

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра _____ «Коммерция и бизнес-информатика»

СОГЛАСОВАНО
Генеральный директор
_____ А.Ю.

Севостьянов
подпись инициалы, фамилия

« ___ » _____ 2020 г.

УТВЕРЖДАЮ
Заведующий кафедрой

_____ инициалы, фамилия

« ___ » _____ 2020 г.

БАКАЛАВРСКАЯ РАБОТА
на тему:

_____ Разработка системы управления контентом сайта _____

по направлению подготовки _____ 38.03.05 – Бизнес-информатика _____

шифр, наименование направления подготовки

Профиль _____ 38.03.05.01- Информационные технологии в бизнесе _____

шифр, наименование направления подготовки

Автор работы _____ К.О. Четвертков _____ Группа ББИ-41 _____

подпись, дата

инициалы, фамилия

Обозначение работы _____ ТГТУ.38.03.05.01.014 БР ДЭ _____

Обозначение документа _____ ТГТУ.38.03.05.01.014 БР ТЭ-ТЛ _____

Руководитель работы _____ Н.А.Инькова _____

подпись, дата

инициалы, фамилия

Нормоконтролёр _____

подпись, дата

_____ Н.В. Дюженкова _____

инициалы, фамилия

Тамбов 2020

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра _____ «Коммерция и бизнес-информатика»

СОГЛАСОВАНО
Генеральный директор
_____ А.Ю.

УТВЕРЖДАЮ
Заведующий кафедрой

Севостьянов

подпись

инициалы, фамилия

подпись

инициалы, фамилия

«__» _____ мая 2020 г.

«__» _____ мая 2020 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

по направлению подготовки _____ 38.03.05 – Бизнес-информатика

шифр, наименование направления подготовки

Профиль _____ 38.03.05.01- Информационные технологии в бизнесе

шифр, наименование направления подготовки

Тема Разработка системы управления контентом сайта

утверждена приказом _____ №86/8-08 от «15» мая 2020 г.

Автор работы _____ К.О. Четвертков _____ Группа ББИ-41

инициалы, фамилия

Обозначение работы ТГТУ.38.03.05.01.014 БР ДЭ

Обозначение документа _____ ТГТУ.38.03.05.01.014 БР ТЭ-ЗД

Срок представления к защите «__» _____ 2020 г.

Исходные данные для исследования Учебники, учебные пособия, монографии, статьи, годовые отчёты предприятия

Перечень подлежащих разработке вопросов:

1 Основные подходы к разработке систем управления контентом сайтов (CMS)

2 Разработка CMS для интернет-маркетинговой организации

Руководитель работы _____ Н.А.
Инькова

подпись, дата

инициалы, фамилия

Задание принял к исполнению _____ К.О.
Четвертков

подпись, дата

инициалы, фамилия

АННОТАЦИЯ

Бакалаврская работа на тему: «Разработка системы управления контентом сайта». Выполнил студент четвертого курса группы ББИ-41 направления подготовки 38.03.05 – «Бизнес-информатика» Четвертков Кирилл Олегович. Руководитель: кандидат экономических наук, доцент кафедры «Коммерция и бизнес-информатика» Федерального государственного бюджетного образовательного учреждения высшего образования «Тамбовский государственный технический университет» Наталья Анатольевна Инькова.

Работа представлена к защите в 2020 году.

По своей структуре бакалаврская работа состоит из введения, двух разделов, заключения и списка использованных источников.

Цель работы: Создание системы управления контентом сайта с использованием современных методов разработки.

Результат работы: Разработана индивидуальная система управления контентом сайта, отвечающая всем необходимым требованиям.

Бакалаврская работа содержит 54 страницы, 31 рисунков.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
....	
1. Основные подходы к разработке систем управления контентом сайтов (CMS).....	6
.....	
1.1 Сравнительный анализ существующих систем управления контентом сайтов	6
.....	
1.2 Анализ современных средств и технологий разработ ки CMS.....	2 9
2. Разработка CMS для интернет-маркетинговой организации.....	4 0
2.1 Технологии, используемые в разработке проекта	4 0
2.2 Описание функций проекта	4 8
ЗАКЛЮЧЕНИЕ.....	5
...	3
СПИСОК ИСПОЛЬЗОВАННЫХ	5
ИСТОЧНИКОВ.....	4

ВВЕДЕНИЕ

С ростом роли сети Интернет растет и необходимость представления тех или иных услуг или продуктов посредством представления их через сайт. Вариантов сайтов великое множество: от самого простого сайта-визитки до интернет-магазина (как средства продажи товаров и услуг через Интернет) и корпоративного портала, состоящего из множества возможностей: веб-интерфейс для доступа сотрудников к корпоративным данным и приложениям, онлайн чаты, интеграция с другими системами, система управления задачами и т.д. «Легкие» сайты (не более 20 страниц) возможно сделать и самостоятельно, не тратя на это денежных средств, а вот крупные информационные сайты или даже интернет-магазины, например, требуют гораздо более широких функциональных возможностей как для пользователей сайтов, так и для их администраторов. Чтобы, например, удалить пункт меню в рукописном сайте, нужно найти код и поправить его. Для работы же с CMS в большинстве случаев не нужно обладать особыми навыками, достаточно прочесть помощь или инструкцию.

Основной целью проекта является разработка системы управления контентом для интернет-маркетинговой организации.

В бакалаврской работе были поставлены и решены следующие задачи:

- Проведен сравнительный анализ существующих систем управления контентом сайтов;
- Рассмотрены современные средства и технологии, которые можно использовать при разработке системы управления контентом;

- Описаны функции проекта;

1 Основные подходы к разработке систем управления контентом сайтов (CMS)

1.1 Сравнительный анализ существующих систем управления контентом сайтов

CMS расшифровывается как «Content Management Software» («программное обеспечение для управления содержимым»). В нашей стране принято последнюю букву «S» расшифровывать как «System», звучит как «Система управления контентом».

Контент (перевод английского термина content, означающего содержание, содержимое) – информационное наполнение web-сайта (сайта), т. е. все типы материалов, которые находятся на сервере: web-страницы, документы, программы, аудио- и видеофайлы и так далее.

Управление – это процесс планирования, организации, мотивации и контроля, необходимый для того, чтобы сформулировать и достичь цели организации. В нашем случае целью является сайт. Причем сайт не просто отображающий информацию о компании в просторах интернета, а тот сайт, который позволяет самостоятельно заказчику работать над его управлением и поддержкой, не прибегая к постоянной помощи веб-разработчиков.

На рисунке 1 показан процесс взаимодействия пользователя CMS по-средством сайта.

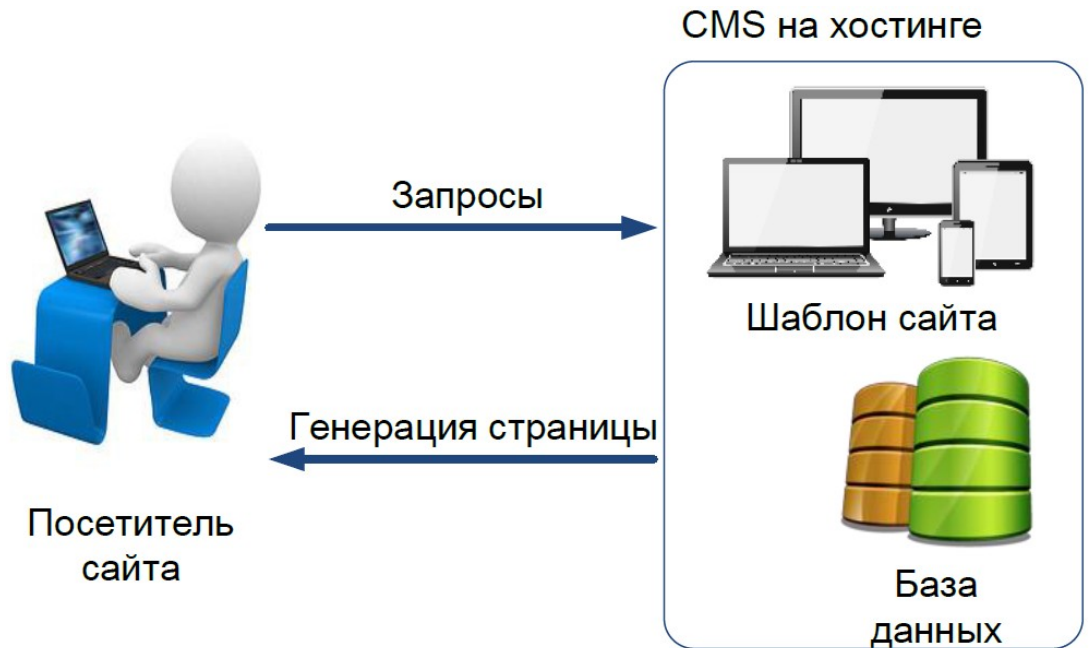


Рисунок 1 - Взаимодействие пользователя и CMS

Одной из особенностей CMS является то, что работа с ней не требует установки какого-либо дополнительного ПО на компьютер, с которого ведется работа с сайтом. Сама программа размещается на сервере провайдера, а пользователю достаточно иметь простой веббраузер (Chrome, Firefox и т.п.).

Как же появились CMS? Обычно заказы на сайты очень похожи, ведь это, по сути, реализация группы страниц в интернете. Прописывать вручную тексты, создавать одни и те же функции из раза в раз и похожие шаблоны сайтов, конечно, не нужно. Кроме того, многие заказчики самостоятельно хотят следить за информацией на своем сайте. А учить их основам программирования PHP, JavaScript и т.п., конечно, не хочется никому. Веб-студии начали разрабатывать свои оболочки для работы с сайтом. Находились и энтузиасты, делающие бесплатные и открытые оболочки. В итоге на данный момент имеются как платные, так и бесплатные системы, с помощью которых можно создавать от самого простого блога до интернет-магазинов и корпоративных порталов.

Существует множество CMS и они все имеют свои особенности. Каждая из них включает в себя разнообразный

функционал, но есть общие возможности, которыми должна обладать каждая CMS.

1. Работа с базой данных сайта (например MySQL). Это первая и самая важная функция, поскольку выдача контента происходит динамически, нет необходимости хранить каждую страницу на сервере, хранение списка пользователей в системе и их регистрационных данных и т.п.

2. Управление структурой сайта – удаление и создание данных.

3. Визуальный редактор WYSIWYG – удобное управление контент-наполнением страниц и разделов (любые страницы, новости, статьи, фото, видео и т.п.), рисунок 2.

WYSIWYG (сокращение от англ. What You See Is What You Get, «что видишь, то и получишь», произносится как «визивиг») – способ редактирования, при котором редактируемый материал в процессе редактирования выглядит так же, как и конечный результат. WYSIWYG-редактор стараются делать максимально приближенным к привычному MS Word. Благодаря этому освоить работу на современных CMS могут самые обычные пользователи.

4. Организация разделения прав пользователей (доступ администратора – полный, доступ контент-менеджера – только изменение страниц, доступ администратора магазина – управление заказами и т. п.).

5. Интеграция с другими информационными системами (например, 1С: Предприятие).

6. Возможность расширения за счет добавления дополнительных модулей.

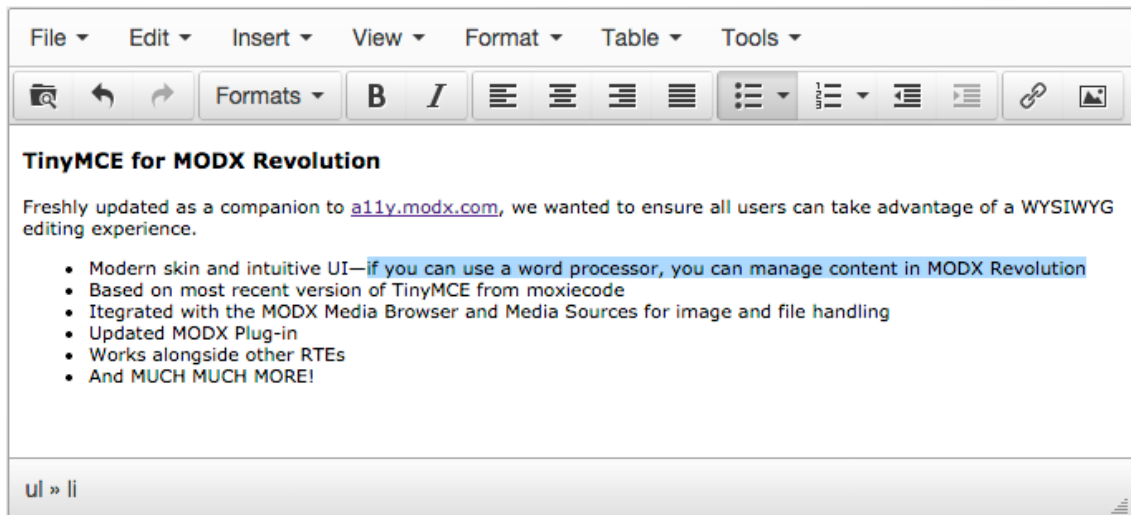


Рисунок 2 - интерфейс визуального редактора Tiny MCE
 К CMS можно применить ряд следующих ниже требований, как и к любому программному продукту.

- кроссбраузерность - корректное отображение готового сайта в различных браузерах;
- масштабируемость - система должна поддерживать любое количество пользователей;
- гибкость - система должна поддерживать любую структуру сайта, должна настраиваться под бизнес-процессы организации и отдельных ее сотрудников, должна поддерживать любое количество шаблонов страниц;
- юзабилити - удобный и интуитивно понятный интерфейс, возможность смены дизайна без отключения основного работающего ресурса;
- многоплатформенность - система должна работать на любом аппаратном и программном обеспечении Web-сервера и

не создавать неоправданно большую нагрузку на оборудование хостинга;

- безопасность - ограничение прав доступа, защищенность от взломов;

- резервное копирование - возможность легко и просто скопировать систему или развернуть резервную копию без потери данных.



Рисунок 3 - основные требования к CMS.

Сайты, созданные на основе CMS, не являются отдельными страницами, так как в основе CMS лежит разделение дизайна и текстового наполнения. Ведь чаще всего нет необходимости менять структуру дизайна на страницах, но есть необходимость изменять текст и иную информацию на них.

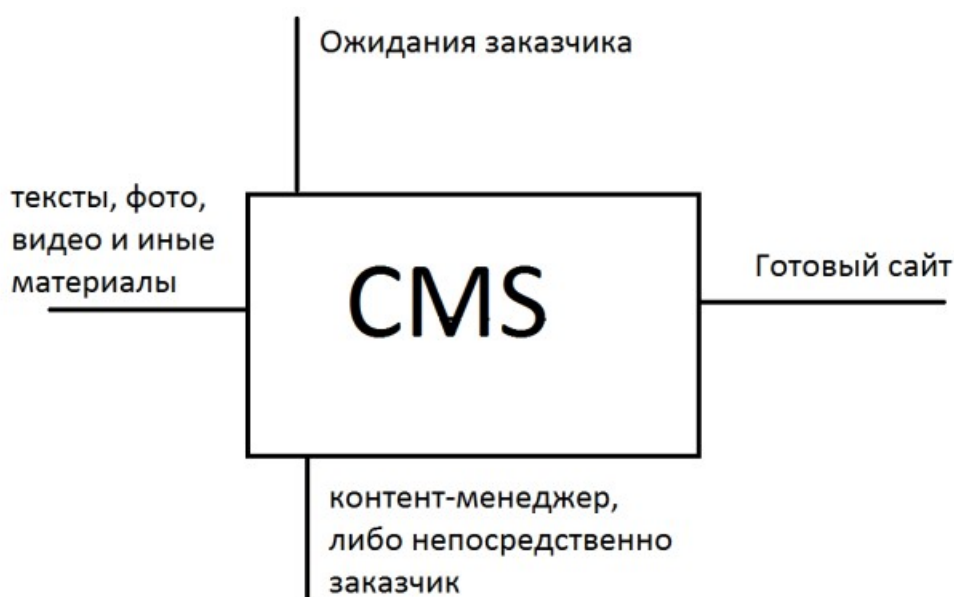


Рисунок 4 - представление организации работы с CMS

Если смотреть с точки зрения обычного заказчика, то разработка сайта на основе какой-либо CMS выглядит согласно рисунку 4 и должна приносить следующие преимущества:

- в работе используется наиболее эффективный инструмент для решения конкретной задачи (в зависимости от вида сайта и требований к его функционалу подбирают оптимальную CMS);

- использование CMS позволяет владельцу сайта самостоятельно создавать и удалять разделы сайта, редактировать различную информацию без привлечения стороннего специалиста;

- работа сайта постоянно тестируется множеством пользователей, а найденные ошибки и уязвимости достаточно оперативно устраняются, при этом сайт работает на самых передовых и проверенных технических решениях;

- временные затраты на разработку сайта существенно снижаются, так как разработчику не надо фиксировать свое внимание только на технических задачах: «как сделать ленту с новостями» или «как научить CMS искать товары в каталоге», а можно сосредоточиться на информационной и визуальной составляющих будущего сайта;

- кроме того, должно быть удобным продвижение сайта и реклама его в поисковиках.

По принципу открытости кода CMS делятся:

- на CMS с открытым кодом;

- CMS с закрытым кодом.

CMS с открытым кодом предоставляет возможность вносить изменения в ядро системы, а также в отдельные ее элементы, таким образом улучшая ее, в соответствии с требованиями заказчика.

CMS с закрытым кодом изменять нельзя. Кроме того, на основе закрытого кода фирма-производитель дает протестировать заказчику функционал продаваемой системы.

Состоят CMS обычно из двух частей:

- back-end - это часть системы, отвечающая за функциональность и хранение информации;
- front-end - это часть системы, обеспечивающая интерфейс с конечным пользователем.

С позиции разработчика сайта. Благодаря наличию уже готовых модулей CMS дают возможность производить изменения в структуре сайтов качественно и оперативно путем компоновки этих модулей, уменьшая вероятность появления ошибок в программном коде. Кроме того, происходит колоссальная экономия времени на написание и поиск однотипных решений.

С позиции пользователя. CMS - это прежде всего удобный интерфейс для редактирования содержимого сайта. Также CMS позволяют не просто показывать посетителям сайта информацию, но и взаимодействовать с ними в режиме диалога: посетители могут зарегистрироваться, общаться на форуме или в чате, оставлять свои замечания и пожелания в гостевой книге, делать заказы и видеть их историю и т.д.

Рассмотрим наиболее популярные системы управления сайтом.

1) CMS Joomla!

Joomla! представляет собой бесплатную систему для создания сайтов. Это проект с открытым исходным кодом. Он очень успешно развивается, в течение семи лет.

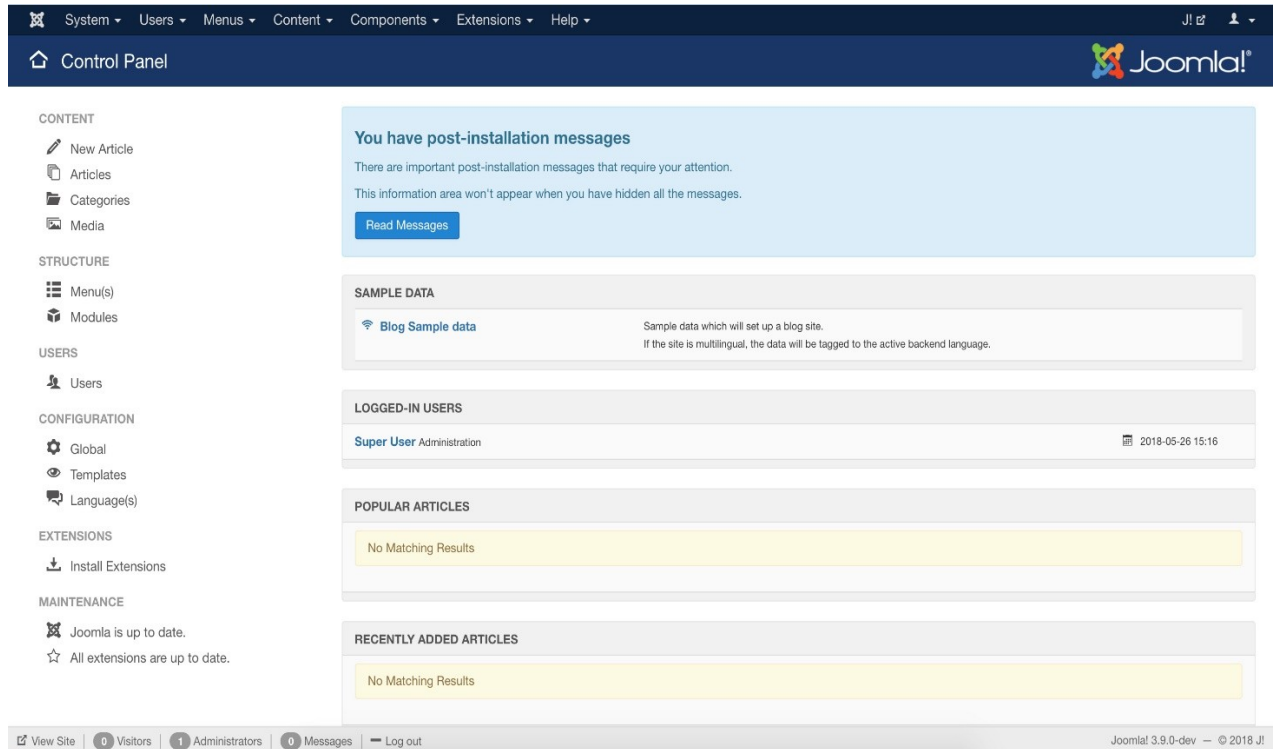


Рисунок 5 – Панель управления CMS Joomla

В 2005 году несколько независимых разработчиков CMS Mambo приняли решение выйти из состава общей группы разработчиков и создать на базе Mambo свою собственную CMS. Так в сентябре 2005 года на свет появилась первая версия системы управления сайтом Joomla! (Джумла). Своим названием Joomla! обязана языку суахили, на котором слово Jumla обозначает: мы все вместе, всем миром или единое целое. Само же название прошло горнило отборов и конкурсов, и только после того как ряд профессиональных фирм в области брендинга и маркетинга дали ему хорошую оценку, эта система управления сайтом получила столь изысканное наименование.

Команда Joomla! совершенствуется с момента своего создания, чтобы лучше удовлетворять потребности пользователей.

CMS Joomla! написана на языке программирования PHP, что делает ее весьма универсальной – ее можно установить как на домашнем компьютере (конечно, для этого нужно будет установить также Web-сервер Apache), так и на практически

любом хостинге. В качестве хранилища данных эта CMS использует сервер баз данных MySQL, который тоже является стандартом дефакто на серверах хостинг-провайдеров.

Общие характеристики:

1) панель управления сайтом, по умолчанию доступна только на английском языке, остальные языки необходимо добавлять вручную;

2) возможность создания, удаления, редактирования страниц, разделов сайта, включаемых блоков;

3) редактируемое меню;

4) WYSIWYG-редактор страниц сайта;

5) распределение прав пользователей на доступ к данным сайта;

6) авторизация посредством LDAP, OpenID, а так же Gmail, что позволяет пользователям, используя имеющиеся данные аккаунта, ускорить процесс регистрации;

7) форум/опросы/голосования для эффективной обратной связи;

8) интеграция с форумом;

9) загрузка изображений при помощи браузера в вашу собственную библиотеку - для последующего использования с любого места сайта;

10) шаблоны сайта;

11) наличие расширений;

12) с версии 3.0 упрощенная установка в 3 шага (ранее 7 шагов);

13) возможность создавать адаптивные сайты;

14) организация рассылок на новости, статьи и т.п.;

15) модуль интеграции с 1С - платный либо дописывать самостоятельно;

16) наличие руководства по установке на русском языке;

17) организация вывода информации в соответствии с приватностью (на страницы можно давать ограниченный уровень доступа);

18) поиск на сайте. Пакет Joomla поставляется с английским морфологическим поиском Snowball Stemmer «Расширения → Smart Search → Опции» (Extensions → Smart Search → Options). Snowball требует расширения Stem PHP и поддерживает 14 языков, включая датский, немецкий, английский, испанский, финский, французский, венгерский, итальянский, норвежский, голландский, португальский, румынский, русский и турецкий. Данные должны быть проиндексированы для того, чтобы получить эту гибкость и скорость. Например, нужно определить строку со словом «коты». Поиск выдаст строки, в которых есть однокоренные слова («котики», «котята» и т.д.) на основе корня «кот». Алгоритм морфологического поиска сводится к поиску однокоренных слов: «рыбалка», «рыболов» – корень слова «рыба».

The screenshot displays the Joomla! 'Articles: Edit' interface. At the top, there's a navigation menu with 'System', 'Users', 'Menu', 'Content', 'Components', 'Extensions', and 'Help'. Below it, the article title is 'What is Joomla?' and the alias is 'what-is-joomla'. The main content area contains HTML code for an article, including a paragraph about Joomla's popularity, a PayPal payment form, and a 'Read More' button. The right sidebar contains metadata fields: Status (Published), Category (News), Featured (Yes), Access (Public), Language (All), Tags, and Version Note. The bottom status bar shows 1 Visitor, 1 Administrator, 0 Messages, and Joomla! 3.8.3.

Рисунок 6 – Редактирование записи в CMS Joomla

Основной минус системы – она сложна для доработок программистами. В большинстве своем программисты не очень лестно отзываються о данной системе, модули не все на русском языке и, кроме того, необходимо дописывать много дополнительного кода в случае, если стандартных возможностей системы недостаточно. Данная система уязвима главным образом благодаря своей популярности. Даже набрав в интернете «как взломать Joomla», можно найти крайне много информации о том, как это сделать.

Самый важный плюс систем – ее бесплатность. Код системы открыт, что является как плюсом, так и минусом одновременно.

Но если не требуется никаких доработок и достаточно стандартного функционала, то данная система вполне подходит!

Большинство западных хостинг-провайдеров давно предоставляют Joomla! в полупредустановленном виде. То есть CMS уже загружена на хостинг, и вам нужно лишь инициировать установку системы на виртуальную площадку. Никаких навыков программирования при этом не потребуется.

Система управления сайтом Joomla! имеет несколько релизов, которые исчисляются простыми цифровыми обозначениями.

Вся эта система построена на библиотеках, которые можно использовать повторно и которые могут расширять ядро и сторонние компоненты. Можно создавать и добавлять свои собственные библиотеки и наборы API для расширения.

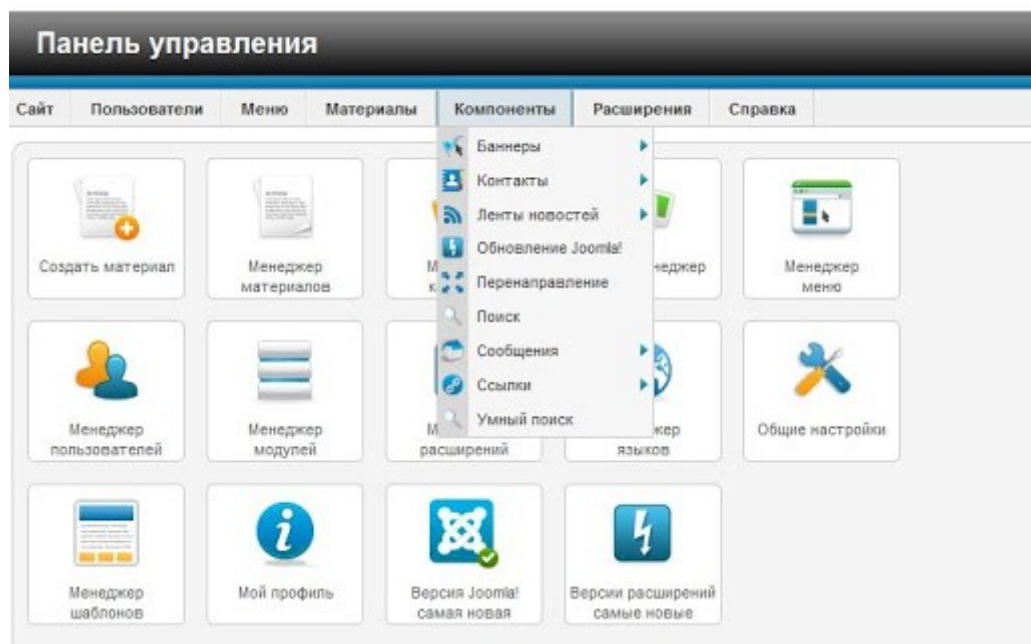


Рисунок 7 – Компоненты в CMS Joomla

Кроме официальных релизов Joomla!, в мире существуют еще так называемые сборки этой CMS, которые могут носить несколько видоизмененные названия. Например, имеются дистрибутивы Joostina, Joomla Lavra Edition, JoomFish или JoomlaTune. Эти сборки, или версии CMS построены на ядре Joomla!, но частично переработаны сторонними разработчиками.

Главное достоинство Joomla — гибкость, которая достигается за счёт открытости исходного кода и большого количества плагинов для расширения базовой функциональности. Бесплатное распространение движка сделало его столь популярным среди веб-мастеров и помогло сформировать пользовательское сообщество, которое в том числе занимается созданием дополнительных материалов: шаблонов, плагинов, инструкций

Один из недостатков Джумлы — высокий процент взломанных сайтов. Правда, связано это в первую очередь с популярностью системы. Движок часто используют начинающие веб-мастера, которые не сразу разбираются с настройками безопасности. На коммерческих системах угроз меньше из-за закрытого ядра, Джумла же для любителей несанкционированного доступа — это открытая книга. Кроме того, уязвимости часто скрываются в расширениях, поэтому к их выбору нужно подходить с большим вниманием.

Проблемы могут возникнуть и при обновлении системы. Главная сложность, это несовместимость версий движка и плагинов. Особенно это касается бесплатных дополнений, разработчики которых не всегда успевают вовремя выпустить обновления или вовсе прекращают поддержку продукта. Из-за этого появляются сбои в работе. Поэтому перед обновлением движка рекомендуется делать резервную копию и отключать все плагины, а затем включать их по очереди, проверяя работоспособность.

Недовольны Джумлой и многие профессиональные разработчики, которые считают, что в структуре CMS есть недоработки, усложняющие её кастомизацию, а излишний код в шаблонах и самом движке приводят к медленной загрузке страниц. Впрочем, на популярность Joomla эти недостатки если и

влиают, то минимально — систему продолжают применять для реализации самых разных идей.

2) CMS Wordpress

CMS WordPress появилась на основе идеи создания удобной, хорошо структурированной персональной системы публикаций, основанной на языке программирования PHP, использующей для хранения информации СУБД MySQL. Первый релиз Wordpress состоялся в сентябре 2003 года.

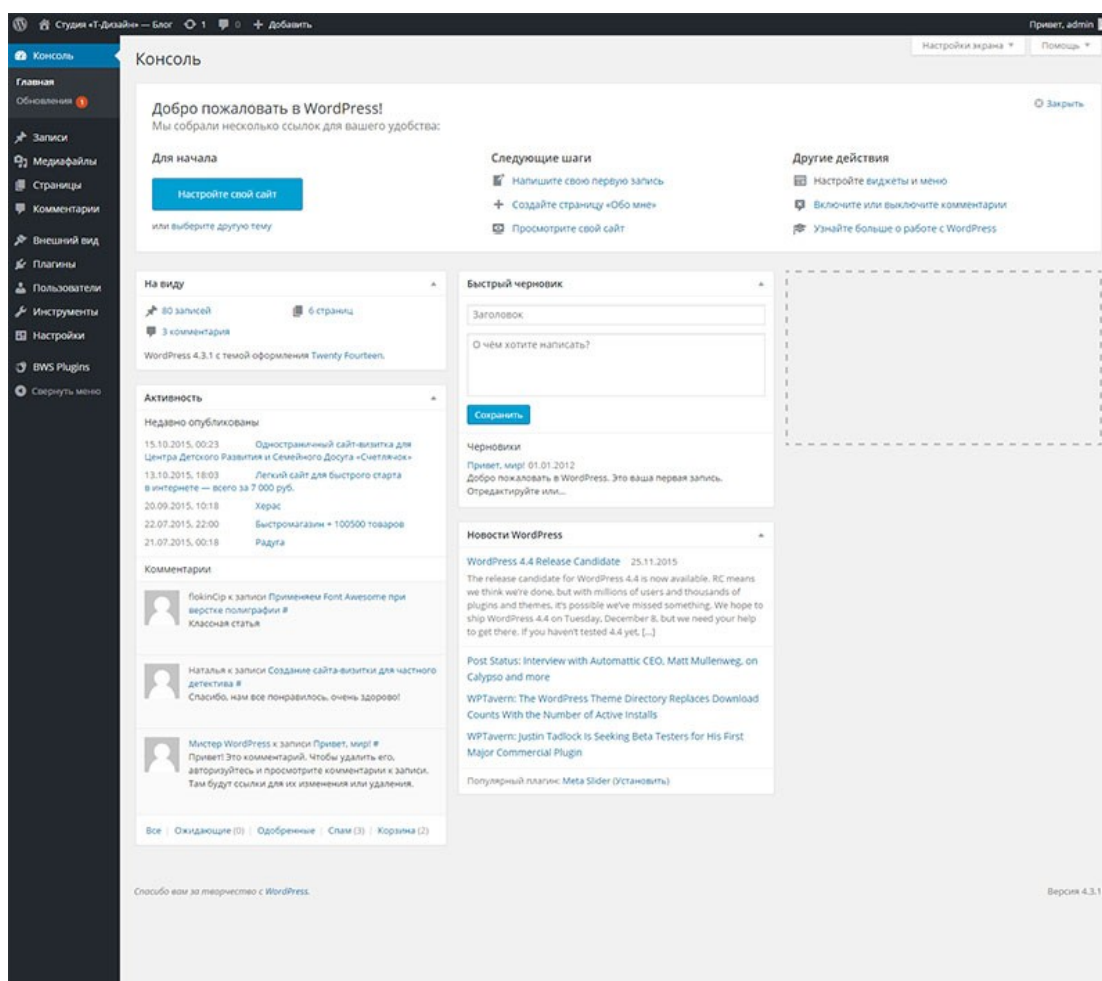


Рисунок 8 – Панель управления CMS Wordpress

Официальный потомок – b2, разработанный Мишелем Валдриги. Несмотря на то что это современное программное обеспечение, корни его развития начинаются с 2001 года. В 2002 году к проекту разработки b2 присоединились Мэтт Муленвег и Майк Литтл. Создателем WordPress считается Мэтт Муленвег,

который сменил имя b2 на WordPress и получил права на торговую марку.

В дальнейшем улучшенные версии WordPress выходили практически ежегодно. В 2005 году, когда Мэтт Муленвег основал собственную компанию Automattic и целиком посвятил себя разработке проектов с WordPress, появились Strayhorn (версия 1.5 – управление страницами и шаблонами) и Duke (2.0 – новая панель администратора и загрузка изображения).

Особенности WordPress:

- Простота установки и первоначальной настройки;
- Возможность регистрации посетителей сайта, а также поддержка многопользовательского режима, при котором каждый зарегистрированный пользователь имеет возможность вести свой собственный блог на сайте;
- Возможность комментирования материалов сайта посетителями, а для администратора сайта – возможность администрирования комментариев, фильтрации «спама». В стандартную поставку WordPress не входит поддержка собственного форума, однако при необходимости вы можете легко установить дополнительный модуль, который позволит добавить на сайт форум и настроить его;
- Поддержка медиа-форматов (аудио-, видеоизображений), возможность загрузки их на сайт и корректного отображения сайта на страницах;
- Организация почтовой подписки для посетителей сайта – любой заинтересовавшийся содержанием вашего сайта может оставить свой e-mail и получать на него новые материалы;
- Поддержка расширений стандартного функционала в виде дополнительных модулей (плагинов). Наличие постоянно пополняемой библиотеки модулей позволяет решить практически любую задачу разработчика сайта на WordPress. А

если решения все-таки не найдено, то можно доработать существующий модуль или написать свой собственный;

- Поддержка дизайн-тем, позволяющих легко менять как внешний вид сайта, так и способы вывода данных. Для WordPress разработано множество дизайн-решений, как платных, так и бесплатных, благодаря чему дизайн сайта будет практически уникальным;

- Можно сэкономить трафик, включив режим gzip в настройках WordPress. Wordpress автоматически упаковывает содержимое страниц для отправки с помощью gzip, если браузер посетителя сайта поддерживает эту функцию. Для работы функции gzip необходимо включить модуль mod_gzip в настройках вашего веб-сервера Apache;

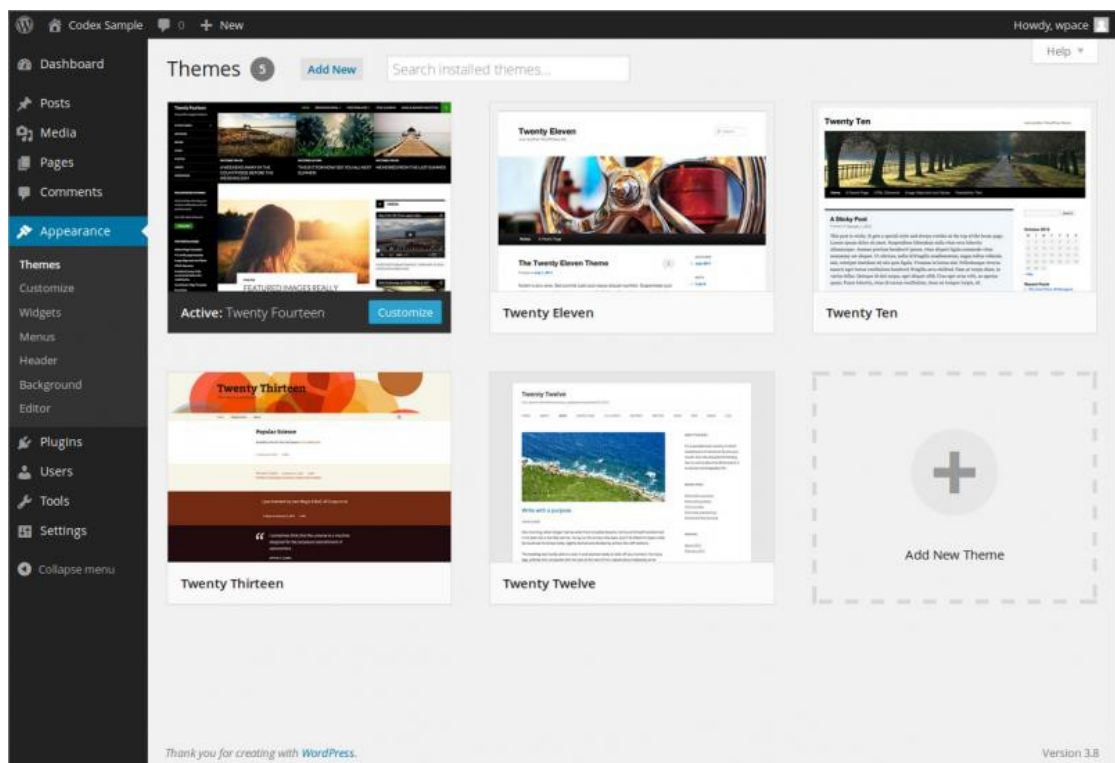


Рисунок 9 – Выбор шаблона темы в CMS Wordpress

Требования к платформе. Требуются PHP и MySQL.

WordPress предназначен для установки на вашем собственном Web-сервере или на сервере вашего хостера, который предоставляет вам полный контроль над вашим блогом.

Основное меню панели управления состоит из следующих функциональных разделов:

- «Консоль» – главная страница, на которой собирается общая информация об обновлениях, новостях, комментариях;
- «Записи» – все размещённые на сайте материалы с возможностью их редактирования, добавления новых и удаления ненужных;
- «Медиафайлы» – менеджер изображений, видеозаписей, таблиц и файлов другого типа, загруженных на хостинг;
- «Страницы» – раздел для управления статичной информацией – например, страницей «О компании», условиях использования или размещения рекламы. Предлагает ту же функциональность, что и «Записи», только опубликованный контент не попадает в ленту блога;
- «Комментарии» – все обсуждения на сайте и инструменты модерации;
- «Внешний вид» – средства добавления и настройки тем оформления, в том числе встроенный редактор файлов шаблонов;
- «Плагины» – раздел для добавления, удаления и настройки расширений;
- «Пользователи» – управление всеми зарегистрированными посетителями, регулировка прав доступа;
- «Инструменты» – служебные возможности WordPress, например, импорт и экспорт;
- «Настройки» – основные параметры работы сайта: название, язык, почта, формат файлов, правила формирования адресов и т.д;

С выходом версии 5.2.4 необходимости в правке кода стало меньше благодаря появлению блочного редактора Gutenberg. После создания и предварительной настройки сайта он станет

для вас основным инструментом, с помощью которого вы будете создавать страницы и записи. Классический редактор предлагал ограниченную функциональность для управления контентом. Для добавления дополнительных возможностей пользователям приходилось устанавливать плагины, что не всегда удобно, в том числе из-за увеличения нагрузки на сервер. Редактор Gutenberg устраняет эту проблему, предлагая функциональные блоки с контентом:

- заголовок, абзац, цитату, список;
- изображение, галерею, аудио, видео и файл;
- код и элементы разметки;
- виджеты и вставки с популярных сервисов.

По сути, Gutenberg позволяет пользователям WordPress делать то, что давно предлагают конструкторы сайтов – сосредоточиться на том, как будет выглядеть контент, не отвлекаясь на сложные настройки конфигурации. В режиме визуального редактирования вы сразу оцениваете, как будет отображаться страница или запись, и можете вносить изменения в интерфейс, не прибегая к правке кода.

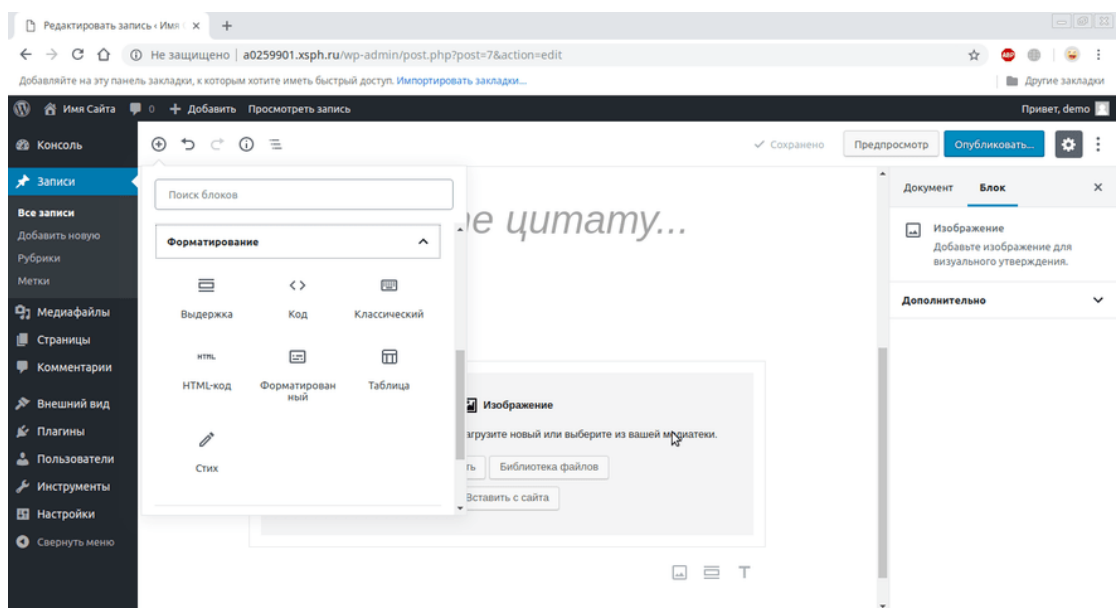


Рисунок 10 – Редактор текста Gutenberg в CMS Wordpress

WordPress — отличный пример того, как пользовательское сообщество может способствовать развитию продукта. Некогда блогový движок благодаря появлению большого количества плагинов превратился в универсальную CMS, которая легко устанавливается и предлагает огромный потенциал.

Среди основных достоинств отметим:

- Удобная панель управления, на освоение которой уйдёт не больше часа;
- Редактор Gutenberg, предоставляющий новые возможности для изменения интерфейса сайта и управления записями в визуальном режиме;
- Огромное количество шаблонов и расширений, делающих WP универсальной системой;
- Мощная база знаний и развитое сообщество пользователей;
- Постоянные обновления и появление дополнительных материалов движка.

WordPress не идеален. Созданные на нём сайты часто подвергаются взлому, но это не проблемы движка, а следствие его распространённости: им пользуются миллионы веб-мастеров, среди которых многие не озадачиваются глубокой проработкой вопросов безопасности.

Ещё один потенциальный недостаток — снижение производительности из-за использования большого количества плагинов. С этим сталкиваются все начинающие веб-мастера. WordPress предлагает миллион возможностей, всё хочется попробовать, но в какой-то момент дополнения начинают слишком грузить сервер. Проблема решается переносом сайта на более мощную и дорогую инфраструктуру или отказом от плагинов — в конце концов, открытый исходный код WordPress позволяет реализовывать на нём отдельные функции без

установки дополнений. Правда, для этого потребуются навыки программирования или бюджет для привлечения специалистов.

Иногда проблемы создают обновления системы. Ни в коем случае нельзя ставить их в автоматическом режиме — только вручную, после создания резервной копии. Обновление движка может привести к тому, что бесплатные плагины перестанут работать из-за несоответствия версий. С платными решениями такие сложности обычно не возникают — разве что иногда приходится недолго ждать их обновлений.

3) CMS 1С-Битрикс

Система «1С-Битрикс: Управление сайтом» универсальна. Она может успешно работать как на сайтах-визитках (самый нижний класс в «табели о рангах» современных сайтов), так и на больших производительных порталах с большим числом посетителей, выдерживая нагрузку до 30–40 тысяч посетителей в час. Разный уровень потребностей пользователей реализуется разными редакциями: от самой простой «Старт» до самой совершенной «Энтерпрайз».

На рисунке 11 представлена часть сравнительной таблицы функционалов разных редакций 1С-Битрикс:Управление сайтом.

	Старт	Стандарт	Малый бизнес	Бизнес	Энтерпрайз
	4 320 руб. <small>5-400 руб.</small>	12 720 руб. <small>15-900 руб.</small>	28 720 руб. <small>35-900 руб.</small>	58 320 руб. <small>72-900 руб.</small>	от 319 200 руб. <small>от 399-000 руб.</small>
	Купить	Купить	Купить	Купить	Купить
	Заказать разработку	Заказать разработку	Заказать разработку	Заказать разработку	Заказать разработку
CMS					
Главный модуль	✓	✓	✓	✓	✓
Сайты24	✓	✓	✓	✓	✓
Число сайтов	2	∞*	∞*	∞*	∞*
Число страниц	∞	∞	∞	∞	∞
Управление структурой	✓	✓	✓	✓	✓
Информационные блоки	✓	✓	✓	✓	✓
Highload-блоки	✓	✓	✓	✓	✓
Поиск	✓	✓	✓	✓	✓
Перевод	✓	✓	✓	✓	✓
Форумы и Блоги		✓	✓	✓	✓
Фотогалерея 2.0		✓	✓	✓	✓
Социальная сеть				✓	✓
Обучение, тестирование				✓	✓
Веб-мессенджер				✓	✓
Бизнес-процессы				✓	✓
Почта				✓	✓
Техподдержка				✓	✓
Календари				✓	✓

Рисунок 11 – Фрагмент таблицы сравнения функционалов в CMS 1С-Битрикс

Рассмотрим основные функции и преимущества.

1) Главный модуль.

Обеспечивает общее функционирование системы, взаимодействие всех модулей продукта и позволяет создавать, поддерживать и управлять неограниченным числом сайтов. Включает в себя: административный раздел, рабочий стол панели управления, управления пользователями и правами доступа, авторизацию, автосохранение форм, русские имена файлов и папок, автоматическое определение часового пояса, короткие ссылки, технологию обновлений SiteUpdate, адаптивный интерфейс, почтовые сообщения, кэширование, защиту от автоматических регистраций и др.

2) Управление структурой.

Предназначен для управления информационным наполнением сайта, разделами, меню и правами доступа. Редактирование страниц осуществляется с помощью встроенного визуального редактора. Управлять структурой можно из административного доступа «Менеджер файлов» или же публично, нажав на кнопку «Структура». Встроенная поисковая оптимизация SEO. Яндекс-карты.

3) Медиабiblioteca.

Это быстрый и удобный менеджер для работы с медиаданными. Целый набор отличных инструментов позволяет быстро и просто создать многоуровневую структуру медиаколлекций. В дальнейшем содержимое этих коллекций постепенно создается при редактировании сайта, причем не только его отдельных страниц, но и информационных блоков. То есть теперь, когда вам нужно добавить изображение на страницу сайта, в новости или в каталог, вам достаточно выбрать его из медиабiblioteca.

4) Многосайтовость.

Означает, что купив одну лицензию, можно поддерживать несколько сайтов.

5) Информационные блоки.

Модуль «Информационные блоки» предназначен для управления различными блоками однородной информации. На базе информационных блоков можно реализовать каталоги товаров, блоки новостей, справочники и т.д.

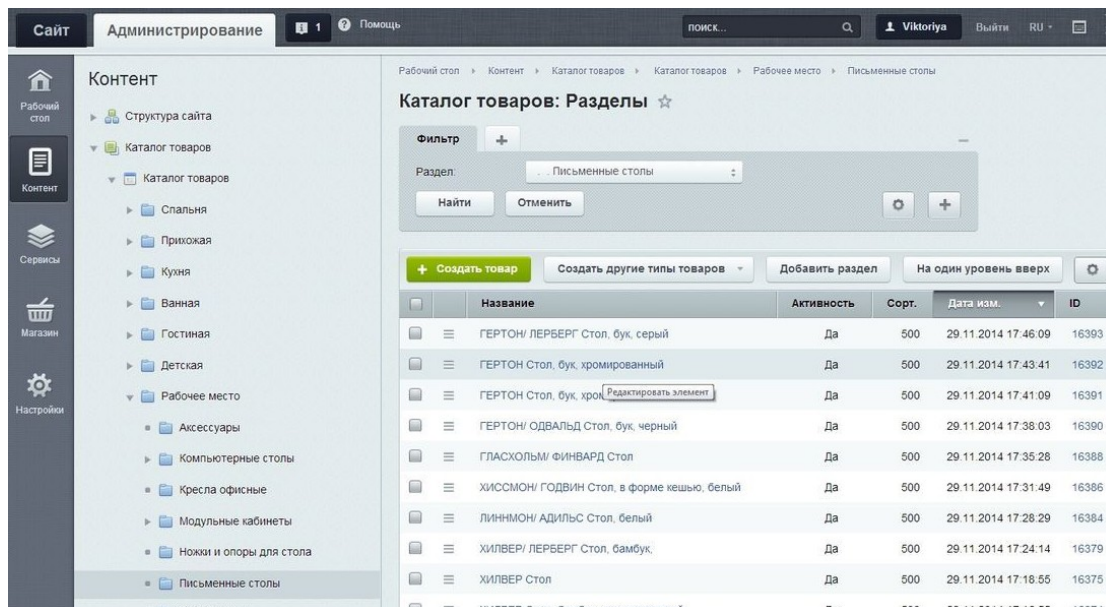


Рисунок 12 – Модуль «Информационные блоки» в CMS 1С-Битрикс.

6) Безопасность.

Высокая степень безопасности. На данный момент не известно случаев, чтобы система была взломана.

7) Форум.

Позволяет создать необходимое количество форумов для организации и поддержки сообществ, обсуждения статей и материалов, формирования постоянной аудитории сайта и достижения других целей.

8) «Интернет-магазин».

Поможет быстро создать свой собственный интернет-магазин и начать продажу товаров и услуг в Интернете. Он уже содержит все основные функциональные элементы, позволяющие начать торговлю в Сети. Это удобная корзина с возможностью добавления туда произвольных товаров и услуг. Это персональный раздел покупателя с возможностью просмотра сделанных заказов и настройкой профилей. Это гибкая процедура создания и оформления заказов. Это мощный административный раздел для настройки параметров магазина и управления заказами клиентов. И многое-многое другое.

9) Поиск.

Осуществляет индексирование и поиск информации на сайте. Работает мгновенный поиск по заголовкам – новостей, блогов, групп, разделов и т.д. Поиск стал социальным – на поиск влияет мнение сообщества – учитывается нажатие кнопки «Мне нравится», что делает результаты поиска информативнее и релевантнее. Существенно облегчен и упрощен поиск нужных документов, ценных с точки зрения сообщества тематических обсуждений, статей, товаров и прочего материала на сайте.

10) Опросы.

Предназначен для проведения онлайн-опросов, голосований и анкетирования посетителей сайта. Модуль может быть использован для сбора информации об аудитории сайта, о потенциальных клиентах компании, для проведения маркетинговых исследований при планировании рекламных кампаний и т.д.

11) Подписка и рассылки.

Предназначен для организации списков рассылки. Посетители смогут подписываться на рассылки новостей сайта по привычной схеме.

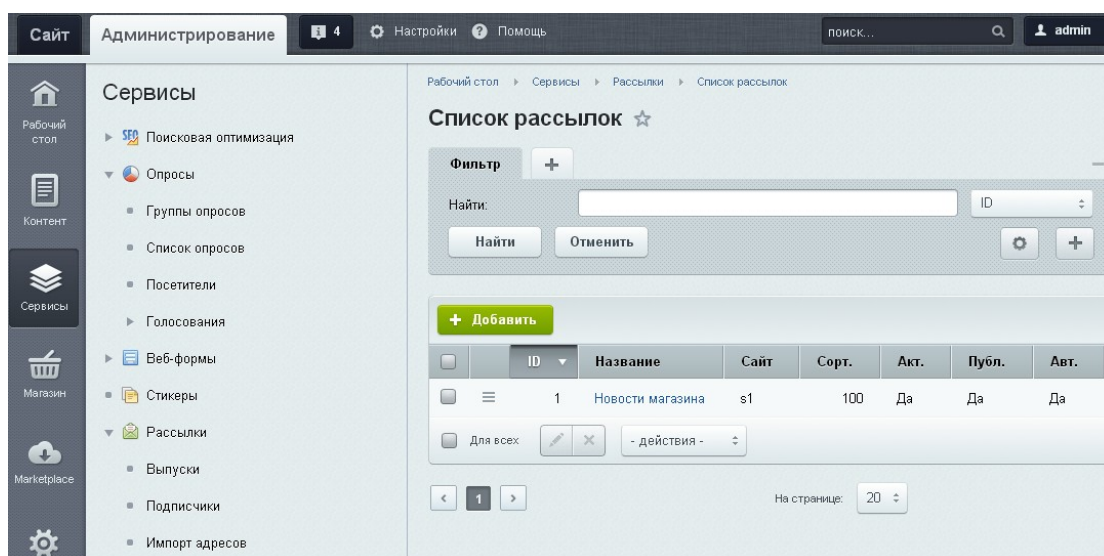


Рисунок 13 – Сервис «Подписка и рассылки» в CMS 1С-Битрикс.

Интеграция с Яндекс.Маркетом. Существует три варианта подключения: выгрузка на Яндекс.Маркет товаров, оплата за клики по товарам, переход в магазин, продажа на Яндекс.Маркете: оформление идет внутри магазина и оплата идет напрямую магазину, продажа на Яндекс.Маркете: продажа идет через Яндекс.Маркет, там же происходит оплата, Яндекс гарантирует надежность и возврат средств. Что для этого необходимо: подписать договор с Яндекс.Маркет, настроить защищенный протокол (https), заполнить форму Яндекс.Маркет в «1С-Битрикс». «1С-Битрикс» полностью интегрирован со всеми сценариями работы Яндекс.Маркета: публикация каталога, продажа, возможность оплаты на Яндекс.Маркете с уведомлением магазина о том, что заказ оплачен, гибкая настройка бизнес-процессов магазина под статусы Яндекс.Маркета, проверка корректности настройки.

И это еще не полный список возможностей и модулей системы 1С- Битрикс.

Из-за требовательности к ресурсам хостинга нужно быть внимательными к возможностям провайдера. Постоянно обновляемый список рекомендуемых провайдеров есть на сайте «1С-Битрикс: Управление сайтом». В то же время система мультиплатформенная и может работать на серверах с ОС, поддерживающей веб-сервер Apache (IIS) с базами данных MySQL, MSSQL, Oracle.

«1С-Битрикс: Управление сайтами» — самодостаточная система, которой можно пользоваться «из коробки». Выбирая лицензию, веб-мастер оценивает возможность её применения для реализации своего проекта. Если редакция предоставляет все необходимые инструменты, то остаётся только установить движок и приступить к настройке сайта. Среди других плюсов CMS:

- визуальный режим редактирования;
- встроенная система разграничения пользовательских прав;
- высокий уровень безопасности;
- интеграция с другими продуктами 1С, очень полезная при организации магазинов и корпоративных сайтов;
- возможность создания собственных модулей для расширения функциональности.

Несмотря на все достоинства, пользователи при работе с 1С-Битрикс обычно сталкиваются с двумя проблемами: высокой стоимостью лицензий и необходимостью изучить HTML и CSS хотя бы на базовом уровне. Больше 5000 рублей за лицензию с минимальными возможностями и несколько десятков тысяч на создание интернет-магазина даже с небольшим ассортиментом товаров — это и правда дорого. Поэтому прежде чем покупать продукты Битрикс, нужно убедиться в том, что без них создать сайт не удастся. Например, для лендинга и небольшого магазина можно без проблем использовать конструкторы — выйдет и дешевле, и проще с точки зрения освоения и управления.

CMS оснащена визуальным редактором, однако для полноценной настройки сайта его возможностей будет мало. Поэтому единственный выход — улучшать навыки программирования, читать документацию. Опять же, при создании небольших проектов это может оказаться неэффективной тратой времени — поэтому обязательно нужно изучить альтернативные инструменты.

«1С-Битрикс: Управление сайтами» позиционируется как универсальная система, на которой можно создавать в том числе персональные сайты, но стоимость лицензий, сложность разработки и функциональность прямо говорят о том, что эффективнее всего потенциал CMS раскрывается при запуске

средних, крупных и огромных проектов. Это действительно самый популярный коммерческий движок в рунете, но это не значит, что он подходит для всех. С созданием лендингов, корпоративных сайтов и небольших магазинов отлично справляются и конструкторы. Они стоят заметно дешевле: за те деньги, которые просят на редакцию с функциональностью для магазина, на конструкторе можно работать несколько лет, получая все обновления. На 1С же через год придётся заплатить 60% от первоначальной стоимости, чтобы версия движка не устаревала.

Чтобы понять, какая CMS больше подойдет тому или иному проекту, нужно поставить ряд вопросов и ответить на них.

1. Имеющийся бюджет?
2. Какие необходимы функциональные модули?
3. Нужна ли дальнейшая поддержка ядра (обновление системы) данного сайта?
4. Сайт будет поддерживаться собственными силами или нет?
5. Нужна ли надежная защита сайта?

По первому пункту бесплатные системы все-таки условно бесплатны. Так как бесплатен основной функционал. После ответа на второй вопрос станет понятно, нужно ли все-таки вкладывать силы, время и деньги в дополнительные возможности системы или нет. То же самое касается и платных систем, ведь от того, какие функциональные модули необходимы, зависит стоимость продукта.

К вопросу поддержки ядра данного сайта, безусловно, выигрывают платные системы. Разработчики платных систем постоянно борются с существующими ошибками, вносят дополнительные возможности в систему, несут ответственность за свой продукт. Если что-то случится с сайтом, они просто

обязаны в рамках поддержки ответить на поставленные вопросы. Минус только один, если постоянные обновления необходимы, раз в год нужно будет продлевать лицензию. Если система устраивает и так, то покупать продление совершенно не обязательно.

Вопрос о том, как будет поддерживаться сайт, очень важен. Если самостоятельно, тогда в штате должен быть человек, который может работать с системой. Если же таких условий нет, нужно понимать, где будет осуществляться поиск подрядчика на работы. Для платных систем существует множество авторизованных партнеров, таким образом, нет привязки к конкретному разработчику. Конечно, веб-студий работающих с Joomla!, Wordpress и др. бесплатных CMS не меньше, но на основном сайте каждой из этих систем нет четкого списка фирм, в которые можно обратиться за поддержкой.

По вопросам надежности: утверждать, что бесплатные системы слишком легко взламываются, некорректно, но в платных системах над этим вопросом работают постоянно. Вопросы защиты платных систем стоят остро! И в случаях взлома разработчики обязательно устранят уязвимые места. Чего нельзя четко утверждать в бесплатных системах.

По вопросам документирования: любая система, будь она платной или бесплатной, имеет документацию. Другое дело, что в платных системах большое внимание уделяется документированию, поскольку это один их маркетинговых ходов. Но и бесплатные системы без документирования, можно сказать, хорошо описаны в форумах и блогах, ведь на бесплатных системах работает множество разработчиков, причем из разных стран мира.

Как видно из сравнения, ярких минусов ни у платных ни у бесплатных систем нет, выбирается та или иная система

индивидуально для конкретного заказчика. Ну а в случае, если подойдет стандартный функционал какой-либо бесплатной системы и не нужны обновления и поддержка, то выбор очевиден.

1.2 Анализ современных средств и технологий разработки CMS

При разработке проекта перед веб-разработчиками нередко встает выбор между готовой CMS и фреймворком, на котором будет написана индивидуальная CMS.

Фреймворки — это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений.

С точки зрения бизнеса разработка на фреймворке почти всегда экономически эффективнее и качественнее по результату, нежели написание проекта на чистом языке программирования без использования каких-либо платформ. Разработка без использования платформы может быть правильным решением только в двух случаях — либо проект совсем простой и не требующий дальнейшего развития, либо очень нагруженный и требует очень низкоуровневой оптимизации (например, веб-сервисы с десятками тысяч обращений в секунду). Во всех других случаях разработка на программной платформе быстрее и качественнее.

Если сравнивать фреймворки с другими классами платформ — SaaS, CMS или CMF — то фреймворки значительно эффективнее использовать в проектах со сложной бизнес-логикой и с высокими требованиями к скорости работы, надёжности и безопасности. Но в простых и типовых проектах без значимых требований скорость и стоимость разработки на фреймворке будет выше, нежели на SaaS или готовой CMS.

Многие известные системы управления сайтом написаны на фреймворках. Так, например, CMS 1С-Битрикс использует PHP фреймворк Bitrix Framework. Довольно новая система управления сайтом October CMS написана на популярном PHP фреймворке Laravel.

В зависимости от целей, систему управления сайтом можно разработать, используя в качестве бэкенда почти любой веб-фреймворк. Рассмотрим наиболее популярные современные веб-фреймворки, которые можно использовать в качестве бэкенда при разработке собственной системы управления контентом, а также фронтенд фреймворки для разработки проекта без перезагрузки страниц в браузере (Single Page Application):

- Laravel (PHP);
- Django (Python);
- Ruby On Rails (Ruby);
- Spring (Java);
- React, Angular, Vue.js (JavaScript).

1) Laravel

Фреймворков в настоящее время очень много — десятки, а может быть, и сотни. Самым же популярным продуктом такого рода является Laravel. Laravel - это бесплатный PHP фреймворк с открытым исходным кодом, созданный Тейлором Отвеллом для разработки веб-приложений по архитектурному шаблону MVC.

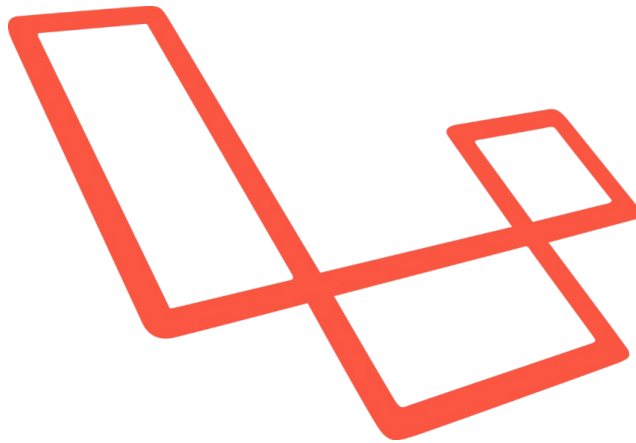


Рисунок 14 - Фреймворк Laravel

Фреймворк Laravel очень популярен среди западных разработчиков веб-приложений. Удобная система миграций помогает упростить развертывание и обновление веб-приложения. В Laravel есть много полезных функций, позволяющих сделать процесс разработки веб-приложений быстрым, простым и качественным.

Рассмотрим основные преимущества и недостатки Laravel:

- Laravel написан на PHP — самом популярном языке программирования динамических сайтов;
- Laravel не имеет каких-либо специфических системных требований и работает на мощностях практически всех присутствующих на рынке хостинг-провайдеров;
- он предоставляет практически всю базовую функциональность сайтов, включая мощную и полностью готовую к работе систему разграничения доступа, средства для отправки электронной почты и исключительно удобный механизм миграций;
- все входящие в него инструменты уже сконфигурированы наилучшим для большинства случаев образом;
- существует много дополнительных библиотек, написанных сторонними разработчиками и расширяющих функциональность Laravel;

- почти все необходимое можно реализовать небольшим количеством программного кода.

К недостаткам можно отнести следующее:

- для русскоязычных разработчиков, без знания английского языка или его слабым знанием, к минусам фреймворка можно отнести довольно небольшое количество статей, примеров кода, переводов официальной документации. Для тех, кто знает английский на уровне чтения технической документации, данный минус можно опустить;

- синтаксический сахар в Laravel как плюс, так может быть и минусом. Очень легко привыкнуть к нему и позабыть, как пишутся чистые запросы и функции;

- нарушение обратная совместимости между версиями фреймворка.

2) Django

Django считается лучшим веб-фреймворком, написанным на Python. Этот инструмент удобно использовать для разработки сайтов, работающих с базами данных.



Рисунок 15 – Фреймворк Django

Django создали разработчики издания Lawrence-Journal World. Этой газете понадобился сайт, чтобы публиковать новости в интернете. Программисты Эдриан Головатый и Саймон Виллисон создали веб-приложение и сделали его публичным.

Вокруг Django быстро сформировалось активное сообщество. Фреймворк стал стремительно развиваться усилиями волонтеров. Значительную роль в успехе Django сыграли несколько известных

сайтов, которые использовали этот фреймворк. В их число входят Pinterest, Dropbox, Spotify, сайт The Washington Post. В настоящее время сообщество Django включает более 11 тыс. разработчиков из 166 стран мира.

Неопытные специалисты считают Django одной из многих систем управления контентом (CMS). На самом деле это программный инструмент, с помощью которого можно создавать и запускать веб-приложения.

Фреймворк Django справляется с большим количеством задач и повышенными нагрузками. Его применяют для создания:

- CRM-систем;
- CMS;
- Коммуникационных платформ;
- Сервисов бронирования номеров;
- Платформ управления документооборотом.
- У Django есть недостатки, как у любого инструмента.
- Не подходит для небольших проектов;
- Поведением некоторых компонентов Django не всегда просто управлять;
- Некоторые модули Django, например, ORM или формы, сложно заменить.

3) Ruby On Rails

«Ruby on Rails» (или просто «Rails») – это фреймворк для разработки веб-приложений на языке программирования Ruby. Со времен своего дебюта в 2004 году Ruby on Rails довольно быстро стал одним из самых мощных и популярных инструментов создания динамических веб-приложений. Rails используется множеством различных компаний: Airbnb, Basecamp, Disney, GitHub, Hulu, Kickstarter, Shopify, Twitter и Yellow Pages. Помимо этого, существует множество компаний, занимающихся разработкой веб-приложений и специализирующихся на Rails,

таких как ENTP, thoughtbot, Pivotal Labs, Hashrocket и HappyFunCorp, плюс бесчисленное множество независимых консультантов, преподавателей и индивидуальных разработчиков.



Рисунок 16 - Фреймворк Rails

Во-первых, Ruby on Rails - это открытый исходный код, доступный на условиях лицензии MIT, и, как следствие, его можно загружать и использовать совершенно бесплатно. Rails также обязан своим успехом изящному и компактному дизайну. Используя податливость языка Ruby, лежащего в его основе, Rails фактически определяет предметно-ориентированный язык для разработки веб-приложений. В результате множество часто встречающихся задач веб-программирования, таких как динамическое создание разметки HTML, определение моделей данных и маршрутизация URL, легко решаются в Rails, а конечный код приложений получается кратким и читаемым.

Rails также быстро адаптируется к новым тенденциям в веб-технологиях. Например, Rails одним из первых полностью реализовал архитектурный стиль REST структурирования веб-приложений. И когда в других фреймворках появляются новые успешные приемы, создатель Rails, Дэвид Хейнмейер Ханссон, и рабочая группа Rails не стесняются использовать чужой опыт. Пожалуй, наиболее драматичным примером является слияние Rails с конкурирующим веб-фреймворком Merb, благодаря

которому Rails получил модульный дизайн Merb, стабильный API и улучшенную производительность.

Наконец, вокруг Rails сплотилось необычайно увлеченное и разнообразное сообщество: сотни разработчиков, представительные конференции, огромное количество гемов (пакетов, законченных решений конкретных задач, таких как постраничный вывод и выгрузка изображений), богатый набор информативных блогов и рог изобилия форумов и IRC-каналов. Большое количество Rails-программистов также облегчает работу с (неизбежными) ошибками, возникающими в процессе разработки: алгоритм «Ищи в Google сообщение об ошибке» практически всегда помогает найти подходящую статью в блоге или сообщение на форуме.

4) Spring

Spring – это свободно распространяемый фреймворк, созданный Родом Джонсоном (Rod Johnson) и описанный в его книге «Expert One-on-One: J2EE Design and Development». Он был создан с целью устранить сложности разработки корпоративных приложений и сделать возможным использование простых компонентов JavaBean для достижения всего того, что ранее было возможным только с использованием EJB. Однако область применения Spring не ограничивается разработкой программных компонентов, выполняющихся на стороне сервера. Любое Java-приложение может использовать преимущества фреймворка в плане простоты, тестируемости и слабой связанности.



Рисунок 17 – Фреймворк Spring

Фреймворк Spring обладает весьма широкими возможностями. Но в основе практически всех его особенностей лежат несколько фундаментальных идей, направленных на достижение главной цели – упрощение разработки приложений на языке Java.

Разработчики многих фреймворков утверждают, что их продукты упрощают те или иные аспекты разработки. Но целью фреймворка Spring является упрощение разработки приложений Java вообще. Это требует некоторых пояснений. Так как же фреймворк Spring упрощает разработку на языке Java?

В своем устремлении на сложности, связанные с разработкой на языке Java, фреймворк Spring использует четыре ключевые стратегии:

- легковесность и ненасильственность благодаря применению простых Java-объектов (POJO);
- слабое связывание посредством внедрения зависимостей и ориентированности на интерфейсы;
- декларативное программирование через аспекты и общепринятые соглашения;
- уменьшение объема типового кода через аспекты и шаблоны;
- Практически все возможности фреймворка Spring уходят корнями в эти стратегии.

5) Angular

Angular представляет фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время

Angular это не новая версия AngularJS, а принципиально новый фреймворк.



Рисунок 18 – Фреймворк Angular

Angular заимствует некоторые лучшие аспекты разработки на стороне сервера и использует их для расширения возможностей разметки HTML в браузере. Таким образом закладывается основа, которая упрощает и облегчает создание приложений с расширенной функциональностью. Приложения Angular строятся на базе паттерна проектирования MVC («модель — представление — контроллер»), который ориентирован на создание приложений, обладающих следующими характеристиками:

- простота расширения: если разработчик понимает основы, ему будет легко разобраться даже в самом сложном приложении Angular, а это означает, что он сможет легко расширять приложения для поддержки новой полезной функциональности;

- удобство сопровождения: приложения Angular просты в отладке, в них легко исправляются ошибки, а это означает простоту сопровождения кода в долгосрочной перспективе;

- удобство тестирования: в Angular реализована хорошая поддержка модульного и сквозного тестирования. Следовательно, разработчик сможет находить и устранять

дефекты до того, как пользователи приложения столкнутся с ними;

- стандартизация: Angular работает на основе внутренней функциональности браузера, не создавая никаких препятствий для разработки. Это позволяет создавать веб-приложения, соответствующие стандартам, в которых задействована новейшая функциональность (например, различные API HTML5), популярные инструменты и фреймворки.

Angular — библиотека JavaScript с открытым кодом, финансированием разработки и сопровождения которой занимается компания Google. Она использовалась в ряде крупнейших и сложнейших веб-приложений.

6) Vue.js

Vue.js или просто Vue - это прогрессивный JavaScript-фреймворк, который используется для создания пользовательских интерфейсов.

Vue.js - это фреймворк, который используется для решения задач именно уровня представления, его просто интегрировать с другими библиотеками и проектами. Иными словами, Vue.js - это инструмент, который можно внедрять постепенно. Возможность внедрения связана с тем, что Vue.js стремится к прогрессивности: поддержку Vue можно добавить в уже существующий проект, благодаря чему его функциональность будет значительно расширена. И это отличает его от других фреймворков. Вообще простота интеграции - это одно из основных преимуществ данного фреймворка, особенно в сочетании с возможностью интеграции с бэкенд фреймворками. Другие сильные стороны Vue.js - это простота в освоении и хорошая документация, а также высокая производительность. Также Vue.js можно использовать для создания одностраничных приложений.

Работа над фреймворком Vue.js началась в 2013 году сотрудником Google Эваном Ю (Evan You). Используя AngularJS в работе над несколькими проектами, Эван решил, что можно взять часть этого фреймворка и создать что-то более легкое и простое в использовании.



Рисунок 19 – Фреймворк Vue.js

Vue реактивен, производителен и прост в освоении. Фреймворк не зависит от jQuery. Компоненты Vue легко использовать в уже существующем проекте.

Vue имеет хорошую документацию, касающуюся в том числе [API](#).

И, конечно, не стоит забывать про лицензию MIT, которую имеет Vue.js – значит, фреймворк можно использовать где и когда угодно.

Более того, в том, как работать с ним, может быстро разобраться даже тот, кто никогда до этого не имел дела с JS-фреймворками.

Фреймворк используется многими китайскими компаниями, в частности такими известными, как Alibaba, Baidu и Xiaomi. Связано это, скорее всего, с тем, что Эван Ю активно взаимодействует с китайским JavaScript-сообществом.

По словам Эвана Ю, Vue.js стал популярен, во-первых, после того, как его начало использовать сообщество Laravel, а во-

вторых, когда разработчики поняли, что могут использовать Vue.js вместо React (у этого фреймворка, разрабатываемого компанией Facebook, возникали проблемы с лицензией).

Фреймворк Vue.js многие разработчики рассматривают как один из самых перспективных на данный момент фреймворков, т.к. он гибок, быстро работает и активно развивается.

Таким образом, можно сделать вывод, что для небольших проектов, выгоднее и намного быстрее развернуть готовую CMS. Для высоконагруженных проектов, где, например, каждые сутки обращаются миллионы пользователей или же, например, магазин, где больше несколько сотен тысяч товаров, стоит разрабатывать индивидуальную систему, рассчитаную на большие нагрузки и готовую к масштабированию. Но нет смысла разрабатывать систему на чистом языке программирования, когда существует огромное количество различных фреймворков – инструментов для разработчика и каркас для приложения. Каждый год разрабатываются новые фреймворки, которые имеют ряд преимуществ и вытесняют имеющиеся популярные. Многие фреймворки разрабатываются при поддержке таких гигантов как Google (Angular) или, например, Facebook (React), другие же создаются энтузиастами и затем спонсируются крупными корпорациями.

Каждый год устраиваются конференции, где презентуются новые версии фреймворков, а сообщества набирают численность в сотни тысяч разработчиков.

2 Разработка CMS для интернет-маркетинговой организации

2.1 Технологии, используемые в разработке проекта

Для расширения сферы услуг интернет-маркетинговой организации потребовалась разработка собственной небольшой CMS-системы. Рассмотрим основные требования перед началом разработки будущей системы управления контентом сайта:

- простота в использовании. Необходимо добавлять, удалять и обновлять контент буквально в два клика;
- легковесность и производительность. Система должна отвечать на запросы, загружать и обновлять контент в кратчайший промежуток времени при разных нагрузках;
- масштабируемость. Система должна иметь возможность модифицироваться и расширяться дополнительными компонентами, которые могут потребоваться заказчику в будущем для более расширенной работы;
- гибкость. Инструменты, необходимые контент-менеджеру для любых целей, возможность создания новостного, статейного разделов, иерархическая зависимость разделов. Также система должна быть дружелюбной для SEO, то есть иметь необходимые для таких целей дополнительные поля;

- пользователи. Неограниченное число пользователей, возможность добавления нового пользователя, например для редактора новостей и редактора статей собственные учетные записи. Так же право добавления нового пользователя только для администратора системы;

- headless CMS. Данная система должна иметь независимый бэкенд, то есть возможность обращаться к системе абсолютно с любых устройств посредством RESTful API.

Перед разработкой продукта важную роль играют его основные цели и задачи, которые определяют ради чего собственно и создается данное приложение. После определения основных целей, основной частью разработки является проектированию архитектуры приложения, базы данных, а также связь приложения с базой данных. Но не менее важным процессом является выбор оптимальных инструментов разработки, а также среда разработки программиста.

Многие команды разработчиков, взвесив преимущества и недостатки заранее определяют при разработке типичного приложения с набором инструментов и шаблоном проектирования.

Попробуем разобраться в особенностях Headless CMS. Headless CMS — это система управления контентом, которая отделяет слой данных от слоя отображения. При помощи такой системы можно управлять содержимым любых клиентов в клиент-сервер архитектуре: веб-сайт, мобильное приложение, устройство Internet of Things. Централизируя и распространяя содержимое в универсальном формате, система дает возможность управлять всеми платформами из одного интерфейса и гибко персонифицировать контент для каждой из них.

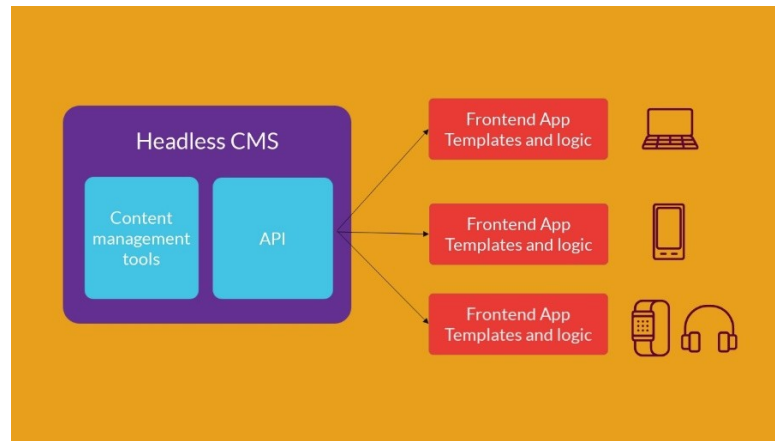


Рисунок 20 – Схема представления Headless CMS

Преимущества архитектуры Headless CMS:

- Headless CMS отвечает только за управление «чистым» контентом, а ответственность за представление и frontend логику несут разработчики, которые используют контент, подгоняя его под тип площадки. Контент публикуется с помощью Headless CMS и становится доступным для любого приложения через API. Такая архитектура стремительно набирает популярность. Разработчикам она дает большое пространство для творчества, а владельцам позволяет управлять контентом, обновлять дизайн, расширять функциональность и изменять логику приложений;
- главная ценность Headless CMS — возможность работы над контентом для разных платформ в одном интерфейсе. Сайт, мобильные приложения, IoT и т.д. — контент для любой из площадок распространяется в универсальном формате, это и дает свободу разработчикам в выборе технологий и удобство управления для владельцев.
- Headless CMS защищена с точки зрения безопасности. Ее можно скрыть архитектурно так, что клиент в клиент-серверном приложении не сможет обращаться к ней напрямую. Поэтому атака на систему извне будет невозможна;
- разработчикам не требуется обладать специальными знаниями для работы с конкретной Headless CMS. Конечно,

нужно вникнуть в API и разобраться с интерфейсом, но не в масштабе традиционной CMS.

Headless CMS отлично подходят для таких приложений:

- Single-Page Applications (SPA): React, Angular, Ember или Vue;
- динамические приложения: JavaScript, HTML и CSS;
- мобильные приложения: Android, iOS, React Native, Flutter, NativeScript;
- статичные сайты на основе сайт генераторов: Jekyll, Нехо, Hugo.

Разрабатываемая система управления будет иметь вид одностраничного приложения (Single-Page Application). SPA - это тип web-приложений, в которых загрузка необходимого кода происходит на одну страницу, что позволяет сэкономить время на повторную загрузку одних и тех же элементов. То есть, например, при переходе по редактируемым записям системы, страница вместе с js, css файлами не будет загружать каждый раз, вместо этого будет лишь перерисовываться компонент.

При разработке системы будет использоваться набирающий популярность стек MERN. Стек - это сочетание технологий, используемых для создания веб-приложения. Любое веб-приложение будет создано с использованием нескольких технологий (фреймворки, библиотеки, базы данных и т. д).

Стек MERN - это JavaScript-стек, разработанный для упрощения процесса разработки. MERN включает в себя четыре компонента с открытым исходным кодом: MongoDB, Express, React и Node.js. Эти компоненты обеспечивают комплексную среду для работы разработчиков.

Рассмотрим каждую технологию стека MERN.

1) Node.js

Node.js это — среда выполнения JavaScript. Окружение Node.js включает все, что вам нужно для выполнения программы, написанной на JavaScript.

Раньше JavaScript можно было выполнять только в браузере. Node.js позволяет запускать JavaScript программы на любом устройстве, куда можно установить данную платформу. Благодаря платформе Node.js язык JavaScript может делать, то что могут делать другие скриптовые языки, например Python.



Рисунок 21 - Платформа Node.js

На официальном веб-сайте Node определяется как «платформа, основанная на исполняемой JavaScript-библиотеке Chrome, которая позволяет упростить создание быстрых масштабируемых сетевых приложений.

Для работы платформы Node используется виртуальная машина V8, которая задействует Google Chrome для серверного программирования. Благодаря V8 производительность Node возросла. В связи с тем, что Node применяет JavaScript на стороне сервера, появляются следующие преимущества:

разработчики могут создавать веб-приложения на одном языке, благодаря чему снижается потребность в переключении контекста при разработке серверов и клиентов;

популярнейший формат обмена данными JSON является собственным форматом JavaScript.

В Node.js существует специальный API для работы с сервером, файловой системой, управлении памятью и прочим. Для создания серверов существует большое количество фреймворков, наиболее популярным и гибким из которых является Express.js.

2) Express.js

Фреймворк Express выбран в качестве бэкенда разрабатываемой системы благодаря гибкости и простоты в использовании.

Express используется для разработки приложений достаточно давно и благодаря своей стабильности прочно занимает позицию одного из самых популярных фреймворков Node.js.



Рисунок 22 - Фреймворк Express

Для этого фреймворка существует большое количество подробных инструкции и описаний, которые составлены разработчиками, проверившими его эффективность на практике. Поэтому именно с Express обычно рекомендуется начинать работу, когда разработчик намерен научиться создавать приложения на платформе Node.js.

Основная особенность этого фреймворка заключается в том, что для Express характерен небольшой объем базового функционала. Все остальные необходимые функции нужно будет добирать за счет внешних модулей. По сути, Express в чистом виде - это сервер и у него может не быть ни одного модуля.

Благодаря такому минимализму разработчик изначально получает в свое распоряжение легкий и быстрый инструмент, который он может расширять и развивать.

При этом немаловажно, что выбор модулей для Express не связан ни с какими ограничениями: ни с количественными, ни с функциональными.

В результате, этот фреймворк обеспечивает разработчику возможность решать любые задачи, не ограничивая его при этом в выборе средств.

С одной стороны, не может не радовать тот факт, что отсутствие готовых универсальных решений фактически означает, что каждое создаваемое приложение будет уникальным.

С другой стороны, разработчику нужно самостоятельно отбирать и организовывать модули, а это предполагает большой объем работы и соответственно, требует от разработчика больше времени и усилий.

К преимуществам фреймворка Express можно отнести следующее:

- простота;
- гибкость;
- хорошая масштабируемость;
- развитое сообщество;
- подробная документация;
- широкий выбор подключаемых модулей.

К недостаткам:

- большой объем ручной работы;
- используется устаревший подход callbacks функций.

3) MongoDB

RESTful API приложения будет написан в связке Express.js с нереляционной СУБД MongoDB.



Рисунок 23 – СУБД MongoDB

MongoDB реализует новый подход к построению баз данных, где нет таблиц, схем, запросов SQL, внешних ключей и многих других вещей, которые присущи объектно-реляционным базам данных. В отличие от реляционных баз данных MongoDB предлагает документо-ориентированную модель данных, благодаря чему MongoDB работает быстрее, обладает лучшей масштабируемостью, ее легче использовать.

Одним из популярных стандартов обмена данными и их хранения является JSON (JavaScript Object Notation). JSON эффективно описывает сложные по структуре данные. Способ хранения данных в MongoDB в этом плане похож на JSON, хотя формально JSON не используется. Для хранения в MongoDB применяется формат, который называется BSON (БиСон) или сокращение от binary JSON.

BSON позволяет работать с данными быстрее: быстрее выполняется поиск и обработка. Хотя надо отметить, что BSON в отличие от хранения данных в формате JSON имеет небольшой недостаток: в целом данные в JSON-формате занимают меньше места, чем в формате BSON, с другой стороны, данный недостаток с лихвой окупается скоростью;

3) React

В качестве фронтенда будущей системы используется JavaScript библиотека React от компании Facebook.

Сравнивать ReactJs с Angular или другими MVC фреймворками не имеет смысла, так как ReactJs – это только представление. React – это язык шаблонов в сочетании с несколькими функциями, которые позволяют отрисовать HTML, т.е. результат работы React – это HTML.

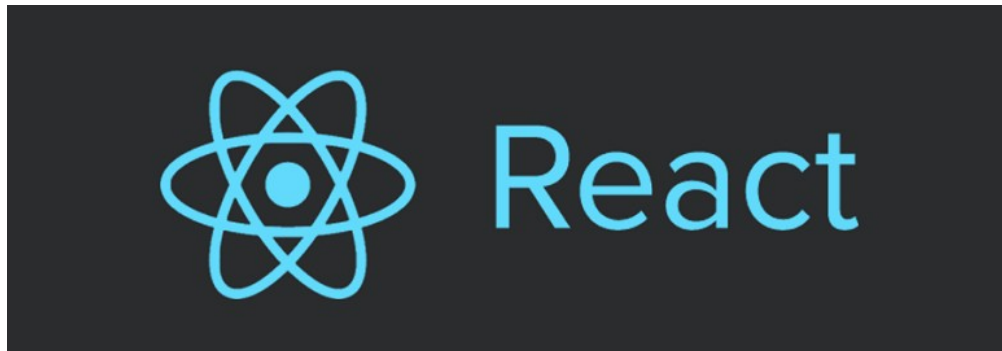


Рисунок 24 – JavaScript библиотека React

ReactJs постоянно работает с DOM, перерисовывая его при изменении условий (та часть DOM, которую меняет ReactJs, называется компонентом). Ранее подобная практика сильно бы отразилась на производительности приложения, но разработчики ReactJs подошли к решению данного вопроса кардинально: они полностью переписали DOM на Javascript.

Важной особенностью ReactJs является использование JSX. Это надстройка на JS, позволяющая использовать про-XML синтаксис в Javascript коде. JSX – это сочетание javascript и html, которые в связке являются непривычным синтаксисом для большинства разработчиков. Стандартом считается разделение JS части от разметки, что усложняет слежение за изменениями HTML. JSX позволяет видеть все процессы в одном месте, не отвлекаясь на сложности грамотного и валидного кода. После компиляции JSX получается чистый JS;

Так же стоит уточнить, что приложение системы управления контентом будет разрабатываться на TypeScript.

TypeScript представляет язык программирования на основе JavaScript.



Рисунок 25 - язык программирования TypeScript

Развитие TypeScript началось в конце 2012 года. Данный язык зародился в Microsoft Андерсом Хейлсбергом, который разработал такие известные языки как Delphi, C#.

Следует отметить, что TypeScript - это строго типизированный и компилируемый язык, чем, возможно, будет ближе к строготипизированным языкам. Хотя на выходе компилятор создает все тот же JavaScript, который затем исполняется браузером. Однако строгая типизация уменьшает количество потенциальных ошибок, которые могли бы возникнуть при разработке на JavaScript.

В качестве инструмента разработки данного приложения используется текстовый редактор Visual Studio Code. [VSCode](#) - популярный бесплатный редактор кода, созданный Microsoft. Данный редактор довольно быстрый и легко расширяется дополнительными плагинами.

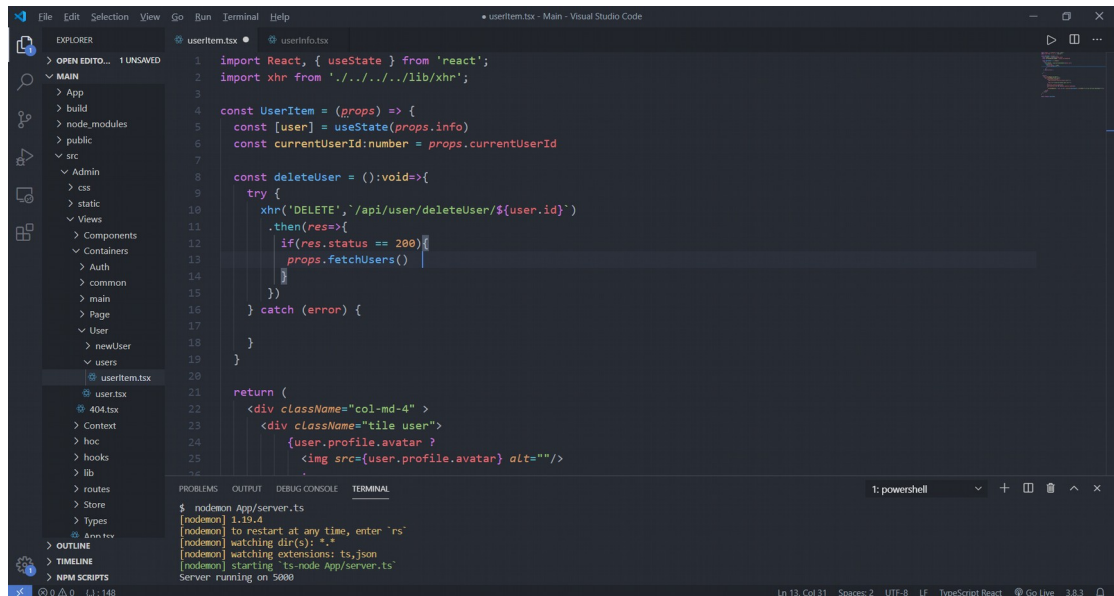


Рисунок 26 – текстовый редактор Visual Studio Code

Определившись с инструментами и подходами можно приступить к разработке проекта.

2.2 Описание функций проекта

Результатом разработки является система управления контентом сайта, соответствующая первоначальным требованиям. Рассмотрим данную систему, а так же ее основной функционал.

При переходе в панель администратора системы управления контентом программа проверяет токен доступа пользователя с помощью стандарта JSON Web Token. Если специальный ключ находится в локальном хранилище браузера, а также срок его действия не истек, программа позволяет нам перейти в панель администратора, в противном случае, система переводит на форму авторизации, где после успешной авторизации новый сгенерированный ключ добавляется в память браузера.

Форма авторизации проверяет валидность полей, а также соответствия введенных данных с имеющимися учетными записями.

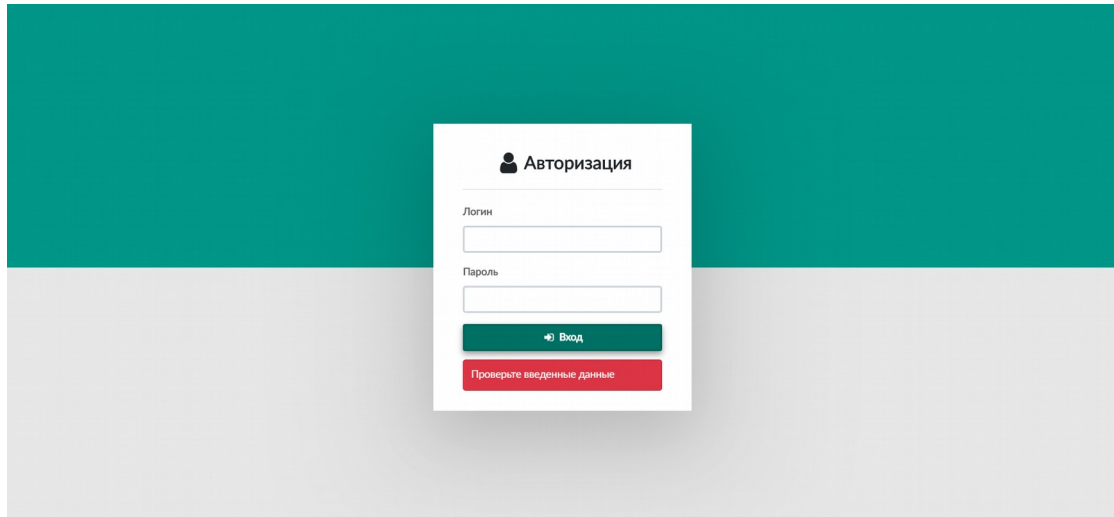


Рисунок 27 – Форма авторизации

При успешной авторизации пользователь перенаправляется в панель администрирования.

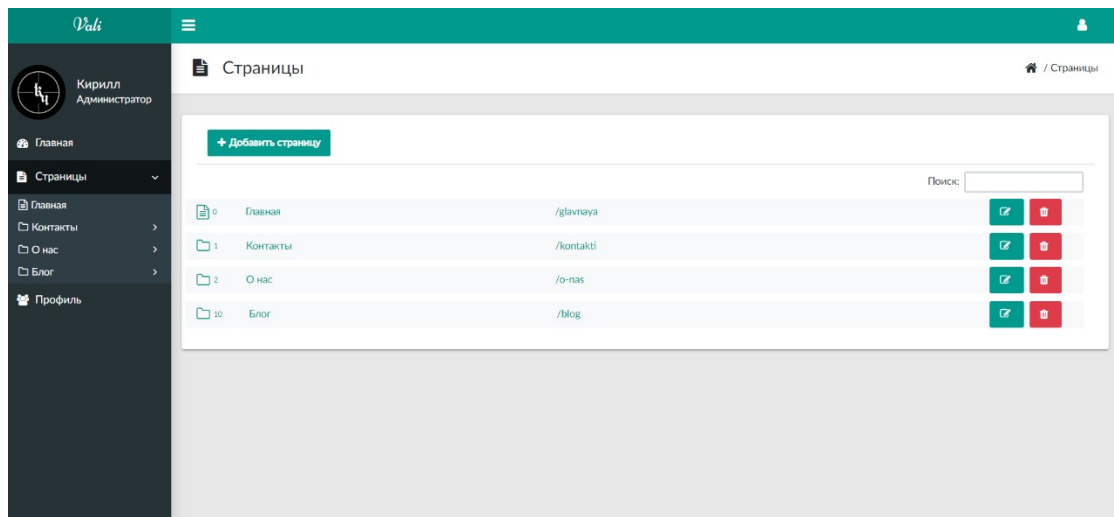


Рисунок 28 – Компонент системы “Страницы”

Интерфейс панели реализован максимально просто, так, что является интуитивно понятным. Все находится буквально под рукой. Слева находится сайдбар с меню компонентов приложения и пользователя, который в настоящее время залогинен.

Находясь в компоненте “Страницы” пользователь видит список страниц, доступных на верхнем уровне. По уровням можно перемещаться, переходя по элементам, имеющим иконку папки соответственно. Такие страницы условно называются контейнерами, то есть они являются категориями и имеют

дополнительные списки страниц внутри себя. Находясь на определенном уровне иерархии страниц, пользователю предоставляется возможность создать страницы на уровне, в котором он находится.

При большом количестве страниц пользователю предоставлено поле для поиска страниц. Благодаря поиску пользователю будут найдены все страницы, которые удовлетворяют вхождению строки, набираемой в поле поиска.

Так как уровней вложенности может быть очень много, в компоненте “Страницы” находится навигация “хлебные крошки”, чтобы пользователь понимал на каком уровне он находится в текущий момент. Так же для удобной навигации в меню слева в иерархическом виде предоставлен способ перемещения по разделам в меню (Рисунок 29).

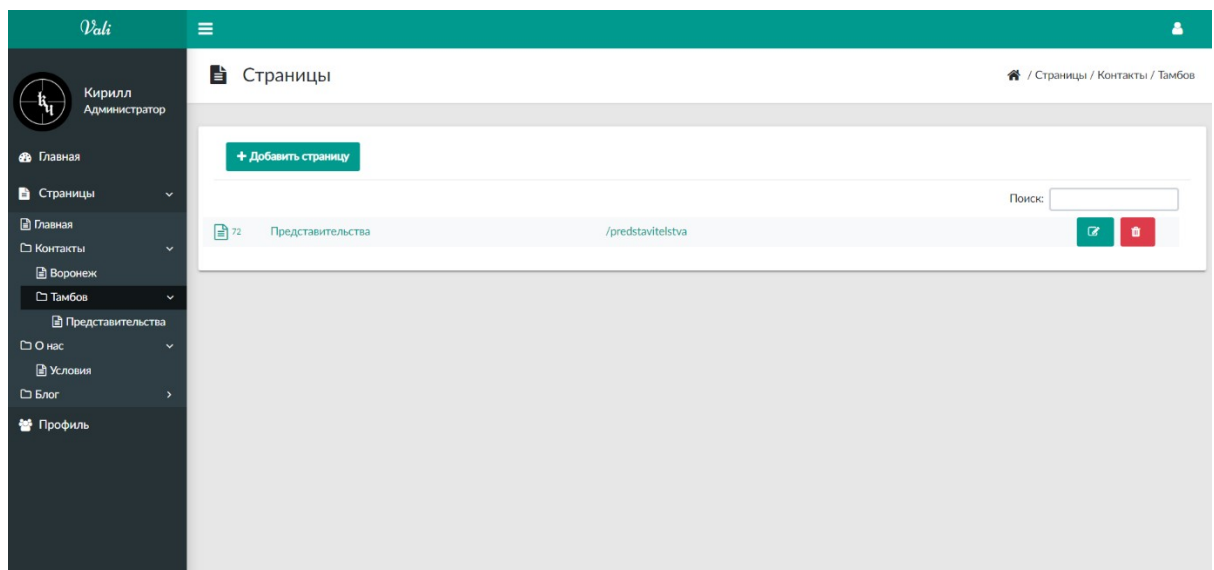


Рисунок 29 – Иерархический вид категорий в меню

Для добавления новой страницы существуют все необходимые поля и конопки. Рассмотрим каждый элемент.

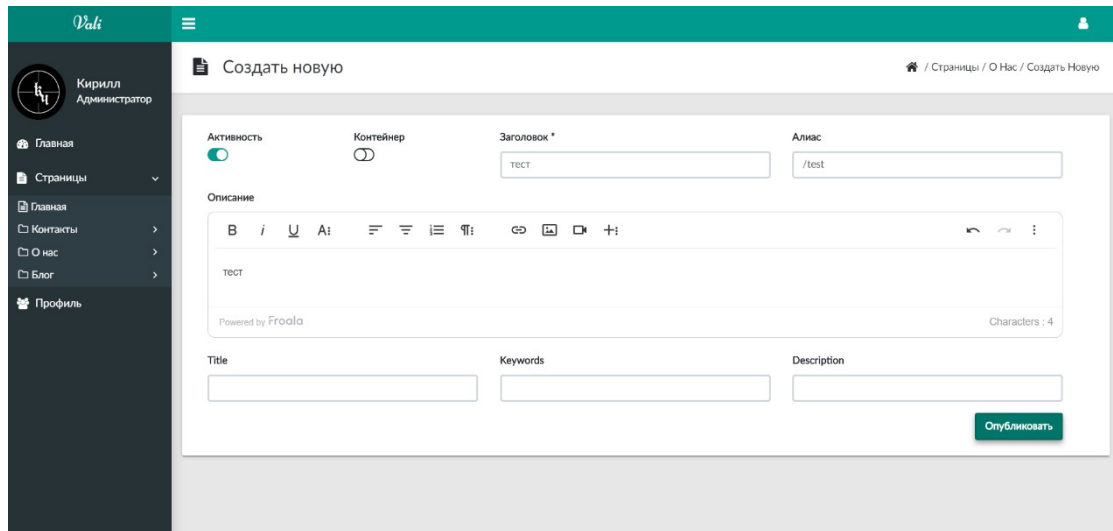


Рисунок 30 – Создание новой страницы

- активность. По умолчанию неактивно. Определяет выводится ли текущая страница;
- контейнер. Определяет является ли текущая страница родительской;
- заголовок. Является обязательным полем при создании страницы;
- алиас. Транслитный вариант заголовка, который является частью ЧПУ (Человеку Понятный Урл) страницы;
- описание. Является телом страницы, здесь находится контент. Для редактирования контента используется веб-редактор WYSIWYG для форматирования текста Froala Editor. Данный редактор довольно удобен, так как имеет большое количество инструментов, в том числе редактор HTML и JavaScript кода;
- title, keywords, description. Являются полями для метатегов соответственно.
- В правом верхнем углу системы находится дополнительное меню для выхода пользователя из системы, а так же пункт перехода в компонент “Профиль”.

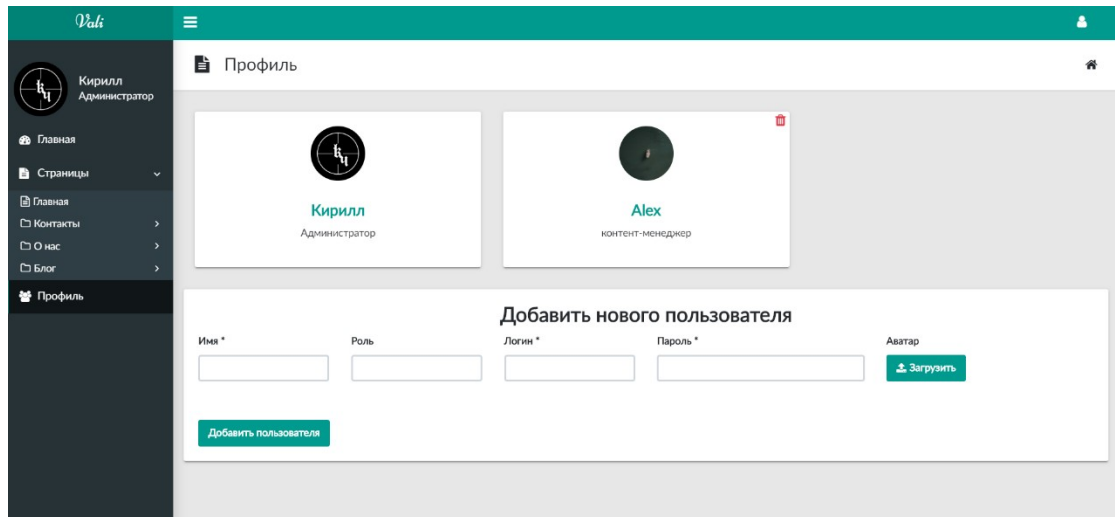


Рисунок 31 – Компонент “Профиль”

Весь функционал компонента отображается только для администратора системы. Таким образом только администратор может удалять пользователей и добавлять новых.

При добавлении нового пользователя доступны следующие поля:

- имя будущего пользователя, которое является обязательным для заполнения;
- роль. Определяет каким родом деятельности занимается пользователь;
- логин и пароль. Обязательные поля, которые используются для авторизации в системе;
- аватар. Позволяет загрузить индивидуальный аватар пользователя.

Так же данная система построена по принципу Headless CMS, то есть имеет RESTful API для CRUD операций. Таким образом с данной системой можно работать на любой платформе, имеющей фронтенд.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была разработана система управления контентом на стеке технологий MERN. Система нацелена на редактирование страниц сайта.

Таким образом достигнута основная цель – разработана система управления контентом сайта, а также выполнены следующие задачи:

- проведен сравнительный анализ существующих систем управления контентом сайтов;

- рассмотрены современные средства и технологии, которые можно использовать при разработке системы управления контентом;
- описаны функции проекта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Гениатулина Е.В. CMS - системы управления контентом// НГТУ — Новосибирск.- 2015.
- 2 Фреймворки в веб-разработке [Электронный ресурс] URL: https://web-creator.ru/articles/about_frameworks (Дата обращения: 30.04.2020)
- 3 ТОП-10 фреймворков для веб-разработки в 2019 [Электронный ресурс] URL: <https://proglib.io/p/web-frameworks-2019/> (Дата обращения: 30.04.2020)
- 4 Дронов В. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS// БХВ-Петербург.— 2017.
- 5 Почему Django — лучший фреймворк для разработки сайтов [Электронный ресурс] URL: <https://ru.hexlet.io/blog/posts/pochemu-django-luchshiy-freymvork-dlya-razrabotki-saytov> (Дата обращения: 02.05.2020)
- 6 Хартл Майкл. Ruby on Rails для начинающих. Изучаем разработку веб-приложений на основе Rails// ДМК Пресс. - 2017.
- 7 Фримен Адам. Angular для профессионалов// Питер .- 2018
- 8 Обзор Vue.js [Электронный ресурс] URL: <https://timeweb.com/ru/community/articles/obzor-vue-js-1> (Дата обращения: 02.05.2020)
- 9 М. Кантелон, М. Хартер, ТЈ Головайчук, Н. Райлих. Node.js в действии// Питер. - 2015.
- 10 Пауэрс Шелли. Изучаем Node. Переходим на сторону сервера// Питер. - 2017.