

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

САЙТ МЕЖОТРАСЛЕВОГО ЦЕНТРА ВЫСОКОТЕМПЕРАТУРНЫХ
ТЕПЛОФИЗИЧЕСКИХ ИССЛЕДОВАНИЙ КОНДЕНСИРОВАННЫХ
МАТЕРИАЛОВ

Выпускная квалификационная работа бакалавра
по направлению 44.03.04 Профессиональное обучение (по отраслям)
профиля «Энергетика»
профилизация «Компьютерные технологии автоматизации и управления»

Идентификационный номер ВКР: 132

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующий кафедрой ИС

_____ Н. С. Толстова

« ____ » _____ 2016 г.

САЙТ МЕЖОТРАСЛЕВОГО ЦЕНТРА ВЫСОКОТЕМПЕРАТУРНЫХ
ТЕПЛОФИЗИЧЕСКИХ ИССЛЕДОВАНИЙ КОНДЕНСИРОВАННЫХ
МАТЕРИАЛОВ

Выпускная квалификационная работа бакалавра
по направлению 44.03.04 Профессиональное обучение (по отраслям)
профиля «Энергетика»
профилизация «Компьютерные технологии автоматизации и управления»

Идентификационный номер: 132

Исполнитель:

студент группы КТэ-401

И. А. Пичугина

Руководитель:

ассистент кафедры ИС

М. Ю. Черноскутов

Нормоконтролер:

Т. В. Рыжкова

Екатеринбург 2016

РЕФЕРАТ

Пояснительная записка к выпускной квалификационной работе выполнена на 54 страницах, содержит 3 таблицы, 39 рисунков, 29 источников.

Ключевые слова: WEB-ТЕХНОЛОГИИ, САЙТ, ТЕПЛОФИЗИКА, НАУКА, РАЗРАБОТКА.

Объект — разрабатываемое оборудование и исследования, проводимые в лаборатории теплофизики.

Предмет — продвижение оборудования и исследований при помощи web-технологий.

Цель — разработан web-сайт лаборатории теплофизики для заинтересованных в этом лиц.

Для достижения цели работы были поставлены следующие *задачи*:

- рассмотреть и выбрать методы и технологии создания web-сайтов;
- на основе выбранных метода и технологии подобрать ПО, позволяющее эффективно использовать сайт-визитку;
- создать макет сайта-визитки;
- наполнить необходимыми первоначальными данными.

СОДЕРЖАНИЕ

Список сокращений	4
Введение.....	5
1 Предпроектная подготовка.....	7
1.1 Обсуждение проекта с заказчиком.....	7
1.2 Обзор и анализ литературы.....	9
1.3 Frontend	11
1.3.1 HTML5	11
1.3.2 CSS3.....	12
1.3.3 JavaScript	16
1.4 Backend.....	18
1.4.1 Python	19
1.4.2 Ruby	20
1.4.3 Java.....	21
1.4.4 PHP	22
1.5 Системы управления содержимым (CMS)	23
1.6 Система контроля версий.....	26
2 Создание сайта-визитки.....	29
2.1 Выбор технологий.....	29
2.1.1 Denwer	29
2.1.2 Joomla	30
2.1.3 Git и Smart Git.....	31
2.3 Разработка проекта.....	32
2.3.1 Установка программного обеспечения.....	32
2.3.2 Заполнение сайта-визитки данными	41
Заключение	48
Список использованных источников	49
Приложение	52

СПИСОК СОКРАЩЕНИЙ

1. ДНВР — джентльменский набор Web-разработчика.
2. ПО – программное обеспечение.
3. API – Application Programming Interface.
4. BOM – Browser Object Model.
5. CCS3 – Cascading Style Sheets, version 3.
6. CMS – Control Management System.
7. DOM – Document Object Model.
8. DNS – Domain Name System.
9. HTML5 – Hyper Text Markup Language, version 4.
10. JS – JavaScript.
11. LTS – Long Term Support.
12. MySQL – My Structured Query Language.
13. PHP – Personal Home Page Tools.
14. SSL – Secure Sockets Layer.
15. STS – Short Term Support.
16. VCS – Version Control System.
17. WAMP – Windows, Apache, MySQL, PHP.
18. WHATWG – Web Hypertext Application Technology Working Group.
19. XHTML – Extensible Hyper Text Markup Language.

ВВЕДЕНИЕ

Интернет развивается очень стремительно. Растёт количество изданий, которые посвящаются Сети. Теперь Интернет — это источник огромного количества информации для любого пользователя, независимо от его интересов.

Уже невозможно представить жизнь без компьютера и Интернета, ведь за любой неизвестной, интересной информацией люди обращаются к его помощи.

Всё необходимое человек находит на каких-либо сайтах, которые несут определённую смысловую нагрузку. Это сайты-портфолио, сайты-визитки, социальные сети и множество других видов сайтов. В свою очередь они создаются людьми, которые хотят принести пользу пользователям. Получается цепочка: пользователь–сайт–пользователь. Все вносят свой вклад в Интернет: конечный пользователь выявляет ошибки, говорит о недостатке или избытке информации, а начальный — исправляет все недочёты.

Хотелось бы углубиться в понятие сайта-визитки. Сайт-визитка (англ. business card website) — небольшой сайт, как правило, состоящий из одной (или нескольких) веб-страницы, и содержащий основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные [1]. Проще говоря, сайт-визитка — это небольшой информационный сайт, на котором располагаются основные сведения об организации, описание услуг и стоимости.

Этот тип сайтов на данный момент является очень распространённым в Интернете, потому что его создание не занимает очень много времени и при этом он позволяет показать всю самую необходимую информацию.

Процесс создания сайта-визитки совсем не отличается от процесса создания какого-либо другого информационного Интернет-ресурса:

1. Обсуждение проекта с заказчиком.
2. Описание технического задания будущей разработки.
3. Создание макета или выбор из имеющихся.
4. Установка системы управления содержимым.
5. Заполнение сайта-визитки данными и информацией.
6. Сдача проекта заказчику [12].

Объектом выпускной квалификационной работы является разрабатываемое оборудование и исследования, проводимые в лаборатории теплофизики.

Предмет — продвижение оборудования и исследований при помощи web-технологий.

Цель — разработать web-сайт лаборатории теплофизики для заинтересованных в этом лиц.

Для достижения поставленной цели в работе были поставлены следующие *задачи*:

- рассмотреть и выбрать методы и технологии создания web-сайтов;
- на основе выбранных метода и технологии подобрать ПО, позволяющее эффективно использовать сайт-визитку;
- создать макет сайта-визитки;
- наполнить необходимыми первоначальными данными.

1 ПРЕДПРОЕКТНАЯ ПОДГОТОВКА

1.1 Обсуждение проекта с заказчиком

Во время обсуждения проекта, заказчик должен рассказать, что именно он хочет видеть на сайте и каким он должен быть в его представлении.

Внешний вид сайта-визитки не должен быть броским, при этом основные элементы должны быть выделены для привлечения внимания посетителей сайта. Также, должны присутствовать блоки для регистрации пользователя, входа пользователя, форма для отправки электронных писем сотрудникам лаборатории с сайта. Обязательным содержимым будут являться блоки с контактами лаборатории, сотрудники лаборатории теплофизики, сфера деятельности, имеющееся оборудование, предоставляемые услуги.

После обсуждений с заказчиком было создано дерево сайта (рисунок 1) и предъявлены требования к административной части сайта-визитки:

- возможность расширения функционала;
- возможность изменения содержания каждой страницы (название раздела, описание, ключевые слова);
- возможность добавления статей;
- возможность изменения кода при необходимости;
- вёрстка HTML5.

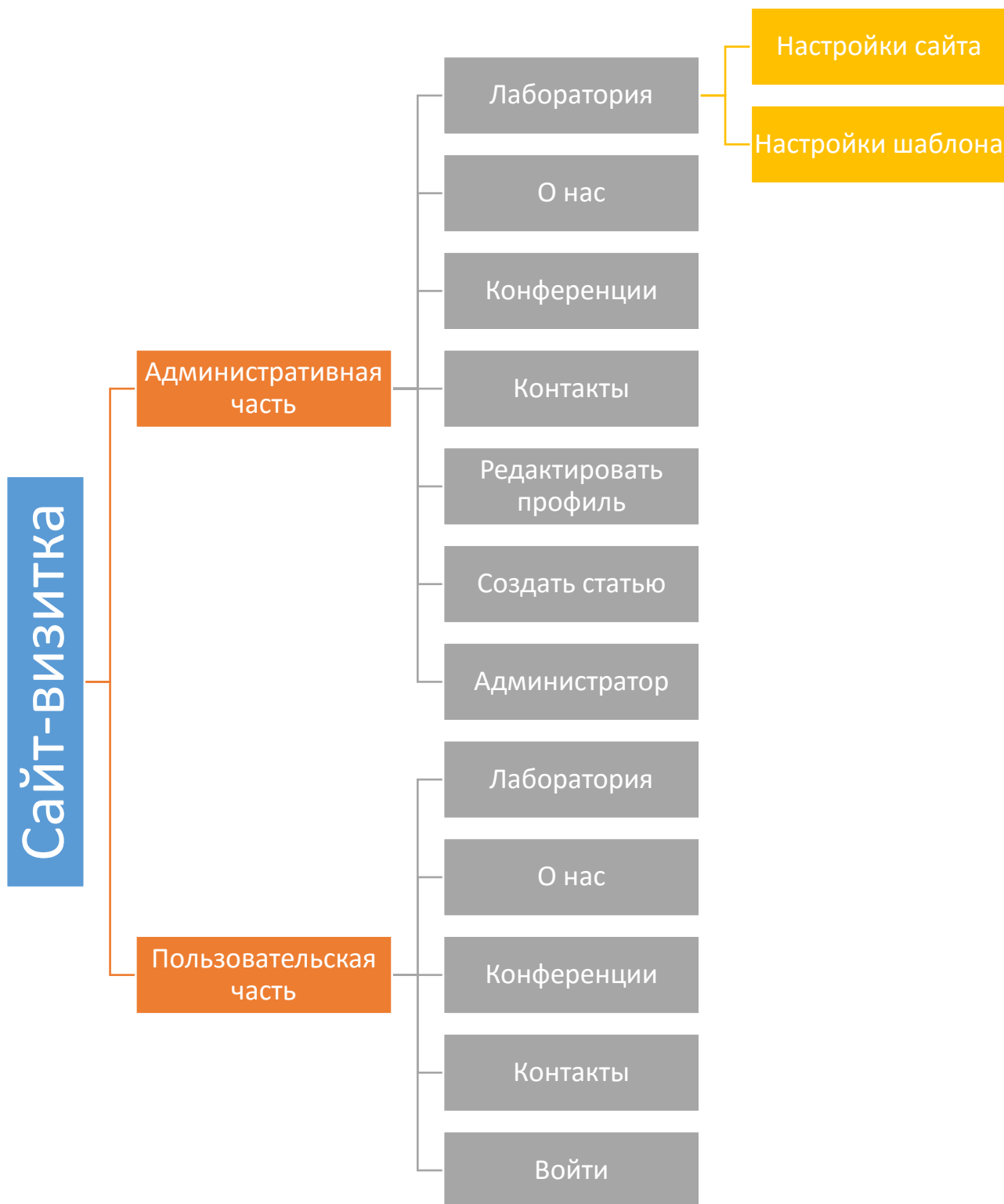


Рисунок 1 — Структура сайта-визитки

1.2 Обзор и анализ литературы

Литература и источники информации по теме должны соответствовать таким требованиям:

- научности;
- учёта возрастных и познавательных возможностей обучающегося;
- систематичности и последовательности.

Хаген Граф «10 легких шагов к освоению Joomla! 3» [1]. В этой книге речь идет о Joomla! 3, первой версии с краткосрочной поддержкой, относящейся к линейке 3.x. Линейка 3.x приспособлена к работе с мобильными браузерами и обладает новым пользовательским интерфейсом.

Эта книга, состоящая из десяти глав, поможет освоить установку, настройку и обслуживание Joomla, то, что необходимо для создания и поддержания работоспособности сайта.

Книга рассказывает о том, как построить сайт при помощи Joomla: как установить эту систему управления содержимым, как ее настроить и как публиковать информацию, содержит множество иллюстраций. «10 легких шагов к освоению Joomla! 3» — десятая книга о Joomla! в ряду созданных автором. Самая первая была посвящена еще системе управления содержимым Mambo — прародительнице «героини» этой книги. Работы Хагена Графа о Joomla переведены по меньшей мере на девять языков и пользуются значительной популярностью: количество загрузок многих из них исчисляется десятками тысяч.

Марк Декстер, Луис Лэндри «Joomla! Программирование» [2]. В этой книге, как и в предыдущей, описывает процесс создания сайта при помощи системы управления содержимым Joomla, но даёт представление о расширениях, компонентах и внедрении других языков программирования.

Материал этой книги направлен не только на начинающих пользователей, но и на опытных. Если у пользователя нет никаких знаний об использовании и применении Joomla, то книга сможет дать все базовые

определения и основы работы с каким-либо компонентом системы. Для опытных пользователей будет плюсом тот момент, что можно найти нужную информацию очень быстро, благодаря чётко построенной иерархией содержания. Кроме того, в книге рассматриваются различные варианты выбора проектных решений, исходя из общей архитектуры Joomla, чтобы стало понятно, почему конкретная программа была построена именно так, а не иначе.

Jennifer Niederst Robbins «Learning web desing» [7]. Эта книга хороша тем, что изучение web-дизайна начинается издалека. Объясняется как работает дизайнер, какие языки программирования и программы необходимо знать дизайнеру, принципы работы Интернета и отображения оформления web-страниц, стандарты внешнего вида сайтов.

Также, в конце каждой главы читателю предлагается проверить себя по материалу пройденной главы и выполнить некоторые задания, такие как ответить на вопросы по теории, решить тест или написать код на предложенном языке программирования. Такой подход позволяет лучше усвоить пройденный материал.

Jon Duckett «HTML & CSS. Desing and Build Websites» [8]. Каждая страница книги содержит в себе новую тему с простыми и визуальными объяснениями, которые сопровождаются примерами кода программ. В конце каждой главы содержатся доскональные примеры.

Книга построена по принципу «от простого к сложному», что позволяет с каждым шагом всё глубже познавать и практиковать написание кода. Практически вся информация сопровождается иллюстрациями, на которых показан конечный результат. Это даёт возможность читателям видеть, правильно ли он сделал задание.

Также, в книге описаны все возможные ошибки, которые можно совершить при написании кода и возможности их устранения.

1.3 Frontend

Frontend — это клиентская часть сайта, то есть пользовательский интерфейс. Именно с этой частью контактирует пользователь.

Аспектами frontend-разработки являются создание макета сайта, вёрстка шаблонов и шаблонов для CMS, привязка к пользовательскому интерфейсу при помощи специальных скриптов, которые отвечают за визуализацию и анимацию.

Во времена зарождения Интернета frontend-частью являлись статичные HTML-страницы. Динамичность страницы осуществлялась за счёт gif-анимации и некоторых тегов, например, бегущая строка. На сегодняшний день клиентская часть web-страниц является динамичной за счёт того, что для разработки пользовательского интерфейса используются такие языки программирования, как HTML, CSS и JavaScript, в частности современные стандарты HTML5, CSS3 и множество библиотек для JavaScript. Таким образом, часть нагрузки по отображению и перестроению страниц легла на клиентскую часть.

1.3.1 HTML5

HTML5 (англ. HyperText Markup Language, version 5) — язык для структурирования и представления содержимого всемирной паутины. Это пятая версия HTML [18]. Разработчиками языка являются Консорциум Всемирной паутины и WHATWG (Web Hypertext Application Technology Working Group — сообщество людей, которые заинтересованы в развитии Интернета). Целью разработки HTML5 является улучшение уровня поддержки мультимедиа-технологий с сохранением обратной совместимости, удобного восприятия кода и простоты анализа.

Долгое время использовались стандарты HTML 4.01 и XHTML 1.1, но они были неудобны, так как приходилось использовать смеси особенностей.

Поэтому был создан HTML5, являющийся единым языком разметки, который сочетает в себе синтаксические нормы HTML и XHTML. Он не только расширяет, улучшает разметку документов, но и добавляет единый API для сложных web-приложений.

В HTML5 появилось множество новых семантических элементов и тегов. Теги `<section>`, `<article>`, `<header>` и `<nav>` разработаны для обогащения содержания документа и страницы.

Тег `<section>` задаёт раздел документа, может применяться для блока новостей, контактной информации, глав текста, вкладок в диалоговом окне и др.

Тег `<article>` задает содержание сайта вроде новости, статьи, записи блога, форума или др.

Тег `<header>` задает «шапку» сайта или раздела, в которой обычно располагается заголовок.

Тег `<nav>` задает навигацию по сайту. Если на странице несколько блоков ссылок, то в `<nav>` обычно помещают приоритетные ссылки [3].

1.3.2 CSS3

Web-страница состоит из HTML или XHTML разметки, которая описывает содержимое страницы, и CSS, которые «говорят» браузеру, как отображать контент пользователям, которые просматривают страницу.

CSS (Cascading Style Sheets) — каскадные таблицы стилей, которые описывают внешний вид документа, написанного с помощью языков разметки HTML или XHTML.

CSS позволяет разработчику определить стиль отображения страницы — шрифты, размер элементов, цвета, а затем применить эти стили к выбранным частям HTML-кода с помощью селекторов. Селектор — это некоторое имя стиля, для которого добавляются параметры форматирования. В качестве селектора выступают теги, классы и идентификаторы.

Самым простым способом добавления CSS стилей на web-страницу является *встроенный стиль*. Он применяется к HTML-элементу с помощью атрибута `style`. В этом типе стиля не применяются селекторы [10].

Правила CSS пишутся на формальном языке CSS и формируют таблицу стилей. Подключаться к документу они могут двумя способами:

- таблицы могут располагаться в web-документе или отдельным файлом, которые имеют расширение `.css`;
- CSS является текстовым файлом, в котором прописаны все необходимые правила.

Каскадные таблицы пишутся по определённым правилам. Каждое правило из таблицы состоит из двух частей — селектора и блока объявлений. Селектор располагается в левой части правила и определяет к каким частям документа будет применяться. Соответственно, блок объявлений расположен справа, пишется в фигурных скобках, может включать одно или несколько объявлений и представляет собой сочетание свойств и значений правил. Выглядит это таким образом:

```
селектор, селектор {  
    свойство: значение;  
    свойство: значение;  
}
```

Существует несколько видов селекторов:

1. *Универсальный селектор*. Используется для одновременного использования стиля для всех элементов web-страницы. Для обозначения универсального селектора используют символ звездочки (*).

2. *Селектор классов*. Классы применяются для определения стиля индивидуального элемента веб-страницы или задать разные стили для одного тега.

3. *Селектор идентификаторов*. Идентификатор (ID селектор) называет элемент уникальным именем, которое используется для изменения его стиля и обращения к нему через скрипты.

4. *Селектор атрибутов:*

- *простой селектор атрибутов.* Устанавливает стиль для элемента, если присутствует специфичный атрибут тега. Его значение в данном случае не важно;

- *атрибут со значением.* Устанавливает стиль для элемента в том случае, если задано определенное значение специфичного атрибута;

- *значение атрибута начинается с определенного текста.* Стиль для элемента устанавливается в том случае, если значение атрибута тега начинается с указанного текста;

- *значение атрибута оканчивается определенным текстом.* Устанавливает стиль для элемента в том случае, если значение атрибута оканчивается указанным текстом;

- *значение атрибута содержит указанный текст.* Возможны варианты, когда стиль следует применить к тегу с определенным атрибутом, при этом частью его значения является некоторый текст. При этом точно не известно, в каком месте значения включен данный текст — в начале, середине или конце;

- *одно из нескольких значений атрибута.* Некоторые значения атрибутов могут перечисляться через пробел, например, имена классов;

- *дефис в значении атрибута.* В именах идентификаторов и классов разрешено использовать символ дефиса (-), что позволяет создавать значащие значения атрибутов `id` и `class`;

5. *Контекстный селектор.* Можно одновременно установить стиль для отдельного тега, а также для тега, который находится внутри другого. Контекстный селектор состоит из простых селекторов, разделенных пробелом.

6. *Соседний селектор.* Соседними называются элементы веб-страницы, когда они следуют непосредственно друг за другом в коде документа. Для управления стилем соседних элементов используется символ плюса (+), который устанавливается между двумя селекторами.

7. *Дочерний селектор.* Дочерним селектором считается такой, который в дереве элементов находится прямо внутри родительского элемента.

8. *Селектор псевдоклассов.* Псевдоклассы определяют динамическое состояние элементов, которое изменяется со временем или с помощью действий пользователя, а также положение в дереве документа. Примером такого состояния служит текстовая ссылка, которая меняет свой цвет при наведении на нее курсора мыши. При использовании псевдоклассов браузер не перегружает текущий документ, поэтому с помощью псевдоклассов можно получить разные динамические эффекты на странице.

9. *Селектор псевдоэлементов.* Псевдоэлементы позволяют задать стиль элементов не определенных в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста. Вначале следует имя селектора, затем пишется двоеточие, после которого идет имя псевдоэлемента. Каждый псевдоэлемент может применяться только к одному селектору, если требуется установить сразу несколько псевдоэлементов для одного селектора, правила стиля должны добавляться к ним по отдельности [3].

Применение каскадных таблиц стилей к HTML-документам строится на принципах наследования и каскадирования.

Принцип наследования состоит в том, что свойства CSS наследуются элементами потомками, если они объявлены для элементов-предков.

Принцип каскадирования применяется, если происходит конфликт значений правил при использовании нескольких. Для разрешения таких конфликтов используются правила приоритета:

- самым низким приоритетом обладает стиль браузера;
- заданный стиль браузера пользователем обладает следующим по значимости приоритетом;
- стиль, заданный автором страницы, обладает приоритетом выше, чем предыдущие;

- стили, объявленные в теге `<style>`, имеют приоритет ещё выше;
- если стиль включает в себя сопроводительным словом «*!important*», то он является самым значимым.

До появления CSS оформление web-страниц производилось средствами HTML, что было не особо удобно. Но с появлением CSS стало возможным оформлять несколько документов и изменять их внешний вид одной каскадной таблицей стилей, что явно упростило написание кода.

Преимущества CSS:

- учёт нескольких стилей для отображения страницы на разных устройствах;
- внешний вид данных и страницы кэшируется, загружаются только данные, что уменьшает время загрузки страницы сайта;
- для изменения дизайна страницы необходимо поправить только лишь CSS-файл;
- дополнительные возможности оформления.

Также существует недостаток, который заключается в некорректном отображении содержимого и оформления в различных браузерах, особенно в устаревших.

1.3.3 JavaScript

JavaScript — это прототипно-ориентированный сценарный язык программирования. Имеет схожести с такими языками программирования как C, Perl и Python. Многие думают, что JavaScript является упрощённой версией Java, но на самом деле кроме названия ничего общего у них нет.

Изначально язык JavaScript применялся для различных клиентских сценариев, проверки данных, введённых пользователем в какую-либо форму или интерактивности web-страниц. Сегодня же этот язык используется для создания динамического содержимого web-страниц в совокупности с HTML.

JavaScript можно разделить на три части: ядро (ECMAScript), объектная модель браузера (Browser Object Model), объектная модель документа (Document Object Model). ECMAScript — основа для построения скриптовых языков. Её спецификация описывает инструкции, ключевые слова, типы данных, объекты, операторов, выражения. BOM является прослойкой между ядром и объектной моделью документа. Основная функция BOM — это управление окнами в браузере и управление их взаимодействием. DOM — это интерфейс программирования приложений для документов гипертекстовой разметки. Согласно объективной модели документа, сам документ может быть представлен как дерево объектов, которые обладают определёнными свойствами и позволяют выполнять различные манипуляции, такие как генерация, добавление, получение, изменение, изменение связей и удаление узлов [10].

Для работы со сценариями подходит практически любой текстовый редактор, который даёт возможность создавать текстовые файлы и сохранять их в формате .htm или .html, например, Блокнот или NotePad++. Также существуют специально разработанные редакторы для создания web-страниц, такие как Adobe Dreamweaver или Visual Studio.

Сценарии могут внедряться в HTML-документ несколькими способами:

- расположение внутри страницы;
- расположение внутри тега;
- вынесение в отдельный файл.

Расположение внутри страницы. Чтобы добавить JS-код на страницу, используется тег <script>. Рекомендуется его помещать внутрь контейнера <head>. Код примет вид:

```
<script type="application/javascript">  
    alert('Hello, World!');  
</script>
```

Расположение внутри тега. Спецификация HTML описывает набор атрибутов, которые используются для обработки событий. Пример:

```
<a href="delete.php" onclick="return confirm('Вы уверены?');">
    Удалить
</a>
```

Вынесение в отдельный файл. Скрипт пишется в отдельном файле с расширением .js, а затем подключается к документу следующей конструкцией:

```
<head>
    <script src="http://Путь_к_файлу_со_скриптом">
    </script>
</head>
```

Широко используемый элемент `<script>` имеет атрибуты:

- *async* — асинхронно загружает скрипт;
- *defer* — откладывает загрузку скрипта до момента поной загрузки страницы;
- *language* — устанавливает язык программирования скрипта;
- *src* — задаёт адрес скрипта из внешнего файла для импортирования в текущий документ;
- *type* — определяет тип содержимого основного тега.

1.4 Backend

Backend — это программно-аппаратная часть, отвечающая за обработку команд от пользователя, доступ к базам данных и отправку данных пользователю, то есть это серверная часть клиент-серверной архитектуры (программный код), которая выполняется не на машине пользователя, а на удалённом сервере.

Производятся все эти операции при помощи множества языков программирования, например, Python, Ruby, Java, PHP.

1.4.1 Python

Python — высокоуровневый язык программирования, ориентирован на повышение производительности разработчика и читаемости кода [20]. Синтаксис языка минималистичен, однако, стандартная библиотека содержит большой объём полезных функций, например, средства для работы с сетевыми протоколами, возможность писать кроссплатформенные приложения, модули для работы с разными типами данных.

Основными характеристиками Python являются:

- динамическая типизация;
- автоматическое управление памятью;
- полная интроспекция;
- механизм обработки исключений;
- поддержка многопоточных вычислений;
- высокоуровневые структуры данных.

Python начали разрабатывать в конце 1980-х годов в одном из голландских институтов. Из-за начала поздней разработки влияние оказали уже существующие на тот момент языки программирования: C, C++, Fortran, Java и многие другие. Из каждого языка брались некоторые особенности, которые появлялись при программировании на Python.

Язык программирования Python обладает чётким и последовательным синтаксисом, по этой причине код программ, написанных на Python, легко читается и воспринимается.

Преимущества языка программирования Python являются:

- библиотеки с широкими возможностями;
- высокая скорость выполнения программ;
- интерпретируемый язык программирования;
- подходит для написания скриптов (сценариев);
- обладает хорошо продуманной системой для объектно-ориентированного программирования.

Недостатки:

- низкое быстродействие;
- глобальная блокировка интерпретатора;
- отсутствие коммерческой поддержки средств разработки;
- ограниченные средства для работы с базами данных.

1.4.2 Ruby

Ruby — мультипарадигмальный высокоуровневый язык программирования для быстрого и удобного объектно-ориентированного программирования. Был создан под влиянием таких языков программирования, как Perl, Eiffel и Smalltalk.

Основными характеристиками языка программирования Ruby являются динамическая типизация и автоматическое управление памятью.

Преимущества Ruby:

- открытая разработка;
- кроссплатформенность;
- внедряется в HTML-разметку;
- современные методы манипуляции строками и текстом;
- лёгкое изучение языка за счёт простого синтаксиса;
- расширение возможностей средствами библиотек;
- встроенный отладчик.

Недостатки:

• недостаточное количество литературы и источников, посвящённых Ruby;

- менее производителен по сравнению с другими языками для web-разработки;
- медленная разработка и обновление языка.

1.4.3 Java

Java — кроссплатформенный объектно-ориентированный язык программирования. Приложения Java транслируются в байт-код. Трансляция программы в байт-код происходит при помощи виртуальной машины Java — это программа, которая обрабатывает байтовый код и передаёт инструкции оборудованию. Благодаря этому свойству приложения могут работать на любой виртуальной Java-машине независимо от архитектуры компьютера.

Язык Java начали разрабатывать в 1990 году, первая версия была выпущена в начале 1996 года. Начиная с этого времени Java совершенствуется по сей день. Последняя версия была выпущена в марте 2014 года, в которой в очередной раз, как и в предыдущих версиях, появились нововведения:

- ссылки на методы;
- в интерфейсах поддержки функциональности по умолчанию ключевым словом является *default*;
- усовершенствованный API для работы с датами;
- функциональные интерфейсы.

Как и любой язык программирования, Java имеет свои преимущества и недостатки. Преимущества:

- работа в любой виртуальной Java-машине;
- безопасность;
- надёжность;
- автоматическое освобождение памяти;
- автоматическая генерация документации на основе комментариев.

Недостатки:

- громоздкость языка за счёт многословности;
- высокая ресурсоёмкость;
- медлительность из-за высокой нагрузки.

1.4.4 PHP

PHP — скриптовый язык общего назначения, применяемый для web-разработки с открытым исходным кодом.

На сегодняшний день PHP является одним из самым распространённых языков для создания web-сайтов, на нём написано более 80% web-ресурсов. Это обосновывается широким набором встроенных средств разработки приложений:

- автоматическое извлечение параметров и переменных окружения web-сервера;
- взаимодействие с большим количеством баз данных;
- автоматизированная отправка HTTP-заголовков;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами;
- обработка загруженных файлов на сервер.

С использованием PHP на данное время разработано множество приложений, которые используют данный язык программирования (рисунок 2).

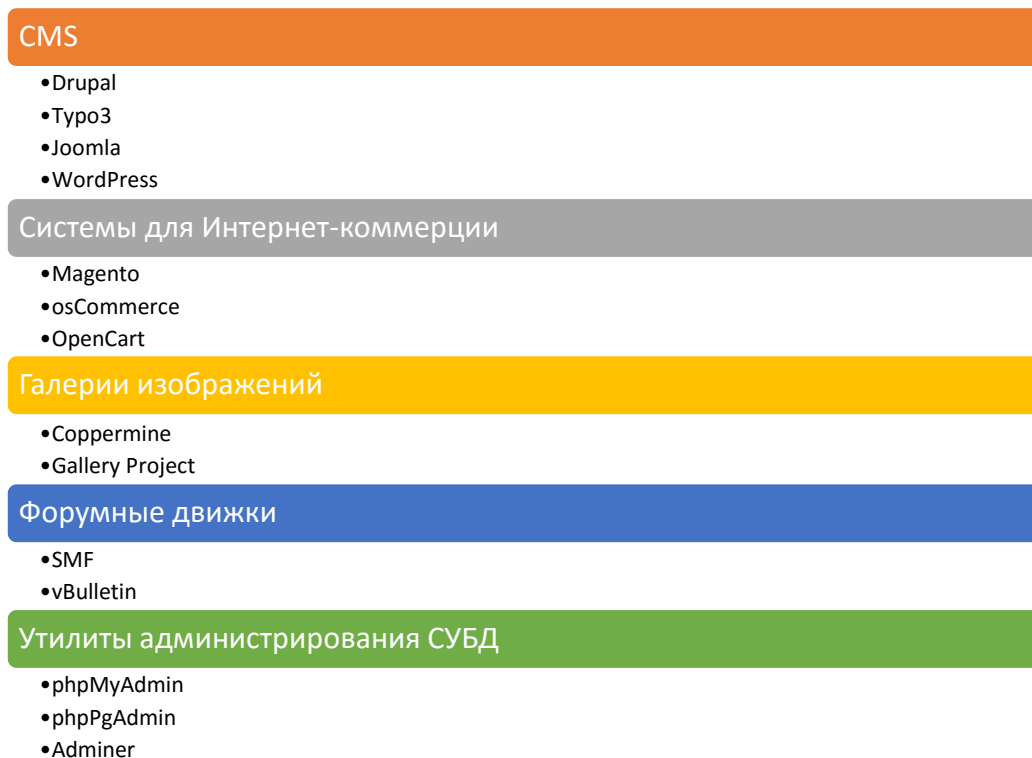


Рисунок 2 — Приложения, использующие PHP

1.5 Системы управления содержимым (CMS)

Система управления содержимым (CMS) — это информационная система, которая используется для организации совместного процесса создания, редактирования и управления содержимым [15].

Основными функциями CMS являются:

- использование инструментов для создания содержимого;
- организация совместной работы с содержимым;
- хранение данных, версий, управление потоками данных и документов;
- публикация содержимого;
- удобное представление навигации.

Администратор сам может выбирать и настраивать необходимые компоненты благодаря модульной архитектуре CMS. К модулям можно отнести:

- меню;
- блог;
- новости;
- поиск по сайту;
- гостевая статистика.

Аналитические порталы строят рейтинги, разделяя CMS на коробочные коммерческие CMS (рисунок 3), свободные CMS (рисунок 4) и студийные CMS. Так как студийные CMS не распространены и не имеют спроса, то их рейтинг не приводится.

Коробочные CMS	Shop-Script
	1С-Битрикс
	UMI.CMS
	NetCat
	HostCMS
	CS-Cart
	AMIRO.CMS
	DIAFAN.CMS
	CMS S.Builder
	ABO.CMS
	SiteEdit
	DJEM

Рисунок 3 — Список коммерческих коробочных CMS

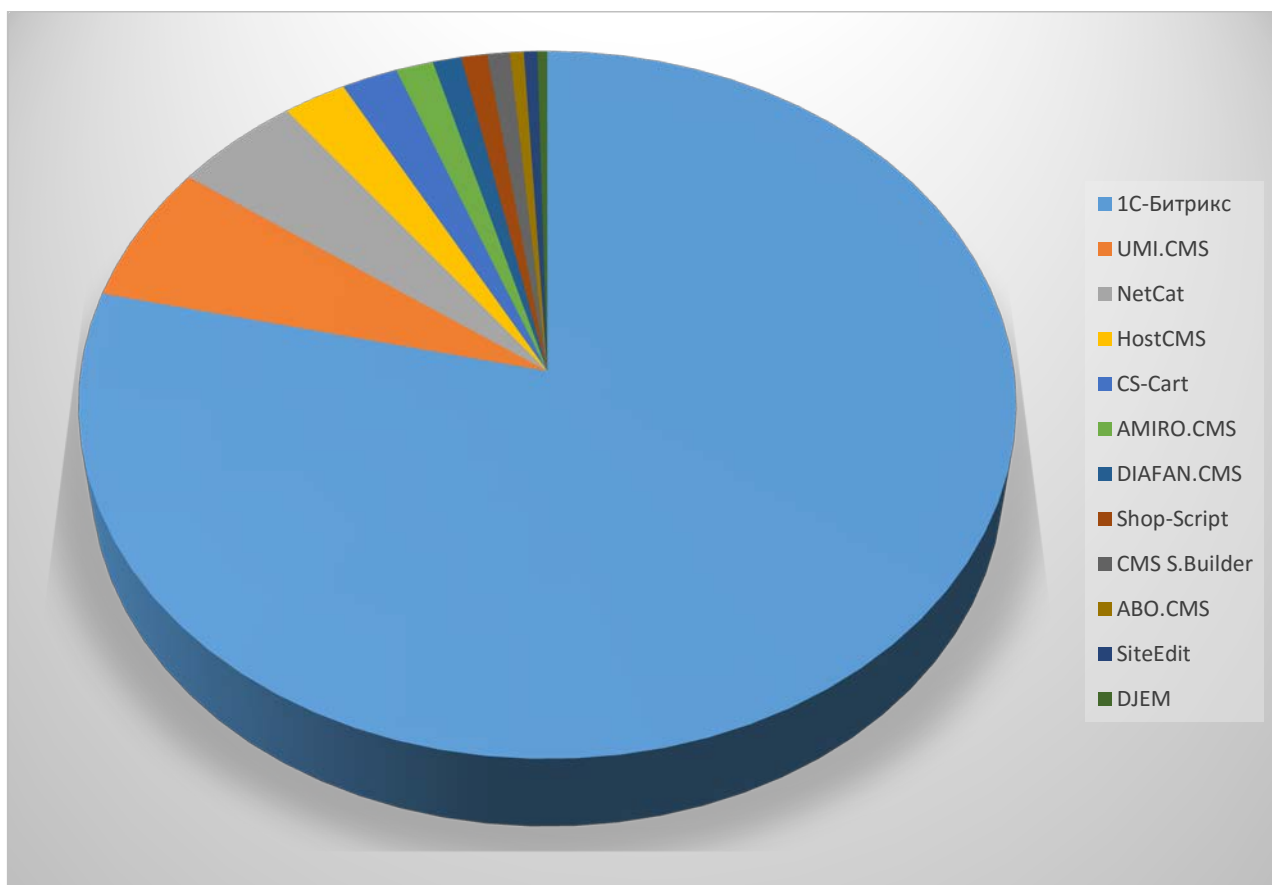


Рисунок 4 — Рейтинг коммерческих коробочных CMS

Свободные
CMS

Drupal

Joomla!

MODX

WordPress

Opencart

Magento

TYPO3 CMS

ImageCMS Corporate Free

PrestaShop

CMS Made Simple

Concrete5

Рисунок 5 — Список свободных CMS

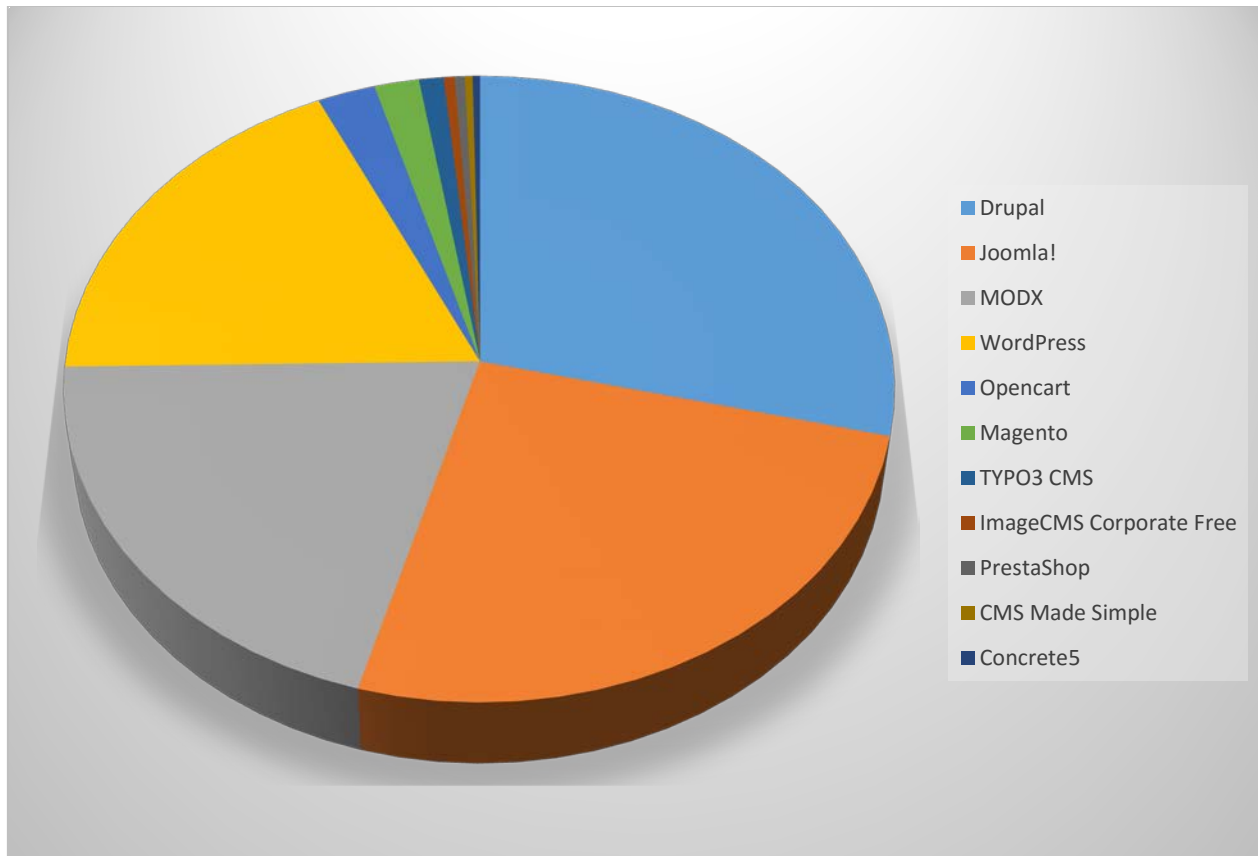


Рисунок 6 — Рейтинг свободных CMS

1.6 Система контроля версий

Помимо всего прочего, в работе использовалась система контроля версий. Система контроля версий (Version Control System) — это программное обеспечение, которое облегчает работу с изменяющейся информацией. Она позволяет хранить несколько версий одного документа, возвращаться к ранним версиям документа, определять, кем и когда изменялся документ [14].

Существует несколько типов систем контроля версий.

Локальные системы контроля версий. Самым простым и распространённым контролем данных является обычное копирование файлов в другой каталог, но такой подход чаще всего даёт сбой. Для решения этой проблемы были разработаны системы управления содержимым с простой базой данных, в которой хранятся все изменения файлов (рисунок 7).

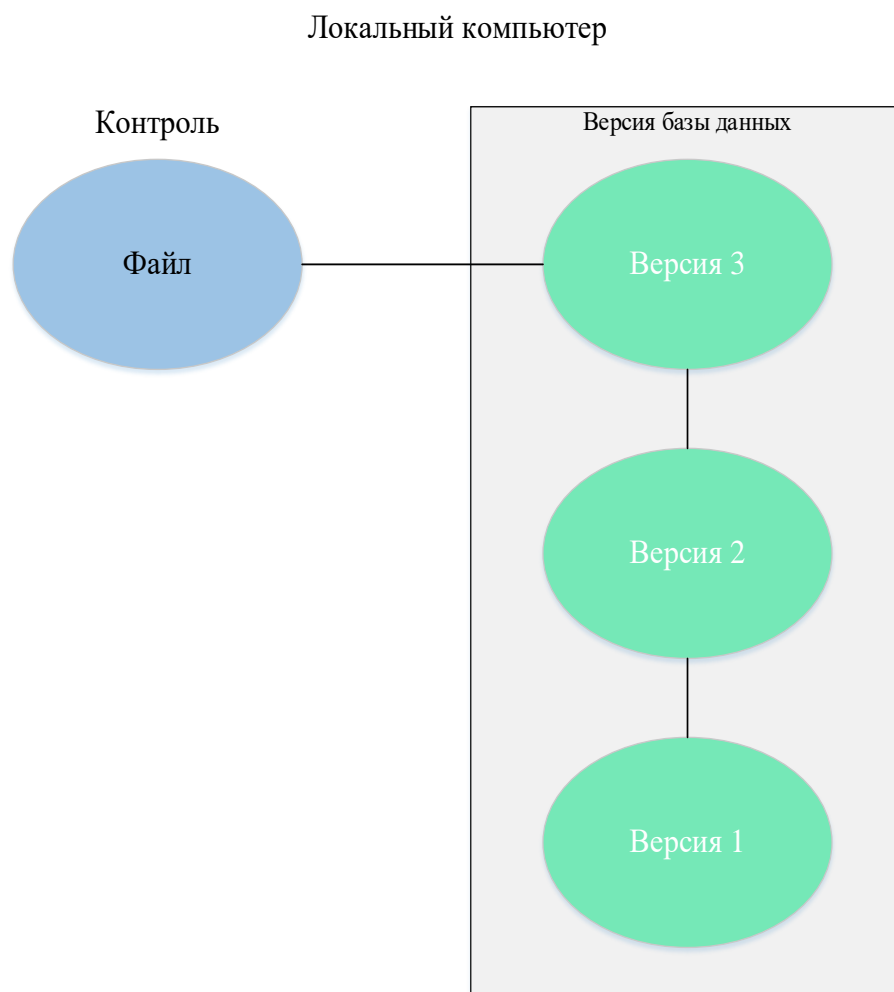


Рисунок 7 — Схема локальной системы контроля содержимым

Централизованные системы контроля версий. Другой проблемой было взаимодействие с другими разработчиками. Проблема решила после создания централизованных систем контроля версий. Принцип работы заключался в том, что все файлы хранятся на центральном сервере. Каждый разработчик мог изменять файлы с любого компьютера, которые выгружались из базы данных (рисунок 8).

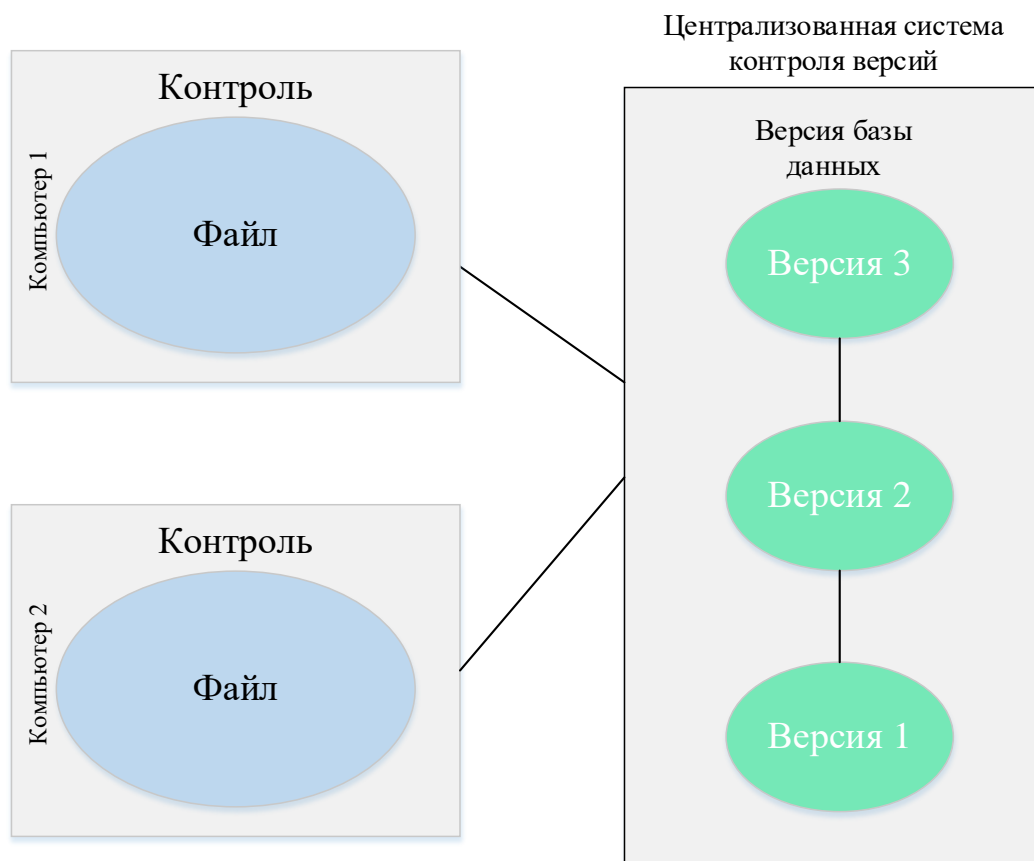


Рисунок 8 — Схема централизованного контроля версий

Такая архитектура позволяет видеть, кто и чем занимается в проекте, чёткое администрирование и разделение обязанностей, по сравнению с локальными базами.

Из недостатков можно выделить уязвимость центрального сервера. Если сервер выключается, то разработчики ничего не смогут сделать и все новые данные, которые не были сохранены, будут потеряны, а также при выходе сервера из строя, теряется информация о всех версиях и восстановить её невозможно.

Распределённые системы контроля версий. Для того, чтобы не потерять в самый нужный момент всю информацию на сервере, используются распределённые системы контроля версий. Принцип работы заключается в том, что при выходе сервера из строя, на котором шла работа, любое клиентское хранилище может быть возвращено на сервер при помощи копирования для восстановления базы данных. При сохранении новых версий файлов, клиент создаёт копию всех данных (рисунок 9).

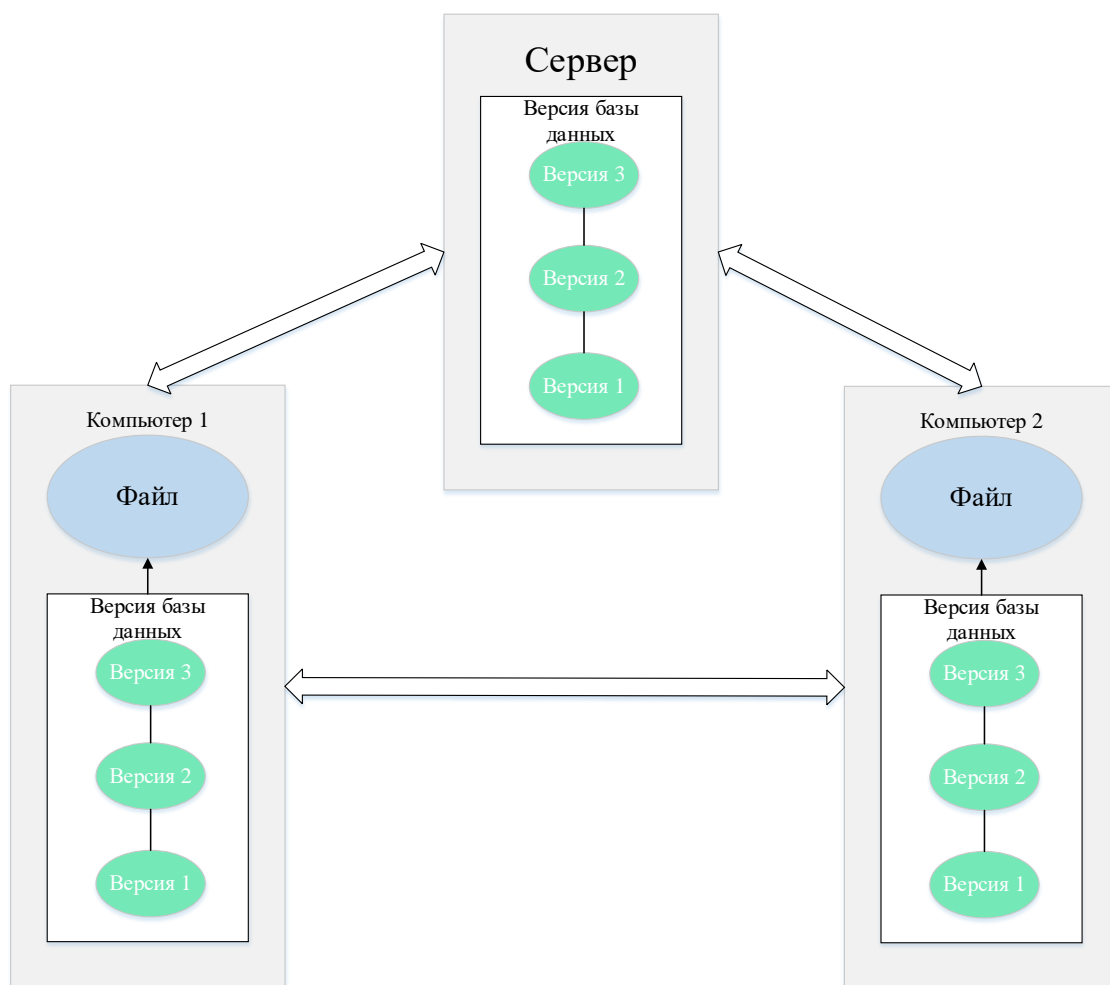


Рисунок 9 — Схема распределённой системы контроля версий

В большинстве из приведённых систем можно работать с несколькими хранилищами, то есть разрешается одновременно работать с разными группами людей в одном проекте, что позволяет вести несколько рабочих процессов [27].

2 СОЗДАНИЕ САЙТА-ВИЗИТКИ

2.1 Выбор технологий

2.1.1 Denwer

Денвер (от сокр. Д.н.в.р или ДНВР — джентльменский набор Web-разработчика) — набор дистрибутивов (локальный сервер WAMP) и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет-страниц) на локальном ПК (без необходимости подключения к сети Интернет) под управлением ОС Windows.

Денвер 3 включает в себя пакеты Apache 2.2.22 + SSL, PHP 5.3.13 + XDebug, MySQL 5.5, phpMyAdmin 3.5, многопроектность и работу с flash-накопителем. Для работы с ним необходимо, чтобы ПО соответствовало требованиям (таблица 1).

Таблица 1 — Системные требования для Joomla 3.x

Программное обеспечение	Рекомендуемая версия	Минимальная версия
PHP	5.3.1 +	5.3.1 +
Поддержка баз данных		
MySQL	5.1 +	5.1 +
MSSQL	10.50.1600.1 +	10.50.1600.1 +
Поддержка web-сервера		
Apache	2.x +	2.x +
Nginx	1.1	1.0
Microsoft IIS	7	7

Особенностью, отличающей Denwer от других WAMP-дистрибутивов, является автоматическая правка системного файла hosts, являющегося локальным аналогом DNS-сервера, что позволяет обращаться к локальным сайтам, работающим под управлением Денвера, по именам, совпадающим с именем папки, расположенной в каталоге home Денвера.

2.1.2 Joomla

Joomla — это система управления контентом (CMS), которая помогает сделать динамический и функциональный web-сайт с минимальными затратами. Система написана при использовании таких языков программирования, как PHP и JavaScript. Разработчиком CMS является The Joomla Project Team.

Joomla представляет собой бесплатную систему для создания веб-сайтов с открытым исходным кодом, что позволяет при желании отредактировать содержимое исходных файлов или изменить порядок выполнения программы [2]. Этот проект очень успешно развивается, в течение семи лет (таблица 2). Является одной из самых востребованных систем управления контентом [14].

Слово «joomla» происходит из языка суахили, является производным от слова «jumla», что в переводе значит «все вместе». Разработчики этим названием хотели показать свой подход к развитию системы.

Версии Joomla делятся на стандартную (STS) и долгосрочную (LTS) поддержку. Срок стандартной поддержки — 7 месяцев, долгосрочной — 22 месяца. Стандартная поддержка обновляется каждые 6 месяцев.

Таблица 2 — Версии Joomla

Версия	Дата выпуска	Дата окончания поддержки
1.0	22.09.2005	22.07.2009
1.5 (LTS)	22.01.2008	09.2012
1.6	10.01.2011	08.2011
1.7	19.07.2011	02.2012
2.5 (LTS)	24.01.2012	31.12.2014
3.0	27.09.2012	05.2013
3.1	24.04.2013	10.2014
3.3	30.04.2014	релиз 3.4
3.4	25.02.2015	релиз 3.5
3.5	21.03.2016	релиз 3.6
3.6	2016	релиз 3.7
3.7	2016	релиз 3.8

CMS Joomla! состоит из минимального инструментария для начальной установки и может пополняться по мере необходимости. Такой подход не загромождает административную панель ненужными элементами и позволяет снижать нагрузку на сервер, также, экономит место на хостинге.

Система управления содержимым является многоязычной, что позволяет отображать фронтальную и административную части на любом языке. Необходимый языковой пакет устанавливается средствами администрирования.

В данной работе использовалась версия Joomla 3.5.1. Её преимущества перед предыдущими версиями:

- адаптирована к мобильным устройствам;
- содержит множество шаблонов;
- имеет новый пользовательский интерфейс;
- использует JQuery JavaScript Lbriray;
- установка выполняется за 3 шага;
- языковой пакет можно установить из менеджера расширений;
- улучшения в Smart Search.

2.1.3 Git и Smart Git

В нашем случае для работы использовалась система контроля версий Git. Git является кроссплатформенным программным обеспечением, написанным на Си, Perl, Bash и командной оболочке UNIX. Впервые была выпущена 7 апреля 2005 года [17].

Чтобы не возникало трудностей при сохранении версий документов, использовался визуальный клиент системы управления версиями Git, который также является кроссплатформенным. В работе использовалась самая новая версия визуального клиента 7.1.3.

Выбор этой системы перед другими обосновывался следующими преимуществами:

- высокая производительность;
- удобность встраиваемых команд и скрипты;
- интуитивно понятный интерфейс;
- подсветка нового содержимого.

2.3 Разработка проекта

Разработка сайта-визитки будет в соответствии с этапами создания любого информационного Интернет-ресурса. Для создания сайта-визитки необходимо установить локальный сервер, систему управления содержимым и плагины. В роли локального сервера будет выступать Denwer, а CMS — Joomla, системой контроля версий будет Git.

2.3.1 Установка программного обеспечения

Для установки Denwer необходимо запустить файл установщика. После запуска появится окно установки (рисунок 10).

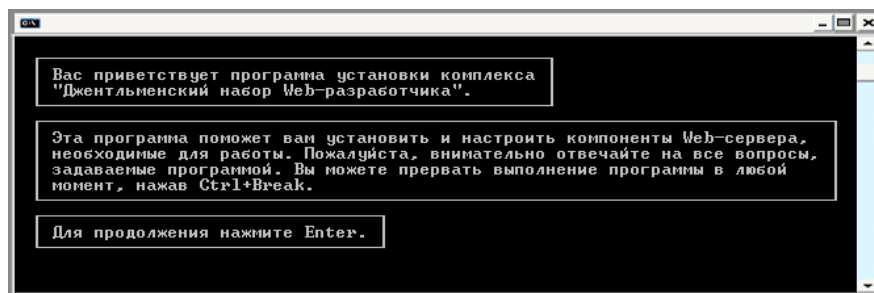


Рисунок 10 — Окно установки Denwer

Затем необходимо выбрать каталог, в который будет установлен локальный сервер (рисунок 11).

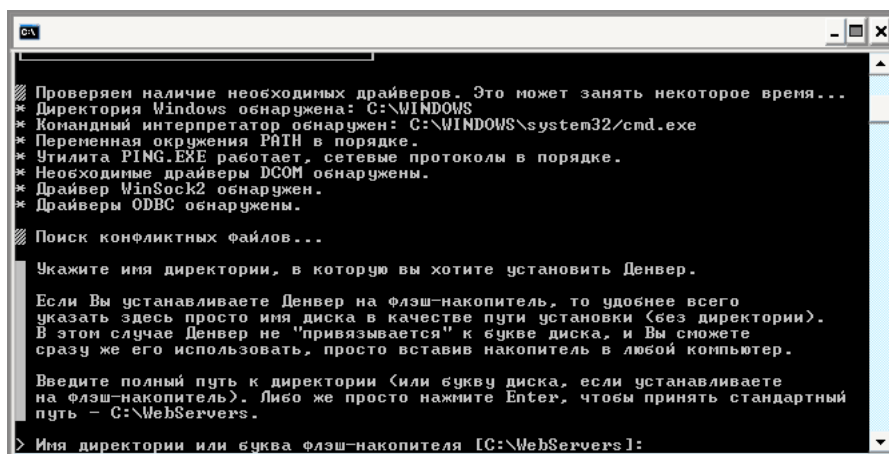


Рисунок 11 — Выбор каталога для установки Denwer

После этого необходимо задать имя виртуального диска (рисунок 12).

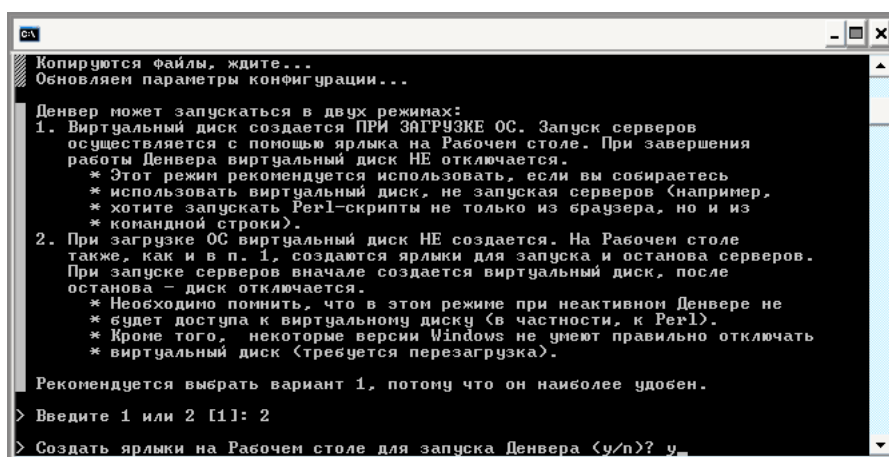


Рисунок 12 — Выбор имени виртуального диска

После всех этих операций будет произведено копирование файлов программы и выбор одного из вариантов запуска программы (рисунок 13).

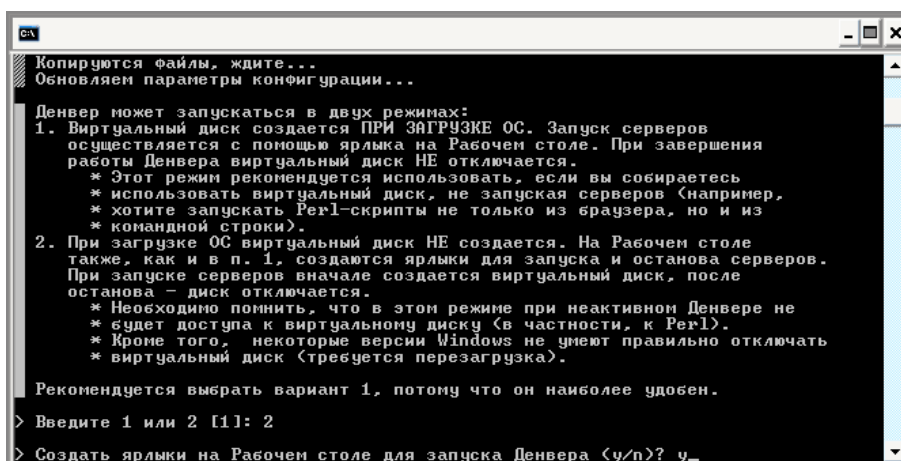


Рисунок 13 — Выбор варианта запуска программы

Если в течение установки не возникло никаких ошибок и препятствий, то в браузере откроется окно (рисунок 14).

Денвер успешно установлен

Чтобы начать использовать Денвер, проделайте следующие действия:

1. Запустите Денвер, воспользовавшись ярлыком **Start Denwer** на Рабочем столе. Если вы не создавали ярлыки, то можно запустить Денвер по команде `C:\Denwer\denwer\Run.exe`.
2. Откройте браузер и перейдите по адресу <http://localhost>.
3. Вы должны увидеть главную страницу Денвера.
4. *Примечание:* если вы используете Skype, убедитесь, что он не занимает порты 80 и 443, необходимые для работы Денвера (Инструменты - Настройки - Соединение - Использовать порты... должно быть отключено). Либо же просто завершите работу Skype.
5. **Внимание!** Если после запуска Денвера <http://localhost> не открывается, проверьте, не блокируется ли Денвер вашим антивирусом или фаерволом. Например, были замечены проблемы с NOD32 в Windows XP (в нем нужно добавить процесс `Z:\usr\local\apache\bin\httpd.exe` в список исключений, это можно сделать в окне IMON/Настройка/Разное/Исключение).

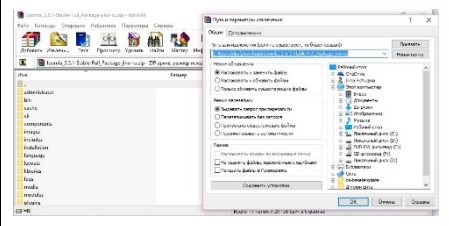
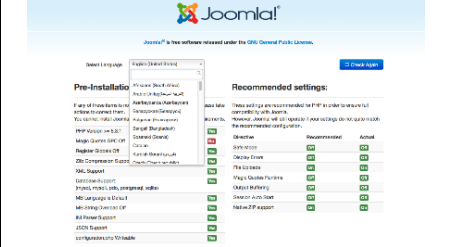

Если по каким-то причинам Денвер не заработал, свяжитесь, пожалуйста, с разработчиками: <http://forum.dklab.ru/denwer/bugs/>. Прикрепите к сообщению следующую информацию:


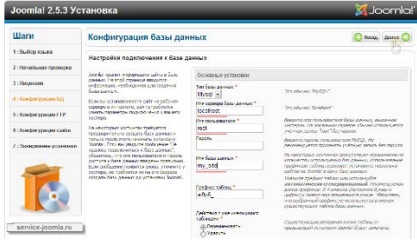

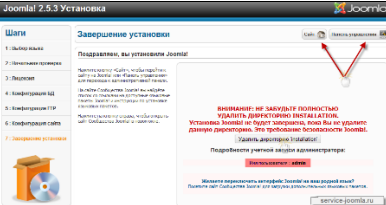

1. При каких условиях проявился баг? Что вы сделали перед тем, как его зафиксировали?
2. Точную версию Вашей OS (можно получить по команде `windows`, запущенной в Командной строке).
3. Файл `netstat.txt`, получившийся в результате работы команды `netstat -nb > C:\netstat.txt` (кстати, это

Рисунок 14 — Окончание установки Denwer

После установки Denwer приступаем к установке CMS Joomla, которая представлена в технологической карте (таблица 3).

Таблица 3 — Технологическая карта установки CMS Joomla

Название операции	Приёмы и действия	Контроль выполнения
Веб-установщик Joomla!	<ol style="list-style-type: none"> 1. Скачать Joomla 2.5 с сайта http://joomla.org; 2. Распакуйте файлы в общую корневую папку документов веб-сервера (например: /htdocs); 	
Предварительная проверка установки	<ol style="list-style-type: none"> 1. В поисковой строке браузера наберите localhost/; 	
Проверка совместимости с сервером	<ol style="list-style-type: none"> 1. Если нет никаких предупреждений, то переходим к следующей вкладке, нажимая <i>далее</i>; 	

<p>Лицензия GNU/GPL</p>	<ol style="list-style-type: none"> 1. Прочитать лицензию; 2. Согласиться с ней, нажав кнопку <i>далее</i>; 	
<p>Настройка базы данных</p>	<ol style="list-style-type: none"> 1. Задать тип базы данных; 2. Указать имя сервера базы данных, в нашем случае это <i>localhost</i>; 3. Напишите имя пользователя <i>root</i>; 4. Поля <i>Пароль</i> и <i>Префикс таблиц</i> оставьте пустыми; 5. В ячейке <i>Имя базы данных</i> дайте любое название ; 	
<p>Настройка конфигурации сайта</p>	<ol style="list-style-type: none"> 1. В главном окне конфигурации необходимо описать сайт и администратора; 2. Напишите имя вашего сайта; 3. Задайте <i>логин администратора</i> и <i>пароль</i>, затем введите его ещё раз для подтверждения; 4. Ниже кликнуть по кнопке <i>Установка демо-данных</i>, чтобы упростить работу с Joomla!; 5. Кликните по кнопке <i>далее</i>; 	
<p>Завершение установки</p>	<ol style="list-style-type: none"> 1. Для завершения установки необходимо <i>удалить директорию Installation</i>; 	
<p>Вход в панель управления</p>	<ol style="list-style-type: none"> 1. В поисковой строке браузера наберите http://localhost/joomla/administrator 	

После установки CMS, необходимо установить систему контроля версиями Smart Git.

Вся работа начинается с создания удалённого *git-репозитория*. Для его хранения использовался публичный сервис *github*, в котором необходимо зарегистрироваться. Выбор пал именно на этот сервис, потому что он является самым популярным хранилищем для разработчиков.

В странице профиля находится функция *Create a repository* (рисунок 15). Переходим по ссылке.



Рисунок 15 — Создание нового репозитория

Появляется окно для создания хранилища, в котором нужно указать имя, описание (необязательно), права доступа. Кликаем *Create repository*. (рисунок 16).

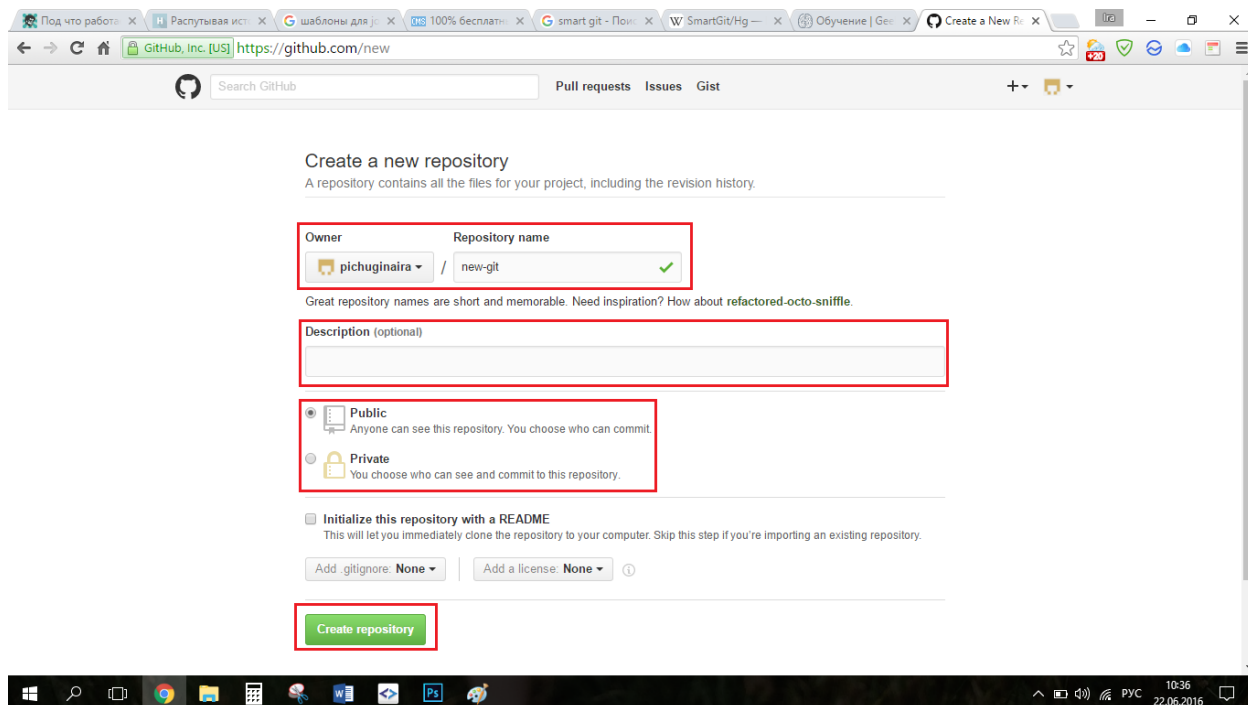


Рисунок 16 — Создание репозитория

Далее устанавливаем саму программу. Устанавливается как любая другая. После завершения установки открываем её и открывается окно с элементами управления (рисунок 17).

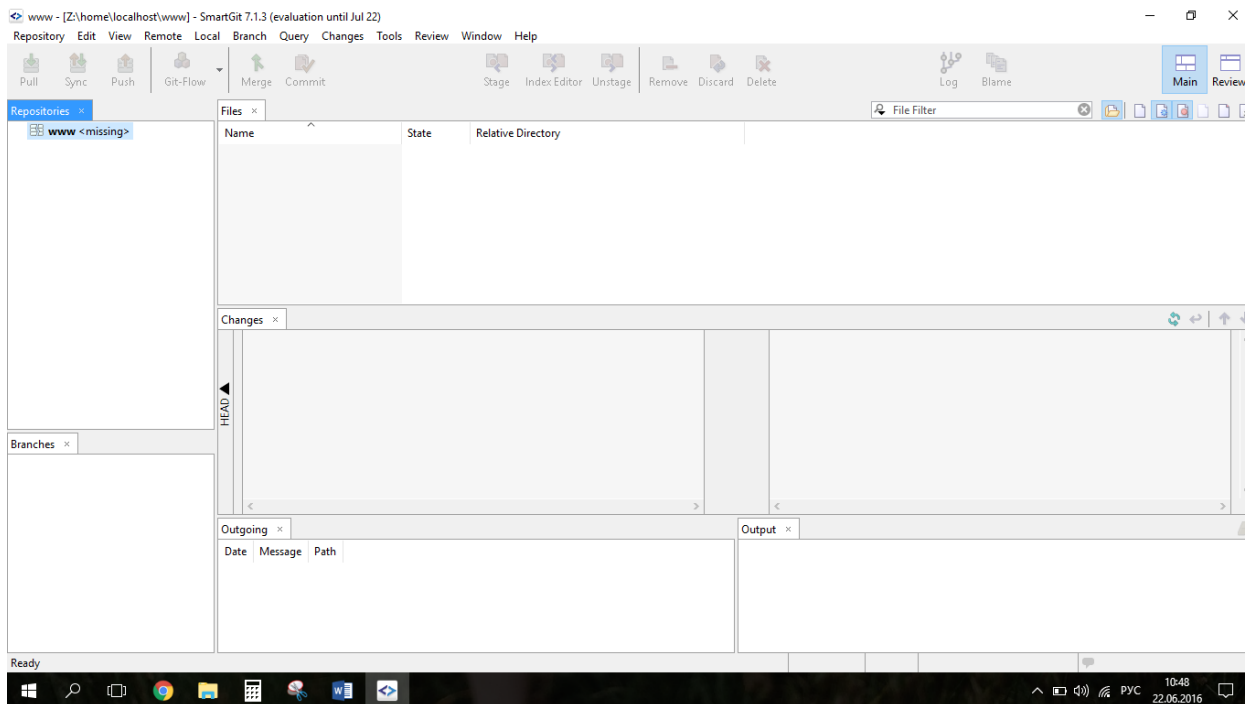


Рисунок 17 — Внешний интерфейс Smart Git

Для того, чтобы создать локальную версию данных, в первую очередь нужна ссылка на удалённый репозиторий, которая была указана при его завершении создания на сайте *github* (рисунок 18).



Рисунок 18 — Ссылка на удалённое хранилище

Для создания локального репозитория в программе Smart Git необходимо перейти на вкладку *Repository* → *Clone repository* (рисунок 19).

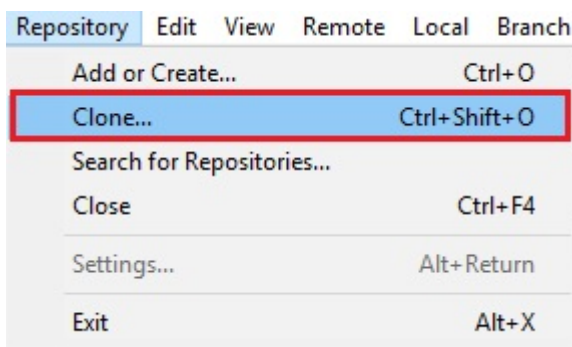


Рисунок 19 — Кнопка для создания локального репозитория

Появляется окно, в котором необходимо указать ссылку на удалённый репозиторий (рисунок 20).

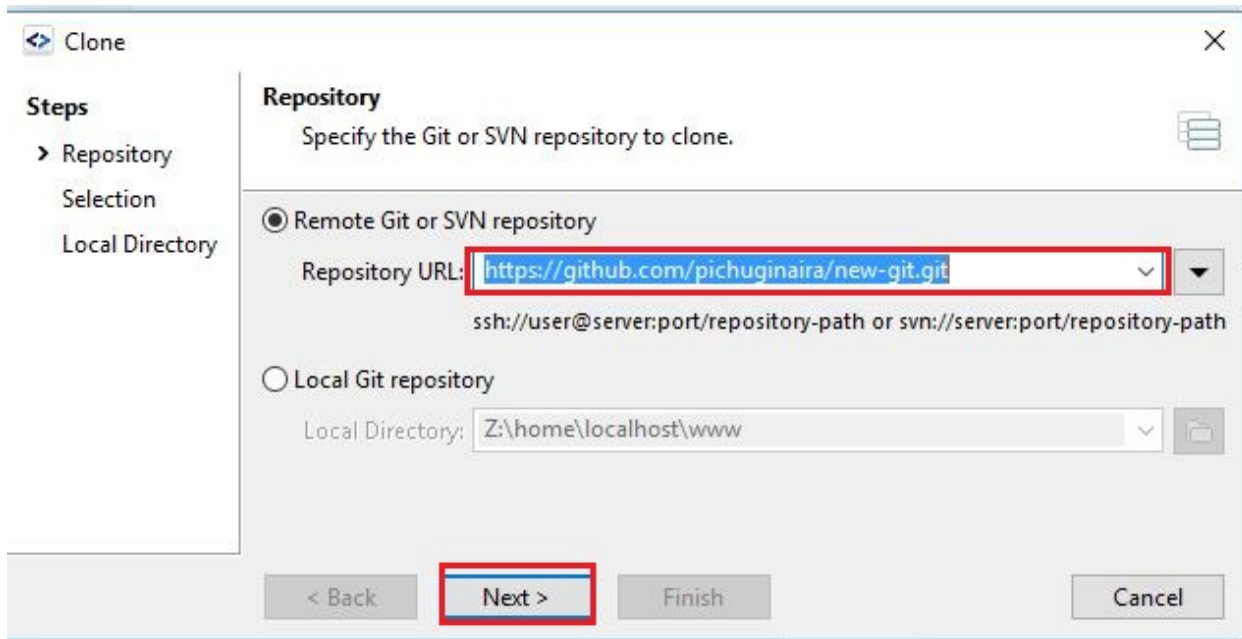


Рисунок 20 — Указание на удалённый репозиторий

Затем указываем папку, в который будет храниться локальный репозиторий (рисунок 21).

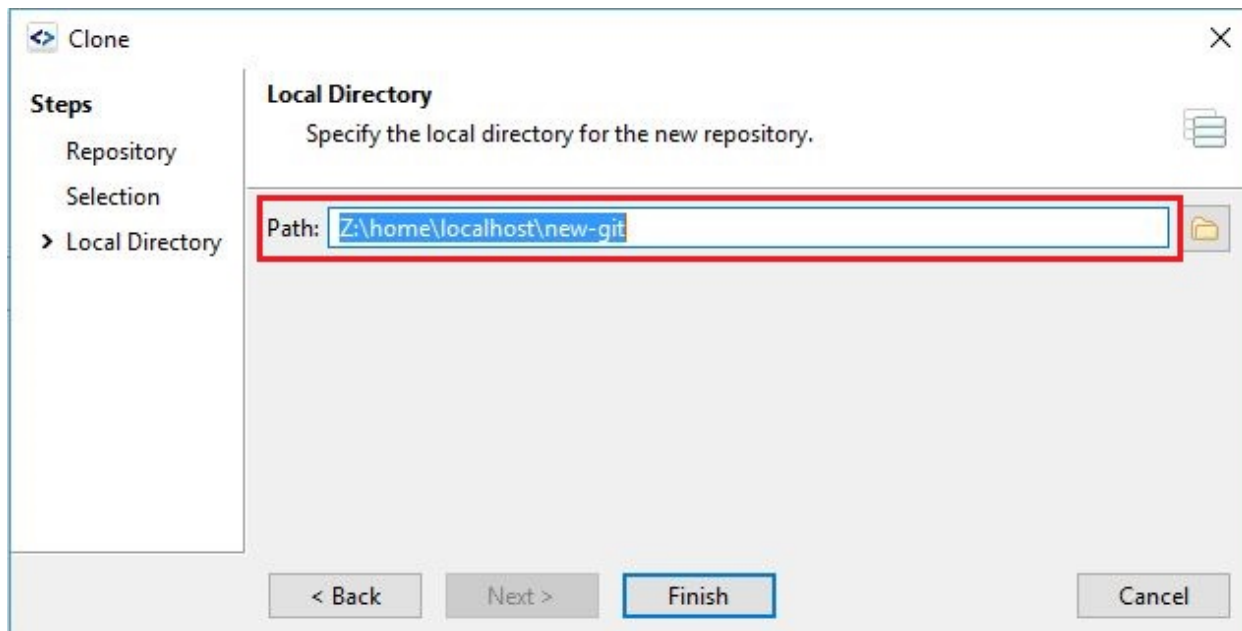


Рисунок 21 — Путь к папке для хранения локального репозитория

Чтобы убедиться, что создание локального сервера прошло успешно и папка для хранения создана, необходимо кликнуть по *Open in explorer* (рисунок 22).

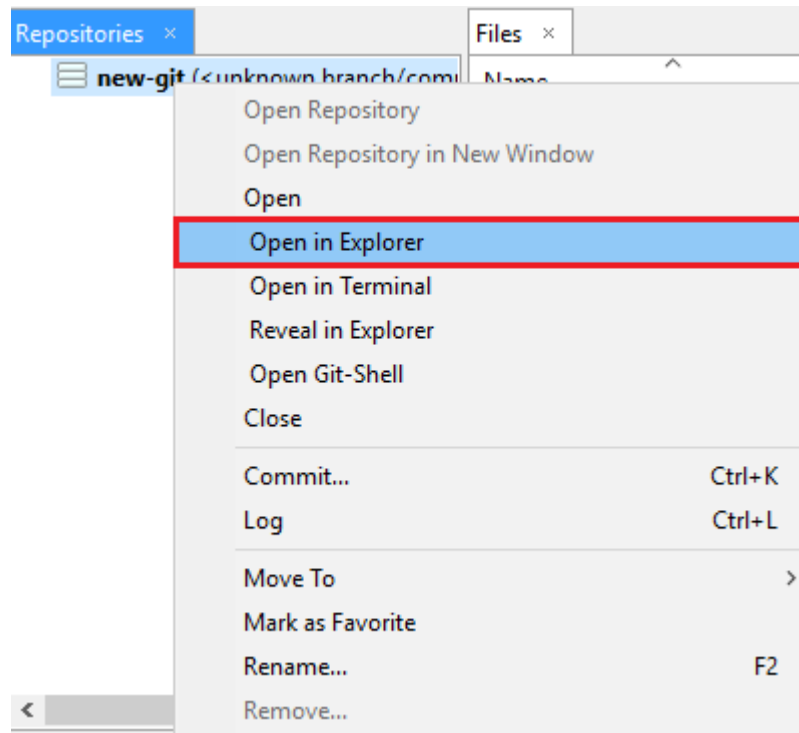


Рисунок 22 — Открытие папки локального репозитория

Эта папка находится именно в том каталоге, который был указан при установке (рисунок 23).

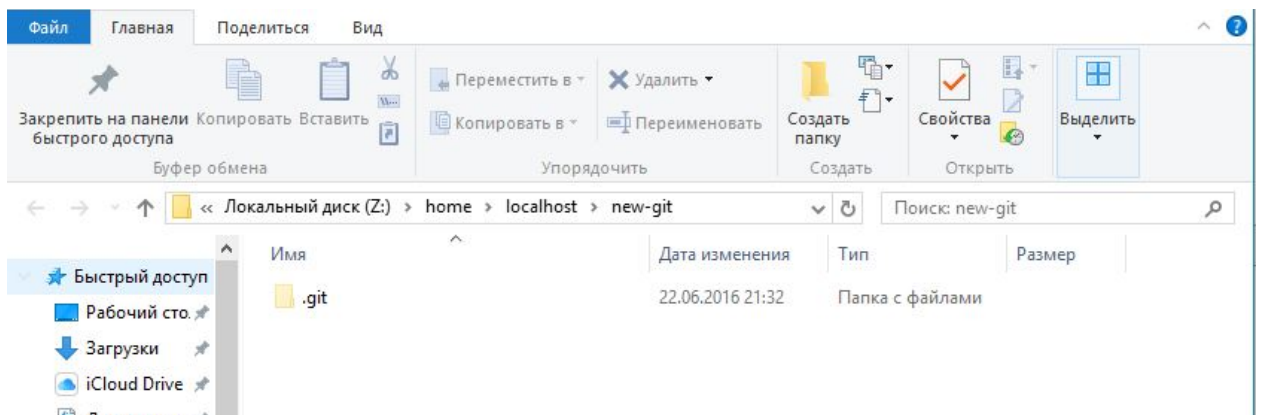


Рисунок 23 — Папка локального репозитория

Далее необходимо только создавать копии с изменениями, о которых Smart Git будет оповещать. Производятся эти изменения при помощи нажатия кнопки *Commit* (рисунок 24).

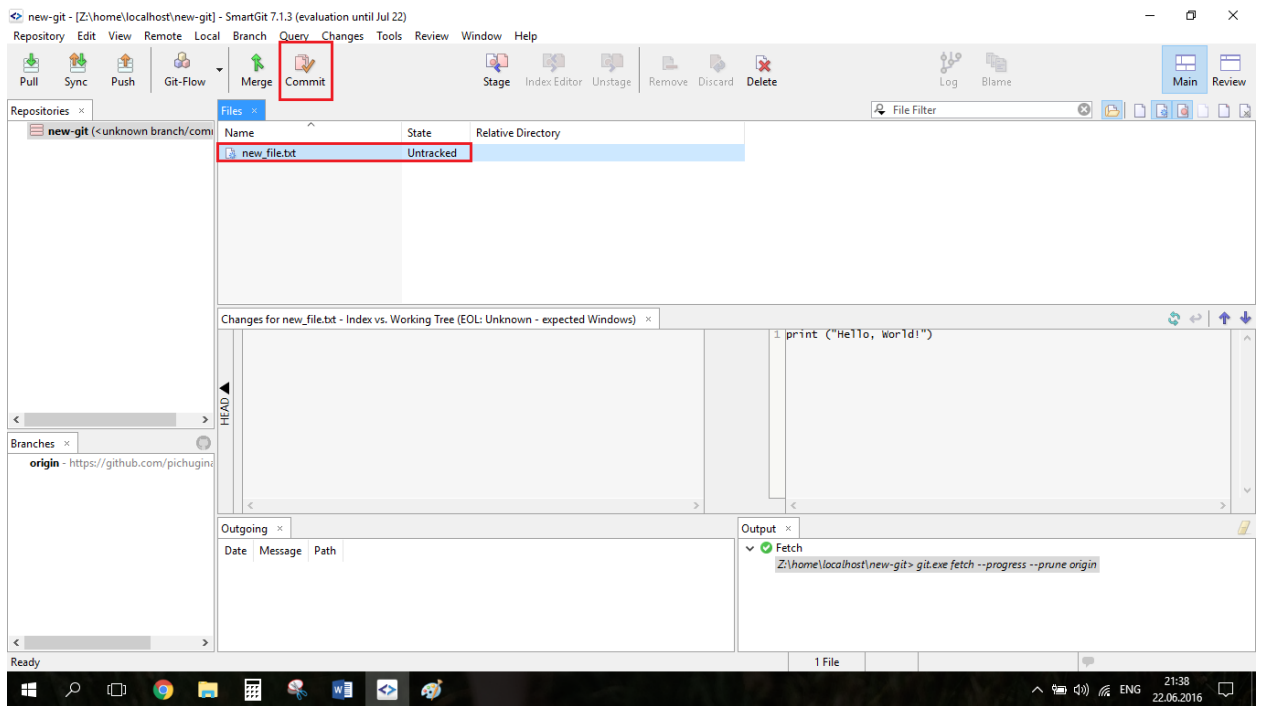


Рисунок 24 — Оповещение о добавлении нового файла

После нажатия *Commit* появляется окно, в котором предлагается задать имя для сохранения. Задаём имя и кликаем *Commit* (рисунок 25).

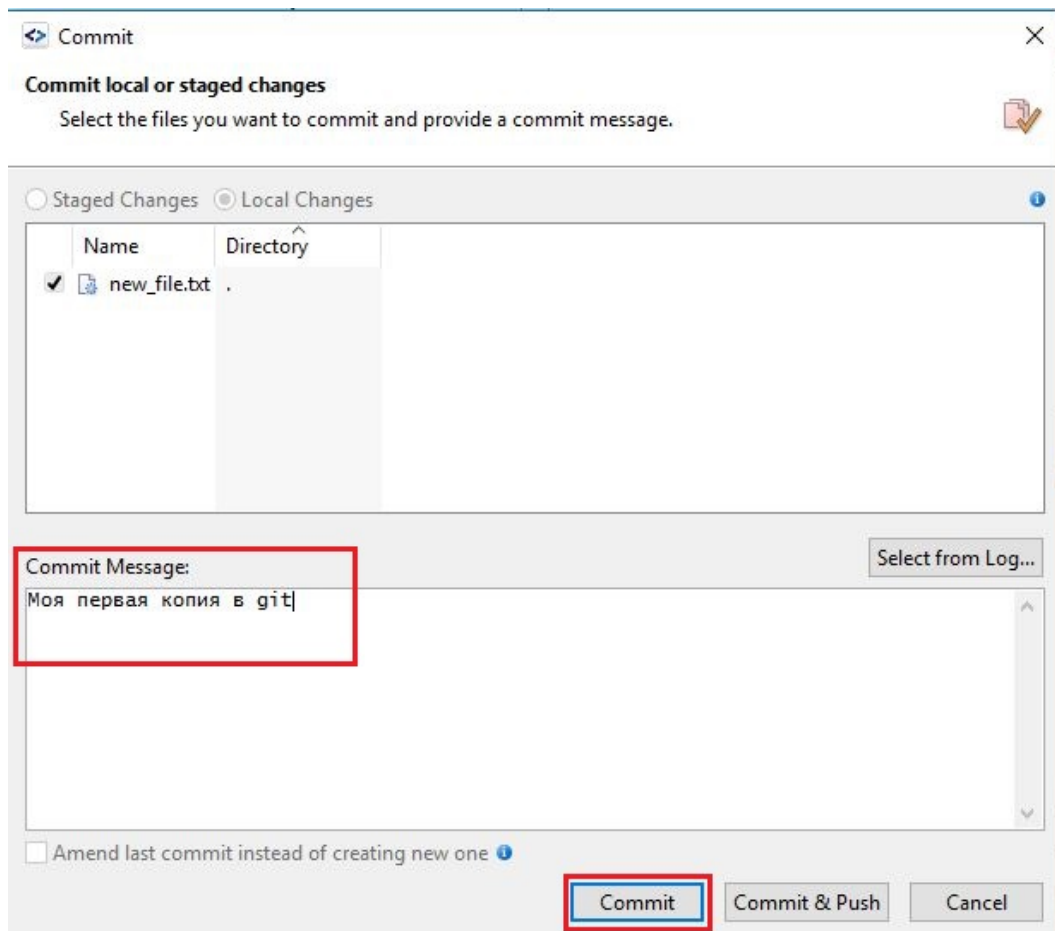


Рисунок 25 — Создание копии в репозиторий

Теперь необходимо локальный репозиторий отправить на удалённый репозиторий. Для этого нажимаем *Push*, выбираем пункт *Current branch 'master'*, а затем *Push* (рисунок 26). Таким образом локальная копия отправляется на удалённый сервис.

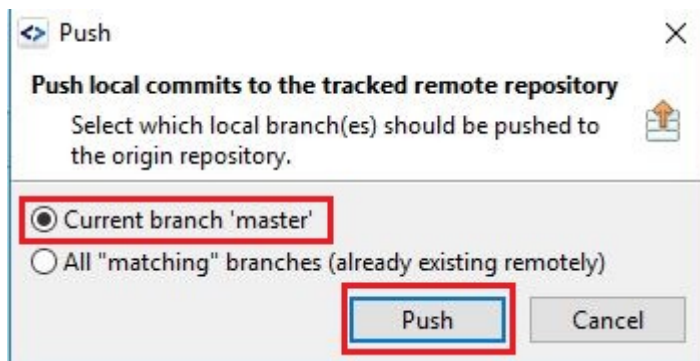


Рисунок 26 — Отправка данных на удалённый репозиторий

После произведённой операции Smart Git оповещает об успешно проведённой операции.

Данные операции выполняются после каждой созданной копии.

2.3.2 Заполнение сайта-визитки данными

Для начала работы с Joomla, необходимо зайти в панель управления CMS. Набирая в браузере в поисковой строке localhost/joomla/administrator, попадаем на начальную страницу. Здесь нужно ввести логин и пароль администратора, которые задавались при установке системы управления содержимым (рисунок 27).

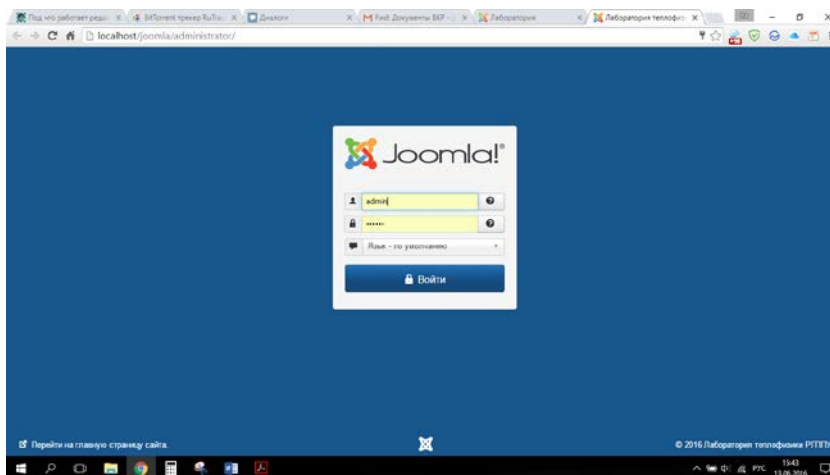


Рисунок 27 — Окно входа в панель управления Joomla

После ввода логина и пароля попадаем на главную страницу CMS (рисунок 28). Слева находятся все разделы, которые необходимы для разработки сайта, в центральной части отображаются последние обработанные и измененные файлы, также сразу же предлагается хостинг для размещения создаваемого сайта.

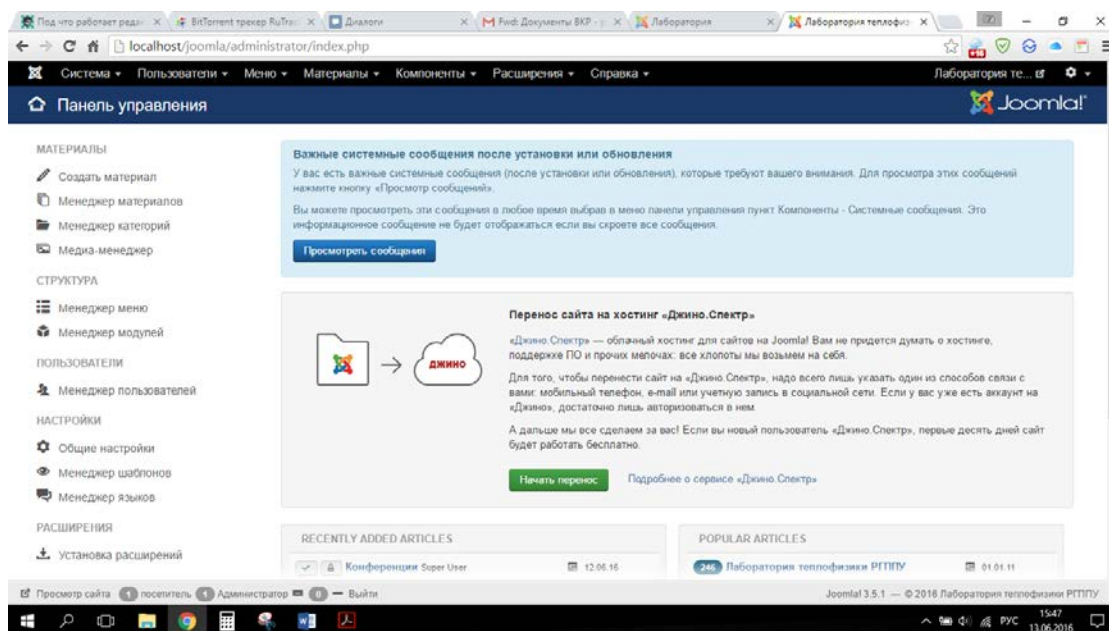


Рисунок 28 — Внешний вид интерфейса Joomla

Чтобы начать работу с будущим сайтом, нужно установить шаблон. В открытом доступе их достаточное количество, но многие из них платные, стоимость их варьируется в диапазоне от 19\$ до 69\$. Многие сайты, которые распространяют шаблоны для Joomla, предлагают подписку, в стоимость которых включено определённое количество скачиваемых шаблонов.

Но можно найти и бесплатные шаблоны, их достаточно, чтобы выбрать нужный по определённой тематике.

Также, Joomla 3.5.1 имеет два встроенных шаблона: Veez3 (рисунок 29) и Protostar (рисунок 30).

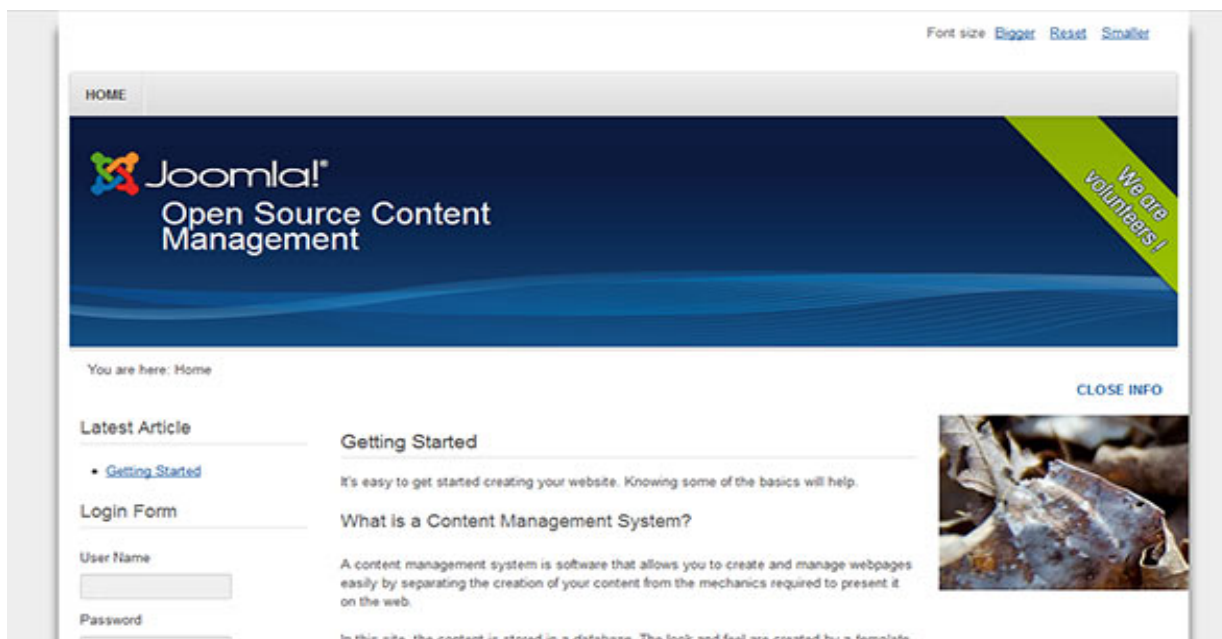


Рисунок 29 — Тема Beez3

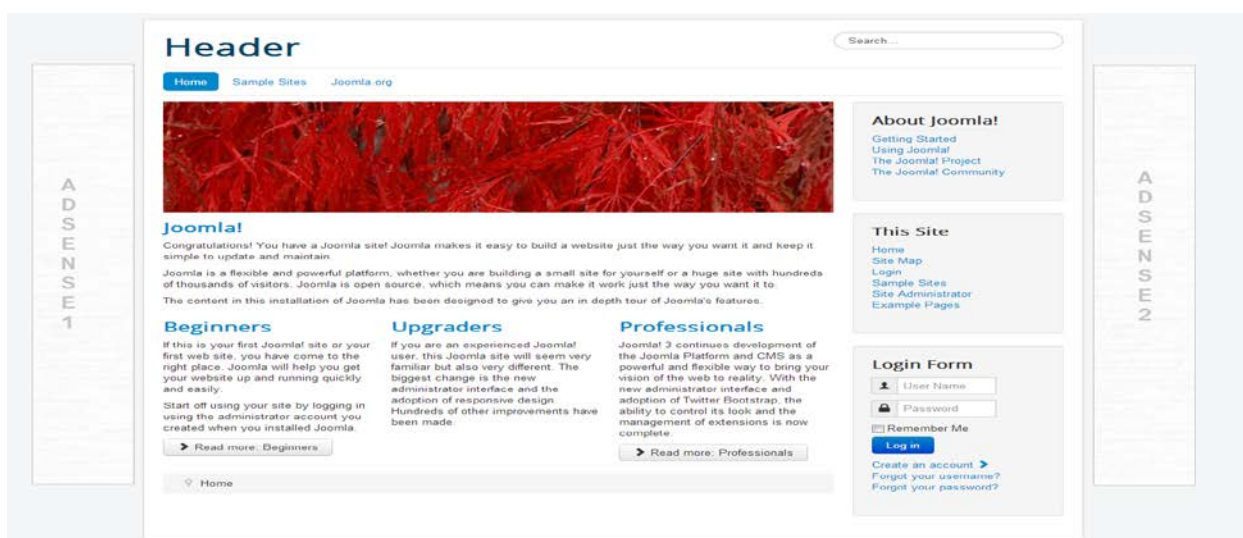


Рисунок 30 — Тема Protostar

Так как администратор имеет возможность изменять шаблоны, то был выбран стандартный Protostar. После изменения цвета фона, удаления ненужной информации, добавления модулей, логотипа университета и изменения изображения на главной странице, был получен следующий внешний вид главной страницы сайта-визитки лаборатории теплофизики РГПУ (рисунок 31).

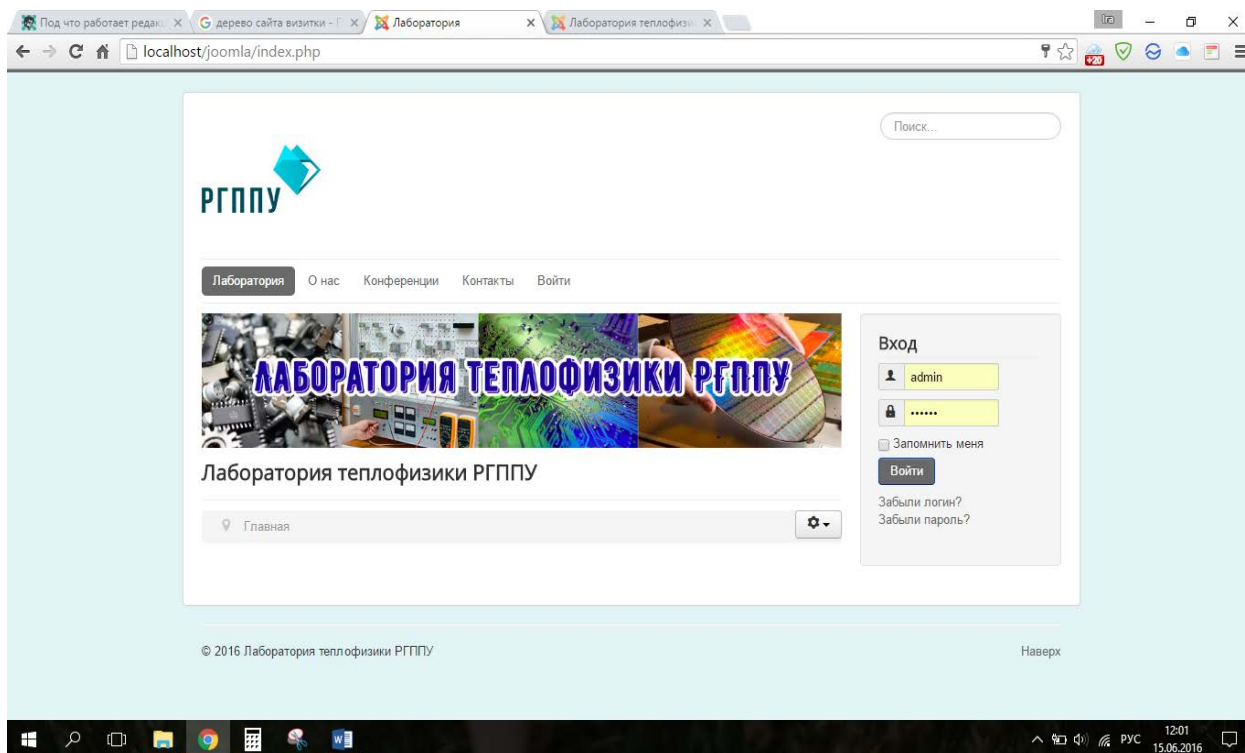


Рисунок 31 — Главная страница сайта лаборатории теплофизики

Для создания меню необходимо перейти во вкладку *Меню* → *Менеджер меню* → *Создать меню* (рисунок 32)

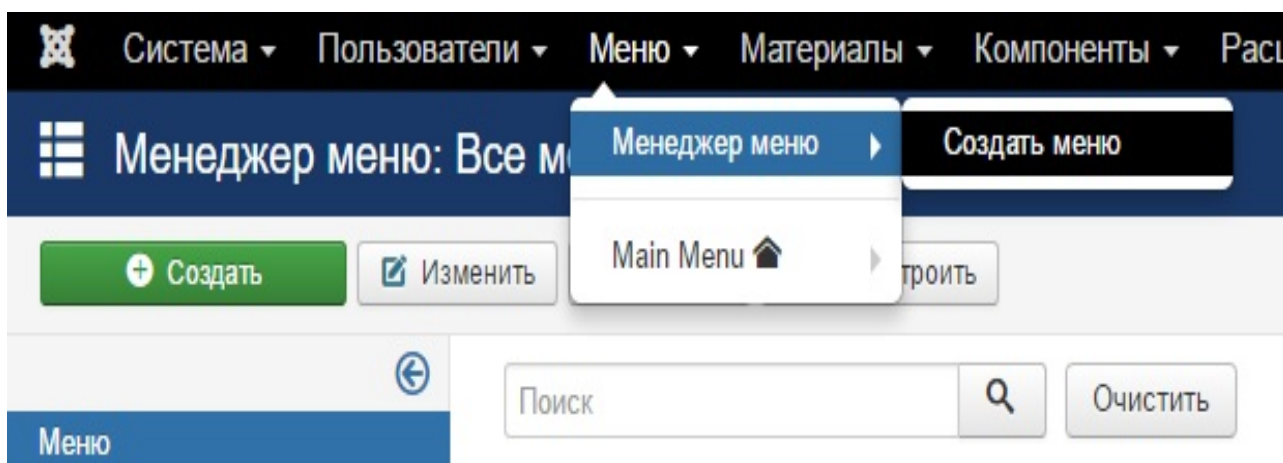


Рисунок 32 — Вкладка Меню

Откроется страница для создания меню, где необходимо задать имя меню, его тип и описание (необязательно) (рисунок 33).

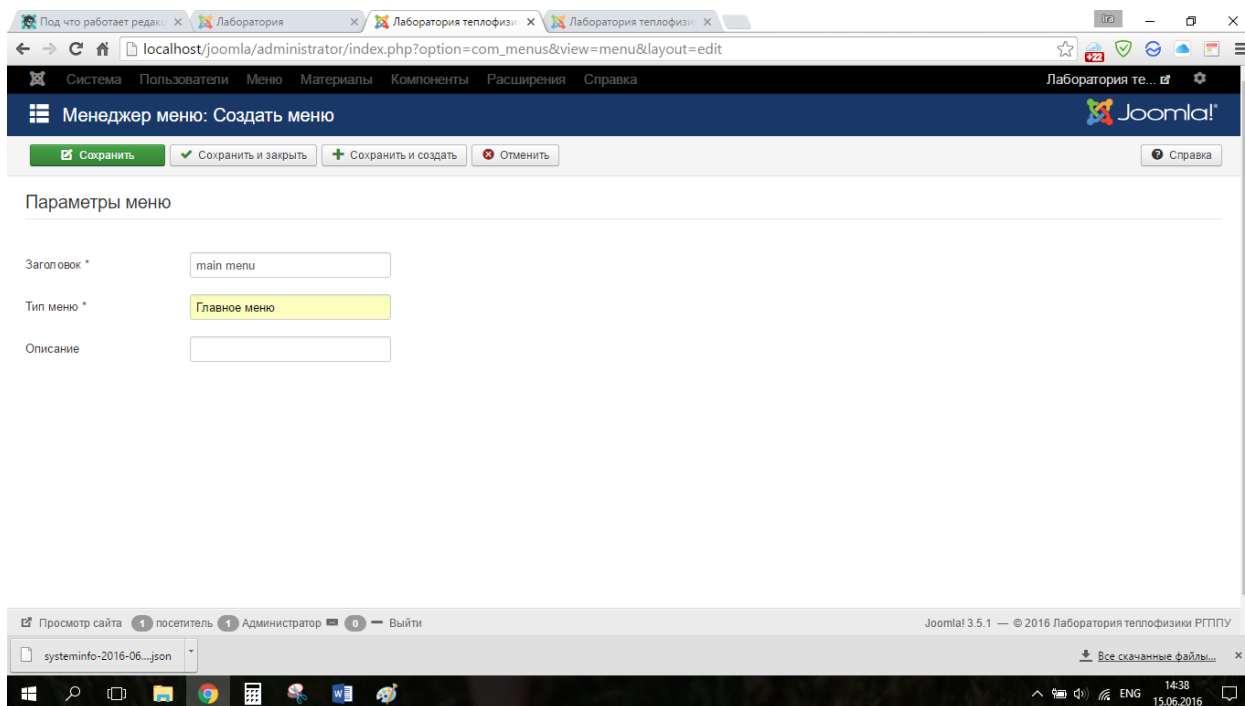


Рисунок 33 — Параметры меню

Таким образом было создано новое меню (рисунок 34), которое в дальнейшем можно заполнить всеми необходимыми полями.

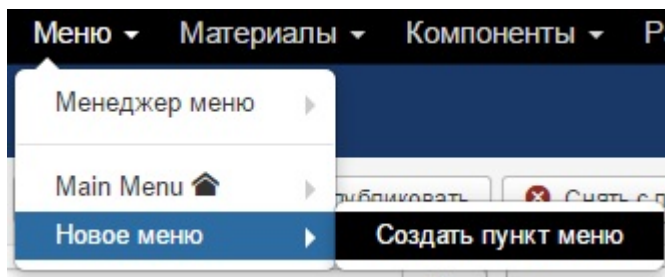


Рисунок 34 — Новое меню

Для создания нового материала необходимо из панели управления перейти в раздел *Материалы* и кликнуть на *Создать материал* (рисунок 35).



Рисунок 35 — Раздел Материалы

При переходе на создание нового материала, откроется страница с редактором, позволяющим оформлять текст в соответствии с требованиями. Для завершения создания новой статьи, необходимо нажать *Сохранить* (рисунок 36).

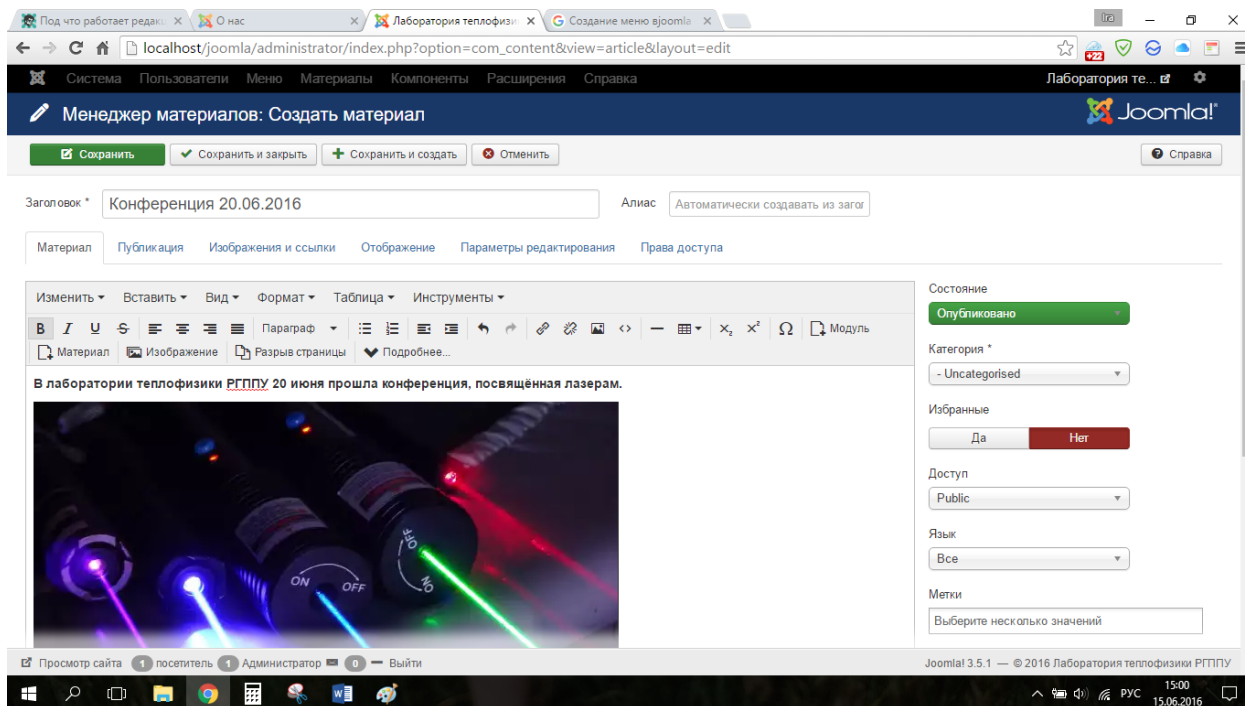


Рисунок 36 — Создание новой статьи

После создания всех необходимых меню и модулей добавляем информацию на страницы сайта. Во вкладке *О нас* будет находиться информация о сотрудниках лаборатории. В панели управления в разделе *Материалы* открываем *Менеджер материалов* и выбираем необходимый материал (рисунок 37).

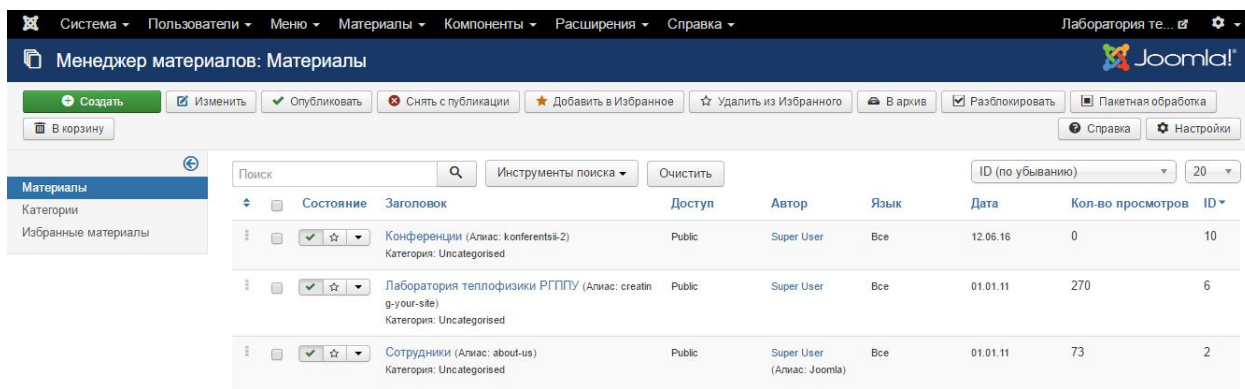


Рисунок 37 — Менеджер материалов

После заполнения раздела и его сохранения, переходим на предпросмотр сайта, нажав в правом верхнем углу по названию создаваемого сайта (рисунок 38), и выглядеть статья будет следующим образом (рисунок 39).



Рисунок 38 — Кнопка для перехода к предпросмотру сайта

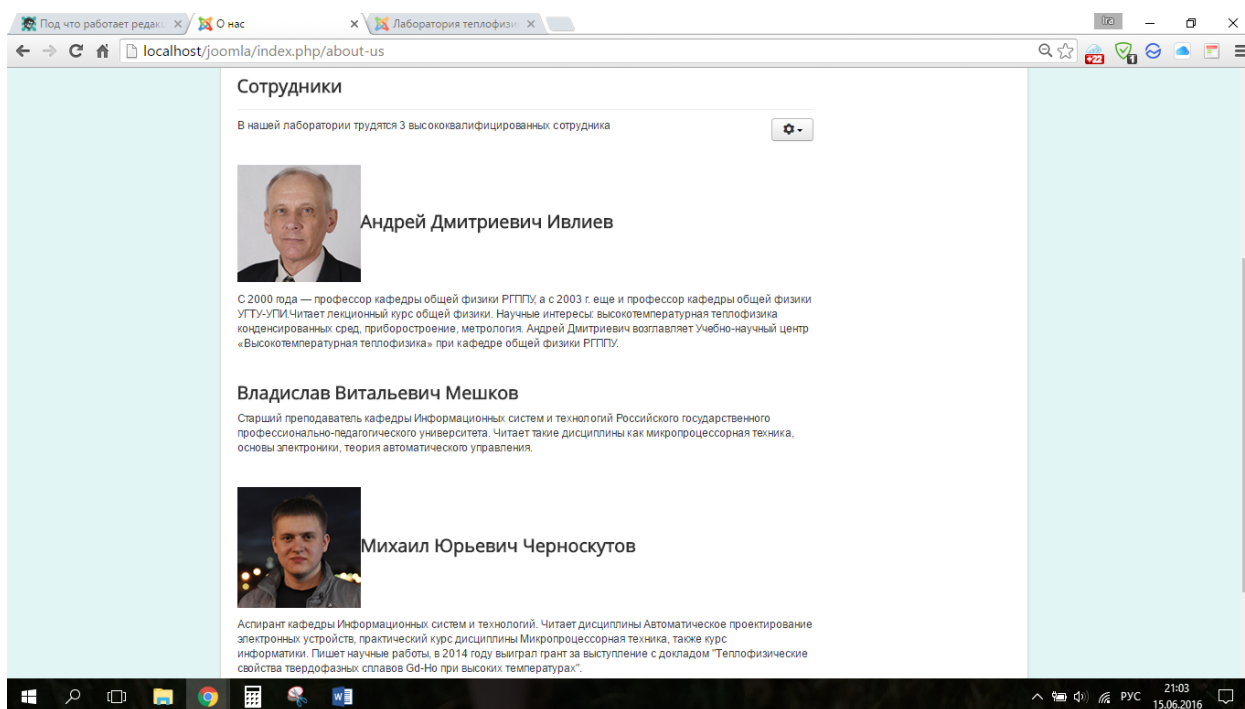


Рисунок 39 — Заполненный раздел О нас

Точно также заполняются и другие раздела сайта-визитки лаборатории теплофизики, такие как Контакты, Конференции и форма для входа.

ЗАКЛЮЧЕНИЕ

Сайт-визитка является одним из ведущих видов web-ресурсов. Их количество возрастает с каждым днём, так как даже самая небольшая компания представляет себя в виде информационного ресурса. По этой причине количество технологий разрастается, а существующие на данный момент улучшаются и становятся более функциональными.

В ходе выполнения выпускной квалификационной работы была достигнута основная цель — разработка web-сайта лаборатории теплофизики РГППУ. Созданный сайт предоставляет всю необходимую информацию об оборудовании, имеющимся в лаборатории, предоставляемых услугах, сотрудниках и их контакты. Сайт-визитка создавался в целях продвижения лаборатории.

В ходе работы были рассмотрены существующие компьютерные технологии, программное обеспечение и методы представления информации на web-ресурсах. В роли системы управления контентом выступила CMS Joomla, система управления версиями Git и локальным сервером является Denwer.

Процесс создания сайта-визитки производился в соответствии с поставленными задачами, в результате которых было сделано:

- обсуждение проекта с заказчиком;
- проанализированы существующие источники и литература по данной теме;
- рассмотрены и выбраны технологии и методы организации и осуществления web-ресурса;
- выбран шаблон сайта-визитки;
- заполнение сайта-визитки необходимым содержанием.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Граф Х. 10 легких шагов к освоению Joomla! 3 [Текст] / Граф Х., перев. Баскинов А. — Ф.: Сосоате, 2012. — 110 с.
2. Декстер М., Лэндри Л. Joomla!: программирование [Текст] / Декстер М., Лэндри Л. Пер. с англ. — М.: ООО «И.Д. Вильямс», 2013. — 592 с.: ил. — Парал. тит. англ.
3. Мержевич Влад. Вёрстка веб-страниц [Текст] / Мержевич В. — htmlbook.ru, 2011. — 378 с.
4. Никсон Р. Создаём динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS. 2-е изд. [Текст] / Никсон Р. — СПб.: Питер, 2013. — 560 с.: ил.
5. Пётр Ташков. Веб-мастеринг. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрытка [Текст] / Ташков П. — СПб.: Питер, 2010. — 512 с.: ил. — (Серия «На 100%»).
6. Bintu Harwani. Foundations of Joomla. 2nd edition [Текст]. — Apress, 2015. — 512 p.
7. Jennifer Niederst Robbins. Learning web design. 4th edition [Текст]. — O'Reilly Media, Inc., 2012. — 624 p.
8. Jon Duckett. HTML & CSS. Design and build websites [Текст]. — John Wiley & Sons, Inc., 2011. — 514 p.
9. Lynn Beighley, Michael Morrison. Head first PHP & MySQL [Текст]. — O'Reilly Media, Inc., 2009. — 814 p.
10. Rachel Andrew. The CCS3 anthology. 4th edition [Текст]. — SitePoint, 2012. — 443 p.
11. Zakas N. The Document object model. 2nd edition [Текст]. — Professional JavaScript for Web Developers, 2009. — 317 p.
12. Блог. Сайт-визитка — что это? [Электронный ресурс]. — Режим доступа: <http://webciti.ru/blog/sait-vizitka> (дата обращения: 4.06.2016).

13. Википедия. Сайт-визитка [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Сайт-визитка> (дата обращения: 4.06.2016).
14. Википедия. Система управления версиями [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_версиями (дата обращения: 8.06.2016).
15. Википедия. Система управления содержимым [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_содержимым (дата обращения: 14.06.2016).
16. Википедия. CSS [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/CSS> (дата обращения: 7.06.2016).
17. Википедия. Git [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Git> (дата обращения: 8.06.2016).
18. Википедия. HTML5 [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/HTML5> (дата обращения: 6.06.2016).
19. Википедия. JavaScript [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 7.06.2016).
20. Википедия. Python [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Python> (дата обращения: 14.06.2016).
21. Википедия. Smart Git/Hg [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/SmartGit/Hg> (дата обращения: 6.06.2016).
22. Всё о Joomla! [Электронный ресурс]. — Режим доступа: <http://joomla.ru/> (дата обращения: 4.06.2016).
23. Денвер — локальный сервер [Электронный ресурс]. — Режим доступа: <http://www.denwer.ru/> (дата обращения: 6.06.2016).
24. Для тех, кто делает сайты [Электронный ресурс]. — Режим доступа: <http://htmlbook.ru/html5> (дата обращения: 6.06.2016).

25. Как создать свой сайт на Joomla! Уроки для «чайников» [Электронный ресурс]. — Режим доступа: <http://joomlalessons.ru> (дата обращения: 15.11.15).

26. НОУ Интуит. Архитектура Joomla. Базовые сведения [Электронный ресурс]. — Режим доступа: <http://www.intuit.ru/studies/courses/3731/973/lecture/14904> (дата обращения: 25.12.2015).

27. Geekbrains. Git. Быстрый старт [Электронный ресурс]. — Режим доступа: <https://geekbrains.ru/courses/66> (дата обращения: 6.06.2016).

28. Git. Fast version control [Электронный ресурс]. — Режим доступа: <https://git-scm.com/book/ru/v1/Введение-О-контроле-версий> (дата обращения: 8.06.2016).

29. CMS magazine. Аналитический портал рынка веб-разработок [Электронный ресурс]. — Режим доступа: <http://www.ratingruneta.ru/cms/> (дата обращения: 12.06.2016).

ПРИЛОЖЕНИЕ

**Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический
университет»**

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
направление 44.03.04 Профессиональное обучение (по отраслям)
профиль «Энергетика»
профилизация «Компьютерные технологии автоматизации и управления»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Н. С. Толстова
« _____ » _____ 2016 г.

**ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра**

студентки 4 курса, группы КТэ-401 Пичугиной Ирины Александровны

1. Тема Разработка сайта лаборатории теплофизики
утверждена распоряжением по институту от 1Т г. № 1Т
2. Руководитель Чернокутов Михаил Юрьевич, ассистент кафедры ИС
3. Место преддипломной практики Уральский горнозаводской колледж имени Демидовых г. Невьянск
4. Исходные данные к ВКР Хаген Граф «10 легких шагов к освоению Joomla! 3»; Марк Декстер, Луис Лэндри «Joomla! Программирование»; Jennifer Niederst Robbins «Learning web desing»; Jon Duckett «HTML & CSS. Desing and Build Websites»
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
Теоретический раздел. Обсуждение проекта с заказчиком, рассмотрение технологий
Практический раздел. Выбор технологий, их описание
Методический раздел. Описание установки программного обеспечения, разбор интерфейса
Список используемых источников информации
Приложение
6. Перечень демонстрационных материалов
Презентация, выполненная средствами MS PowerPoint

