РЕФЕРАТ

Содержание

[ВВЕДЕНИЕ 3](#_Toc486088377)

[1 Общая характеристика ООО «Перфоманс» 4](#_Toc486088378)

[1.1 Основные виды деятельности 4](#_Toc486088379)

[1.2 SWOT-анализ ООО «Перфоманс» 8](#_Toc486088380)

[2 Программное обеспечение ООО «Перфоманс» 12](#_Toc486088381)

[2.1 Технологии и инструменты для разработки 12](#_Toc486088382)

[3 Разработка сайта для тур-фирмы «Азия Бас» 16](#_Toc486088383)

[3.1 Верстка основных страниц сайта 16](#_Toc486088384)

[3.2 Создание темы для CMS Drupal 7 20](#_Toc486088385)

[3.3 Расчет экономической эффективности создания сайта 58](#_Toc486088386)

[Заключение 65](#_Toc486088387)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 66](#_Toc486088388)

ВВЕДЕНИЕ

Целью данной работы является – сайт для тур-фирмы ООО «Азия Бас Тур».

Для достижения цели необходимо выполнить следующие задачи:

1. Изучить и проанализировать цикл разработки сайта в веб-студии «Перфоманс»;
2. на основе технического задания на создание веб-сайта и psd-макетов, сверстать основные страницы сайта, использую html для разметки, css-препроцессор less для написания стилей и JavaScript, в основном библиотеку jQuery для создания динамических эффектов на сайте;
3. создать тему для CMS Drupal 7.54 используя статичную верстку, т. е. добавить к сайту динамику, бэкэнд;
4. разместить каталог сайта на хостинге заказчика;
5. протестировать, исправить ошибки и доработать функционал сайта, по желанию заказчика;

1 Общая характеристика ООО «Перфоманс»

1.1 Основные виды деятельности [33]

Перфоманс – крупнейшая веб-студия Хабаровского края. Существует с 2007 года и занимается созданием, продвижением сайтов, дизайном, интернет-рекламой.

Разработка сайтов

Создание, продвижение, развитие – это слова, описывающий полный цикл любой веб-компании, которая занимается разработкой и созданием сайтов. В каждое слово включено много действий. Действия могут меняться местами, в зависимости от текущего состояния сайта, но все начинается с его создания.

В Хабаровске несколько веб-студий, предлагающих разработку сайтов, но качественное создание сайтов предлагает именно Перфоманс.

Стандарты качества, которых придерживается веб-студия «Перфоманс»:

* В основе – достоверная информация

Перед тем, как начать работу, разработчик должен знать о целях заказчика, решившего создать сайт. И в качестве источника информации будет являться не только заказчик, но и вся область отрасли.

Качество создания сайта, будет зависеть от качества собранных данных. Собранную информацию, необходимо зафиксировать и проанализировать. Как итог – техническое задание, где освещены не только карта сайта и программная платформа, но и абсолютно все аспекты и задачи будущего сайта.

* Организованность

У каждого производства должен быть тот, кто отвечает за конечный его результат. Также у каждого проекта должен быть такой человек – руководитель проекта со стороны разработчика или проект-менеджер. Проект-менеджер – это сотрудник, который настраивает и обеспечивает взаимодействие всех участников команды разработки (программист, дизайнер).

* Каждый сайт уникален

Уникален каждый человек, заказывающий создание сайтов. Хабаровск – деловой город, здесь много компаний, но каждая из них имеет свою индивидуальность. Мы умеем углядеть эту особенность компании, даже если она не бросается в глаза и не видна самому заказчику. Ведь эксклюзивность – это конкурентоспособность, прежде всего. Для этого нужно быть искренне заинтересованным в разработке и создании неповторимого сайта, иметь пытливый ум, не лениться во все вникать и иметь нестандартное мышление. Мы относимся к каждому веб-сайту, как к замечательной возможности реализовать что-то новое и отличное от других.

* Дизайн: говорящие идеи

Всё, что существует в техническом задании, в мыслях заказчика и разработчика, визуализируется в конкретном концепт-макете сайта. Дизайн сайта должен учитывать все, что было увидено и услышано ранее. Дизайнер может создать что-то типовое или что-то эксклюзивное – главное, чтобы осуществлялись все поставленные задачи. Опять же, в своей работе дизайнер ориентируется не на работу Хабаровских веб-студий, а как минимум на весь Рунет. Дизайн говорит на международном языке, встречает посетителя, информирует его, направляет именно туда, куда гость сайта сам желает попасть.

* Программирование – арсенал решений

Для качественного программирования сайта имеется большой арсенал технических решений. Компания Перфоманс разрабатывает сайты для самых популярных систем управления содержимым. Также за все время работы студии накопился архив, проверенных решений. Но даже если нет готового решения, мы сумеем быстро сориентироваться и найти его, ведь в распоряжении всегда мировая коллекция накопленных знаний в сети интернет.

Продвижение сайтов

Для того, чтобы успешно продвинуться на рынке, компания должная донести до потребителей информацию о себе. Интернет – это самое популярный и доступное СМИ. Поэтому, создав сайт, а в дальнейшем продвинув его – получается отличность возможность для развития бизнеса.

Причины обратиться за продвижением в веб-студию «Перфоманс»

1. Опыт работы более 10 лет

Компания «Перфоманс» работает уже больше 10 лет. За все время у веб-студии появилось много довольных клиентов. Благодаря качественному продвижению сайтов, много людей находят сайты компаний клиентов и приобретают их товары и услуги.

2. Оплата за результат.

Клиенты «Перфоманс» платят только за реальный результат. Результатом будет являться вывод ключевых запросов сайта на верхние позиции поисковых систем.

3. Прозрачная схема работы.

Наши методы работы прозрачны и понятны. Вы будете знать, чего мы хотим добиться, и какой объем работ проделаем для этого. Мы используем только законные «белые» схемы поискового продвижения сайтов, это гарантирует высокие позиции на протяжении долгого периода. Каждый месяц вы будете получать полный детальный отчет о проделанной работе и ее результатах.

4. Комплексный подход.

Веб-студия «Перфоманс» работает в полном цикле. Перед тем, как приступить к продвижению сайта, специалисты проверяют текущий и, по необходимости, дают рекомендации по улучшению. Работа ведется над контентом сайта, дизайном и юзабилити. Это необходимо для повышения конверсии.

5. Технологии индивидуальности.

Если использовать шаблонный подход к проекту, то его выполнение не даст эффективного результата. Для каждой компании необходим индивидуальный подход. После анализа деятельности заказчика, веб-студия «Перфоманс» занимается разработкой стратегии раскрутки сайта.

К каждому проекту прикрепляется личный менеджер. Он поможет решить вопросы, связанные с поисковой оптимизацией сайта, продвижением и раскруткой сайта, а также любые технические и маркетинговые вопросы.

Виды продвижения:

* Поисковое (СЕО) зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили, и их конкурентности. Составим для вас бюджет исходя из задач и статистики по вашим конкурентам. Появление в топ-10 поисковых систем Яндекс и Google в течении 2-3 месяцев. Сроки обусловлены технологическими особенностями.

Это основной вид поискового продвижения, который обеспечит вам максимум клиентов.

* Контекстное (Яндекс.Директ и Google.AdWords)

Цена зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили и их конкурентности.

Сроки — старт сразу в день пополнения баланса

Снятие денег за фактический переход на ваш сайт.

Вас видно наверху в поиске и на тысячах сайтов, состоящих в рекламной сети.

* В социальных сетях

Цена зависит от количества словосочетаний, по которым вы хотите, чтобы вас находили и их конкурентности. Можно настроить списание средств за показ или за переход.

Сроки — старт сразу в день пополнения баланса

Точный выбор аудитории для показа: пол, возраст, регион, интересы и т.д.

* Медийная реклама

Цена зависит от площадки, количества показов, тематики. Списывается за показы.

Сроки — ближайший свободный период

обычно старт в течение 1-2 недель.

В основном федеральные площадки: Яндекс.почта и Mail.почта, тематические разделы.

* Специализированные площадки

Цена зависит от площадки, количества показов, тематики. Списывается за показы.

Сроки ­ ближайший свободный период

обычно старт в течение 1-2 недель.

* Любая другая реклама
  + - Подготовим медиаплан по эффективному продвижению вашей компании:
    - массовые рассылки;
    - целевые рассылки;
    - статьи на разных площадках;
    - реклама в тематических роликах на YouTube;

1.2 SWOT-анализ ООО «Перфоманс»

Первым шагом в проведении SWOT анализа является определение сильных и слабых сторон товара или услуги.

1. Параметры, оказавшиеся лучше, чем у конкурентов – это сильные стороны продукта.
2. Параметры, оказавшиеся хуже, чем у конкурентов – это слабые стороны продукта

Таблица 1 – Сильные и слабые стороны

|  |  |
| --- | --- |
| Сильные стороны | Слабые стороны |
| Возможность удаленной работы сотрудников | В периоды «затишья» доход студии существенно снижается |
| Большой опыт работы веб-студии (9 лет) | Некоторые сотрудники находятся на большом расстоянии друг от друга, а соответственно в разных часовых поясах, что иногда мешает их взаимодействию |
| Опыт работы с крупными клиентами:World Class, Транснефть, Toyota, Дальэнергомаш, Камаз, Yamaha | Небольшое количество сотрудников |
| Хорошее продвижение сайта в поисковых системах |  |
| Индивидуальный подход к клиентам |  |

Далее определяем возможности и угрозы развития.

Таблица 2 – Возможности и угрозы развития

|  |  |
| --- | --- |
| Возможности | Угрозы |
| Работа с заказчиком не привязана к определенному региону | Наличие большого количество мелких студий, которые предоставляют услуги за низкие цены |
| Растущий рынок интернет-рекламы | Конструкторы сайтов |
| У заказчика всегда будет потребность в новых кдиентах | Современные системы автоматизации позволяют заказчикам настраивать и вести рекламу в интернете самостоятельно |

Таблица 3 – SWOT-анализ сильных сторон

|  |  |  |
| --- | --- | --- |
| Описание сильных сторон | Каким образом сильные стороны влияют на удовлетворенность клиента? | Каким образом сильные стороны влияют на отличие продукта от конкурента? |
| Большой опыт работы (более 9 лет) | Клиенту важен результат, а опытная веб-студия может создать очень качественный сайт, это мотивирует клиента вернуться | Сайт создается быстрее за счет архива готовых решений и опытности сотрудников студии |
| Опыт работы с крупными клиентами | Заказчик может сопоставить свой сайт, с сайтами этих крупных клиентов и гордиться, что создал сайт именно в этой веб-студии | Ценность сайта выше |
| Хорошее продвижение сайта в поисковых системах | Увеличение количества заявок на услуги с сайта | Более высокие позиции в выдаче |

Таблица 4 – SWOT-анализ слабых сторон

|  |  |  |
| --- | --- | --- |
| Описание слабых сторон | Каким образом слабые стороны влияют на удовлетворенность клиента? | Каким образом слабые стороны влияют на отличие продукта от конкурента? |
| В периоды «затишья» доход студии существенно снижается | На удовлетворенность клиента влияния нет, так как клиентов мало | Обычно периоды затишься начинаются у веб-студий почти в одно и то же время |
| Некоторые сотрудники находятся на большом расстоянии друг от друга, а соответственно в разных часовых поясах, что иногда мешает их взаимодействию | Клиенту может срочно потребоваться, что-то доработать или исправить, а так как работа может вестись в разное время, клиент не будет удовлетворен | Если над проектом работают несколько программистов, они могут не согласованно делать работу, что повлияет на структура проекта |

Подведем итоги:

1. Крупные фирмы будут чаще обращаться в веб-студию «Перфоманс», так как у нее имеется неплохое портфолио и большой опыт работы.

2. Компания постоянно проводит стажировки программистов, отбирая из них лучших в свой штат, тем самым пополняя его

3. Сайты разрабатываются для системы управления содержимым Drupal, которая имеет некоторые преимущества перед более популярными CMS, благодаря своей гибкости. Это позволяет создавать сайты любого типа.

2 Программное обеспечение ООО «Перфоманс»

2.1 Инструменты для разработки

1. CMS Drupal 7.54; [34]

CMS(Система управления содержимым или контенотом), предназначенная для создания, управления и редактирования материалами сайта. [50]

Основные функции CMS: [50]

* + - * Предоставление инструментов для создания содержимого, организация совместной работы над содержимым,
      * Управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов и т. п.,
      * Публикация содержимого,
    - Представление информации в виде, удобном для навигации, поиска.

1. СУБД MySql 5.1; [9]
2. PHP 5.6;

PHP – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML. [32]

1. Apache HTTP-сервер – свободный веб-сервер;

Веб-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Веб-сервер – сервер, который принимает от клиентов (веб-браузеров) HTTP-запросы, и выдающий им HTTP-ответы. [50]

1. Html5;

Пятая версия языка разметки гипертекста html.

1. Less;

Less – css-препроцессор или динамический язык стилей, который был создан на основе языка Sass.

CSS – язык описания внешнего вида сайта, который написан на языке разметки (html)

1. JavaScript;

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

1. jQuery 1.10;

jQuery – самая популярная библиотека языка JavaScript, основанная на взаимодействии JS и html. Библиотека jQuery позволяет легко работать с DOM документа, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. [50]

1. Yarn/npm/bower.

Yarn/npm/bower – пакетные менеджеры.

Система управления пакетами – набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов программного обеспечения. [50]

1. Adobe Photoshop Creative Cloud 2015

Adobe Photoshop – редактор растровой графики, разработанный фирмой Adobe Systems, также имеет несколько векторных инструментов и работу с 3d. [50]

1. Sublime Text 3

Sublime Text ­ кроссплатформенный проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python. [50]

1. Gulp

Gulp ­ таск-раннер для автоматического выполнения часто используемых задач (например, минификация, тестирования, объединения файлов), написанный на языке программирования JavaScript. Программное обеспечение использует командную строку для запуска задач, определённых в файле Gulpfile. [50]

1. Node JS

Node или Node.js ­ программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера. [50]

1. Git

Git ­ распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. [50]

1. GitHub

GitHub ­ крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub. [50]

1. ConEmu

ConEmu – это «графический хост» для консольных приложений Windows. При запуске нового консольного приложения в ConEmu реальное окно консоли не отображается, а весь вывод и ввод перехватывается и перенаправляется в ConEmu. [50]

1. PhpStorm

JetBrains PhpStorm – коммерческая кросс-платформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

Интегрированная среда разработки (англ. Integrated Development Environment) — система программных средств, используемая программистами для разработки программного обеспечения. [50]

1. OpenServer

Open Server Panel – это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. [50]

1. Google Chrome

Google Chrome (с англ. - «хром») – браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink. [50]

1. Bootstrap

Bootstrap (также известен как Twitter Bootstrap) ­ свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. [50]

2.2 Прочее программное обеспечение

2.3 Причины выбора CMS Drupal

Таблица – Сравнительная характеристика самых популярных CMS [32]

|  |  |  |  |
| --- | --- | --- | --- |
|  | Wordpress | Joomla | Drupal |
| Сайт | ru.wordpress.org | www.joomla.org | www.drupal.ru |
| Удобство для контент-менеджера | + |  |  |
| Качество визуального редактирования |  |  | + |
| Функциональность |  | + | + |
| Встроенные модули | + | + | + |
| Поддержка мультимедийного контента | + |  | + |
| Надежность |  |  | + |
| Скорость работы | + |  |  |
| Какие сайты разрабатываются? | сайт-визитка, блог, корпоративный сайт, промо-сайт, портал с несложным функционалом | блог, сайт-визитка, корпоративный сайт с несложным функционалом, промо-сайт, интернет-магазин с несложным функционалом | сайт-визитка, блог, интернет-магазин со сложным функционалом, корпоративный сайт, порталы и сервисы, социальная сеть |

3 Разработка сайта для тур-фирмы «Азия Бас»

3.1 Верстка основных страниц сайта

Что требуется установить для работы:

* Node;
* Git;
* yarn;
* gulp;
* bower;
* sublime text 3

Для начала верстки, необходимо настроить автоматическую сборку, Livereload (автоматическая перезагрузка страницы), чтобы ускорить процесс разработки. Переходим в консоль ConEmu, в ней нужно эмулировать консоль GitBush и прописать следующие команды, после выполнения каждой нужно нажимать клавишу Enter, после этого можно будет настраивать проект и создавать базовую структуру проекта. [35]

Операции в консоли:

* mkdir asiabus – создание корневой директории;
* cd !$ - переход в созданную директорию;
* mkdir app – папка для содержимого проекта;
* yarn init –y – создание файла package.json и yarn.lock
* touch gulpfile.js .gitignore .bowerrc bower.json

Далее, при помощи yarn, нам необходимо установить модули для gulp и прописать их таски (работу) в файле gulpfile.js [43]

Вводим следующую команду:

yarn install –D gulp gulp-less path gulp-connect gulp-livereload gulp-wiredep

Этой командой у нас устанавливаются зависимости в папку node\_modules, которая создается автоматически, флаг –D указывает, что эти зависимости должны попасть в файл package.json, в массив devDependencies, то есть в зависимости для разработки.

Далее настраиваются таски этих модулей в файле gulpfile.js.

В файле .bowerrc указываем директорию, в которую будут устанавливаться, библиотеки, скачанные при помощи пакетного менеджера bower. Настройка таска в gulpfile.js (bower) и специальные комментарии в index.html будет подключать bower\_components автоматически, используя содержимое файла package.json.

Все зависимости устанавливаются в папку app/bower\_components, и записываются в массив dependencies файла bower.json. [45]

Bower install -S [имя зависимости] (например jquery или bootstrap)

После настройки, в консоли нужно выполнить команду gulp

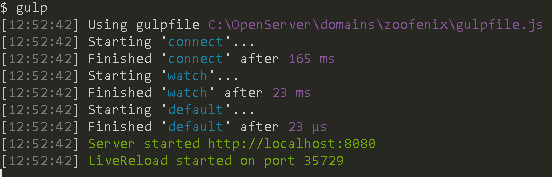


Рисунок 1 – выполнение команды gulp

Запущен локальный сервер с адресом localhost:8080, который можно ввести в браузере, теперь можно создавать структуру приложения.

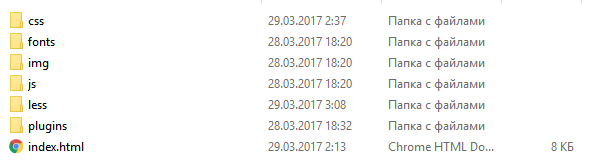


Рисунок 2 – содержимое папки app

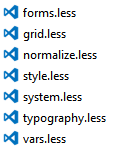


Рисунок 3 – Содержимое папки less

Все стили будут создаваться при помощи css-препроцессора less, настройка модулей gulp-livereload и gulp-less позволяет, при каждом сохранении файла стилей, преобразовывать файлы с расширением .less в файлы с расширением .css и помещать их в папку css. В index.html достаточно подключить файл style.css. При каждом сохранении любого .js, .less, .css и .html файла, браузер автоматически обновляется (при работе с двумя мониторами это очень удобно, так как не нужно переходить в браузер и обновлять страницу после каждого сохранения файлов). [40]

Теперь можно создать пустой git-репозиторий. В корневой папке проекта вводим следующие команды:

* git init – создаем репозиторий;
* git add all – добавляем все файлы в отслеживание;
* git commit –m “first commit” – коммитим (фиксируем) изменения в файлах. [41, 42]

Автоматизация настроена, проект находится под управлением системы контроля версий, можно приступать к верстке, используя psd-макет, который можно открыть в программе Adobe Photoshop. [49]

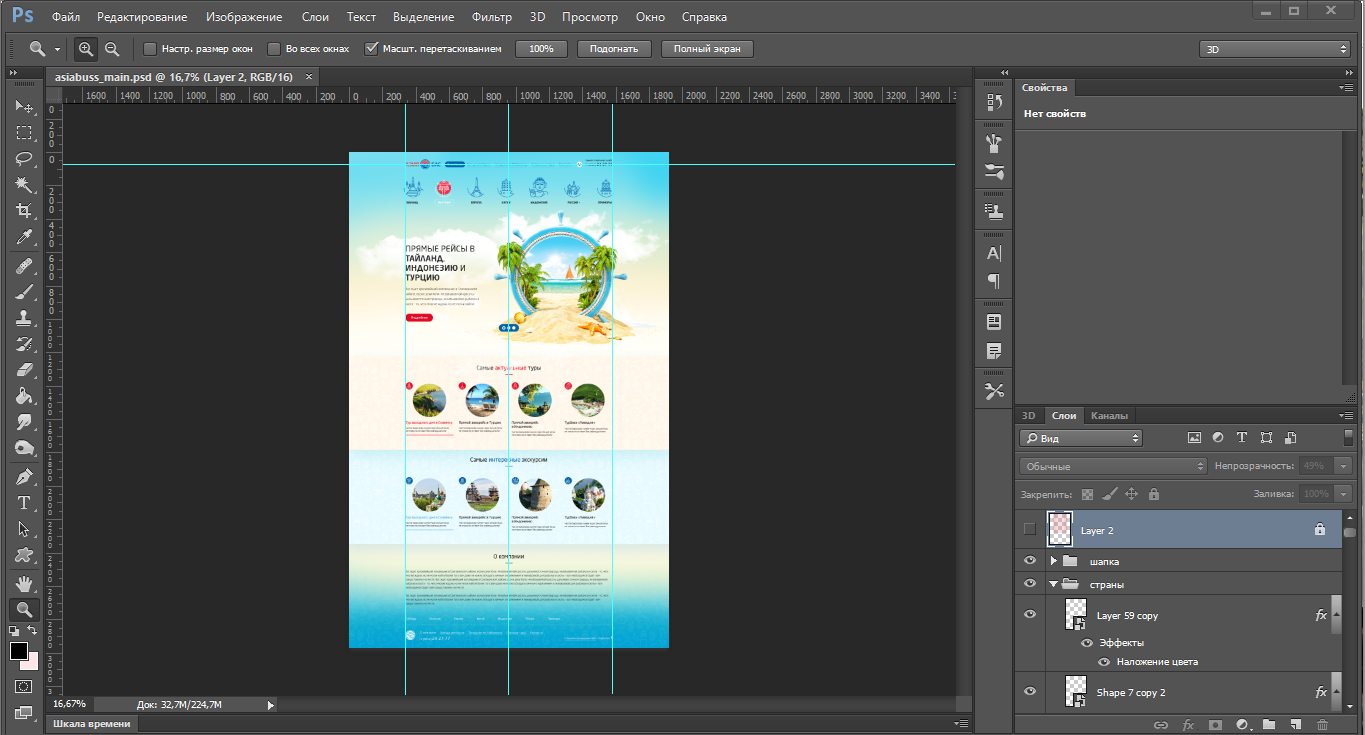


Рисунок 4 – psd-макет в Adobe Photoshop

Далее, в текстовом редакторе sublime text 3 пишутся все less, js и html файлы. [47]

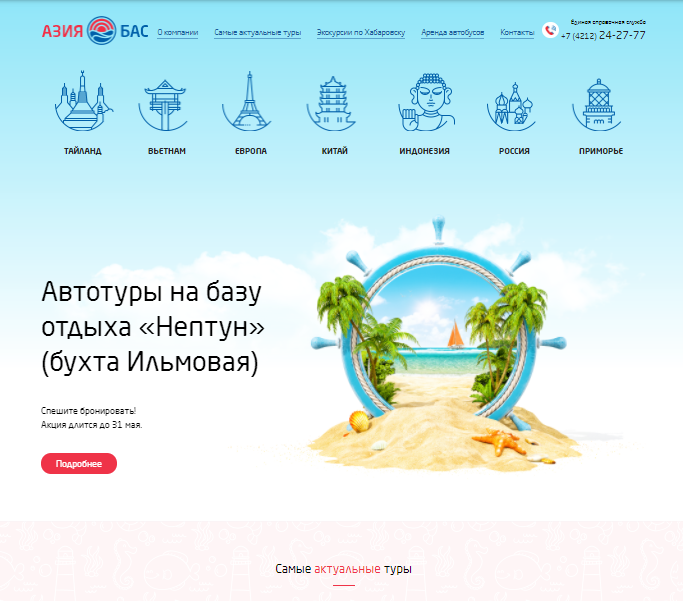


Рисунок 5 – Результат верстки

3.2 Создание темы для CMS Drupal 7

После того, как готова статичная верстка сайта, необходимо сделать сайт динамичным и добавить к нему административную часть, для удобного управления содержимым сайта. Для начала работы нужно запустить локальный сервер Open Server, запустить его и далее перейти в PhpMyAdmin. (Рисунок 6, 7) [2, 3, 4, 5, 7, 9]

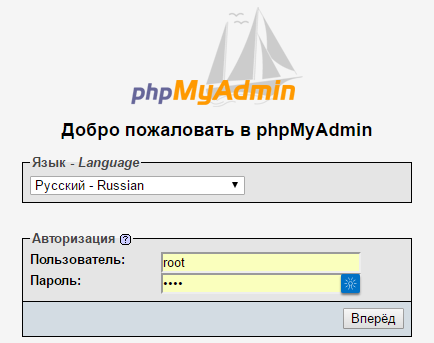


Рисунок 6 – Окно входа в phpMyAdmin

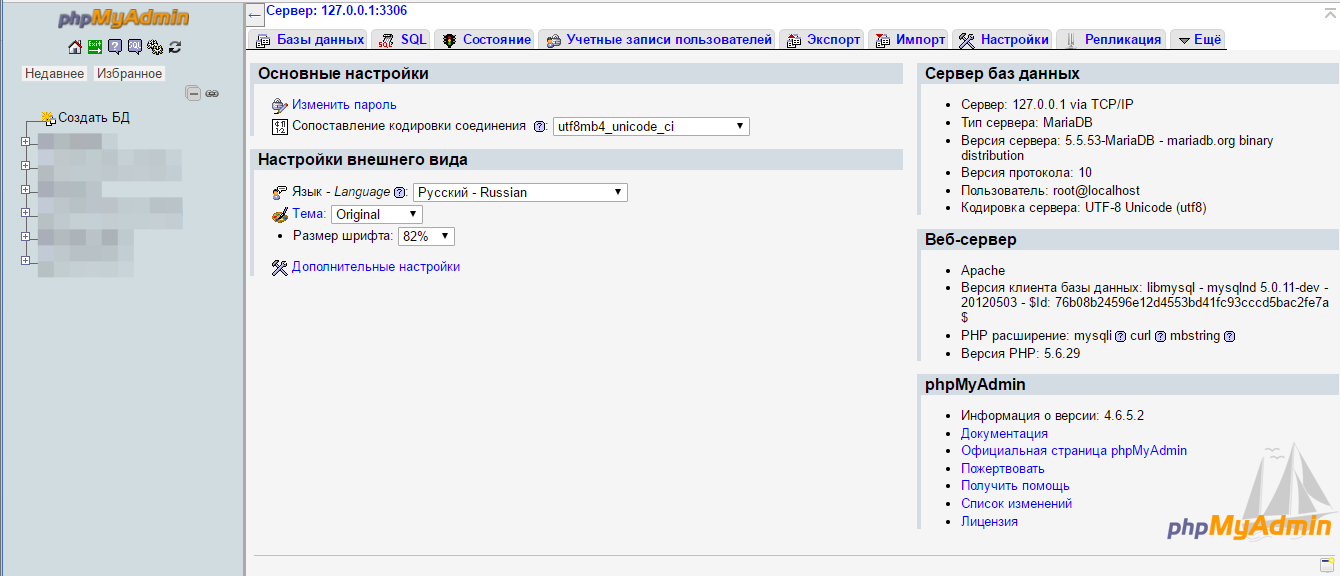


Рисунок 7 – Главное окно phpMyAdmin

Переходим в пункт «Создать базу данных» и создаем новую БД, указав ее название и сравнение. (Рисунок 8)

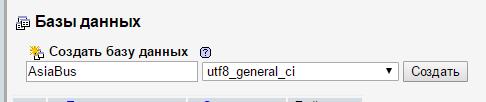


Рисунок 8 – Создание новой БД

База данных готова, создание таблиц, связей и наполнение данными будет происходить автоматически, когда база будет подключена к сайту под управлением Drupal.

Теперь в папке domains корневого каталога Open Server нужно создать папку, в которую далее установится CMS Drupal. (Рисунок 9) []

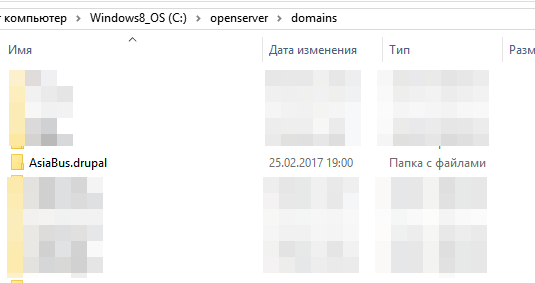


Рисунок 9 – Папка для содержимого сайта

В эту директорию нужно перенести все файлы из архива, в котором находится сам Drupal, после этого в меню Open Server, в пункте «Мои сайты» выбрать соответствующий, после этого откроется браузер и мы увидим окно приветствия Drupal, которое предложит нам установку.

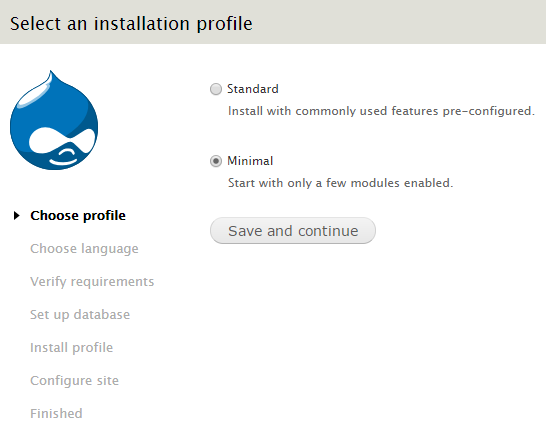


Рисунок 10 – Выбор профиля установки

Выбираем минимальный профиль установки, так как потом будет проще включить нужные модули и функции, чем убирать лишние. (Рисунок 10)

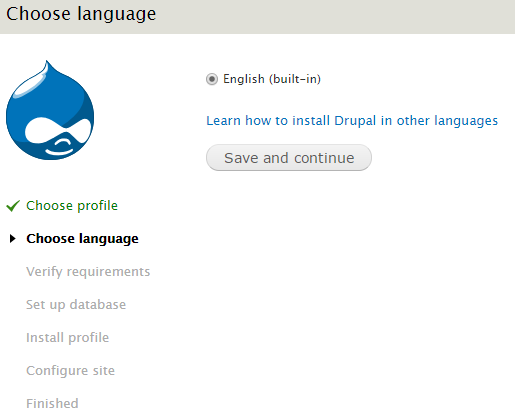


Рисунок 11 – Выбор языка

По умолчанию в Drupal 7 только один язык для выбора, но после установки можно будет установить любой язык. (Рисунок 11)

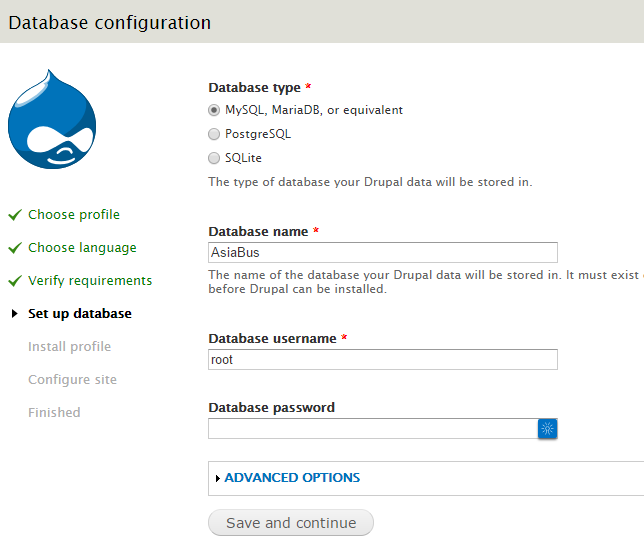


Рисунок 12 – Подключение к базе данных

После подключения к базе данных идет установка минимального профиля. (Рисунок 12)

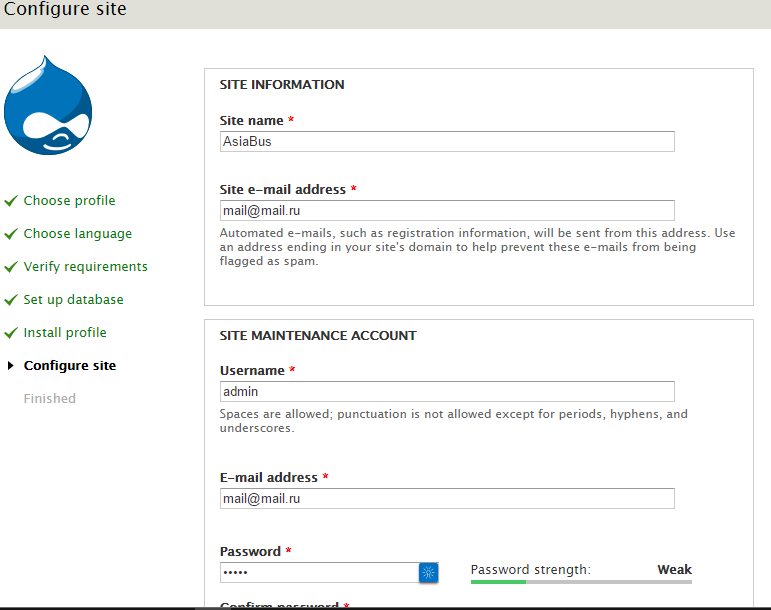


Рисунок 13 – Конфигурация сайта

В данном окне настраивается название сайта, главный администратор сайта и т.д. (Рисунок 13)

CMS Drupal 7.54 Установлена (Рисунок 14)

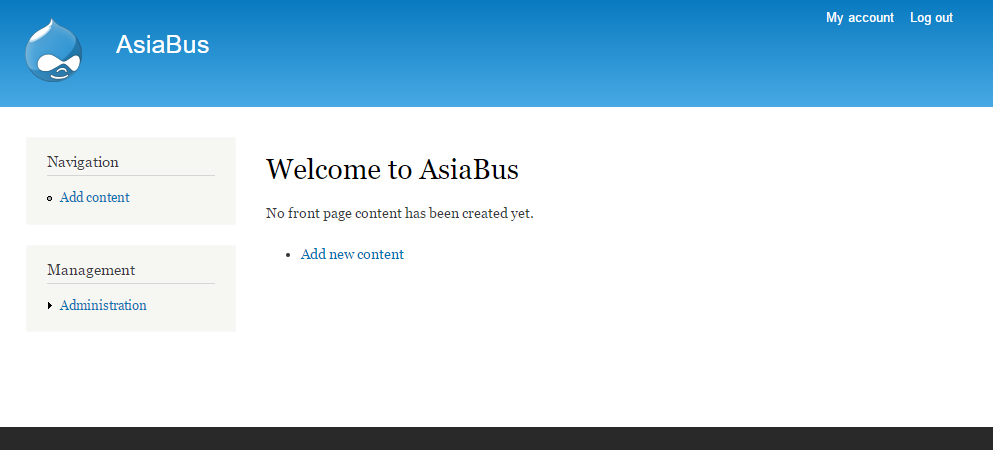


Рисунок 14 – Главная страница сайта

Все модули для разработки, интерфейса пользователя и т. д. будут скачиваться с официального сайта www.drupal.org/project/project\_module [7, 8, 34]

Для настройки интерфейса нам понадобятся следующие модули:

1. Admin menu. Добавляет выпадающее меню для большинства административных и других общих задач (для пользователей с соответствующими правами).
2. Module filter. Фильтрует список модулей.
3. Empty Front Page. Удаление контента по умолчанию с главной страницы.
4. Localization Update. Предоставляет возможность автоматической загрузки и обновления переводов.
5. Administration menu Toolbar style. Улучшенная Панель инструментов.

Переходим пункт меню Administration, далее modules и кликаем в пункт Install new module. (Рисунок 15, 16)

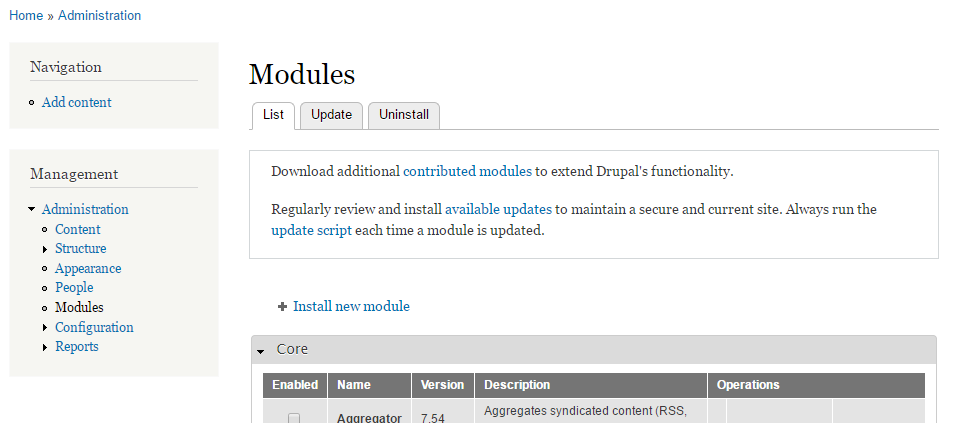


Рисунок 15 – Список модулей

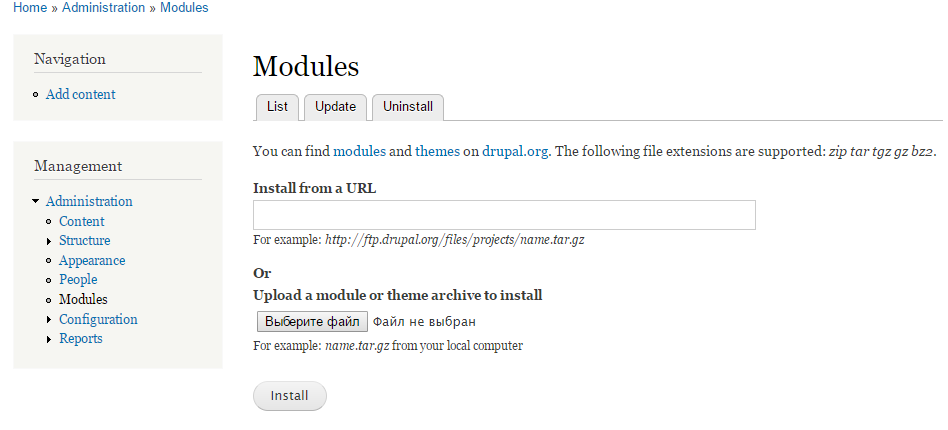


Рисунок 16 – Окно установки модуля

На сайте Drupal.org, в поле имени модуля вводим название модуля из списка выше, потом выбираем нужный в результатах поиска.

На странице модуля можно прочитать его описание, посмотреть документацию, скриншоты и примеры использования.

В самом низу страницы находится список для скачивания модуля для разных версий Drupal. Ссылки также помечены разными цветами. Зеленым цветом выделены ссылки на модули, проверенные официальным разработчиком. Кликаем правой кнопкой мыши (далее ПКМ) по нужной ссылке и выбираем копировать адрес ссылки. (Рисунок 17)

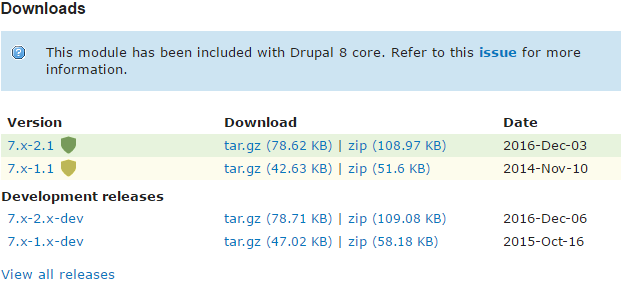


Рисунок 17 – Список ссылок для скачивания

Далее вставляем ссылку, которую мы скопировали ранее в поле Install from url и кликаем по кнопке Install. Далее установка всех модулей, за исключением самописных, будет происходить таким же образом. (Рисунок 18)

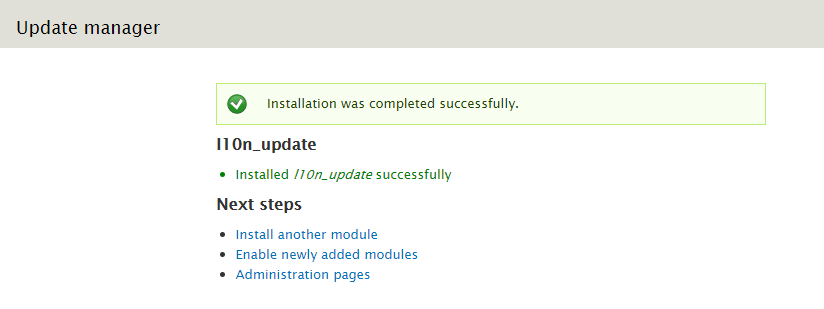


Рисунок 18 – Успешная установка модуля

Теперь, установленный, модуль нужно включить, кликнув Enable newly added modules, выбрать его в списке и сохранить конфигурацию. (Рисунок 19)

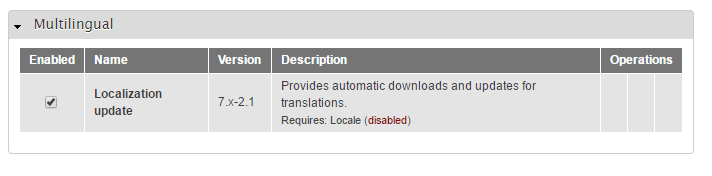


Рисунок 19 – Включение модуля

Теперь можно добавить русский язык, переходим во вкладку configuration, далее languages и выбираем пункт add language (Рисунок 20, 21)

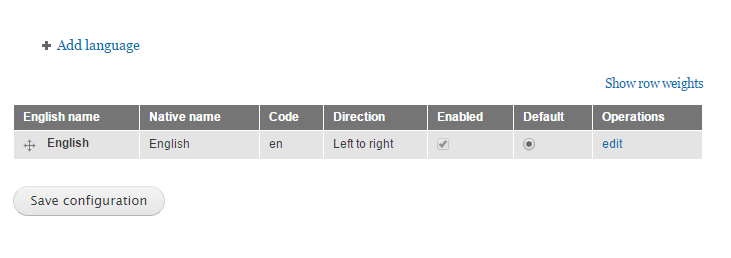


Рисунок 20 – Окно конфигурации языка

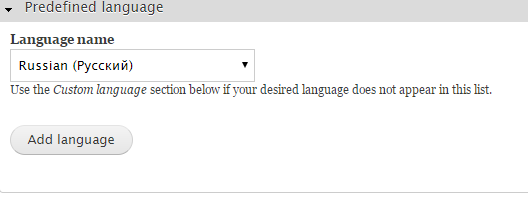


Рисунок 21 – Выбор русского языка из списка

После установки выбираем русский язык по умолчанию и сохраняем конфигурацию (Рисунок 22)

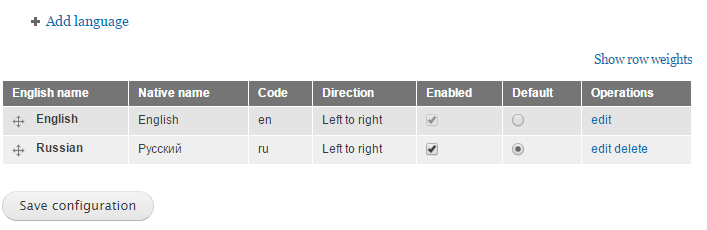


Рисунок 22 – Смена языка по умолчанию

После установки русского языка, все модули, которые уже установлены и будет установлены, будут локализованы, если имеется поддержка данного языка. Далее устанавливаются остальные модули из списка.

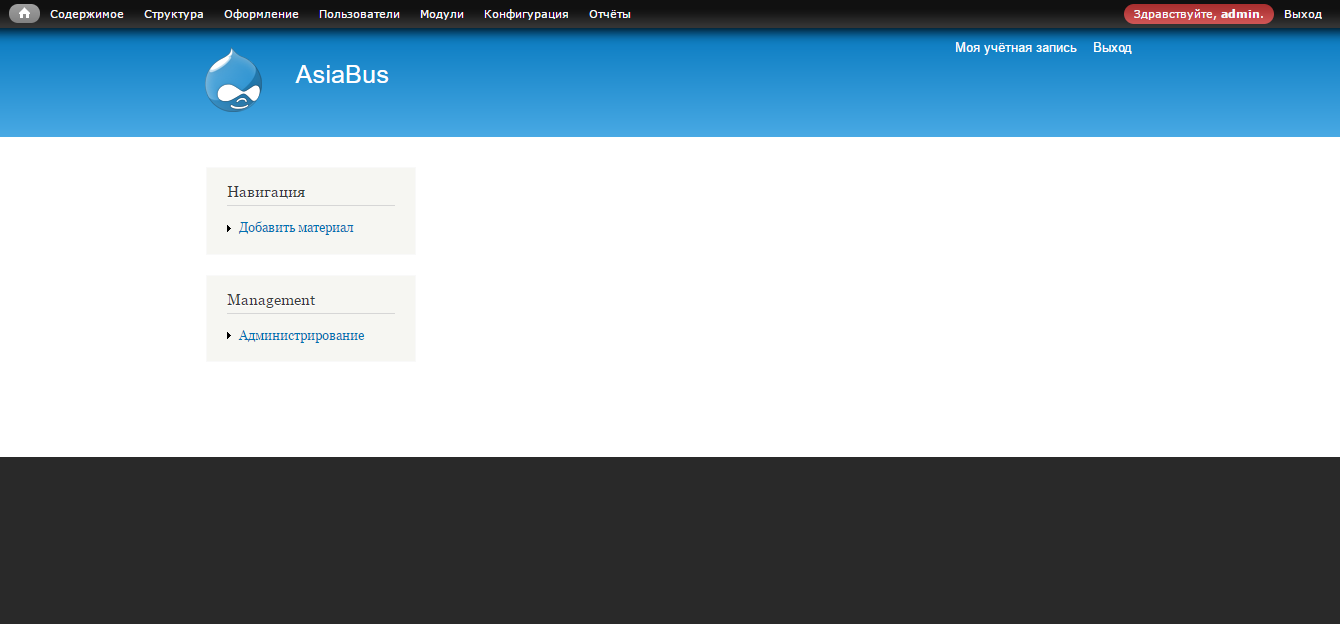


Рисунок 23 – Главная страница после установки модулей

Теперь есть удобная панель администратора сверху, благодаря которой можно быстро получить доступ к любой функции CMS Drupal. (Рисунок 23) Далее можно приступать к темизации.

Темы – это плагины Drupal, которые придают сайту Drupal новый облик.

Тема – важный компонент пользовательского интерфейса (UI) Web-сайта на основе Drupal. В версии Drupal 7 появилась новая методология реализации тем, хотя структура темы существенно не изменилась.

Тема Drupal отделяет логику обработки системы от ее элементов дизайна. Для этого Drupal использует сложную структуру тем, которая состоит из тем, механизмов тем и связей. Компоненты темы взаимодействуют с системами ядра Drupal и элементами дизайна модулей, образуя оригинальный пользовательский интерфейс ― отдельные страницы и формы Drupal. Когда бизнес-логика Drupal отделена от логики представления, код легче поддерживать, а реализацию темы можно менять без необходимости переписывать код, и наоборот. Архитектура среды Drupal показана на рисунке 24.

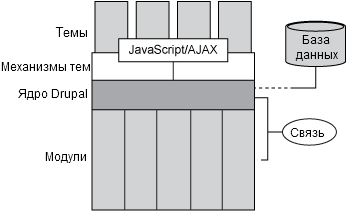


Рисунок 24 – Архитектура среды Drupal

Темы отображают данные, которые ядро Drupal извлекает из базы данных через механизм тем, служащий интерфейсом между ядром Drupal и шаблонами тем.

Механизм тем

Механизмы тем ­ это средства взаимодействия тем с ядром Drupal. Существует несколько механизмов тем, которые можно использовать с Drupal. Вот наиболее популярные из них:

* PHPTemplate
* XTemplate
* Plain PHP
* Smarty Engine

При желании можно использовать свой собственный механизм.

Механизм тем Drupal по умолчанию PHPTemplate, который рассматривается в этой статье, использует для функций тем Drupal типа theme\_filename() отдельные файлы тем с такими именами, как filename.tpl.php. Каждый из этих файлов содержит HTML-основу и PHP-операторы для работы с динамическими данными. Таким образом, при наличии начальных знаний в области PHP можно легко создавать достаточно сложные темы с помощью PHPTemplate, так как они содержат всего лишь небольшие фрагменты кода.

Планирование темы

Тема состоит из нескольких файлов. Их количество зависит от сложности темы. В теме может быть всего три файла или несколько десятков, включая графические элементы и различные сценарии, размещенные в нескольких папках в каталоге theme. На рисунке 5 показаны типичные описания файлов (цвета, изображения, логотип, шаблоны, таблицы стилей и PHP-файлы), которые могут содержаться в главной папке темы. (Рисунок 5)

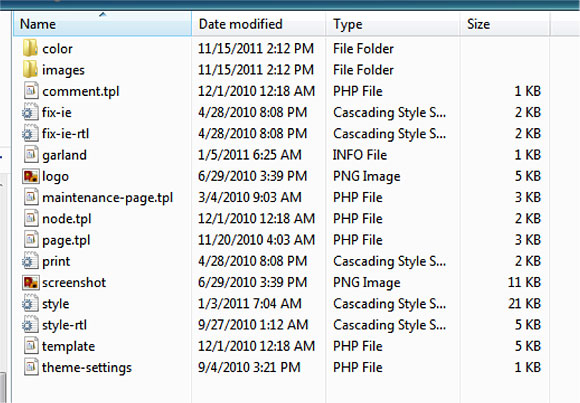


Рисунок 25 – Типичный состав папки темы

Типичный набор папок и файлов в папке темы

Не все эти файлы необходимы для темы, и в некоторых темах нет даже файла каскадных таблиц стилей (CSS), который многие считают абсолютно необходимым. Структуру темы и то, как она работает, легче понять, когда знаешь, как используются все эти файлы. Рассмотрим некоторые файлы из каталога theme и их функции в рамках общей темы.

Файл .info

Файл с расширением .info обязателен, в нем находится вся информация о теме, подключаются файлы скриптов и таблицы стилей, выбираются обласи для вывода блоков страниц. Перед расширением находится название темы, создав этот файл, в админ части Drupal, в разделе оформления в списке тем будет новая тема с таким же названием, как info-файл.

Файлы шаблонов .tpl.php

В каталоге темы есть несколько файлов шаблонов с именами типа [имя шаблона].tpl.php. Эти файлы содержат разметку Extensible HTML (XHTML) и PHP-переменные темы. В некоторых случаях они могут содержать и другие виды выходных данных, такие как RSS. В общем случае каждый файл темы Drupal .tpl.php управляет определенными выходными данными. Встраивать сложную логику в файлы шаблонов неудобно и нерационально, так как это затрудняет их обслуживание. Желательно, чтобы они содержали только прямые XHTML-теги и переменные PHP.

Файл template.php

В файле template.php содержится вся логика условных переходов и обработки выходных данных темы. Здесь можно добавить переменные для хранения телефонов, электронной почты и т.д. то есть значения, которые могут появляться на нескольких страницах сайта. Так же здесь можно прописывать хуки, особые функции, о которых будет подробнее рассказано далее, при создании шаблона главной страницы.

Создание темы

Тему можно создать несколькими способами:

1. с нуля;
2. на основе существующей.

В данной работе будем применен первый способ, но также можно использовать файлы из существующих тем и изменять их под новую тему (шаблоны).

Создание структуры каталогов

Сначала создадим каталог для хранения файлов темы. Самое подходящее место для этого каталога - папка sites/all/themes. Присвоим ему уникальное имя, характерное для нашей темы: оно не должно содержать пробелов.

Новую тему необходимо создавать по следующему пути, /sites/all/themes, следуя от корня сайта. В директории создается новая папка с названием темы. Внутри нее будут лежать все остальные файлы для темы (css, js, изображения, шрифты, шаблоны и т.д.) (Рисунок 26)

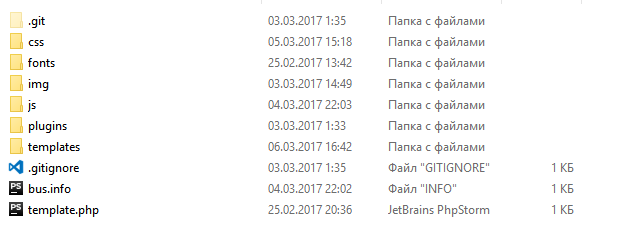


Рисунок 26 – Структура папки с новой темой

В папке css будут храниться файлы стилей, fonts папка для шрифтов, img папка для изображений, js папка для скриптов, plugins папка для сторонних расширений и фреймворков, templates папка для шаблонов страниц и представлений.

Создание файла .info

Файл .info ― это просто текстовый файл, содержащий данные ― как правило, параметры, необходимые для описания структуры, а также содержание и параметры настройки темы. Это текстовый файл, каждая строка которого содержит пару из ключа и значения, причем значение расположено справа, а ключ ― слева от знака равенства (=). (Рисунок 27)

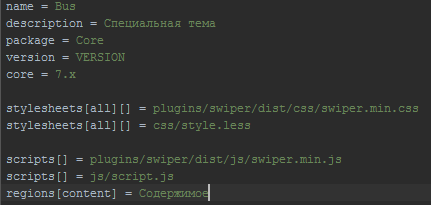


Рисунок 27 – Файл с информацией о теме

Name – имя темы

Description – описание темы

stylesheet – массив таблиц стилей

scripts – массив подключаемых скриптов

regions – область

После того, как создан файл с расширением .info, в админ-панели можно включить нашу тему и поставить ее по умолчанию.

В верхнем меню кликаем по пункту – оформление и в списке тем выбираем – bus, включаем ее и устанавливаем по умолчанию. (Рисунок 28)

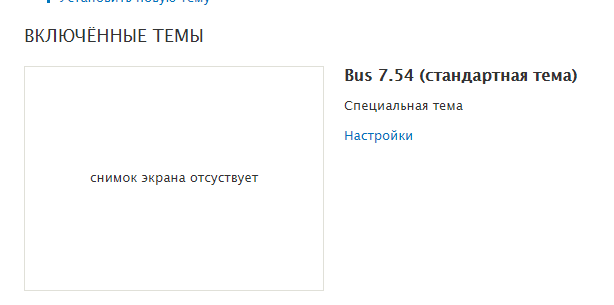


Рисунок 28 – Тема по умолчанию

Важно для оформления административных страниц выбрать одну из базовых тем, в данном случае это будет тема seven, для того чтобы избежать ошибок и конфликтов при создании материалов и т. д.

Области

Ключ regions определяет области блока, доступные для темы. Необходимо определить ключ regions, а затем указать внутреннее машиночитаемое имя в квадратных скобках. После этого нужно в качестве применимого значения указать имя, понятное для человека. Синтаксис:

regions[name] = Content

В нашем случаем всего одна область regions[content] =Содержимое.

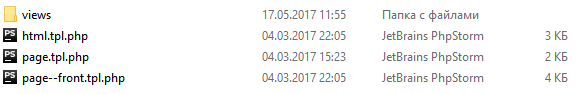


Рисунок 29 – Структура папки tamplates

В данной папке будут храниться все шаблоны документа, внутренних страниц, главной страницы, представлений, форм и т. д. (Рисунок 29)

Файл html.tpl.php в данной теме является корневым шаблоном, в нем указывается тип документа, подключаются стили, скрипты и указывается переменная, в которую будут выводится другие дочерние шаблоны.

Файл page.tpl.php это шаблон для любой внутренней страницы нашего сайта. Здесь располагается вся разметка сайта, которая находится внутри тега body, и указывается регион для содержимого страницы. Вместе с файлом html.tpl.php образуется, так называемый, layout, то есть общая разметка для всех страниц.

page--front.tpl.php шаблон для главной страницы сайта. Так как помимо основного содержимого, в отличии от внутренних страниц сайта, на главной есть другие элементы, (слайдер, блок экскурсий и т. д.), то и сама страница должна отличаться, а потому ей задается более уникальное имя. После сброса кеша на сайте, главная страница будет попадать под более уникальный шаблон (front). Для того чтобы видеть из какого шаблона выводится страница, необходимо изменить настройки сайта. В папке sites/all/themes/default есть файл settings.php, в котором указаны доступы к базе данных, строка подключения и закомментированы некоторые полезные строки.

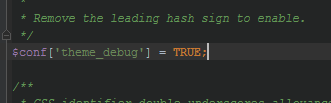


Рисунок 30 – файл settings.php

Убрав комментарий с данной строки, мы сможем увидеть дополнительную информацию о шаблонах, пути к ним и т.д. прямо в браузере, в инструментах разработчика. Крестиком будет указан текущий шаблон. (Рисунки 30, 31)

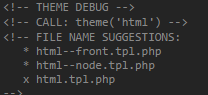


Рисунок 31 – информация о шаблоне

В шаблоне html.tpl.php нет необходимости что-то менять, за исключением типа документа и добавления некоторых meta-тегов.

Далее создается шаблон для главной страницы (page--front.tpl.php) и для внутренних страниц. В каждый из этих шаблонов вставляется верстка, которая повторяется на всех страницах сайта, то есть: хедер (шапка) сайта и футер (подвал). Весь контент, который находится между этими блоками, будет выводится в других шаблонах (нодах, представлениях и т.д.).

После сброса кеша на сайте, мы увидим разметку главной страницы, но без применения стилей, так как drupal, пока, не понимает less. Указав специальную строку в шаблоне (<?php print $messages; ?>), в содержимом сайта будет выводится информация об ошибках, очистке кеша на сайте и т.д. Загрузив страницу без поддержки препроцессора less, мы увидим сообщение о том, что drupal не может загрузить стили и указывает, что нужно для исправления ошибки. Далее устанавливаем модуль LESS CSS Preprocessor, а также, в папку /sites/all/libraries добавляем библиотеку less.php, ссылка на скачивание которой, находится на странице с модулем.

Подключив поддержку стилей, написанных на css-препроцессоре less, в консоли инструментов разработчика браузера Google Chrome, выводятся ошибки, говорящие, что браузер не может найти изображения (логотип в шапке и подвале), так как неверно указан путь. При верстке был указан статичный путь к изображениям, теперь нужно указывать путь относительно корневого каталога сайта (/sites/all/themes/[папка с изображениями]). Для того, чтобы не вводить корневой путь каждый раз, его можно присвоить, как значение переменной, которую так нет необходимости прописывать в ручной. Для этого нам поможет хук preprocess\_page, код которого добавляется в файл template.php.

Как и любой веб-фреймворк, Drupal берет на себя часть обработки HTTP-запроса, вызывая написанные в модулях функции на определенных этапах. Drupal знает, когда нужно вызвать ту или иную функцию некоторого модуля благодаря механизму хуков(hooks). Хук – это спецификация API-функции Drupal с фиксированным именем, сигнатурой и семантикой. Реализация хука – это PHP-функция в некотором модуле Drupal, соответствующая спецификации.По смыслу хук аналогичен интерфейсу в ООП, а реализация хука – реализации интерфейса в некотором классе. Однако возможности ООП PHP не используются для осуществления работы хуков.

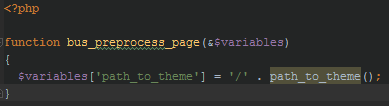


Рисунок 32 – содержимое файла template.php

На рисунке 32 показан хук bus\_preprocess\_page, который срабатывает при загрузке страницы. В массив $variables добавляется элемент path\_to\_theme, и присваивается значение пути до темы, при помощи функции path\_to\_theme().

Теперь в шаблонах внутренней страницы и главной страницы пути до изображений можно указать следующим образом:

<img src=”<?php print $path\_to\_theme; ?>/img/[имя].расширение” alt=””/>

После проделанных выше операций, на странице сайта будут применяться стили и корректно выводится изображения, но на странице все еще нет контента.

В drupal все (каждая страница, блок, элемент блока, форма и т.д.) является сущностью/объектом. Соответственно, у каждой сущности есть свои атрибуты, которые отличают ее от других. В нашем случае примером сущности является тур, атрибутами которого будут: тип сущности (тур), изображения, название, описание, страна. Для того, чтобы добавить содержимое на сайт (туры, экскурсии, страны и т.д.) в drupal есть возможность создания своих сущностей или типов материалов.

Для примера создадим тип материала автобус:

1. Переходим в пункт меню структура -> типы материалов -> добавить тип материала (Рисунок 33)

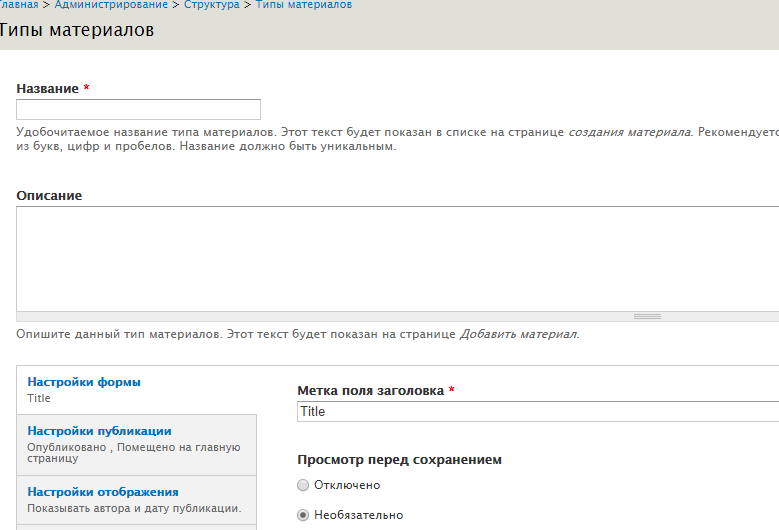


Рисунок 33 – Добавление типа материала

В поле название вводим логичное имя для нашего типа материала (Автобус) и машинное имя bus. Поле описание можно проигнорировать и оставить пустым, метку заголовка лучше исправить на «Заголовок», так как администратору сайта в будущем, проще понимать русский язык. Во вкладке настройка публикации отключаем публикацию на главной странице, в настройке отображения убираем вывод даты и автора публикации, в настройке меню можно убрать добавление пункта в меню, так как для данного типа материала в этом нет необходимости. Сохраняем тип материала и добавляем поля.

1. Добавление полей (Рисунок 34)

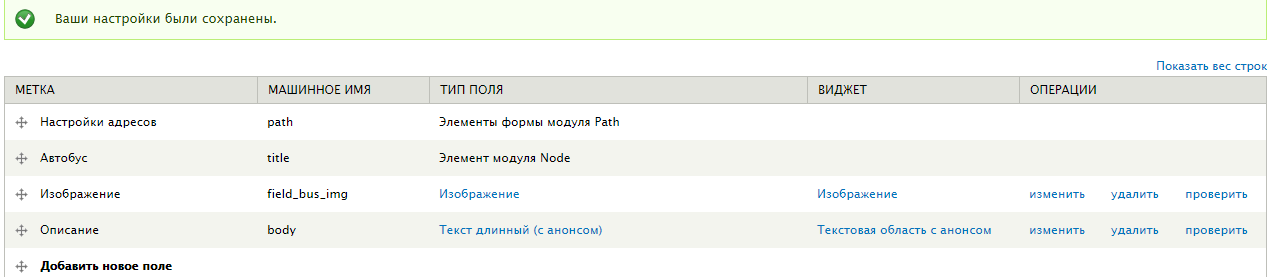


Рисунок 34 – Страница добавления полей

По умолчанию у типа материала уже существуют поля title, body и поле path, которое появляется при включении модуля path. Модуль path является модулем ядра drupal, устанавливать его не нужно. По psd-макету внутренней страницы автобуса можно выделить следующие поля:

* Изображение;
* заголовок;
* описание.

Поля стоимость, минимальное время аренды и количество мест, по желанию заказчика, можно опустить.

В строке добавить новое поле вводим имя для нашего поля (Изображения), машинное имя (field\_bus\_img) (желательно указаывать машинное имя, именно в таком формате field\_[машинное имя материала]\_[имя поля], это поможет избежать дальнейшего конфликта между материалами, так как поле изображений может быть и у других сущностей). В настройках поля выбираем его обязательным, директорию для загрузки изображений также необходимо указать, чтобы структурировать изображения по типам материалов на сервере, и выбираем неограниченное количество для поля, чтобы можно было загрузить несколько фотографий.

Остальные поля уже существуют, можно сохранить тип материала и добавить несколько автобусов.

Переходим в пункт меню Содержимое -> добавить материал -> Автобус

Вводим названия для нового автобуса, изображение для него. В поле описание вводим информацию об автобусе. Но что, если нам необходимо создать список, выделить текст жирным или курсивом, ведь в drupal изначально нет текстового редактора, а обычный пользователь не всегда может знать html-теги. Для этого понадобится модуль CKeditor, который скачивается отдельно. После его установки и включению в пункте меню Конфигурация -> Работа с содержимым -> Текстовые форматы добавляем новый формат, называть его принято FULL text. (Рисунок 35)

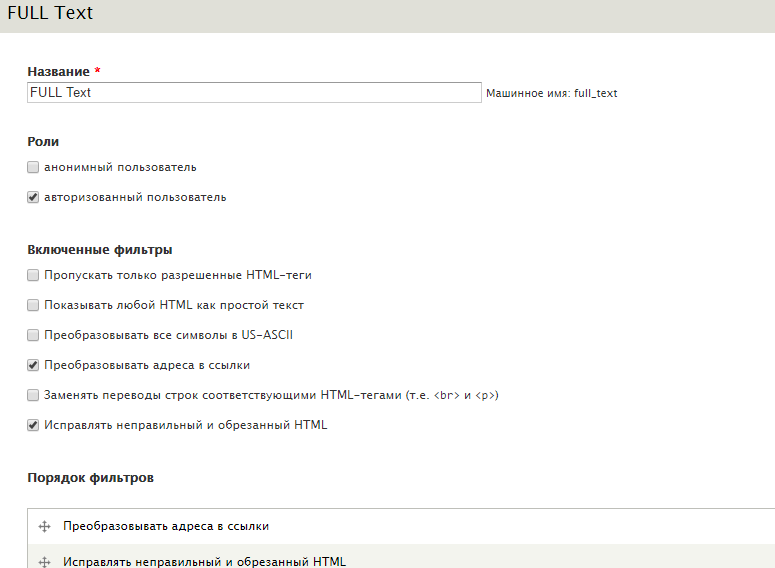


Рисунок 35 – Настройка нового текстового формата

Настраиваем CKEditor (Конфигурация -> Работа с содержимым -> CKEditor), добавляем новый профиль и выбираем для него текстовый формат. (Рисунок 36)

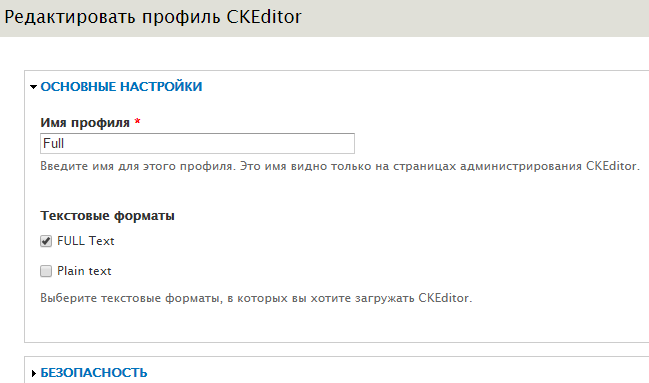


Рисунок 36 – Создание профиля в CKEditor

Теперь при создании материала, в полях с типом – Длинный текст, появился WYSIWYG-редактор.

WYSIWYG (произносится [ˈwɪziwɪɡ], является аббревиатурой от англ. What You See Is What You Get, «что видишь, то и получишь») — свойство прикладных программ или веб-интерфейсов, в которых содержание отображается в процессе редактирования и выглядит максимально близко похожим на конечную продукцию, которая может быть печатным документом, веб-страницей или презентацией. В настоящее время для подобных программ также широко используется понятие «визуальный редактор».

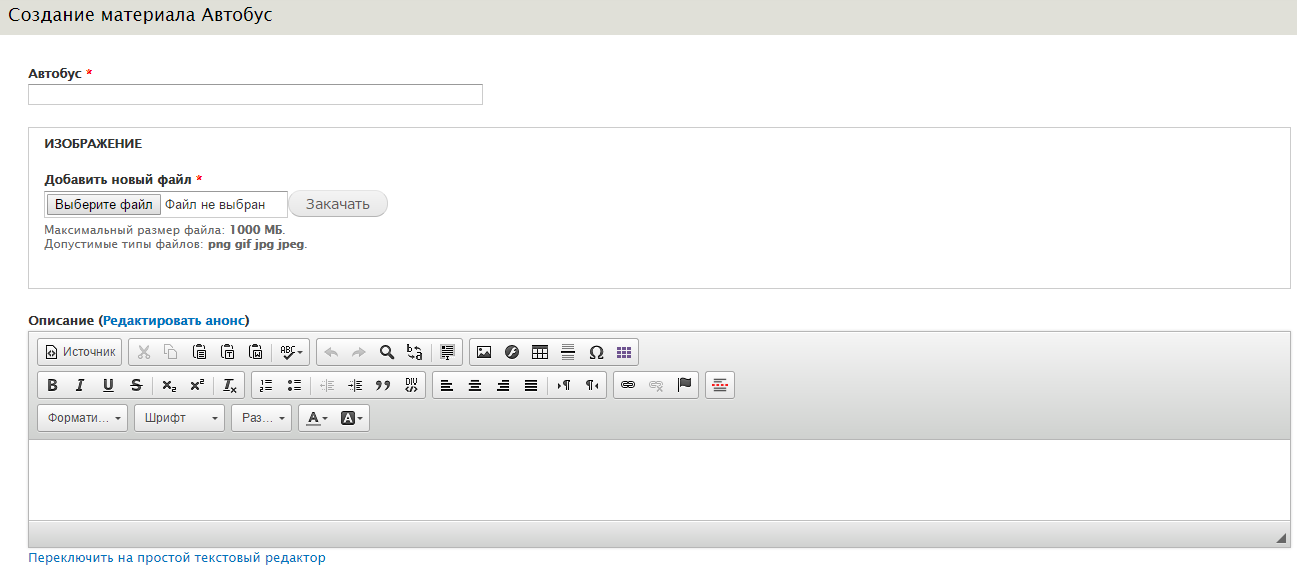


Рисунок 37 – Окно создания материала

После создания материала (Рисунок 37), мы автоматически попадаем на страницу с его полным содержимым. Шаблоном является node.tpl.php из ядра drupal.

Далее все типы материалов будут создаваться по такому же сценарию, отличаться будут только их названия и поля.

В этом разделе будет показан один пример для создания представления, при помощи модуля views, самого популярного модуля для cms Drupal, который, как было сказано выше, будет установлен с официального сайта Drupal. Данный модуль позволяет нам выводить на странице несколько материалов одного типа. Как пример можно привести каталог любого интернет-магазина, где однотипными материалами будут являться товары, которых несколько на странице.

Создадим представление для вывода туров. Для этого нужно пройти в пункт меню Структура –> Представления -> Добавить новое представление. В открывшемся окне указываем название нашего представления, его машинное имя, тип материала (ы данном случае Тур), сортировку необязательно, путь к странице (tours). Выбираем создать страницу или блок, либо и блок и страницу. В нашем случае понадобиться только страница. Создание блока будет разобрано далее, на примере создания слайдера. Хотя создание блока ничем не отличается от создания страницы, на примере слайдера будет рассмотрен еще один модуль, добавляющий новый тип полей, при создании материала, и еще одна особенность блока в том, что его можно вывести в любом месте сайта.

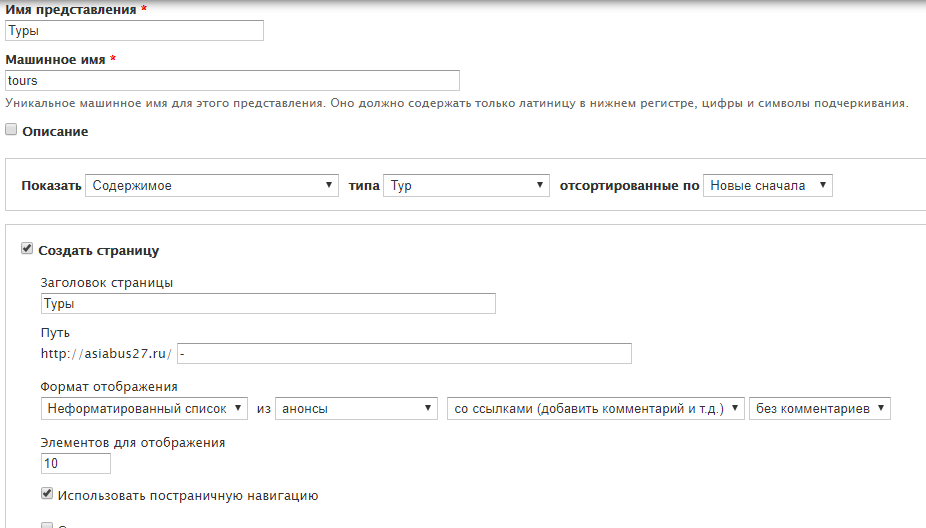


Рисунок 38 – Окно создания нового представления

После кликаем продолжить и переходим на страницу редактирования представления (Рисунок 38). Здесь мы можем указать какие поля выводить, гибко настроить вывод полей, добавив им обертку, классы и т.д., настроить критерии фильтрации, сортировку, количество выводимых материалов и много другое. Для начала настроим формат, у нас уже указан тип вывода – неформатированный список, отображение полей материала. Для нашей темы лучше убрать обертку полей по умолчанию в настройках формата. В критериях фильтрации добавлен фильтр по типу материала, так что на странице будут выводится материалы необходимого типа.

Для того чтобы отображать нужные поля, нужно кликнуть по пункту добавить, справа от раздела – поля. Воспользовавшись поиском или просто выбрав нужный пункт (например, Изображение) добавляем его в вывод. Далее открывается окно настройки отображения, добавленного поля. В этом окне можно настроить метку, отключить ее и этого будет достаточно для этого поля. (Рисунок 39)

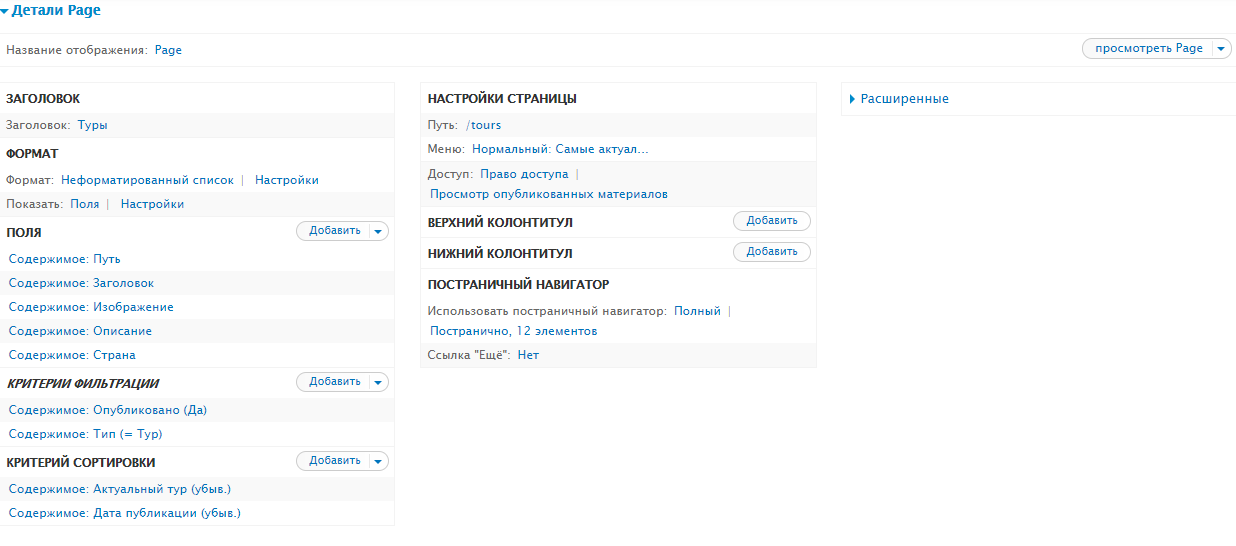


Рисунок 39 – Страница настройки представления

Дальше добавляем следующие поля: заголовок; описание; страна.

Поле путь не добавлялось при создании материала, оно автоматически добавляется ко всем материалам модулем path. Ниже настроек можно кликнуть по кнопке обновить предпросмотр и увидеть уже готовое представление, но мы увидим не совсем то что нужно: картинки разных размеров, путь к материалу, который должен быть ссылкой на полную версию. По макету должно быть по 4 тура в ряду, круглое изображение и должны появляться эффекты при наведении (полоса снизу и замена цвета заголовка). Для того чтобы это настроить, нужно создать новый шаблон для вывода представления. Увидеть текущий шаблон можно в расширенных настройках в разделе другое, информация о теме. Здесь можно увидеть названия шаблона, который используется для вывода представления, полей, материалов и т.д. (Рисунок 40)

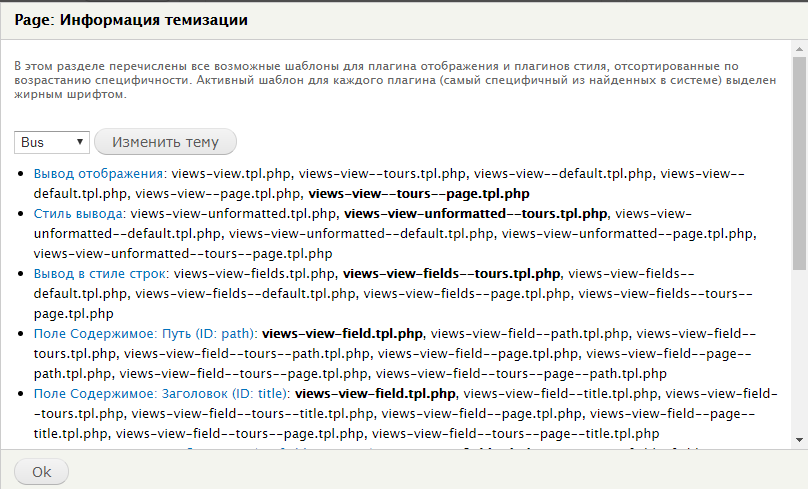


Рисунок 40 – Информация темизации

Для нашего случая достаточно будет создать шаблоны для стиля вывода и вывода в стиле строк, где мы сверстаем вывод отдельного материала и структуру их вывода, соответственно. Жирным выделены названия текущих шаблонов. На рисунке выше уже созданы файлы views-view-unformatted—tours.tpl.php и views-view-fields—tours.tpl.php, эти названия более уникальны чем, названия шаблонов по умолчанию, поэтому представление будет выводится используя новые шаблоны.



Рисунок 41 – Шаблон вывода в стиле строк

На рисунке 41 показана разметка вывода одного материла типа – Тур. Данная разметка была скопирована из уже готовой верстки, только здесь статичный контент заменен динамичным при помощи php. Заменяются поля изображения, пути ссылки, заголовка и описания. Синтаксис php:

print $fields[‘имя поля’]->content;

Данная строка выводит содержимое поля, по его имени.

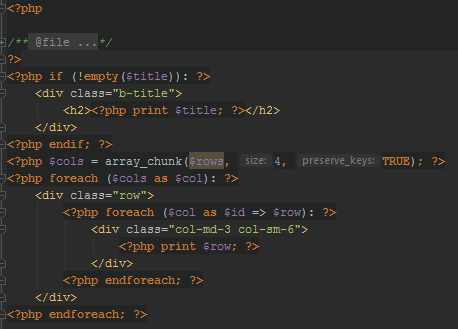


Рисунок 42 ­ Шаблон стиля вывода

На данном изображении (Рисунок 42) размечена структура вывода материалов. Массив $rows содержит все материалы типа – Тур. Для вывода будет использоваться сетка bootstrap, в каждой строке будет находится по 4 блока. Для этого делим массив всех материалов на 4, для того чтобы оборачивать новой строкой группу из четырех туров.

После сохранения последних двух шаблонов, можно будет увидеть результат пройдя на страницу туров. (Рисунок 43)

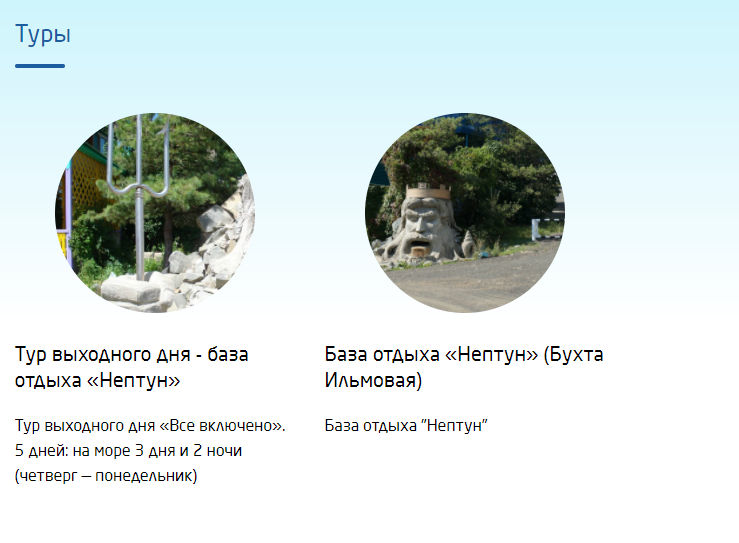


Рисунок 43 – Страница туры

Представления для вывода автобусов и экскурсий создаются, почти идентично.

Создание слайдера

Иногда при разработке сайтов со сложной структурой, возникает желание объединить несколько полей типа материала так сказать в один пакет, при необходимости который мы смогли в один клик добавить необходимое количество этих пакетов. Для подобных случаев существует модуль Field collection (Коллекция полей), который добавляет новый тип поля, при создании материала. Коллекция полей представляет собой новый тип материала, но также является полем другого материала. В данном случае тип материала будет слайдер с одним полем (слайд) типа – коллекция полей. Поле слайд сможет иметь неограниченное количество значений и свои поля.

Создаем новый тип материала – Слайдер – добавляем одно поле, с названием – Слайды (field\_slides), количество значений устанавливаем неограниченное. Теперь, в пункте меню структура появился новый раздел – коллекция полей, а внутри field\_slides, к которому уже можно добавить поле для заголовка, текстового контента и изображения слайда, а также текстовое поле для ссылки. (Рисунок 44, 45)

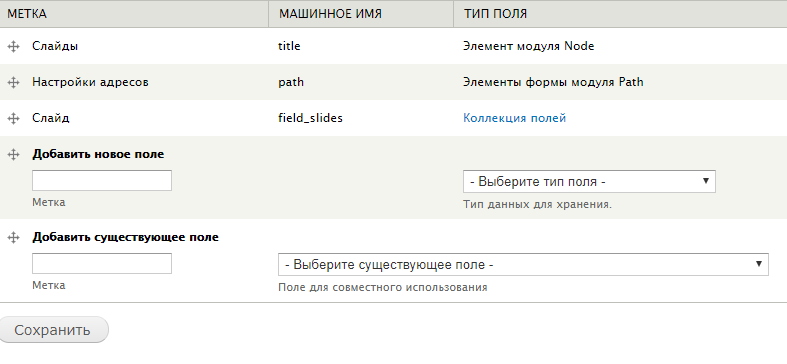


Рисунок 44 – Добавление полей к новому материалу

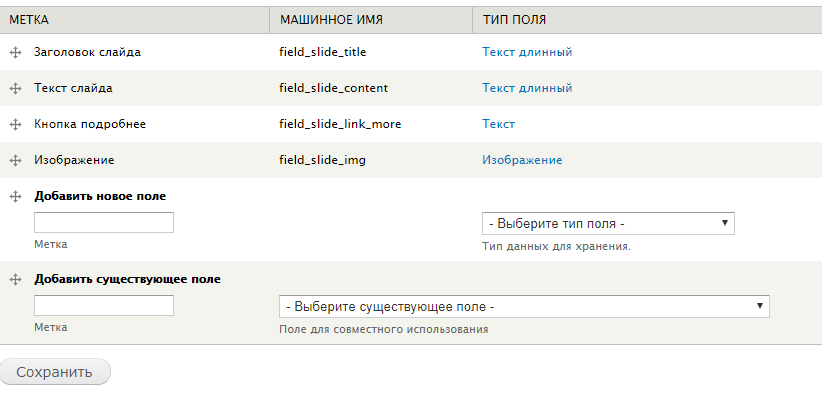


Рисунок 45 – Добавление полей в коллекцию полей

Далее, необходимо создать несколько тестовых слайдов, затем настроить их дальнейшее отображение при помощи модуля view. Настройка представления отличается от предыдущей только тем, что в нем будут отображаться элементы коллекции полей и вместо страницы будет создан блок, который как уже говорилось, может быть отображен в любом месте сайта. Для создания слайдера можно использовать любой jquery-плагин, большое количество которых можно найти в интернете. На данном сайте это будет плагин Swiper.

После настройки шаблонов, слайдер нужно вывести на главной странице под списком стран. Есть несколько способов программного вывода блоков, в Drupal 7 это делается следующим образом: (Рисунок 46)

Вывести только контент блока:

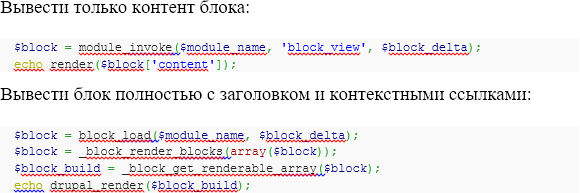


Рисунок 46 – Варианты вывода блоков

Еще существует программный вывод блока с контекстными фильтрами, но для данного слайдера это не нужно.

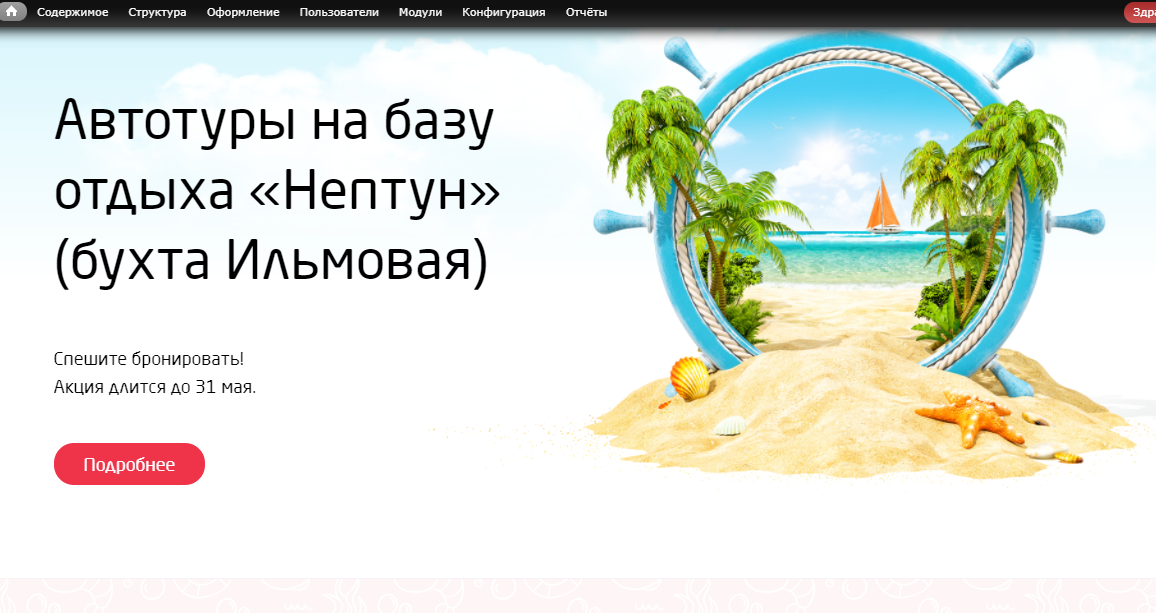


Рисунок 47 – Слайдер на главной странице

Модуль taxonomy позволяет упорядочить материалы по категориям. На сайте asiabus словарем таксономии будет являться список стран, дял его создания нужно пройти в пункт меню Структура -> Таксономия -> Добавить словарь. Ввести название, машинное имя и (необязательно) описание, затем сохранить словарь. (Рисунок 48)

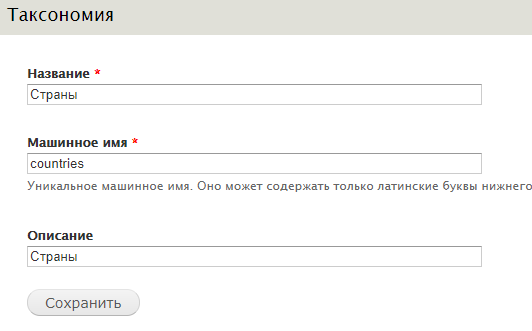


Рисунок 48 – Создания словаря таксономии

Далее в словарь добавляются термины. Для словаря стран – терминами будут страны.

Во внутренних страницах каждого термина должны находиться туры для текущей страны, то есть необходимо связать тип материала и термин таксономии. Для этого при создании полей в материале, существует поле с типом – Ссылка на термин, где необходимо выбрать словарь, на термин которого будет ссылаться материал, а при создании нового материала типа Тур выбирать для него соответствующую страну. (Рисунки 49, 50)

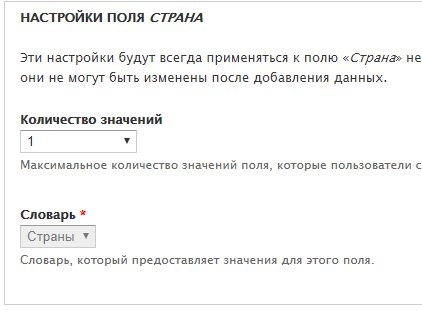


Рисунок 49 – Настройка поля типа – Ссылка на термин

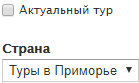


Рисунок 50 – Выбор термина при создании материала

Теперь необходимо отобразить словарь таксономии, то есть список стран на странице, как в макете. Для этого будет использоваться модуль views, только в качестве выводимых значений будут термины таксономии. (Рисунок 51)

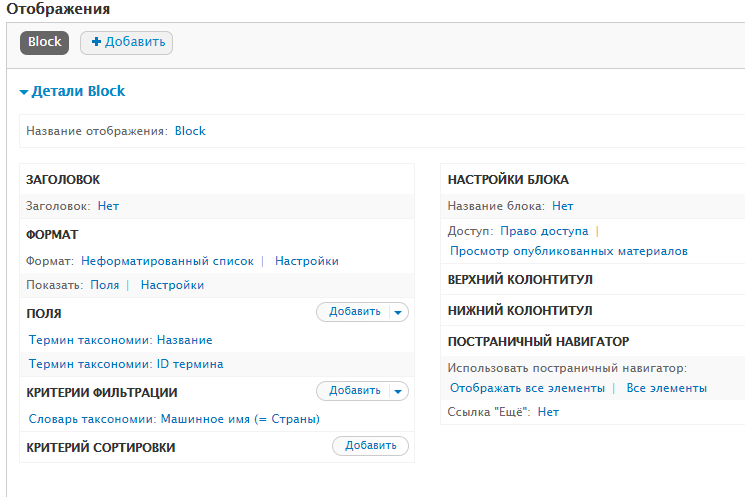


Рисунок 51 – Создание блока страны

После программирования шаблонов блок можно вывести на странице. (Рисунок 52)



Рисунок 52 – Блок страны

Также нужно настроить внутренне отображение термина таксономии – вывод туров для страны. Здесь нужно просто создать новое представление и задать контекстный фильтр по термину таксономии. Шаблоны идентичны шаблонам представления с материалами типа – Тур. (Рисунок 53)

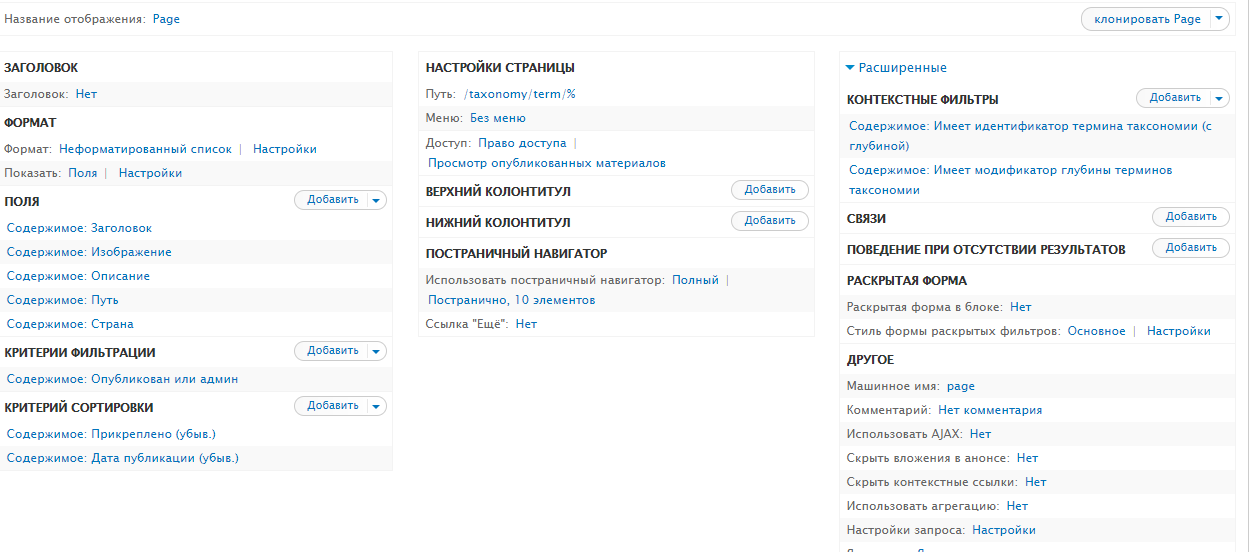


Рисунок 53 – Настройка представления термина таксономии

Навигационная цепочка (дублирующее меню, «хлебные крошки», англ. Breadcrumbs) – элемент навигации представляющий собой путь от корня сайта (главной страницы), который в данный момент просматривает пользователь. Для создания потребуется модуль Path Breadcrumbs. Переходим в пункт структура->path breadcrumbs->Добавить хлебные крошки. Для примера создадим страницу вывода туров. (Рисунок 54)

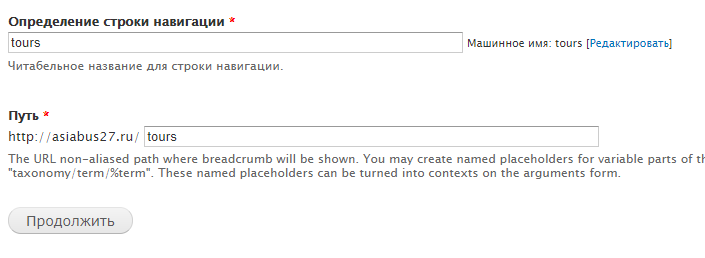


Рисунок 54 – Создание хлебных крошек

Следующие пункты аргументы и правила выбора пропускаются. В настройках строки навигации выбираем отображение ссылки на главную страницу и добавляем в строку ссылку на текущую страницу (название – Туры, путь <none>). Путь ссылки текущей страницы необходимо оставить пустым. (Рисунок 55)

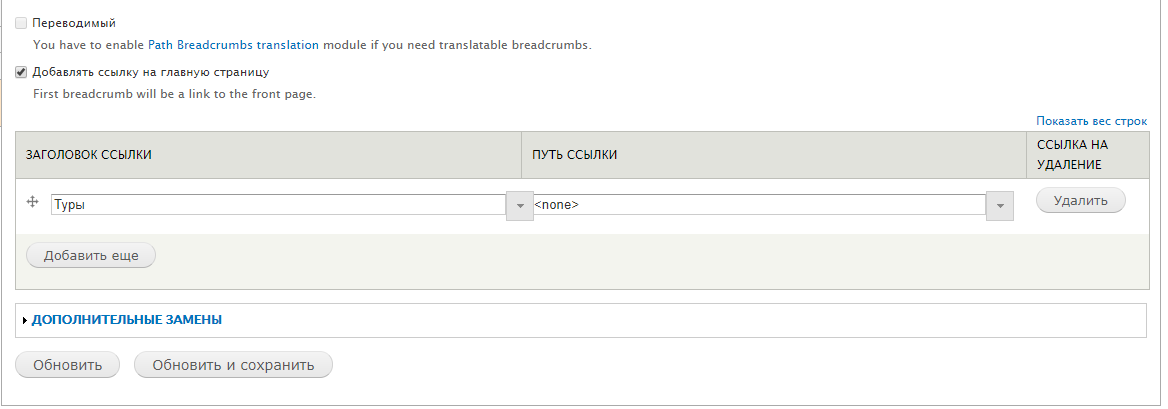


Рисунок 55 – Настройка строки навигации

Далее настроим хлебные крошки на внутренние страницы туров. Путь ко всем материалам можно посмотреть в шаблонах в разделе Конфигурация->Поиск и метаданные->Синонимы URL. В данном случае это будет путь node/%node (для машины). Далее настраиваем аргумент строки (Рисунок 56)

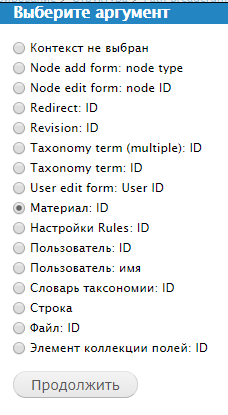


Рисунок 56 – Выбор аргумента

В правилах выбора добавляем критерий по типу материала (тур). (Рисунок 57)

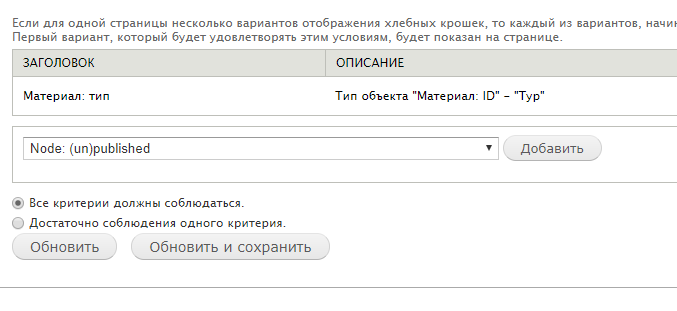


Рисунок 57 – Настройка правил выбора

В настройках строки навигации также добавляем ссылку на главную страницу, ссылку на страницу туров и ссылку на текущий материал. (Рисунки 58, 59)

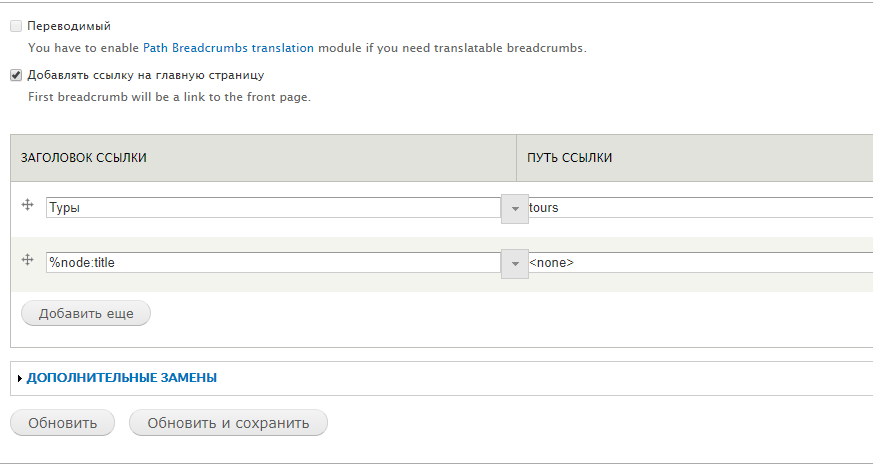


Рисунок 58 – Настройка строки навигации материала

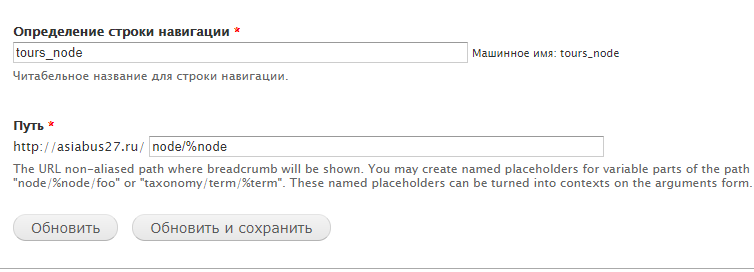


Рисунок 59 – Создание строки навигации

Для создания веб-форм нужен модуль webform, а также дополнительный модуль webform ajax для отправки форм без перезагрузки страницы.

Веб-формы являются новым типом материала и добавить их можно также, как и любой другой материал. (Рисунок 60)

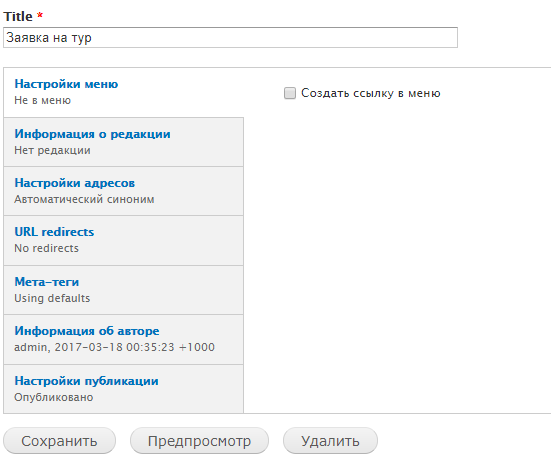


Рисунок 60 – Создание новой формы

Далее добавляем компоненты формы (поля материала). Метка будет выводиться в html-элементе label, если включить его отображение, ключ формы вставиться в значение атрибута name, элемента формы. В заявке на тур есть 3 поля для ввода текста (Имя, Электронная почта, Телефон). Имя и телефон обязательны. В настройках компонента можно изменить его метку (label), заполнитель (атрибут placeholder), сделать поле обязательным (атрибут required), настроить класс компонента и класс обертки. (Рисунок 61)

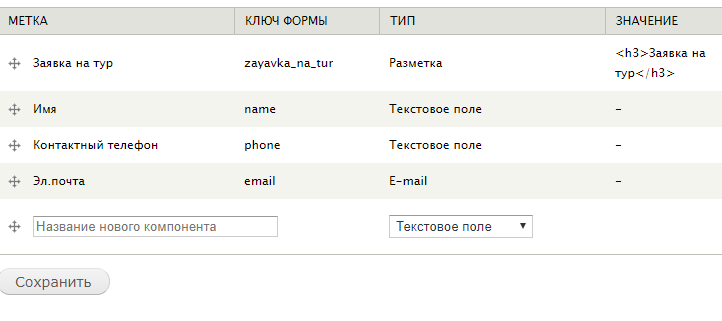


Рисунок 61 – Форма заявки на тур

Далее, в настройках формы вводим текст сообщения о подтверждении успешной отправки формы, включаем режим ajax и вводим текст для кнопки отправки, разрешаем использовать форму в качестве блока, чтобы можно было вывести ее в любом месте сайта. (Рисунок 62)

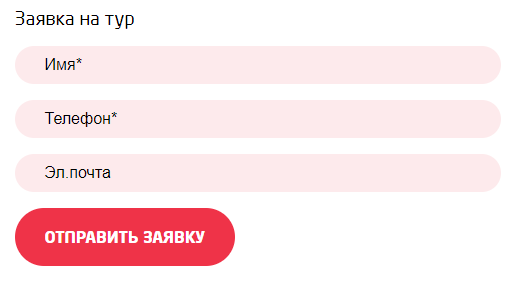


Рисунок 62 – Форма заявки внутри страницы тура

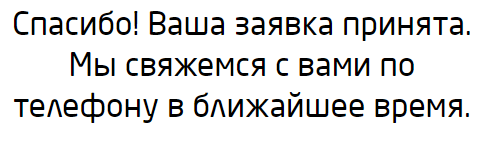


Рисунок 63 – Ответ после успешной отправки формы

На данном этапе создание темы завершено, теперь необходимо перенести весь каталог сайта на хостинг заказчика, в данном случае это будет timeweb. В период разработки сайт будет доступен по адресу <http://asiabus.pf27.ru>. После тестирования сайта и его доработки сайта, его можно перенести на основной хостинг и купить для него нормальное доменное имя. Сейчас на работающий сайт можно пройти по адресу <http://asiabus27.ru>.

3.3 Расчет экономической эффективности создания сайта

Постановка задачи:

Необходимо разработать web-сайт для тур-фирмы «Азия Бас»

Цели разработки:

* повышение прибыли за счет увеличения числа заказов;
* экономия рабочего времени на уведомлении клиента по телефону или почтой;
* экономия рабочего времени на выборке информации из базы данных и ручном заполнении счетов и квитанций.

При анализе экономической эффективности эффективности используются данные об увеличении количества принимаемых и обрабатываемых заказов на туры и экскурсии ООО «Азия Бас». Экономическая эффективность определяется на основе этих данных.

Проводя анализ экономической эффективности, для начала необходимо рассчитать стоимость разработки сайта. Далее следует вычислить эффективность результатов деятельности тур-фирмы «Азия Бас», используя результаты расчетов.

Таблица 5 – График выполнения работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Имя | Исполнитель | Длительность работ в марте 2017 г. | |
| дни | часы |
| 1 | Анкетирование заказчика | Менеджер проекта | 1 | 1 |
| 2 | Разработка технического задания | Менеджер проекта | 2-3 | 6 |
| 3 | Разработка дизайн-концепции сайта: главная страница сайта | Дизайнер, менеджер проекта | 2 | 12 |
| 4 | Разработка дизайна внутренних страниц сайта | Дизайнер, менеджер проекта | 2 | 12 |
| 5 | Программирование, разработка функционала | Программист | 8 | 60 |
| 6 | Тестирование и перенос сайта на хостинг заказчика | Программист, менеджер проекта | 3 | 20 |

Расчет стоимости разработки сайта ООО «Азия Бас»

Виды расходов, которые учитываются при расчете стоимости разработки сайта:

* стоимость материалов и покупных изделий;
* заработная плата (основная);
* заработная плата (дополнительная);
* страховые взносы;
* накладные расходы;
* затраты на машинное время (затраты на электроэнергию).

Затраты приведены в таблице 6.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Наименование | Единица измерения | Количество | Цена за единицу, руб | Стоимость, руб |
| 1 | Ручки, карандаши и т.п. | Шт | 5 | 20 | 100 |
| 2 | Бумага А4 | Упаковка | 1 | 150 | 150 |
| 3 | Картридж | Шт | 1 | 1500 | 1500 |
| 4 | Интернет | Гб | 3 | - | 800 |
| 5 | Итого: | | | | 2550 |
| 6 | Транспортные расходы (10% от п.5) | | | | 255 |
| 7 | Итого: | | | | 2805 |

Таблица 6 – Затраты на материалы и покупные изделия

Расходы на транспорт занимают 10% от общей суммы, и они составляют 255 руб.

В итоге, затраты на материалы и покупные изделия равны

руб

Расчет фонда заработной платы

К этой статье относится основная и дополнительная заработная плата разработчика web-сайта (программиста). Результаты расчета фонда заработной платы представлены в таблице 3.

Таблица 7 – Расчет фонда заработной платы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Должность: программист | Количество рабочих дней в мае 2017г. | Количество проработанных дней в мае 2017г. | Размер дневной оплаты, руб | Заработная плата в месяц, руб (май 2017г.) |
| 1 | Основная заработная плата | 20 | 19 | 2000 | 38000 |
| 2 | Дополнительная заработная плата (10% от основной) | | | | 3800 |
| 3 | Итого фонд заработной платы: | | | | 41800 |

В статью «Дополнительная заработная плата» входят выплаты, предусмотренные трудовым договором с руководством ООО «Азия Бас» на разработку web-сайта. Размер дополнительной заработной платы разработчика web-сайта определяется в размере 10 процентов от основной заработной платы.

руб

Следовательно, разработчику web-сайта в мае 2017 года всего начислено

руб

Таким образом, фонд заработной платы разработчика web-сайта в мае 2017 года составляет 41800 руб.

Отчисления на социальные нужды (страховые взносы)

К отчислениям на социальные нужды относят страховые взносы в ПФР, ФСС, ФФОМС и взносы на страхование от несчастных случаев на производстве и профзаболеваний.

Страховые взносы рассчитываются в размере 32,2 процента от фонда заработной платы, что составит

руб

Тарифы страховых взносов приведены в таблице 4.

Отчисления в пенсионный фонд ЗПФ составляют 24 процента от фонда заработной платы и равны

руб

Отчисления в фонд обязательного медицинского страхования Змс равны

руб

Отчисления на социальное страхование Зсс равны

руб

Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний равны

руб

Численные значения отчислений на социальные нужды (страховые взносы), представлены в таблице 8.

Таблица 8 – Расчет отчислений на социальные нужды (страховые взносы)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Отчисления на социальные нужды  (страховые взносы) | Тарифы страховых взносов, в % | Суммы страховых взносов (руб.) |
| 1 | Отчисления в ПФР |  |  |
| 2 | Отчисления в ФОМС |  |  |
| 3 | Отчисления в ФСС |  |  |
| 4 | Отчисления на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний |  |  |
| 5 | Итого: |  |  |

Накладные расходы

Накладные расходы, косвенные затраты – расходы, затраты, сопровождающие, сопутствующие основному производству, но не связанные с ним напрямую, не входящие в стоимость труда и материалов - дополнительные к основным затратам расходы для обеспечения процессов производства и обращения.

Накладные расходы Зн фирмы составляют 20 процентов (условно) от суммы основной и дополнительной заработной платы

руб

Затраты на машинное время

Как следует из данных таблицы 1, на разработку и последующую отладку web-сайта для ИП «Musica» потребовалось 18 рабочих дней (Дн).

В среднем с учетом перерывов программист работает за компьютером 6 часов в день. Себестоимость одного кВт/ч электроэнергии (С1кВт/ч) для организаций составляет 2 рубля 80 копеек. При проведении расчетов в проекте необходимо в расчеты брать существующие на дату расчета тарифы.

Суммарная мощность энергопотребителей для АРМ программиста складывается из мощности, потребляемой системным блоком персонального компьютера, монитором, принтером и другим периферийным оборудованием, которая составляет 1,2 кВт.

Следовательно, за 6 часов работы программиста суммарное энергопотребление за день составит:

кВт/ч

Таким образом, стоимость машинного времени Змаш, необходимого для разработки web-сайта, составит:

руб

Затраты на машинное время учитываются как затраты на электроэнергию.

В результате выше произведенных расчетов мы получили итоговые затраты на разработку web-сайта (Таблица 9).

Таблица 9 – Итоговые затраты на разработку сайта

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование статей расхода | Сумма, руб |
| 1 | Стоимость материалов и покупных изделий | 2805,00 |
| 2 | Основная заработная плата | 41800,00 |
| 3 | Дополнительная заработная плата | 3800,00 |
| 4 | Отчисления на социальные нужды (30,2% от п.2 и п.3) | 12 624,60 |
| 5 | Накладные расходы (20% от п.2 и п.3) | 7920,00 |
| 6 | Затраты на машинное время (затраты на электроэнергию) | 362,88 |
| 7 | Итого: | 69 312,48 |

Цена программного продукта

Цена программного продукта (web-сайта) Ц определяется итоговыми затратами и прибылью, которая, в свою очередь, составляет 30 процентов (условно) от фонда заработной платы:

руб

Оценка экономической эффективности

Разработка web-сайта преследовала цель экономии рабочего времени специалистов на проведение устных консультаций с покупателями по телефону, уведомление клиентов о принятии их заказов по телефону либо почтой, на работе операторов ЭВМ с базой данных и формировании счетов и квитанций.

Одним из основных методов определения экономической эффективности служит метод сравнения результатов работы до и после проведения мероприятия.

Так как в данный момент сайт находится в процессе продвижения, нет возможности оценить его эффективность, так как он еще не попадает в топ по ключевым запросам. Хотя сейчас на сайте имеется несколько заявок, их количества недостаточно для сравнения с положением дел у фирмы, до создания сайта.

Заключение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Итан Браун. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. – Санкт-Петербург: Питер, 2017. 336 с.
2. Тодд Томлинсон. CMS Drupal 7: руководство по разработке системы управления веб-сайтом, 3-е издание – Москва: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. 560 с.
3. Мелансон Б., Нордин Д., Луиси Ж. Профессиональная разработка сайтов на Drupal 7. – Санкт-Петербург: [«Питер»](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2013. 688 с.
4. Черных А. Drupal 7. - [«Эксмо»](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BC%D0%BE_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2011. С. 208.
5. Колисниченко Д. Н. Drupal 7. Руководство пользователя. - [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. С. 256.
6. Ромашов В. CMS Drupal: система управления содержимым сайта. – Санкт-Петербург :[«Питер»](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)), 2010. С. 256.
7. Robert T. Douglass, Mike Little, Jared W. Smith «Building Online Communities with Drupal, phpBB, and WordPress». ApressDavid Mercer «Drupal: Creating Blogs, Forums, Portals, and Community Websites»
8. John K. VanDyk, Matt Westgate «Pro Drupal Development»
9. В. Васвани. MySQL: использование и администрирование. — Москва: «Питер», 2011. 368 с.
10. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста, 2-е издание. Москва: «Диалектика», 2010. 912 с.
11. Котеров Д., Костарев А. PHP. — Санкт-Петербург.: «БХВ-Петербург», 2005. С. 1120.
12. Костарев А. Ф. PHP 5. – Санкт-Петербург: «БХВ-Петербург», 2008. С. 1104.
13. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования – 3-е издание. — Москва: «Вильямс», 2010. С. 560.
14. Кристиан Дари, Эмилиан Баланеску. PHP и MySQL: создание интернет-магазина. — Москва: «Вильямс», 2010.
15. Джейсон Ленгсторф. PHP и jQuery для профессионалов. – Москва: «Вильямс», 2010. С. 352.
16. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. – 2-е издание. – Москва: «Диалектика», 2010. С. 912.
17. Квентин Зервас. Web 2.0: создание приложений на PHP. – Москва: «Вильямс», 2009. С. 544.
18. Кузнецов Максим, Симдянов Игорь. PHP 5/6. – Санкт-Петербург.: «БХВ-Петербург», 2009. С. 1024.
19. Э. Фримен, Э. Фримен. Изучаем HTML, XHTML и CSS. – Санкт-Петербург: «Питер», 2010. 656 с.
20. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. – Москва: «Диалектика», 2011. 400 с.
21. Питер Лабберс, Брайан Олберс, Фрэнк Салим. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. – Москва: «Вильямс», 2011. 272 с.
22. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. – Москва: «Диалектика», 2010. 656 с.
23. Фримен Эрик, Фримен Элизабет. Изучаем HTML, XHTML и CSS. – 1-е изд. – Москва: «Питер», 2010. С. 656.
24. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. – Москва: «Диалектика», 2011. 400 с.
25. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. – Москва: «Диалектика», 2011. 656 с.
26. Энди Бадд, Камерон Молл, Саймон Коллизон. CSS: профессиональное применение Web-стандартов. – Москва: «Вильямс», 2008. 272 с.
27. Кристофер Шмитт. CSS. Рецепты программирования. – Санкт-Петербург: БХВ-Петербург, 2007. 592 с.
28. Эрик А. Мейер. CSS-каскадные таблицы стилей: подробное руководство. – Москва: Символ, 2006. 576 с.
29. Адам Фримен. jQuery для профессионалов. – Москва: «Вильямс», 2012. 960 с.
30. Джейсон Ленгсторф. PHP и jQuery для профессионалов. – Москва: «Вильямс», 2010. С. 352.
31. Самков Г. jQuery. Сборник рецептов. – Санкт-Петербург: БХВ-Петербург, 2010. С. 416.
32. URL: http://php.net/
33. URL: http://www.pf27.ru/
34. URL: https://www.drupal.org/
35. URL: http://gulpjs.com/
36. URL: https://yandex.ru/referats/?ncrnd=3492
37. URL: http://getbootstrap.com/
38. URL: https://docs.emmet.io/
39. URL: https://htmlacademy.ru/
40. URL: http://lesscss.org/
41. URL: https://github.com/
42. URL: https://git-scm.com/
43. URL: https://yarnpkg.com/lang/en/
44. URL: https://www.npmjs.com/
45. URL: https://bower.io/
46. URL: https://www.jetbrains.com/phpstorm/
47. URL: https://www.sublimetext.com/
48. URL: https://conemu.github.io/
49. URL: https://photoshop-master.ru/
50. URL: https://fonts.google.com/