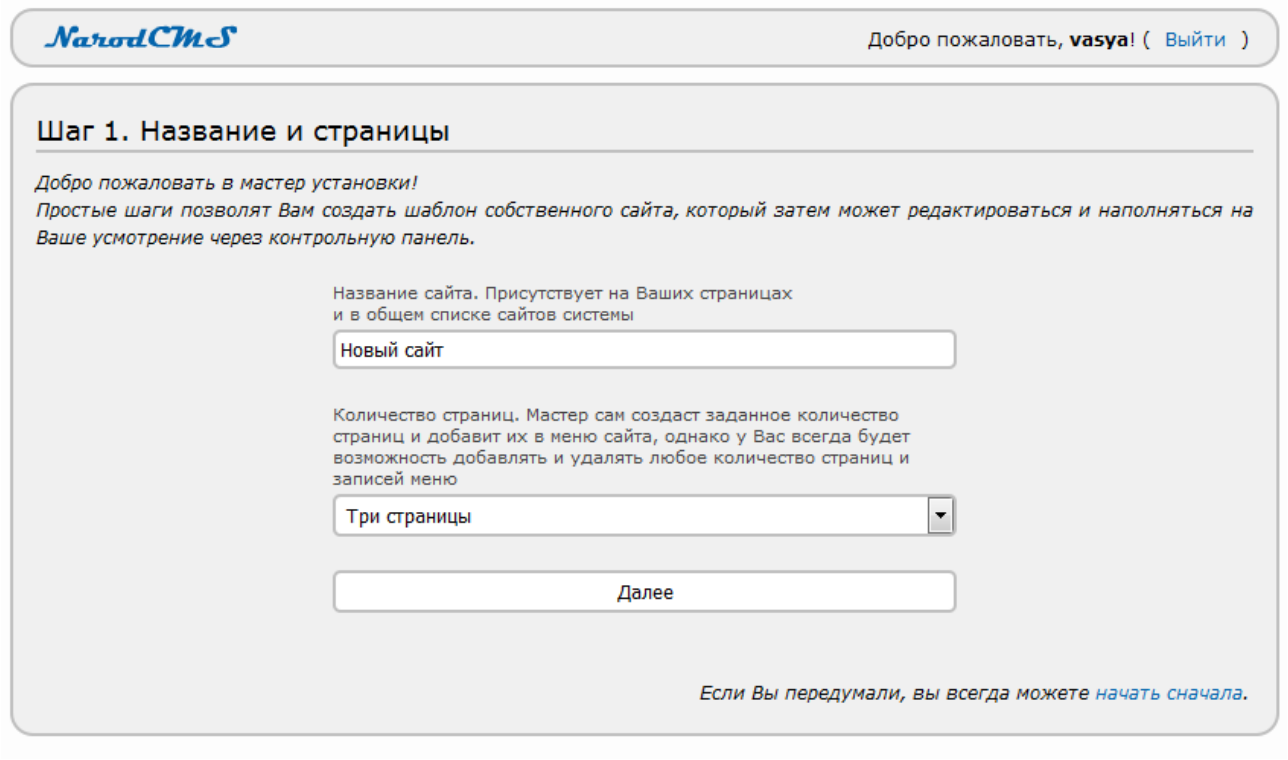

Рисунок 6 – Мастер шаблонов, шаг 1

Шаг 2 предназначен для выбора количества статичных блоков информации и их расположения. На выбор пользователю предлагается четыре варианта:

- верхний и нижний блоки;
- только верхний;
- только нижний;
- без блоков.

Внешний вид мастера шаблонов на шаге 2 представлен на рисунке 7. Пользователю предлагается выбрать один из четырех вариантов, нажав под соответствующей схемой, демонстрирующей расположение блоков, надпись «Выбрать». На схемах расположение блоков обозначается синим цветом.

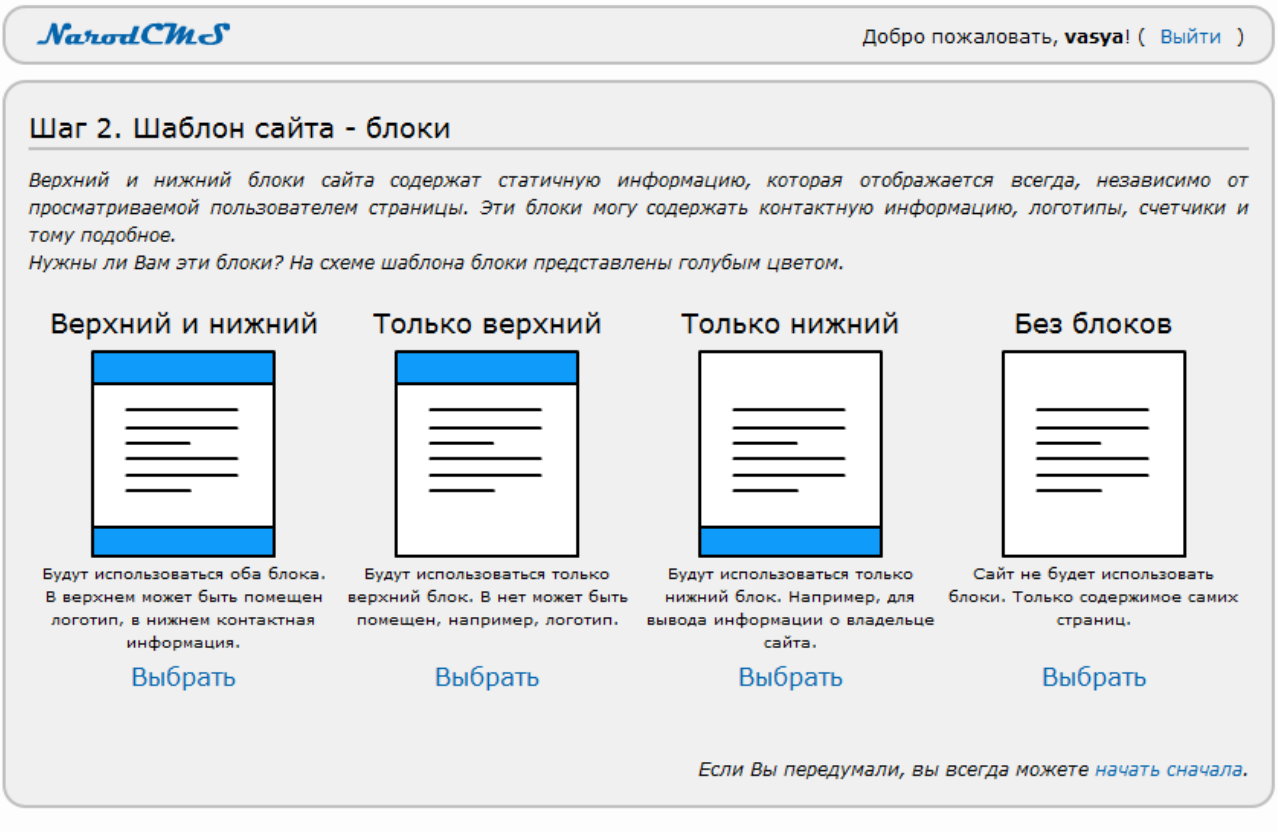

Рисунок 7 – Мастер шаблонов, шаг 2

Шаг 3 предназначен для настройки отображения меню сайта. На выбор доступны три варианта:

- меню слева;
- меню сверху;
- без меню.

Внешний вид мастера шаблонов на шаге 3 представлен на рисунке 8. Пользователю предлагается выбрать один из трех вариантов, нажав под соответствующей схемой, демонстрирующей расположение блоков, надпись «Выбрать». При этом отображаемые схемы учитывают выбор пользователя на предыдущем шаге. На представленных схемах будущее расположение меню обозначается пурпурным цветом, расположение блоков синим цветом.

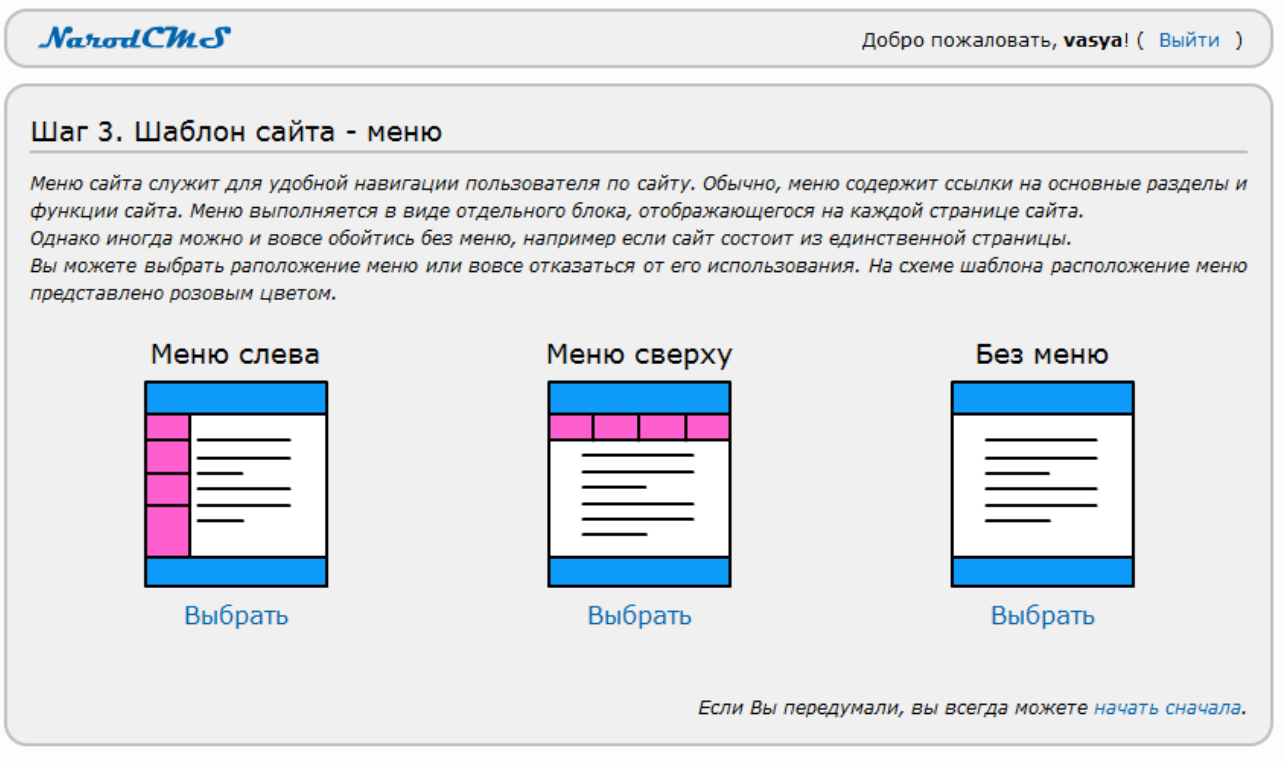

Рисунок 8 – Мастер шаблонов, шаг 3

Шаг 4 позволяет пользователю выбрать цветовое оформление ресурса. На выбор предлагается три цветовые схемы:

- «Стандартная»;
- «Пастель»;
- «Готика».

Внешний вид мастера шаблонов на шаге 4 представлен на рисунке 9. Пользователю предлагается выбрать один из трех вариантов, нажав под соответствующей схемой, демонстрирующей будущее цветовое оформление шаблонов, надпись «Выбрать».

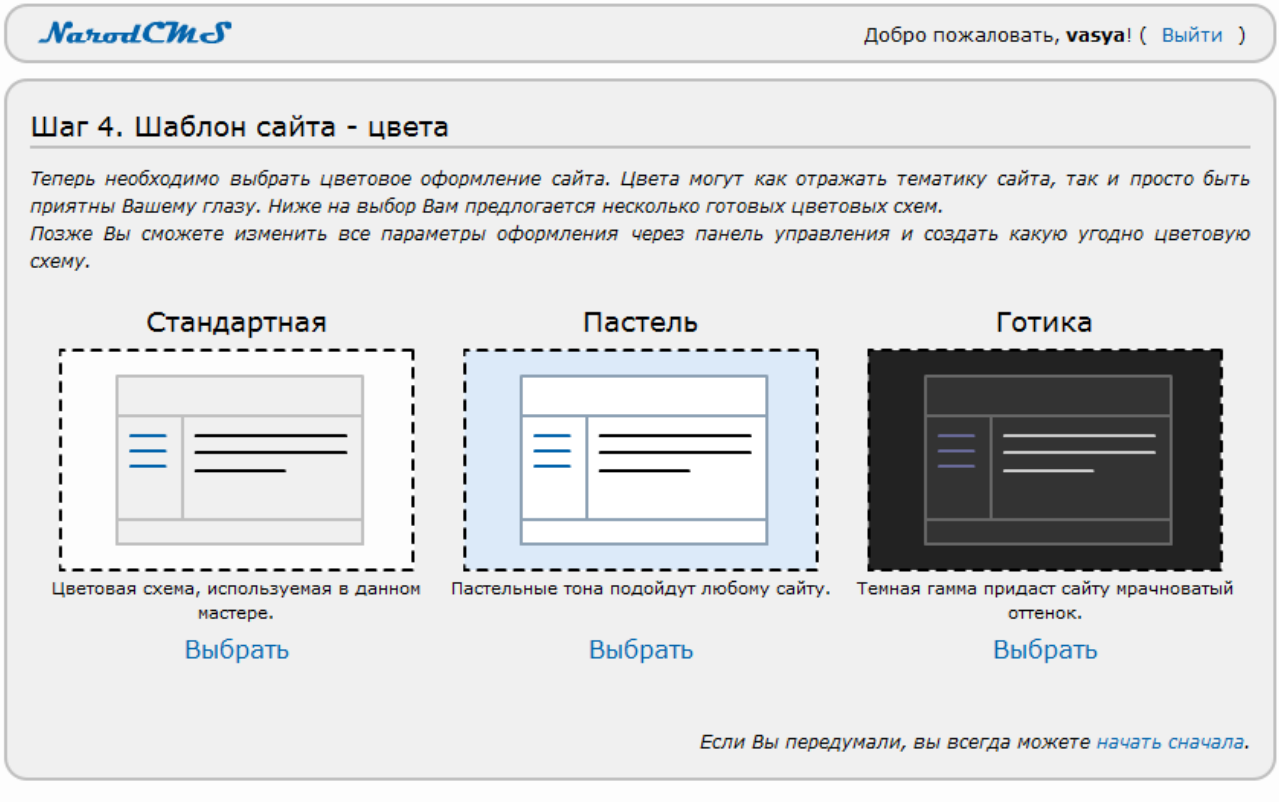

Рисунок 9 – Мастер шаблонов, шаг 4

Шаг 5 не содержит вариантов выбора. Он подразумевает завершение работы мастера нажатием кнопки «Создать сайт» либо переход назад к шагу 1.

После завершения работы мастера шаблонов пользователь перенаправляется в панель управления.

Внешний вид мастера шаблонов на шаге 5 представлен на рисунке 10.

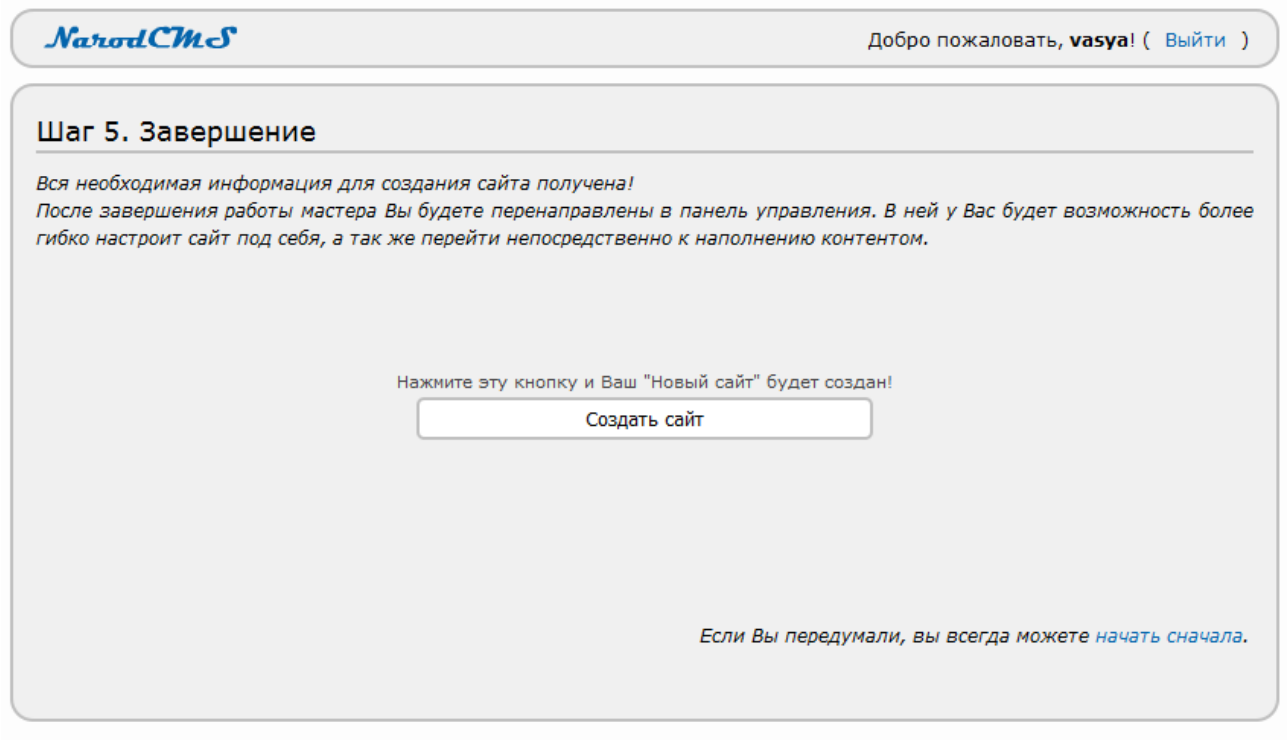

Рисунок 10 – Мастер шаблонов, шаг 5

3.3.3 Модуль контрольной панели

Модуль контрольной панели является основным инструментом пользователя для организации работы web-ресурса. Данный модуль предоставляет необходимый и достаточный набор функций для удобного наполнения ресурса контентом, изменения шаблона представления контента, а так же ряд дополнительных функций.

Интерфейс модуля состоит из шести основных разделов:

- страницы;
- блоки;
- меню;
- шаблон;
- файлы;
- опции.

Раздел «Страницы» предназначен для управления страницами ресурса, то есть его основным контентом. В данном разделе выводится список уже

существующих страниц, а так же ссылки для их создания, редактирования и удаления.

Внешний вид модуля в разделе «Страницы» представлен на рисунке 11. На нем изображен список из трех созданных страниц, ссылки для их удаления и редактирования, кнопка для создания новой страницы.

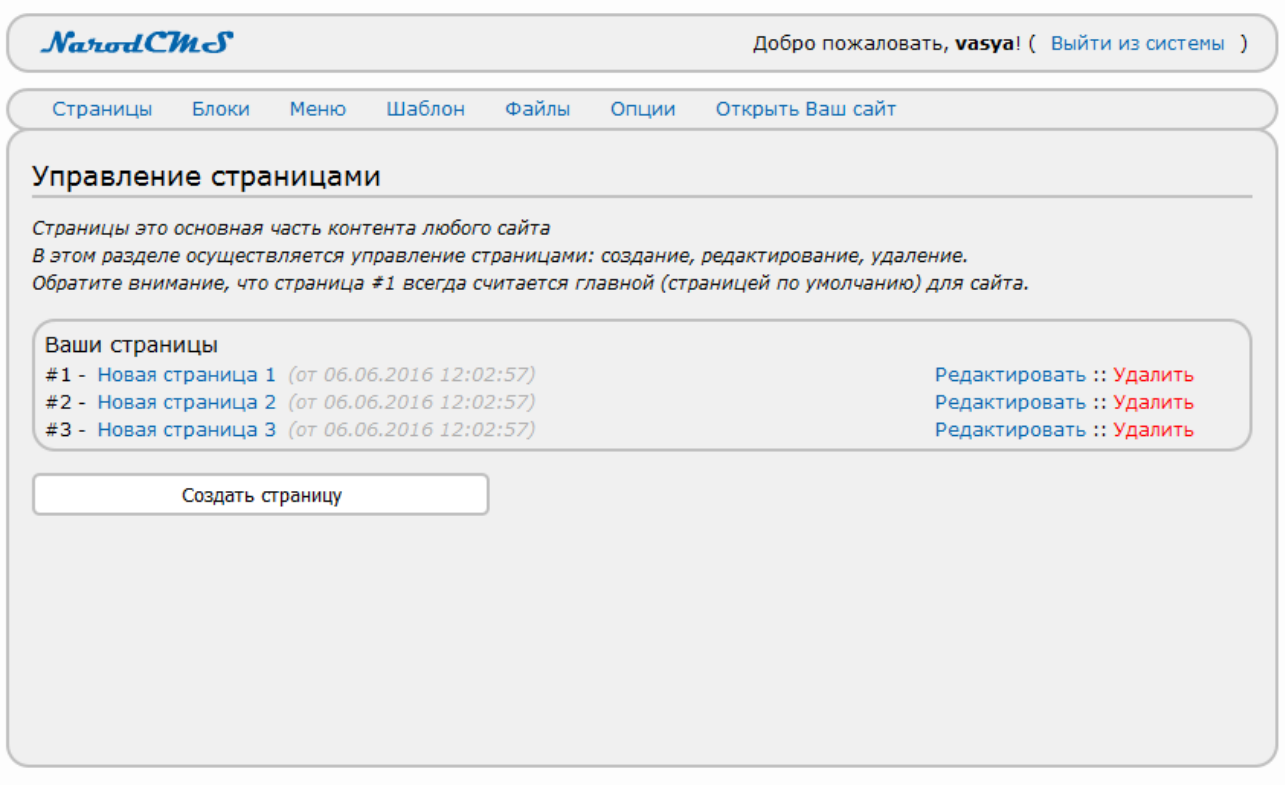

Рисунок 11 – Список страниц

Если нажать кнопку «Создать страницу», то откроется интерфейс, представленный на рисунке 12. Здесь пользователю предлагается ввести заголовок будущей страницы и ее текст.

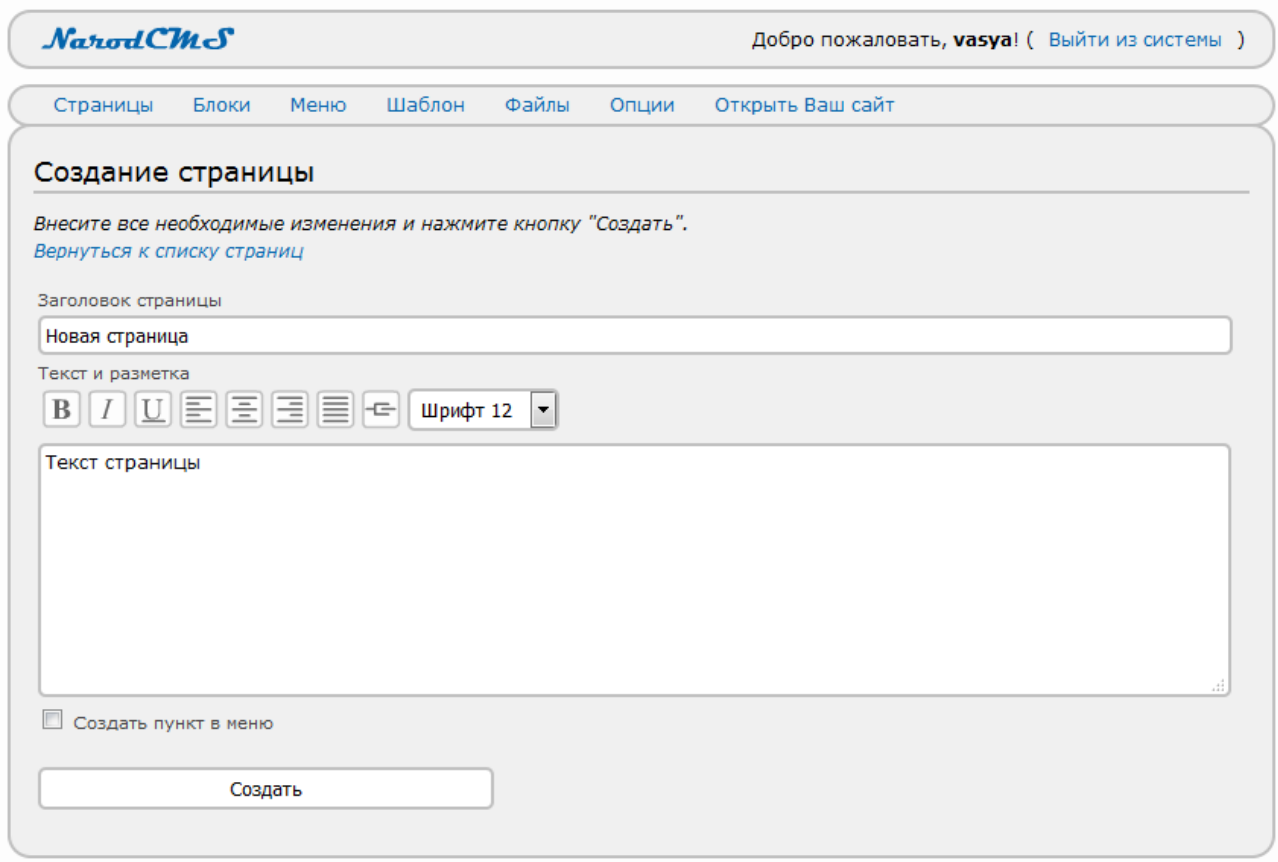

Рисунок 12 – Создание страницы

В тексте при этом можно использовать специальный внутренний язык разметки, позволяющий настроить отображение текста. Этот язык разметки позволяет настроить размер текста, его выравнивание на странице, а так же сделать его жирным, наклонным или подчеркнутым. Для доступа к разметке пользователю необходимо воспользоваться пиктограммами над полем ввода текста, после чего соответствующий код будет вставлен в общий текст страницы.

Так же пользователь может при создании страницы сразу добавить соответствующий пункт в меню ресурса при помощи флажка «Создать пункт в меню». При создании страницы ей автоматически присваивается порядковый номер, который так же виден в списке страниц.

Редактирование страницы производится аналогичным образом с той разницей, что изменения вносятся в уже существующую страницу.

Раздел «Блоки» предназначен для редактирования статичных блоков информации. Интерфейс и принцип работы с ним схож с созданием и редактированием страницы.

Для переключения между редактированием содержимого верхнего и нижнего блоков используются соответствующие ссылки.

Внешний вид модуля в разделе «Блоки» представлен на рисунке 13.

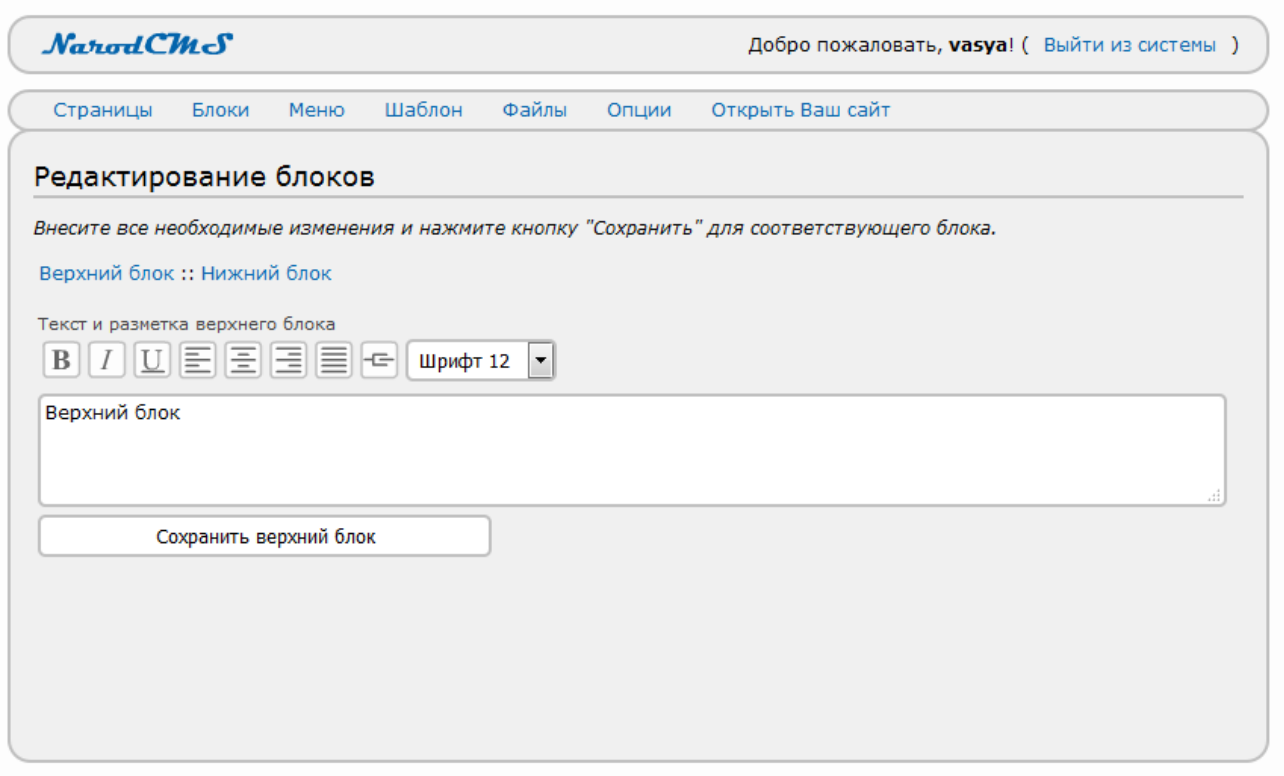

Рисунок 13 – Редактирование блока

Раздел «Меню» служит для управления пунктами меню web-ресурса. Меню является списком ссылок, выводящихся в определенном месте согласно настройке шаблона.

Каждый пункт меню имеет свое название и ссылку в виде адреса URL. Адрес может быть как внутренним (ссылающимся на страницу ресурса), так и внешним.

Каждый пункт меню имеет свой порядковый номер, который присваивается автоматически при создании и удалении пунктов, однако может быть изменен в

процессе работы. Порядковый номер определяет то, в каком порядке будут выводиться пункты меню.

Внешний вид модуля в разделе «Меню» представлен на рисунке 14.

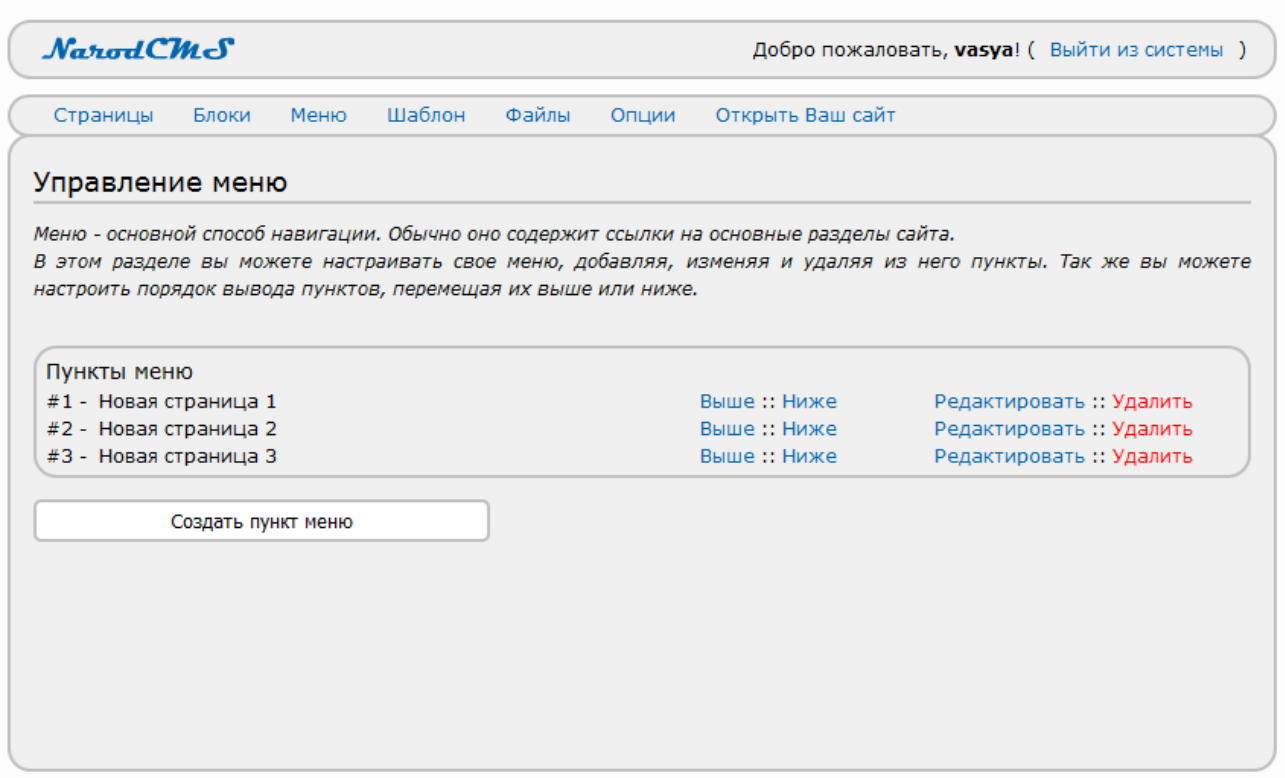

Рисунок 14 – Управление пунктами меню

Ссылки «Выше» и «Ниже» используются для уменьшения или увеличения порядкового номера пункта и соответственно поднятия его выше или ниже относительно текущей позиции вывода.

Кнопка «Создать пункт меню» вызывает интерфейс добавления нового пункта. Он представлен на рисунке 15. Пользователю предлагается заполнить поля с названием пункта и с URL, переход по которому будет осуществляться при выборе данного пункта.

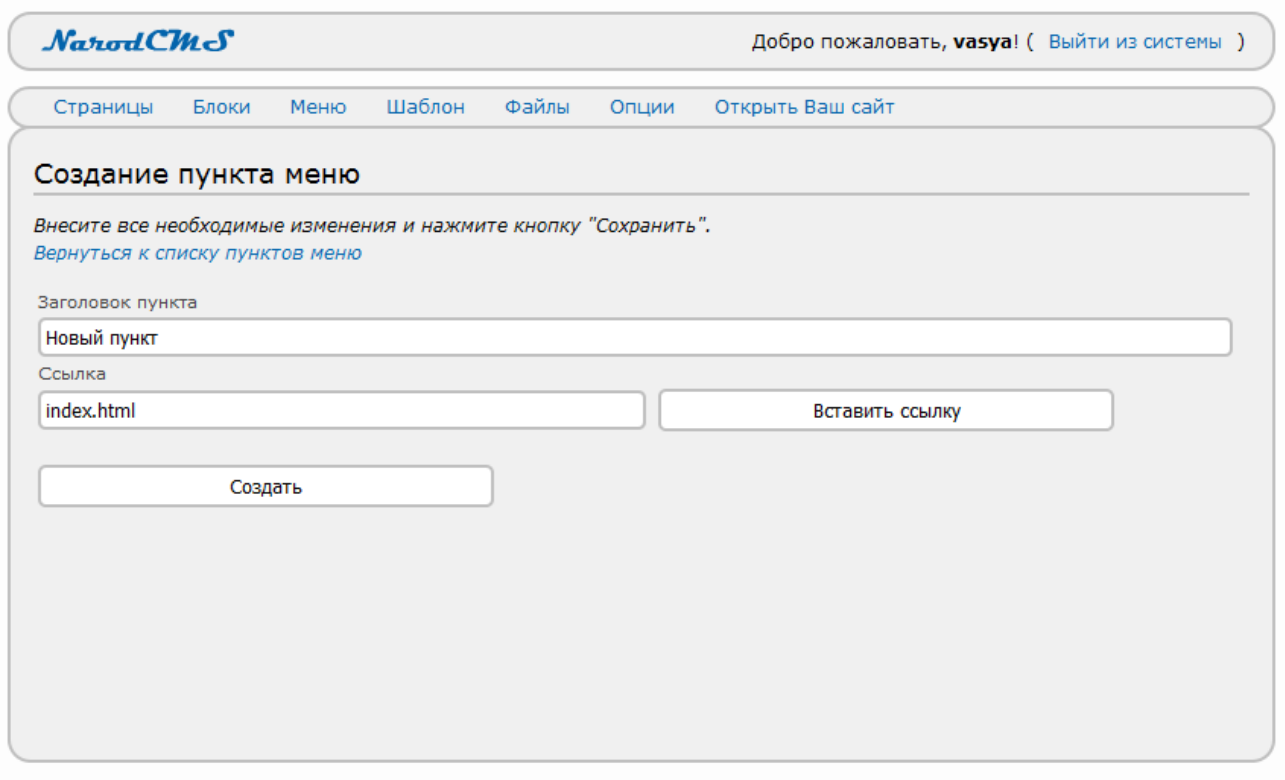

Рисунок 15 – Создание пункта меню

При этом у пользователя существует возможность вызвать специальное всплывающее окно упрощенной вставки ссылки нажатием кнопки «Вставить ссылку». Данное окно позволяет быстро получить ссылку на любую созданную ранее страницу ресурса или загруженный файл и вставить соответствующую ссылку в поле «Ссылка».

Внешний вид всплывающего окна вставки ссылки представлен на рисунке 16.

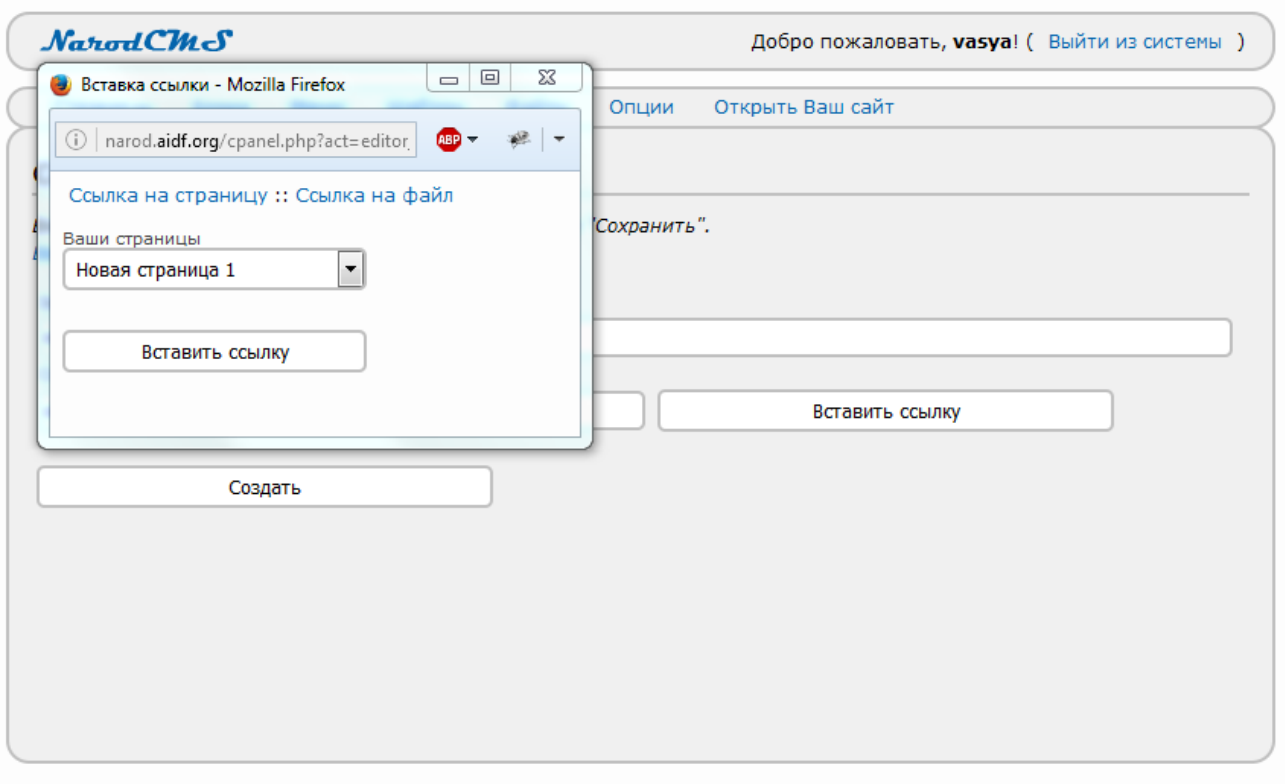

Рисунок 16 – Всплывающее окно «Вставка ссылки»

Раздел «Шаблон» содержит настройки шаблона ресурса. Данные параметры отвечают за внешний вид и оформление создаваемого web-ресурса.

Данный раздел состоит из трех подразделов:

- «Основное»;
- «Цвета»;
- «Размеры».

Подраздел «Основное» позволяет изменить название ресурса, а так же его шаблон путем изменения расположения меню и блоков информации путем выбора нужного пункта из выпадающего списка.

Выпадающий список «Блоки» содержит четыре пункта:

- «Верхний и нижний блоки»;
- «Только верхний»;
- «Только нижний»;
- «Без блоков».

Выпадающий список «Меню» содержит три пункта:

- «Меню слева»;
- «Меню сверху»;
- «Без меню».

Внешний вид подраздела «Основное» представлен на рисунке 17.

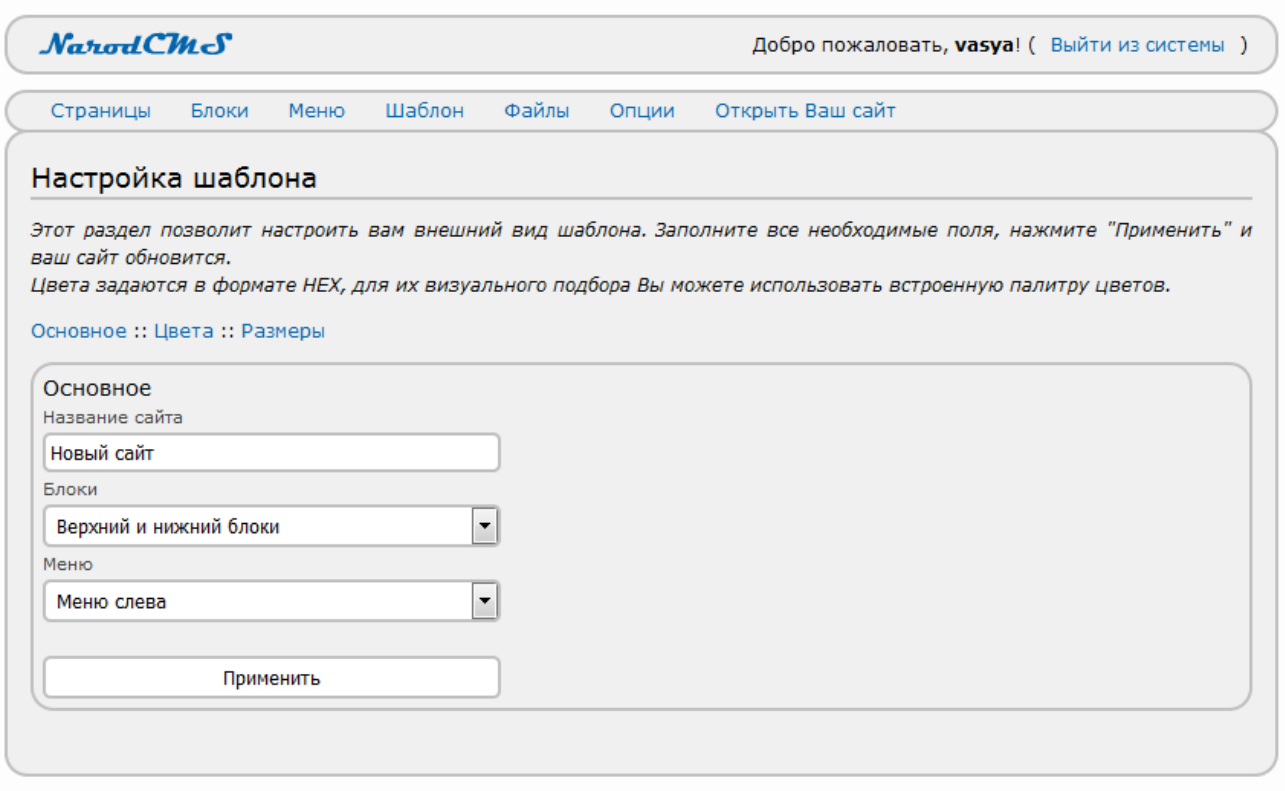

Рисунок 17 – Настройки шаблона, подраздел «Основное»

Подраздел «Цвета» содержит HTML коды цветов, используемых для соответствующих элементов шаблона. К этим элементам относятся:

- текст;
- ссылки;
- фон блоков;
- рамки блоков;
- фон сайта.

Внешний вид подраздела «Цвета» представлен на рисунке 18.

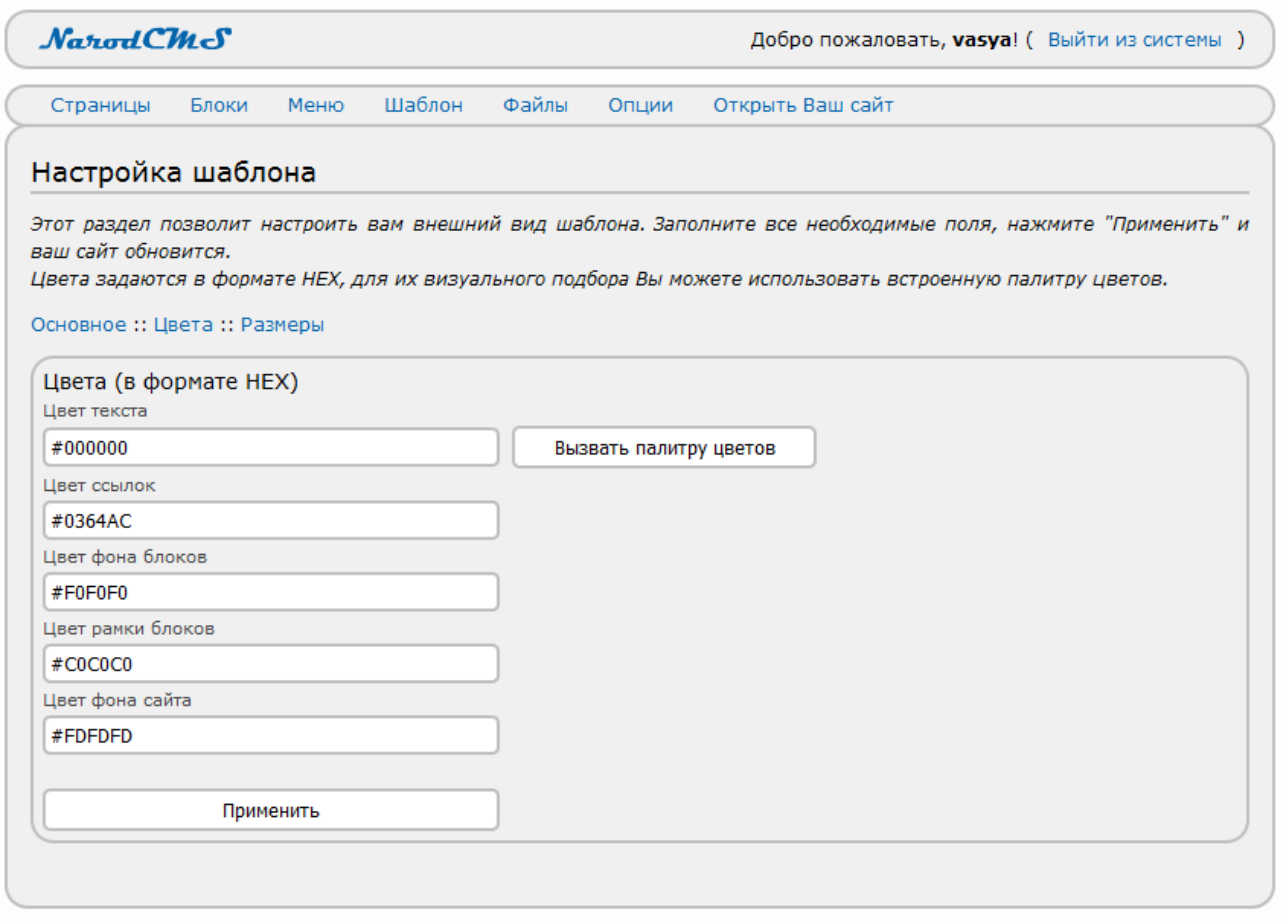

Рисунок 18 – Настройки шаблона, подраздел «Цвета»

При этом пользователь может воспользоваться вспомогательным диалоговым окном «Палитра» для подбора нужного ему цвета и получения его HTML кода. Для того чтобы вызвать это окно, пользователю необходимо нажать кнопку «Вызвать палитру цветов».

Внешний вид всплывающего окна «Палитра» представлен на рисунке 19.

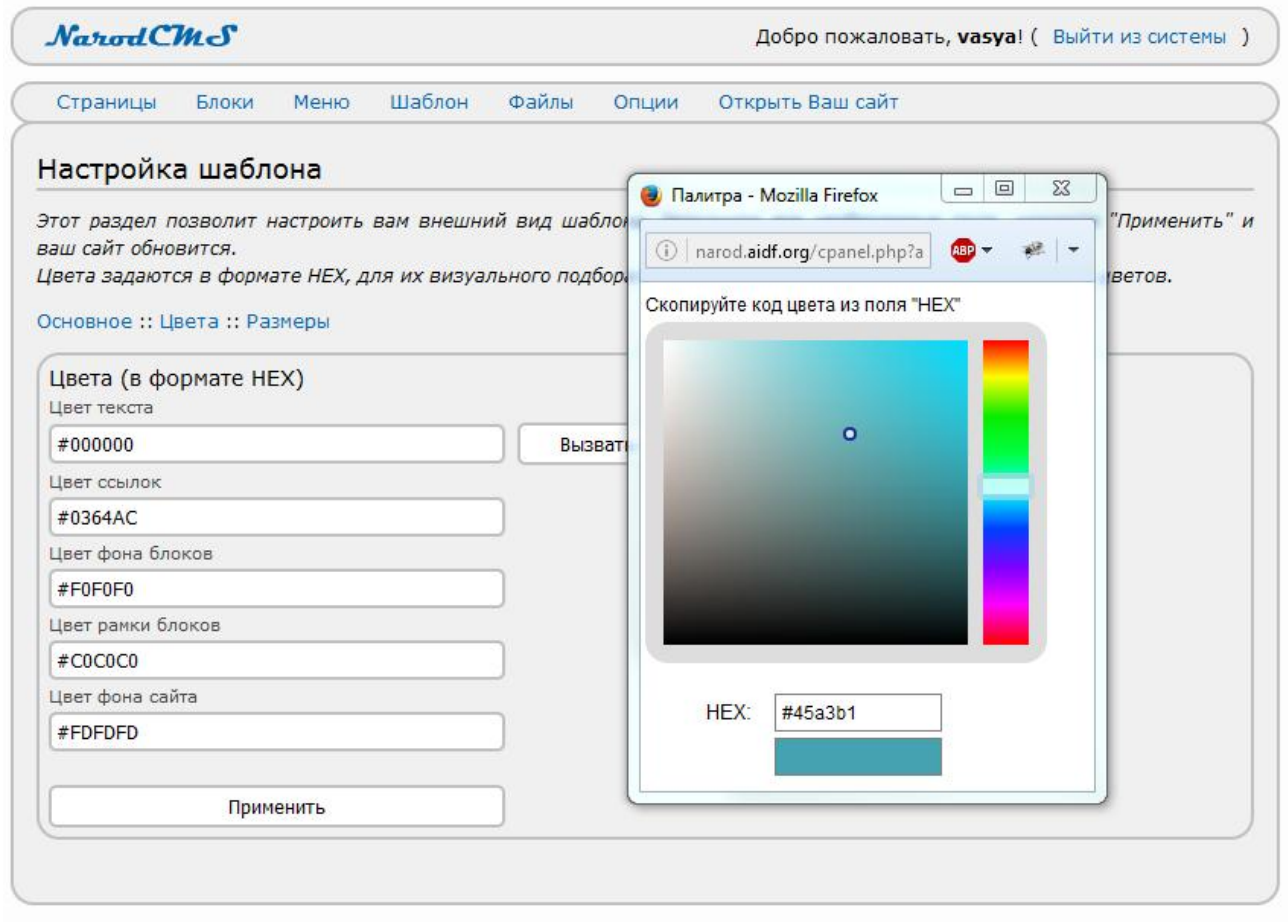

Рисунок 19 – Всплывающее окно «Палитра»

Подраздел «Размеры» содержит размеры основных элементов шаблона. В этом разделе можно задать:

- размер шрифта;
- ширину или высоту меню;
- высоту верхнего и нижнего блоков;
- общую ширину информационной части ресурса.

Внешний вид подраздела «Размеры» представлен на рисунке 20.

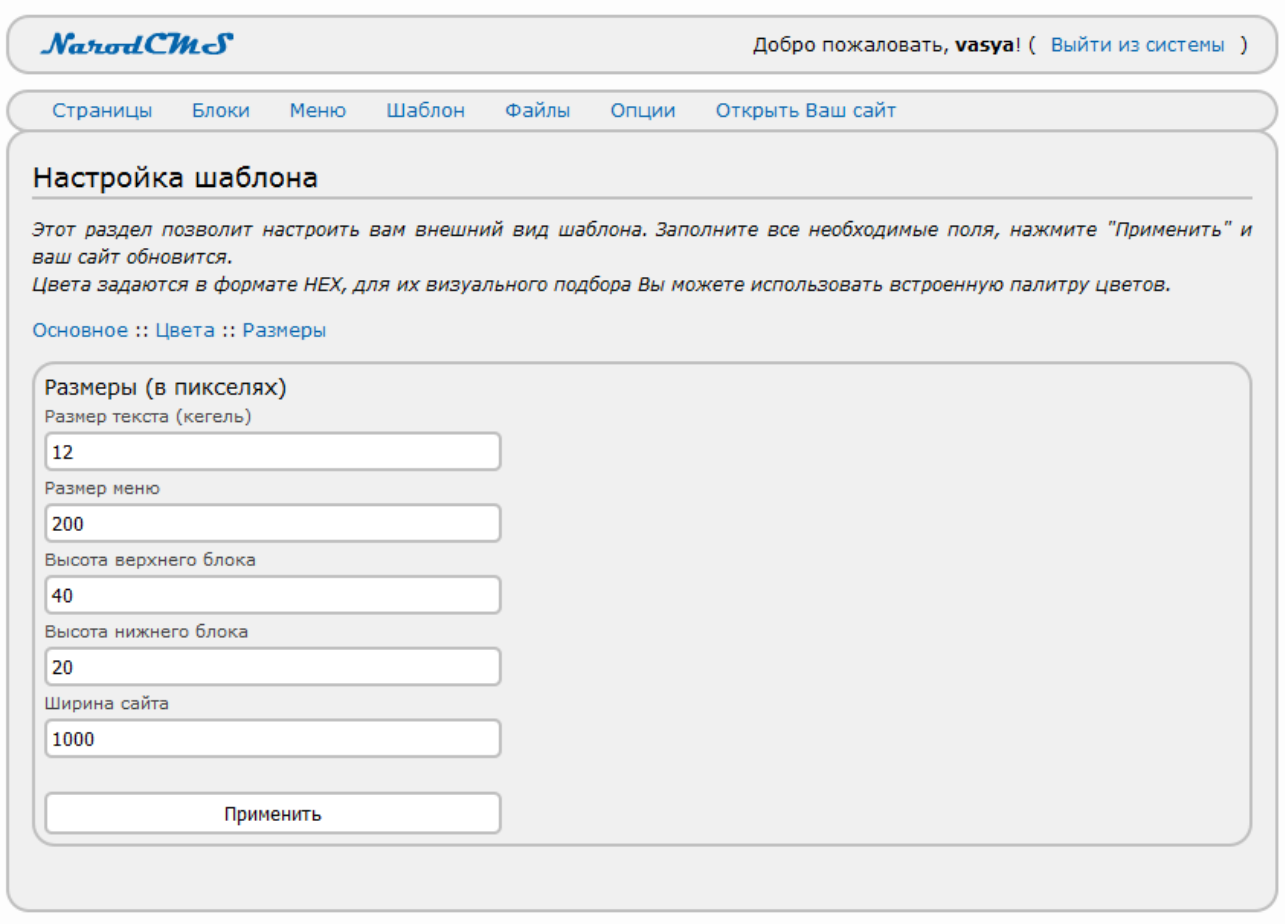

Рисунок 20 – Настройки шаблона, подраздел «Размеры»

Раздел «Файлы» содержит список загруженных на ресурс файлов. Данный раздел предоставляет возможность загружать файлы и удалять уже имеющиеся.

Внешний вид подраздела «Файлы» представлен на рисунке 21.

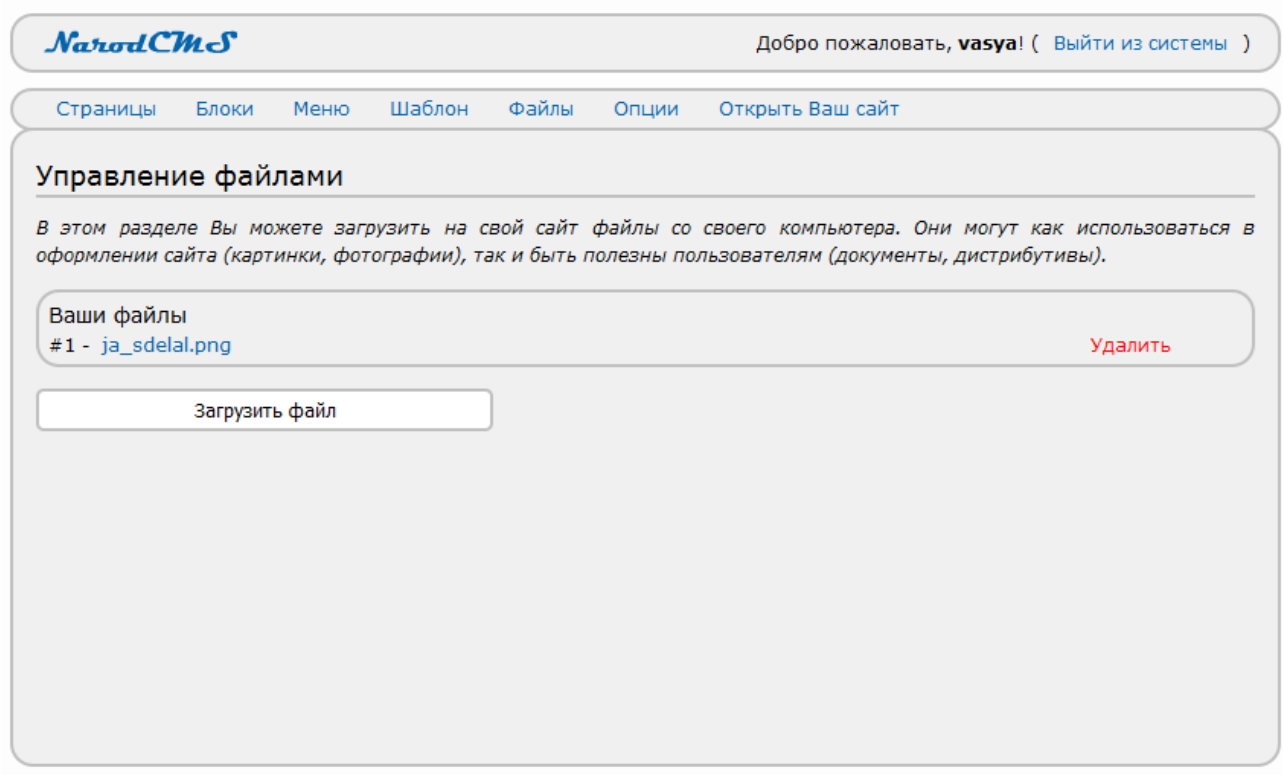

Рисунок 21 – Раздел «Файлы»

Раздел «Опции» содержит ряд дополнительных опций для владельца ресурса, таких как:

- создание архивной копии;
- удаление ресурса;
- удаление учетной записи.

Создание архивной копии ресурса позволяет получить в виде архива полноценную копию ресурса в виде HTML документов и файлов. Данную копию впоследствии можно автономно просматривать без доступа к сети Интернет, либо разместить ее на стороннем хостинге, поддерживающим загрузку HTML документов.

Удаление ресурса позволяет удалить ресурс целиком из системы. При этом с хостинга стираются все файлы ресурса, а информация удаляется из базы данных. После это операции при повторном входе в систему для пользователя будет вызван модуль мастера шаблонов, и ресурс можно будет создать заново.

При этом по желанию пользователя, вместе с ресурсом можно удалить так же и саму учетную запись пользователя. В этом случае продолжить работу с системой управления контентом можно будет только пройдя процедуру регистрации. При это логин удаленной учетной записи снова освобождается для регистрации.

В целях безопасности операция удаления ресурса требует от пользователя ввода его пароля.

Внешний вид подраздела «Опции» представлен на рисунке 22.

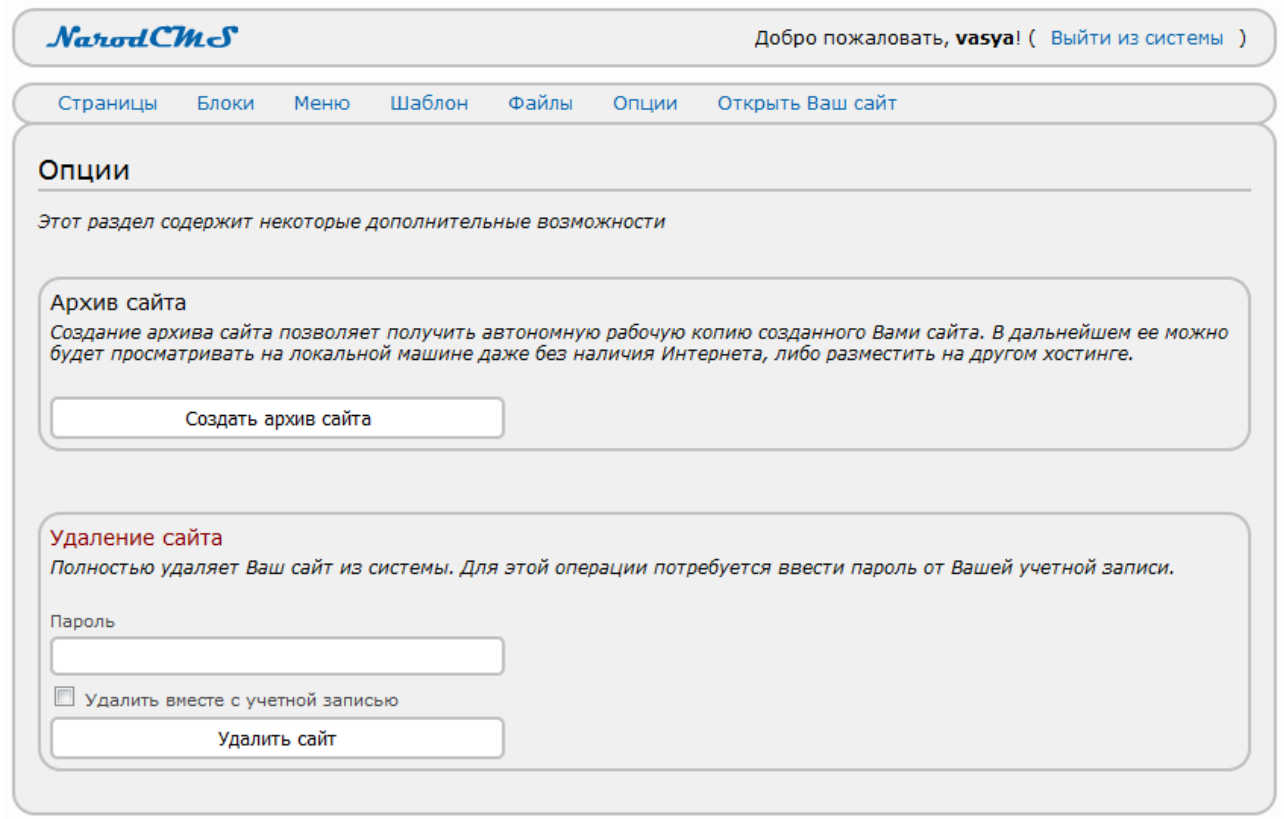

Рисунок 22 – Раздел «Опции»

ЗАКЛЮЧЕНИЕ

Целью данной работы было изучение способов и средств представления информации в телекоммуникационных сетях и сети Интернет. Были рассмотрены различные типы web-ресурсов, выделены их признаки.

В ходе изучения было установлено, что оптимальным и наиболее популярным способом представления информации является представление ее с помощью шаблонов.

В ходе анализа направленности ресурсов и объема информации на них был сделан вывод о необходимости наличия специальных систем управления содержимым, позволяющих управлять информацией и осуществлять поддержку ресурса.

Целью дипломного проекта была разработка подобной системы.

В ходе достижения данной цели были выполнены задачи:

- проведен анализ существующих web-ресурсов в сети Интернет;
- изучены программные и аппаратные средства создания web-ресурсов;
- были сделаны выводы о необходимости разработки специальных систем управления содержимым и рассмотрены их существующие примеры;
- была составлена модель проектируемой системы;
- были выбраны архитектура системы и средства ее реализации;
- были изучены язык разметки HTML, язык программирования PHP и язык запросов SQL;
- были разработаны и реализованы модули системы управления содержимым.

В результате дипломного проектирования была создана система управления содержимым, которая может использоваться для организации web-ресурсов как в корпоративных телекоммуникационных сетях, так и глобальной сети Интернет.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Колисниченко, Д.Н. Drupal 7. Руководство пользователя / Д.Н. Колисниченко. – Москва: Вильямс, 2011. – 256 с.
2. Расторгуев, О.П. Сайт на 1С-Битрикс. Создание, поддержка и продвижение / О.П. Расторгуев. – Санкт-Петербург: Наука и техника, 2012. – 256 с.
3. Сергеев, А.Н. Создание сайтов на основе WordPress. Учебное пособие / А.Н. Сергеев. – Санкт-Петербург: Лань, 2015. – 555 с.
4. Форта, Б. SQL за 10 минут / Б. Форта. – Москва: Вильямс, 2015. – 288 с.
5. Янк, К. PHP и MySQL. От новичка к профессионалу / К. Янк. – Москва: Эксмо, 2013. – 384 с.

ПРИЛОЖЕНИЕ А

Техническое задание

Настоящее техническое задание содержит требования к автоматизированной системе наполнения web-контента.

А.1 Наименование программного продукта и основание для разработки

А.1.1 Наименование

Программному продукту, разрабатываемому на основании данного технического задания, присваивается наименование: «Программный комплекс автоматизированного наполнения Web-контента».

А.1.2 Основание для разработки

Основанием для разработки программного продукта является задание на дипломный проект.

А.2 Назначение разработки программного продукта

Программный продукт предназначен для организации и поддержки в сети Интернет информационного ресурса. Программный продукт будет использоваться для автоматизированного наполнения web-ресурса содержимым.

А.3 Требования к программному продукту

А.3.1 Описание функций программного продукта

Разработанный программный продукт должен иметь следующие функции:

- регистрация и авторизация пользователей системы;
- предоставление панели управления;
- публикация контента.

Регистрация и авторизация пользователей системы необходима для организации независимой работы сразу нескольких.

Предоставление панели управления обеспечивает пользователю полный и наглядный доступ ко всем функциям системы через графический интерфейс системы.

Публикация контента обеспечивает размещение информации пользователя в сети в виде сформированного отдельного web-ресурса, доступного для просмотра пользователям сети.

А.3.2 Требования к надежности

Программный продукт должен выполнять все реализованные функции в течение всего срока эксплуатации.

Программный продукт и его данные должны быть защищены от некорректных действий пользователя.

Программный продукт должен предупреждать пользователя в тех случаях, когда его действия приводят к необратимым последствиям и могут повлиять на работоспособность самого продукта или системы в целом.

А.3.3 Требования к условиям эксплуатации

Требования к условиям эксплуатации данного программного продукта соответствуют требованиям, предъявляемым к условиям эксплуатации персонального компьютера.

А.3.4 Требования к составу и параметрам технических средств

Состав необходимых программному продукту аппаратных средств:

- ПЭВМ на базе процессора Intel Pentium IV 3.2 GHz или выше;
- ОЗУ объемом от 1 Gb;
- жесткий диск объемом от 80 Gb;
- цветной монитор с диагональю от 19’’;
- сетевая карта Ethernet;
- клавиатура;
- мышь.

А.3.5 Требования к программной совместимости

Для успешного функционирования программного продукта в системе должны присутствовать следующие пакеты:

- веб-сервер Apache HTTP Server;
- СУБД MySQL;
- интерпретатор языка PHP.

Взаимодействие с программным продуктом должно осуществляться через любой современный веб-браузер, вышедший после 2006 года.

А.3.6 Требования к носителю программного продукта

Программный продукт должен поставляться на компакт-диске.

А.4 Требования к документации

К программному продукту должно прилагаться техническое описание.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ С.А. Виденин
подпись инициалы, фамилия
«___» _____ 2016г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

в форме дипломного проекта
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Ахметшину Павлу Сергеевичу
фамилия, имя, отчество

Группа ЗКИ 10-08-1 Направление (Специальность) 230102.65
номер код

Автоматизированные системы обработки информации и управления
наименование

Тема выпускной квалификационной работы Программный комплекс для автоматизированного наполнения Web-контента

Утверждена приказом по университету № 4201/с от 28.03.16

Руководитель ВКР Л.С. Троценко, доцент, к.т.н., ИКИТ кафедра ИС
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР: руководство пользователя программного пакета XAMPP; руководство пользователя текстового редактора Notepad++.

Перечень разделов: характеристика объекта разработки; проектирование системы управления содержимым; описание разработанной системы управления содержимым.

Перечень графического материала.

- Представление.
- Цели и задачи работы.
- Назначение систем управления содержимым.
- Сравнение существующих систем.
- Функции разработанной системы.
- Описание шаблона представления информации.
- Модуль учетных записей.
- Модуль мастера шаблонов.
- Модуль контрольной панели.
- Текстовый редактор модуля контрольной панели.
- Пример сформированного web-ресурса.
- Заключение.

Руководитель ВКР _____ Л.С. Троценко
подпись инициалы, фамилия

Задание принял к исполнению _____ П.С. Ахметшин
подпись инициалы, фамилия

« ___ » _____ 2016 г.