

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Разработка web-приложения для начинающих писателей «Writing for Yourself»

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА

Студентки 4 курса 441 группы

дополнительной профессиональной программы Специалист в области
компьютерной графики и WEB-дизайна (WEB-дизайнер)

_____ механико-математического факультета

_____ Задворной Ирины Александровны

Научный руководитель
_____ ст. преподаватель

_____ Л.В. Бессонов

Саратов 2017 год

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Основы веб - разработки	5
1.1 О разработке.....	5
1.2 Этапы разработки.....	7
1.3 PHP для разработки.....	10
1.4 HTML и CSS для разработки.....	14
1.5 CMS Drupal	17
1.6 База данных MySQL.....	20
2 Реализация веб - приложения	23
2.1 Изучение темы	23
2.2 Создание логотипа	24
2.3 Разработка макета и дизайна приложения	26
2.4 Верстка адаптивного интерфейса приложения	31
2.5 Модули системы	34
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	38
ПРИЛОЖЕНИЕ А Вёрстка главной страницы	40

ВВЕДЕНИЕ

Создание веб - приложений для различных целей в последнее время стало общедоступным. Люди продают свои продукты и услуги через Интернет, делятся различной информацией.

Созданный веб - сайт может служить как персональным, так и профессиональным целям. Веб - разработка считалась сложным и дорогостоящим процессом в прошлом, но теперь, благодаря достижениям в области технологий, она стала удобной. Преимущества предоставляются как разработчикам, так и пользователям, и это подтверждается увеличением количества веб - сайтов. Увеличение пространства, предоставляемого пользователю, является причиной того, что эта технология приобретает большую ценность, чем другие технологии.

Важные вещи, которые следует учитывать при построении веб - сайта, включают его общий вид, выбранную тему, используемые цвета, контент, который он отображает, с той легкостью, которую он предоставляет всем своим посетителям при выполнении различных функций для технических целей. Такие аспекты, как простота кодирования, должны учитываться в процессе формирования любого веб - сайта. Общее мнение, популярное среди масс, заключается в том, что разработка веб - страниц — это не дизайн, а кодирование и написание разметки, но веб - разработка в основном касается требований пользователей, ее функциональности и интерфейса.

Планирование любого веб - сайта в первую очередь начинается с того, чтобы получить представление о том, что такое веб - сайт, и о том, что он обычно предлагает посетителю. Компания предоставляет всю необходимую информацию, с которой необходимо работать. Затем в строке идет часть художественной и уникальной дизайнерской идеи, которая должна служить всей цели. Конструкция должна быть как привлекательной визуально, так и полностью функциональной. Как только базовая идея сформирована, она разбивается на разные компоненты и снова пересматривается, чтобы убедиться, что все в порядке, до начала окончательной работы по формированию. Как только все встанет на свои места, начинается процесс развития. Веб - разработка включает в себя все специализированные функции, которые задействованы во всем процессе формирования. Основным требованием

является понимание всех технологий, участвующих в этом процессе. Клиент и разработчик должны иметь обширный диалог, в котором клиент должен подробно объяснить разработчику все его потребности, требования и цели, только тогда может быть сформирован веб - сайт, достаточно компетентный для получения хороших результатов. Затем инструменты и язык, разработанные для создания веб - сайта, используются для создания веб - сайта высокого качества и служат всем целям.

Целью работы является создание полноценного веб - приложения для начинающих писателей «Writing for Yourself».

В ходе разработки будут решены следующие задачи:

- Определение основных терминов, используемых при веб - разработке;
- Рассмотрение особенностей PHP;
- Изучение HTML и CSS для разработки;
- Определение CMS Drupal как основного средства реализации;
- Рассмотрение основных используемых СУБД и определение особенностей разработки с MySQL;
- Проведение поэтапной разработки веб - приложения с изучением особенностей каждого из этапов.

Итоговой задачей исследования данной темы является получение теоретических знаний в этой области и возможности их применения для решения частных и общих задач на практике с созданием в конечном итоге полноценного ресурса.

1 Основы веб - разработки

1.1 О разработке

Начало сети было основано на идее исключительно как технологии, которая будет использоваться для обмена идеями на академическом уровне, но теперь эта технология развивается как открытое публичное место, где рассматривается сочетание как личных, так и профессиональных сайтов. С течением времени Интернет начал заполняться личными веб - сайтами, на которых люди делились своими личными биографиями. После этого начался большой бум различных компаний, которые начали создавать свои веб - сайты, чтобы продвигать свой бизнес, но наряду с ростом их бизнеса сеть росла с экспоненциальной скоростью и с тех пор не прекращалась. Веб начал служить цели онлайн - рынка, где люди покупали и продавали всевозможные продукты и предлагали всевозможные услуги.

Теперь, начиная с жизненного цикла создания веб - сайта, команда, занимающаяся разработкой, оценивает и пересматривает всю информацию и идеи, которые организуются командой дизайнеров в сотрудничестве с клиентами. Веб - разработка требует понимания всех процессов, которые являются частью создания веб - сайта. Команда разработчиков несет наибольшую ответственность за завершение создания веб - сайта. Лица, которые участвуют в процессе разработки веб - сайта, как правило, находят свое уникальное видение сайта, придавая ему отдельную индивидуальность.

Процесс выполняется с помощью различных специализированных языков, таких как HTML, CSS, PHP [1]. У этих языков есть своя специальная логика, и для создания высокоэффективной веб - страницы веб - разработчик должен владеть всеми языками, создание и обслуживание которых попадает под веб - разработку.

Как и в случае с традиционным настольным приложением, веб - приложения имеют разные уровни риска [2]. Личная домашняя страница гораздо менее рискованна, чем, например, веб - сайт для торговли акциями. Для некоторых проектов важны проблемы безопасности и ошибки программного обеспечения. Если время выхода на рынок или техническая сложность вызывают озабоченность, то документация, планирование тестирования, контроль

изменений, анализ требований, архитектурное описание, формальный дизайн и методы строительства могут снизить риск.

Веб - разработка стала очень популярной благодаря новейшим тенденциям и технологическим достижениям. Процесс веб - разработки дает жизнь статическому дизайну и делает его живым. Этот процесс включает в себя различные этапы, такие как выбор имени домена, разработка таблицы стилей, сценарии, навигация и программирование [3]. Уникальный контент загружается, чтобы повысить видимость веб - сайта в поисковой системе. Помимо навыков программирования, веб - разработчику может потребоваться понимание SEO и отслеживания показателей. Первоначальному дизайну сайта могут потребоваться некоторые изменения, чтобы получить хорошие результаты в поисковой оптимизации. Веб - разработчику, возможно, потребуется внести дополнительные изменения в дизайн сайта в соответствии с совместимостью браузера.

Очень важно не забыть основные аспекты разработки приложения, которые должны быть продуманы заранее. Рассмотрим кратко важные детали, а в дальнейшем подробнее каждый этап разработки.

Планирование играет ключевую роль в создании эффективного веб - сайта. Планирование дает скелетный позвоночник или линии, вдоль которых при создании возможно двигаться вперед.

Сбор всей необходимой и актуальной информации по тематике сайта является первым шагом в процессе развития. Информация должна содержать основную цель создания веб - сайта и целевого типа клиента. Это облегчает работу, так как появляется четкое представление о том, как действовать дальше.

Для удобства создается первоначальная структура. Это в основном внутренняя структура веб - сайта. Информация, собранная на первом этапе, используется здесь.

После дизайнер создает схему того, как страница будет выглядеть для всех, кто её посещает. Это имеет первостепенное значение, поскольку оно решает, как посетители, посещающие сайт, будут взаимодействовать с системой.

Написание содержания, которое является релевантным и описательным, играет важную роль для успеха сайта. Эта работа выполняется авторами контента.

Программирование имеет дело с функциональными возможностями, так что сайт работает плавно и в полном объеме без каких - либо сбоев. Итоговый вид должен полностью удовлетворять требованиям клиента, а также привлекать внимание аудитории.

Окончательное тестирование выполняется в конце, когда проверяются все функции, проекты, сценарии, ссылки для обеспечения бесперебойной работы веб - страниц.

После тестирования сайт загружается на сервер. После загрузки сайт нужно время от времени обновлять, чтобы он мог идти в ногу с тенденциями рынка. Добавление новых функций и изменение старых должно выполняться по отзывам текущей аудитории сайта.

Существуют различные способы, которые приводят к эффективной веб - разработке. Далее будут рассмотрены основные технологии, позволяющие достичь успеха при создании ресурса.

1.2 Этапы разработки

Каждое веб - приложение проходит процесс разработки, прежде чем он станет доступен в релизной версии.

Веб - приложение представляет собой часть программного обеспечения, которое разработано для пользователей. Такая информационная система, как и любое другое программное обеспечение, должна пройти последовательный процесс разработки. Процесс в большинстве проектов проходит через почти идентичные фазы. В некоторых разработках две фазы могут быть объединены вместе, чтобы сделать один процесс. В других случаях он может быть разделен между двумя подпроцессами.

Рассмотрим основные шаги в разработке веб - приложения.

Каждая компания придерживается стандартных методологий, инструментов моделирования, стандартов и языков для разработки веб - приложе-

ния. Определим основные этапы проектирования, которые должно пройти любое веб - приложение [4].

Этап №1 — Сбор информации. Первым этапом в этом процессе является сбор информации. Это также самое важное, так как любое недопонимание на данном этапе полностью препятствует всему дальнейшему процессу разработки. На этом этапе необходимо понимание компании и её деятельности. Он также будет включать понимание целей разрабатываемой системы, пожеланий заказчиков, целевой аудитории и возможностей компании. Очень важно задать ряд вопросов на этом этапе. Даже небольшое сомнение в отношении точных требований должно быть устранено на данном этапе. Это позволит убедиться, что в конечном продукте нет ошибок. Наряду с информацией, также должны быть поняты точные требования. С помощью этой собранной информации и требований веб - приложение и будет разработано.

Этап №2 — Анализ требований. После того, как необходимая информация собрана, следующим этапом в этом процессе является анализ требований. На этом этапе проводится полный анализ того, как развитая система будет помогать существующей и заменять её. В то же время необходимо также провести анализ, чтобы увидеть, как веб - приложение будет объединено с существующей системой. Целевая аудитория веб - приложения также должна быть принята во внимание. Если это новое веб - приложение, необходимо проанализировать требования, чтобы получить точное представление о том, как должен выглядеть конечный продукт.

Этап №3 — Планирование. Анализ должен иметь возможность разработать экономически эффективную систему. На этом этапе необходимо подготовить карту сайта для веб - приложения. Весь сайт может быть разделен на темы и подтемы. Разделение окажется полезным для сортировки данных. В то же время, план может быть приобщен для навигации по системе. Планирование должно последовательно учитывать потребности конечных пользователей. Интерфейс должен быть выполнен таким образом, чтобы упростить навигацию между различными страницами и разделами веб - приложения. План также должен принимать различные формы: всплывающие окна и флэш - приложения, приложения электронной коммерции и прочее. Он должен учитывать их.

Этап №4 — Дизайн. После планирования переходят на этап проектирования веб - приложения. На этом этапе учитываются логотип компании, цвета, фон и прочие элементы общей концепции, которые будут использоваться в веб - приложении. Должно быть создано более одного прототипа всего проекта. Прототипы обычно представляются клиенту. Клиент утверждает один из многих прототипов, а затем дальнейшая работа должна быть выполнена по выбранному дизайну. Это важный этап в процессе разработки портала. Прототип должен соответствовать требованию и симпатии клиента. Лучше работать в тесной координации с клиентом на данном этапе, чтобы это не приводило к возникновению проблемной ситуации. На этом этапе команда разработчиков будет учитывать требования к базе данных. Дизайн базы данных будет одновременно завершен.

Этап №5 — Разработка. После того, как дизайн был одобрен, начинается реальный процесс разработки. Обычно процесс начинается с разработки домашней страницы. Существуют прототипы или шаблоны, созданные для других страниц. На этом этапе реализованы все необходимые элементы в веб - приложении. На этом этапе разработчики также будут работать над разработкой базы данных.

Этап №6 — Содержание. Контент является важной частью веб - приложения. Написание текстов производится таким образом, что клиенты привлекаются к продукту или услугам, предлагаемым компанией. Разработка контента может быть назначена профессиональным разработчикам (например, копирайтерам). Их целью будет разработка оригинального, информативного и релевантного контента.

Этап №7 — Тестирование. Как и любое программное обеспечение, веб - приложение также должно пройти жизненный цикл тестирования программного обеспечения. Тестирование веб - приложения является более строгим. Процедура должна учитывать многопользовательский режим работы, а также требования к системе. Кроссбраузерное тестирование также выполняется, чтобы убедиться, что веб - приложение действительно работает во всех браузерах без сбоев. В системе проводятся как автоматические, так и ручные процедуры тестирования. После того, как все внутренние тесты были выполнены, начинается тестирование веб - приложения в реальном времени.

Этап №8 — Размещение. После окончательного утверждения веб - приложения происходит его размещение. Обычно службы по размещению также предлагают регистрацию доменных имен, а также услуги веб - хостинга. Можно найти недорогой веб - хостинг, чтобы снизить стоимость проекта. Приложение загружается на веб - сервер, и выполняются последние тесты. Последний тест гарантирует, что все необходимые файлы были загружены, а система полностью работоспособна.

Этап №9 — Техническое обслуживание и обновление. Это важный шаг во всем процессе. Важно обновить сайт с помощью новых функций. Для чего веб - приложению, возможно, придется снова пройти весь цикл. В случае каких - либо проблем принимаются меры для исправления в ближайшие сроки, так как прежняя версия проекта в этот момент функционирует.

По завершении процесса разработки необходимо будет принять меры для увеличения трафика веб - приложения. Также необходимо будет принять различные меры, чтобы система была включена в веб - каталоги, что поможет ей улучшить трафик и обеспечить посещаемость клиентов. Также можно рассмотреть возможность индексации веб - приложения различными поисковыми системами, поскольку это также поможет в увеличении трафика. Это завершает процесс проектирования системы.

1.3 РНР для разработки

РНР — это серверный скриптовый язык, разработанный в основном для веб-разработки, но также используемый в качестве языка программирования общего назначения. Первоначально был создан в 1994 году, эталонная реализация РНР теперь производится командой разработчиков РНР [5].

РНР - код может быть встроен в разметку HTML или может использоваться в сочетании с различными системами веб - шаблонов, системами управления веб - контентом и веб - каркасами. РНР - код обычно обрабатывается интерпретатором РНР, реализованным как модуль на веб - сервере или в качестве исполняемого файла Common Gateway Interface (CGI). Программное обеспечение веб - сервера объединяет результаты интерпретируемого и исполняемого РНР - кода, который может представлять собой любой тип

данных. PHP - код также может быть выполнен с помощью интерфейса командной строки (CLI) и может использоваться для реализации автономных графических приложений.

Стандартный PHP - интерпретатор, основанный на Zend Engine, является свободным программным обеспечением, выпущенным под лицензией PHP. PHP был широко портирован и может быть развернут на большинстве веб - серверов практически на всех операционных системах и платформах бесплатно.

Язык PHP развивался без письменной официальной спецификации или стандарта до 2014 года, оставив канонический PHP - интерпретатор стандартом. С 2014 года продолжалась работа по созданию формальной спецификации PHP.

Последние годы привели к заметным изменениям в индустрии PHP. Современной версией языка PHP является PHP 7, который позволяет повысить производительность и оптимизировать использование памяти. Благодаря своему улучшенному характеру PHP также становится более подходящим для решений Internet of Things.

Прошло десять лет до того, как была выпущена современная версия PHP. PHP 7 ориентирован на высокую производительность. Производительность приложений с использованием PHP 7 выросла в два раза по сравнению с предыдущей версией PHP 5.6. Многие ошибки, которые ранее были причиной прекращения работы приложения, теперь могут обрабатываться как исключения. Основные популярные рамки готовы для PHP 7. Все эти улучшения языка PHP позволяют увеличить скорость приложения и сократить количество необходимых ресурсов.

PSR-7 улучшает работу с HTTP - запросами, что, в свою очередь, позволяет сделать веб - разработку PHP и MySQL более гибкой. PSR - 7 — это стандарт, который определяет интерфейсы для HTTP - сообщений. Это позволяет достичь большей функциональной совместимости между каркасами и предоставлять более эффективные решения различными способами.

Эксперты предсказывают к 2025 году, что половина американских предприятий проведет более 10 заявок. Хотя PHP не может быть лучшим выбором для любых веб - решений, он по - прежнему занимает лидирующие

позиции в разработке технологических стартапов. С помощью PHP можно быстро создать жизнеспособное приложение в краткосрочной перспективе, чтобы увидеть, каким образом проект должен продвигаться вперед.

В итоге, PHP — это хорошо известная технология, используемая для служб веб - разработки. Во - первых, это открытый источник. Во - вторых, это экономически выгодно, и многие разработчики по всему миру используют его. В - третьих, это одно из самых эффективных средств современной веб - разработки. То, как изменилось PHP, со временем можно проследить. Некоторые изменения были отменены сразу после их введения, но некоторые остались с технологией в течение некоторого времени.

Веб - разработка предоставляет возможность для эффективной онлайн - презентации. Передние разработчики PHP знают, как сделать сайт привлекательным и профессиональным. Использование умной инфографики также позволяет пользователям проводить больше времени на веб - сайте.

Для некоторых пользователей спецэффекты — довольно важный сегмент, независимо от того, посещают ли они сайт или просто играют в игру, а технология PHP используется для воплощения этих функций в жизнь. Одним из примеров такого опыта является прокрутка Parallax. После его введения он дает сайту более анимированный вид. Используя этот метод, можно создать баннеры и изображения, размещенные и прокручиваемые таким образом, чтобы дать 3D - эффект.

Промышленность PHP растет и меняется из года в год. Основные тенденции показывают, что наиболее значительные изменения были направлены на повышение производительности и функциональной совместимости веб - приложений. Это позволяет реализовать более эффективные решения как в амбициозных, так и в обычных проектах, построенных с использованием этого гибкого и совместимого языка программирования.

Подведем итог и отметим основные преимущества PHP:

- Открытый исходный код. PHP свободно доступен под общедоступной лицензией. Наряду с PHP большинство из сопутствующих и необходимых для разработки программ, таких как текстовые редакторы, MySQL и Apache Server также доступны. Он общепризнан как язык с открытым

исходным кодом, предоставляющий лучшие инструменты и конечные решения для удовлетворения всех потребностей разработчиков.

- Перекрестная платформа. PHP позволяет создавать кросс - платформенные приложения, поскольку он лучше всего работает на платформах LINUX, UNIX и Windows. Он легко взаимодействует с MySQL и Apache и совместим с основными веб - серверами и операционными системами, что упрощает его развертывание на нескольких различных платформах.
- Удобство. PHP предлагает пользователям упростить использование языка программирования. Он обеспечивает гибкость в изучении и устранении сложности изучения синтаксиса или архитектуры нового языка. PHP очень надежный и масштабируемый по сравнению с другими языками.
- Быстрая разработка. Создание сайтов с PHP является кратким и простым. Это мгновенный и быстрый язык программирования, удовлетворяющий всем потребностям бизнеса. PHP хорошо работает с сетью, HTML, а доступ к GET и POST — это встроенные средства на этом языке сценариев на стороне сервера.
- Многоуровневая безопасность. PHP предоставляет доступный способ защитить ресурс от вредоносных атак, угроз и вирусов. Он поставляется с несколькими уровнями безопасности для защиты от ошибок и внешних вторжений.
- Высокая производительность. PHP сокращает время разработки и рабочую нагрузку на сервер, используя собственную память. Он работает намного быстрее в обычном веб - стеке и использует веб - сервер Apache, который увеличивает скорость обработки. PHP — это зрелый язык, который сохраняет скорость даже при использовании с другими программными продуктами.
- Гибкость. PHP — идеальный язык программирования, обеспечивающий максимальный уровень гибкости по сравнению с другими языками. PHP лучше всего подходит для пользовательских требований бизнеса, поскольку его модульная система способна адаптировать этот язык к нескольким библиотекам, таким как XML и многое другое.

- Расширяемость. PHP как новаторский язык поставляется с многочисленными расширениями для дополнения набора основных функций. Программный код можно изменять и интегрировать с любыми настройками расширений.
- Многопользовательская работа. PHP имеет возможность обрабатывать веб - сайты с большим количеством трафика. Являясь языком сценариев на стороне сервера, возможно сделать все, что может сделать любая программа CGI, например, отправка или получение файлов cookie, создание динамического содержимого страницы и многое другое.
- Веб - разработка. PHP может использоваться как для интерфейсной, так и для внутренней веб - разработки, что приводит к более быстрому построению системы.
- Совместимость. PHP совместим со всеми известными системами ОС, такими как Unix, Windows. Он обеспечивает универсальность пользователям, предоставляя разнообразие. PHP также поддерживает MacOS, тем самым делая его самым популярным языком программирования.

Если перед разработчиком стоит задача построить динамический веб - сайт или полноценное веб - приложение, то выбор PHP — это абсолютно правильный выбор. Это позволяет без проблем управлять и обновлять ресурс. Предлагая множество преимуществ, эта функциональность богатого языка доступна бесплатно и совместима с различными платформами.

1.4 HTML и CSS для разработки

HTML — это стандартный язык разметки для создания веб - страниц [6]. HTML описывает структуру веб - страниц с использованием разметки. HTML - элементы являются строительными блоками HTML - страниц. Элементы HTML представлены тегами. HTML - теги обозначают фрагменты контента, такие как «заголовок», «абзац», «таблица». Браузеры не отображают теги HTML, но используют их для отображения содержимого страницы.

HTML использует predetermined набор элементов для разных типов контента [7]. Элементы заключаются в открытие и закрытии тегов (за неко-

торыми исключениями). Тэги — это короткие фрагменты текста, указывающие тип содержимого, заключенный в окружность с угловыми скобками (<>). Закрывающий тег выглядит как открытый тег, за исключением того, что перед первой угловой скобкой имеет косую черту (/).

Элементы HTML содержат другие элементы для создания иерархической структуры. Ниже представлен фрагмент простой, но полной html - страницы:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The page title</title>
  </head>
  <body>
    <p>This is a basic page with a paragraph in it.</p>
  </body>
</html>
```

Первая строка сообщает браузеру, что он должен интерпретировать текст как HTML5.

Все остальное содержимое на странице окружено одним набором тегов. Внутри этого есть две секции — голова и тело. Глава содержит метаданные о странице, которая не отображается на самой странице, например заголовок (как показано выше). Этот раздел также может включать ссылки на другие файлы, которые включены в страницу. Тело — это место, где весь контент, отображаемый на странице, похож на элемент абзаца, который мы видим выше.

Иерархическую, вложенную структуру HTML называют Document Object Model, или DOM. Аналог DOM — ветвистое дерево. Возможность доступа к элементам дерева важна, когда необходимо манипулировать HTML с помощью CSS или JavaScript.

HTML обеспечивает семантическую структуру содержимого сайта.

Содержимое страницы + HTML (семантическая структура) + CSS (стилевое оформление) = Структура сайта.

Каскадные таблицы стилей или CSS используются для указания веб - браузерам, как стилизовать и отобразить HTML - структурированный текст. Хотя HTML определяет контент, CSS предоставляет его стилистическое оформ-

ление. Разделение содержимого на стили упрощает чтение кода, а также позволяет легко менять один стиль на другой, создавая несколько стилевых файлов.

При выделении CSS в отдельный файл создаётся новый с расширением файла .css и его связь с <head> HTML - файла описывается с помощью тега <link>.

Первая часть, селектор, сообщает браузеру, к каким элементам на странице применяется правило. Формат правил на примере:

```
selector {  
  property: value;  
}
```

За селектором следует набор фигурных скобок, которые заключают одну или несколько пар «property:value». Свойство, которое нужно установить, и значение для этого свойства, разделенное двоеточием и сопровождаемое точкой с запятой.

В работе используются последние версии HTML и CSS. Отметим их основные особенности.

HTML5 имеет много новых синтаксических функций. HTML5 поставляется с элементами <audio>, <video> и <canvas>. Благодаря этим новым элементам будет очень легко интегрировать мультимедийный и графический контент в Интернет без использования плагинов Flash и сторонних плагинов.

HTML5 также содержит новые элементы, такие как <section>, <article>, <header> и <nav>, что обогащает семантическое значение документа.

Рассмотрим преимущества HTML5:

- Улучшенный код. HTML5 позволяет использовать более чистый код. Можно удалить теги div и заменить их семантическими элементами HTML5.
- Элегантные формы. HTML5 позволяет использовать более удобные формы. Разные типы входов, поиск и разные поля для разных целей.
- Последовательность. Более согласованная структура с точки зрения HTML.
- Поддержка мультимедийных элементов. HTML5 имеет встроенную возможность воспроизведения аудио и видео.

- Кэш офлайн-приложений. HTML5 предлагает автономный инструмент кэширования приложений, который будет загружать страницу, которую посетил пользователь, даже если пользователь временно отключен. Эта функция снижает нагрузку на сервер.

CSS3 используется для организации стиля и компоновки веб-страниц. Это последний стандарт в серии CSS [9]. CSS3 предлагает множество новых тенденций, которые могут повлиять на ваши веб-проекты.

Преимущества CSS3:

- Лучшие результаты поисковой оптимизации. С CSS3 можно сохранить код HTML намного чище, что поможет искателю в получении реального контента с сайта.
- Доступность и удобство использования. CSS3 позволяет использовать более совместимые элементы стиля, включая размер шрифта и высоту линии, через которые веб-страницы могут быть легко использованы людьми с ограниченными возможностями. Можно создавать специальные файлы CSS для печати, мобильных устройств и обычных мониторов, и при этом превращать веб-сайты в полностью мультимедийные приложения.
- Изоляция и дифференциация. CSS3 позволяет вносить изменения и модификации в отдельные модули. Формат CSS3 помогает пользователям отделять стилевое оформление от структуры.

HTML5 и CSS3 представляют широкий спектр возможностей в сфере веб-разработки. Они разрабатывают и расширяют контент и веб-приложения. С помощью этих инструментов можно улучшить веб-интерфейс пользователя.

1.5 CMS Drupal

В последние годы программное обеспечение с открытым исходным кодом Drupal было лучшим выбором для некоторых из ведущих брендов в мире, чтобы получить свою систему управления контентом (CMS) [8]. Написанная на PHP, эта популярная структура не только помогает в проектировании и разработке качественных сайтов, но и экономит деньги на этом процессе. Он насчитывает более 29 000 модулей и более 2000 тем разных сортов, которые

помогают разработчикам в создании пользовательских веб - сайтов. Это хорошо зарекомендовавшая себя CMS, которая широко используется предприятиями разных доменов и отраслей для реализации своих целей в Интернете.

Кроме того, Drupal — это часть программного обеспечения, которая предоставляет множество возможностей для бизнеса. Он может использоваться для разработки различных решений, в том числе от личных веб - сайтов до веб - порталов сообщества, от сайтов электронной коммерции до сайтов социальных сетей. Поскольку у него тысячи проектов и дополнительных модулей, его используют разработчики и дизайнеры для различных целей. Это помогает дизайнерам создавать более тысячи пользовательских шаблонов. Это составляет основу более 2% всех веб - сайтов в мире [10]. Совершенно очевидно, что эта CMS поставляется с большим количеством потенциальных возможностей.

Таким образом, можно использовать возможности Drupal для повышения эффективности разработки. Он должен использоваться для целей развития не только из - за его преимуществ, но и из - за его бесплатных модулей и тем. Он очень настраиваемый, и разработчики могут внести изменения в его дизайн, компоненты, работу и функции, чтобы удовлетворить потребности предприятий с легкостью. Кроме того, его основные функции и функциональные возможности могут быть быстро развернуты, и даже настройки после развертывания Drupal возможны. Это означает, что настройка возможна даже после того, как сайт начинает свою работу.

Кроме того, Drupal — это CMS корпоративного класса, который достаточно хорошо развивается благодаря масштабируемому, надежному и гибким тенденциям. Он также защищен и обеспечивает совместимость, а его способность интегрироваться с другими приложениями делает его одним из лучших продуктов подобного рода. Кроме того, он имеет обширную поддержку API, которая дает разработчикам право создавать собственные модули с легкостью. Ещё одним хорошим плюсом является SEO - дружественный подход, так как дружественные поисковые системы веб - сайта создаются с легкостью, и их перспективы повышаются. Удобство для оптимизации достигается благодаря способности отображать контент в соответствии с нормами, установленными поисковыми системами.

Drupal является одним из тех немногих программ CMS, которые мобильны, помогая бизнесу реализовать свои маркетинговые цели без значительных инвестиций. Таким образом, веб - сайты, созданные с использованием этой платформы, могут легко просматриваться пользователями смартфонов так же, как и пользователями настольных компьютеров, и именно так создается больше потенциальных клиентов. Можно проследить тенденцию, как становится доступным высококачественный и многофункциональный веб - сайт, когда эта популярная CMS используется.

В работе используется версия Drupal 8. Рассмотрим подробнее особенности данной версии.

Для Drupal 8 были переписаны все функции с нуля.

Ключевыми новыми особенностями являются [11]:

- Строительный инструмент. Предыдущие версии функций включали обширные обходные пути, чтобы компенсировать отсутствие поддержки экспорта конфигурации в ядре Drupal. Теперь, благодаря API - интерфейсу конфигурации, встроенному в Drupal 8, эта проблема решается.
- Автоматическая упаковка. Вместо того, чтобы требовать ручного создания отдельных функций, функции анализируют сайт и автоматически упаковывают конфигурацию сайта в набор функций.
- Конфигурация. Практически все функции, которые могут быть выполнены, могут быть настроены под конкретные требования с помощью плагинов.
- Взаимодействие. Возможность создания функций по умолчанию.
- Поддержка наборов функций (пакетов). Функции поддерживают пакеты — наборы функций, разделяющих пространство имен.

Рассмотрим преимущества и недостатки Drupal.

Drupal — это комплексная платформа CMS и электронной коммерции. Это также чрезвычайно гибкая платформа, и практически каждый аспект сайта, каталога и представления продуктов компании или информации полностью настраивается. Использование Drupal начинается с менее определенной настройки, чем во многих CMS, и играет гораздо большую роль в выборе нужных параметров.

Хотя многие разработчики любят гибкость, которую позволяет Drupal, для настройки платформы требуются расширенные знания. Это означает, что это не обязательно удобно для пользователей, у которых нет опыта разработки сайтов.

1.6 База данных MySQL

Рассмотрим базы данных, которые можно использовать с PHP. Базы данных расположены в порядке уменьшения популярности для использования в веб - проектах.

- База данных MySQL. MySQL остается самой популярной базой данных для приложений PHP. Помимо этого это популярная БД с открытым исходным кодом в мире. Корпоративная версия базы данных также доступна и является платной. Многие популярные приложения, такие как SugarCRM, Magento, WordPress и Drupal, используют MySQL. Большие сайты, такие как Википедия, Facebook и Google, также являются его клиентами. MySQL Workbench и phpmyadmin предоставляют веб - интерфейс для управления базой данных. Благодаря своей уникальной архитектуре хранения данных производительность MySQL очень высока. MySQL поддерживает большое количество встроенных приложений, что делает её очень гибкой БД. Использование триггеров, хранимых процедур и представлений позволяет разработчику повысить производительность. MySQL позволяет откатывать транзакции, фиксировать и восстанавливать аварийные ситуации. К плюсам можно отнести триггеры и курсор.
- База данных PostgreSQL. PostgreSQL был запущен в 1995 году. По сравнению с MySQL данная БД богата функциональностью. Имеет расширенные функции оптимизации и сжатия, которые ставят его в один ряд с Oracle. Однако по сравнению с MySQL PostgreSQL считается более медленным. PostgreSQL — это open - source и бесплатная, но очень мощная система управления реляционными базами данных. PostgreSQL поддерживается преданным и опытным сообществом. PostgreSQL украшен множеством отличных сторонних инструментов с открытым исход-

ным кодом для проектирования, управления и использования системы управления. Можно программно расширять PostgreSQL с помощью хранимых процедур.

- База данных Oracle. Одна из самых распространенных баз данных нашего времени, одинаково популярна для серверов на базе Windows и Linux. Трудно встретить организации, которые не используют эту базу данных. Для подключения к приложениям PHP требуется расширение. У Oracle множество функций, которые делают его популярным. Новые версии баз данных Oracle выпускаются с новыми и улучшенными функциями, ориентированными на предприятия и корпорации. Они обеспечивают сочетание высокоуровневой технологии и интегрированных бизнес - решений. Oracle — это база данных, обеспечивающая отличную производительность при сложных задачах.

Существуют и другие базы данных, которые могут быть связаны с PHP. Выбор средства зависит от задач разработки конкретного проекта. Изучив различные параметры базы данных PHP, можно принять обоснованное решение о том, какую базу данных можно использовать в конкретном случае.

Когда речь идет о реляционных базах данных, у разработчиков есть выбор. На более высоком уровне находятся сложные отраслевые продукты, такие как Oracle, SQL Server. Они многофункциональны, нужны специальные администраторы баз данных для работы с ними. Как правило, требуется профессиональная подготовка, прежде чем начать работу. На нижнем уровне представлены такие легкие продукты, как SQLite или встраиваемые базы данных. Им даже не нужен отдельный процесс, и они не нуждаются в какой - либо конфигурации.

MySQL находится между ними. Она многофункциональна и не требует слишком много знаний для использования в одно и то же время. Для разработки ресурса выберем базу данных MySQL. Рассмотрим основные принципы.

MySQL является самым популярным из всех широкомасштабных серверов баз данных. Это многофункциональный продукт с открытым исходным кодом, который предоставляет множество веб - сайтов и приложений в Ин-

тернете [12]. Начало работы с MySQL относительно просто, и разработчики имеют доступ к массиву информации о базе данных в Интернете.

Следует отметить, что с учетом популярности продукта существует множество сторонних приложений, инструментов и интегрированных библиотек, которые очень сильно помогают во многих аспектах работы с этой СУБД.

Несмотря на то, что он не пытается реализовать полный стандарт SQL, MySQL предлагает пользователям множество функций.

MySQL можно установить очень легко. Сторонние инструменты, в том числе визуальные (графические интерфейсы), очень упрощают работу с базой данных. MySQL поддерживает множество функций SQL, ожидаемых от СУБД — прямо или косвенно. В MySQL встроено множество функций безопасности, некоторые из которых достаточно продвинутые. MySQL может обрабатывать множество данных. Отказ от некоторых стандартов позволяет MySQL работать очень эффективно, тем самым обеспечивая прирост скорости.

Определим, почему стоит использовать MySQL. Как и любой автономный сервер баз данных, MySQL обеспечивает большую свободу действий вместе с некоторыми расширенными функциями. Функции безопасности MySQL обеспечивают надежную защиту доступа данных простым способом. Подавляющее большинство веб - сайтов (и веб - приложений) могут просто работать на MySQL, несмотря на ограничения. Этот гибкий и несколько масштабируемый инструмент легок в использовании и удобен в управлении, что очень полезно в долгосрочной перспективе.

Хотя точное количество сайтов, использующих MySQL как базу данных, невозможно определить, но можно оценить, что 80% личного блога используют базу данных MySQL. Крупные компании, такие как Google или Amazon, не используют MySQL в качестве основного хранилища данных, поскольку MySQL не может справиться с бизнес - ситуацией в этом масштабе. Но для миллионов небольших или средних сайтов MySQL обычно является лучшим выбором.

2 Реализация веб - приложения

2.1 Изучение темы

Писательство на протяжении веков было очень эффективным и популярным средством для выражения мыслей. Но раньше издательство своих книг и поиск читателей были затруднены по ряду многих причин.

На самом деле, для написания небольшого рассказа не нужно иметь много навыков. Каждый человек в своей жизни пробует себя на поприще писателя. Не для того чтобы стать знаменитым, а из - за желания поделиться своими сокровенными мыслями. Но если бы стать писателем было так просто, то на слуху у людей было бы гораздо большее количество имен действительно хороших авторов.

Современный писатель также сталкивается с проблемой поиска своего читателя. Но это связано не только с тем, что в современном обществе читает гораздо меньшее количество людей, но и в целом со сложностью распространения своей работы и поиска издателя.

В настоящее время все большее распространение приобретают издательства, предлагающие платную публикацию книг. Но у некоторых начинающих писателей может не хватать средств для того, чтобы воспользоваться подобного рода предложениями, а в прочих издательствах они не раз уже получали отказ. Сложно не согласиться с тем, что писатель без читателей не получает должной отдачи от своих произведений. Да, такими читателями могут стать близкие люди, но они не всегда могут объективно оценить работу.

Для решения данной проблемы и необходима платформа, позволяющая выкладывать свою работу на обозрение широкому кругу читателей, получать рецензии как от профессиональных критиков, так и от обычных пользователей портала, и, как следствие, в дальнейшем получать возможность развивать свой писательский талант.

Портал «Writing for yourself» создан с целью помочь писателям на их творческом пути. Работа системы начинается с публикации первого произведения пользователя. После этого к работе приступают эксперты и рецензенты. При публикации работа автоматически отправляется нескольким экспертам/рецензентам. Она также становится доступна для прочтения обычными

пользователя. После получения советов и правок к работе от экспертов автор вносит исправления в свое произведение до получения им статуса «Готовая работа». Далее автору начисляются очки рейтинга, показывающие его текущий писательский талант.

2.2 Создание логотипа

Логотип — это лицо любого сайта. Это определяет первое впечатление, поэтому его дизайн чрезвычайно важен. При правильном исполнении логотип является мощным активом. Рассмотрим основные правила для разработки эффективного логотипа, которыми воспользуемся на практике.

Предварительные эскизы являются важным первым шагом в разработке эффективного логотипа.

Было разработано множество вариантов логотипов. Первый вариант логотипа представлен в соответствии с рисунком 2.1.



Рисунок 2.1 — Первый вариант логотипа

Предпоследний вариант логотипа представлен в соответствии с рисунком 2.2.



Рисунок 2.2 — Предпоследний вариант логотипа

Логотип должен быть разборчивым на экранах любого размера. Логотип может оказаться не эффективен, если он будет неразборчивым на мобильных платформах. Логотип также должен хорошо выглядеть при использовании в больших форматах.

Также важно подобрать правильное цветовое решение для логотипа. Цвета вызывают чувства и настроения. Например, оранжевый, используемый в финальной версии логотипа, является комбинацией желтого и красного, поэтому часто используется с целью привлечения внимания. Хотя, в отличие от своих производных, оранжевая гамма применяется в логотипах для создания игровой эмоции, стимулирования креатива, энтузиазма, веселья.

Важным фактором успешности создаваемого логотипа — выбор правильного типа шрифта и размера [13]. Стоит рассматривать шрифты с засечками и без, а также курсив, полужирный шрифт и пользовательские шрифты. При выборе логотипа учитываем три основных момента. Не используем наиболее часто используемые шрифты, такие как Comic Sans и Arial, отдаём предпочтение последним тенденциям. Убеждаемся, что шрифт разборчив при различных размерах. Отдаём предпочтение одному шрифту.

Финальная версия веб - приложения «Writing for Youself» выглядит следующим образом в соответствии с рисунком 2.3.



Рисунок 2.3 — Финальный вариант логотипа

2.3 Разработка макета и дизайна приложения

Разработка собственных цифровых продуктов является довольно сложным процессом, требующим объединения усилий самых разных специалистов. Однако на ранних стадиях проекта всегда есть очень важный этап — создание прототипа будущего сайта или приложения. Сделать это можно разными способами. Воспользуемся сервисом Balsamiq Mockups для решения этой задачи. В качестве примера приведём создание макета главной страницы сайта в соответствии с рисунком 2.4.

Цвета, графика и исходный текст — это все элементы, которые работают вместе, чтобы сделать сайт интересным для потенциальных клиентов. К сожалению, невозможно просто скомпоновать веб - сайт с двумя любимыми цветами, бросить графику где - нибудь на странице и назвать её идеально подходящей. Внешний вид сайта может превратить портал как в успешный, так и в провальный проект. Отметим основные моменты, на которые стоит обращать внимание.

Когда посетители приходят на сайт, одна из характеристик, которые больше всего ценятся и бросаются в глаза, — это читаемость. Если в статьях используются технические термины, требуется их объяснение по первой ссылке, чтобы читатель мог понять [14].

Объявления становятся все более раздражающими для пользователей Интернета. Потенциальные клиенты не хотят тратить время на сайты, которые тяжеловесны из - за рекламы, и для большинства компаний лучше не иметь их вообще.

Посетители, скорее всего, с большей вероятностью останутся на странице портала, если им не придется просматривать рекламу, чтобы увидеть контент, который они ищут.

Любая страница, которая занимает больше трех секунд для загрузки, теряет почти половину своего трафика. Это означает, что если сайт работает медленно, можно потерять половину своих потенциальных клиентов и отправить их обратно в результаты поиска, где они выберут одного из конкурентов.

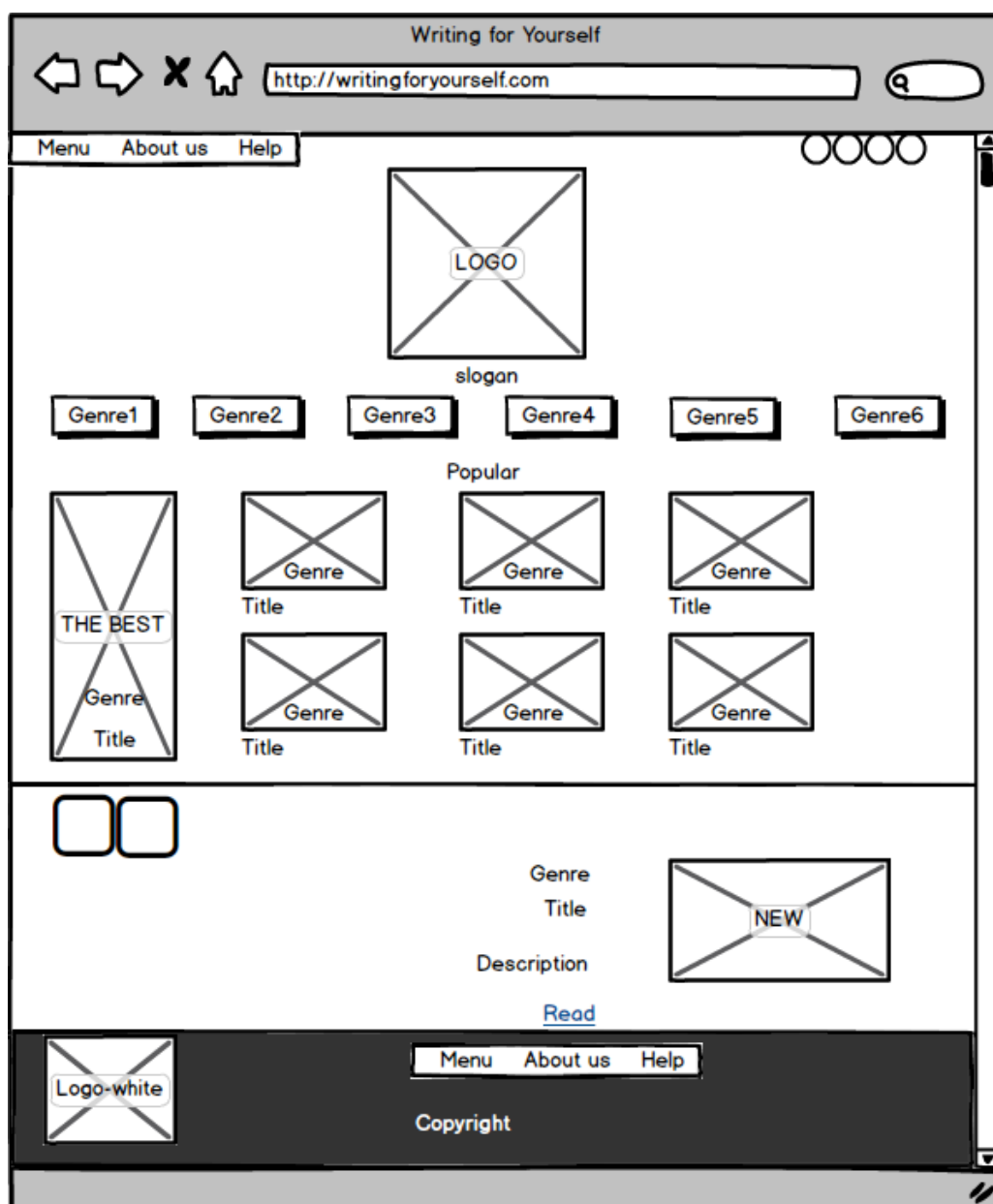


Рисунок 2.4 — Mockup

На мобильных устройствах временные рамки немного более прощающие, а страницы обычно оставляют 74% людей через 5 секунд. Тем не менее, время загрузки всё равно остаётся одним из важных параметров.

Хотя чрезвычайно важно иметь призывы к действию на сайте, слишком большое их количество может быть подавляющим. Если посетители чувствуют, что их загружают, запрашивая их информацию слишком часто, они могут быть смущены тем, что выбрать, или раздражены проявлением излишней настойчивости со стороны сайта.

Ещё одним признаком дурного тона является наличие автоматически проигрывающегося видео - элемента на странице. Многие посетители будут крайне раздражены, если увидят такое дизайнерское решение, и тут же остановят воспроизведение. Пусть автовоспроизведение будет отключено, и посетитель сам решит, смотреть или не смотреть.

Когда для большинства действий на сайте нужна персональная информация посетителей, часть из них будет потеряна. Чаще всего посетители, впервые зашедшие на сайт, хотят найти конкретную информацию и уйти. Позже можно запросить информацию, такую как имя и адрес электронной почты, но в начале процесса главная задача — исключить любые элементы, которые могут стать препятствием между пользователем и контентом.

Если сайт имеет большое количество цветов и графики, он может казаться запутанным для посетителей. Во многих случаях они предпочтут сайт, который будет более простым и в котором легче будет ориентироваться [15].

Это может быть несколько субъективное качество, но всегда полезно, чтобы люди за пределами компании проверяли сайт. Если они находят, что он запутывает или на нем трудно найти информацию, возможно, пришло время переосмыслить дизайн.

Нет ничего более неприятного, чем посещение веб - сайта, на котором невозможно найти то, что пользователь планировал найти. Одним из самых важных качеств является легкая навигация по всему контенту сайта.

Приступим к созданию дизайна портала.

Простота дизайна станет ценным активом данной разработки. Такой подход выбран, так как простота делает веб - сайты более сглаженными, уменьшает путаницу в навигации и помогает достичь желаемых целей и результатов наиболее эффективным способом. В данном случае повысить посещаемость и популярность ресурса при дальнейшем его развитии.

Избавляемся от всех элементов дизайна, которые перегружают главную страницу и оставляем только главные функциональные единицы. Это включает в себя избавление от длинных строк текста, уродливых или ненужных кнопок, форм со слишком большим количеством полей, стрелок и текстовых полей, которые занимают пустое пространство, но ничего не добавляют к функциональности веб - сайта. Воспользуемся принципом Парето, считая,

что только 20% элементов дизайна повлияют на желаемый результат и достижение цели портала [16]. Разделив дизайн на самые простые элементы, достигаем наибольшего влияния, которое оно может оказать.

После этого приступим к созданию навигации для сайта.

Веб - сайт с множеством аккуратных навигационных функций станет основой используемой идеи. Незнакомая навигационная схема может стать препятствием для пользователя, заставив пользователя «переучиваться». Это может создать негативное впечатление.

Для достижения желаемого эффекта воспользуемся следующими принципами. Поместим самую важную информацию на информационных панелях. Сократим резервные и ненужные навигационные ссылки (например, те элементы меню, которые обычно несколько раз дублируются на верхних и нижних уровнях навигационной системы). Избавимся от избыточных элементов внутренних страниц, перенося нагрузку на графические элементы интерфейса. Помещаем кнопки навигации на легко доступные места. Отдельно разрабатываем интерфейс для просмотра работ и редактирования. Используем универсальные обозначения и символы для кнопок навигации.

Необходимо также решить проблемы с читаемостью контента, так как сайт в первую очередь основан на главной функции — возможности ознакомиться с работами начинающих писателей и оценить их.

Воспользуемся следующими, определёнными для системы принципами. Выбор отдаётся в пользу читаемого и удобного для пользователей шрифта. Выбираем самые легкие для глаз и вызывающие наименьшую усталость. Создаём инструменты для выделения важного текста с использованием специальных параметров форматирования.

Цветовая палитра сайта представлена в соответствии с рисунком 2.5.

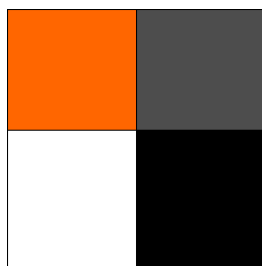


Рисунок 2.5 — Colors

Дизайн верхней части главной страницы представлен в соответствии с рисунком 2.6.



Рисунок 2.6 — Header, Main Page

Дизайн блока самых популярных произведений на главной странице представлен в соответствии с рисунком 2.7.



Рисунок 2.7 — Most Popular Book, Main Page

Дизайн блока новых произведений на главной странице представлен в соответствии с рисунком 2.8.



Рисунок 2.8 — New Book, Main Page

Дизайн нижней части главной страницы представлен в соответствии с рисунком 2.9.

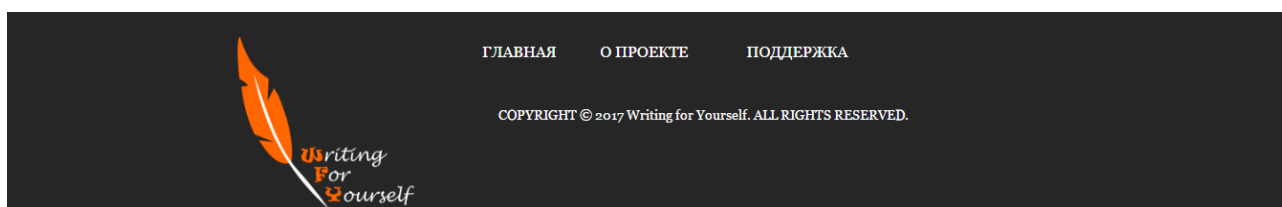


Рисунок 2.9 — Footer, Main Page

2.4 Верстка адаптивного интерфейса приложения

Интерфейсы — одни из важнейших аспектов современного дизайна. Пользователи зависят от интуитивно понятных, гибких интерфейсов для доступа к контенту и услугам в Интернете, а в последние годы тенденции в этой области сильно изменились [17].

Компьютеры больше реагируют на ввод оптических мышей и клавиатур, а также голосовые команды, сенсорные и различные другие материальные средства. Дисплеи также существенно изменились. Дисплеи с низким разрешением уступили место более гладким дисплеям высокой четкости, которые можно просматривать на многих платформах и устройствах. Концепция адаптивных интерфейсов проистекает из непрерывного изменения в мире проектирования. По мере изменения устройств, возможностей и тенденций также изменились устройства, которые мы используем для доступа к Интернету.

Вместо того, чтобы создавать веб - страницы и контент для реагирования на различные устройства, веб - разработчики пытаются создать интерфейсы, которые отвечают отдельным пользователям. Эти интерфейсы будут адаптироваться «на лету» к текущему пользователю и собирать данные с течением времени, чтобы предвидеть действия и предпочтения каждого пользователя.

На данный момент разработчики вынуждены пытаться создавать веб - страницы, которые работают на разных устройствах с различными возможностями и размерами экрана. Разработчикам приходится бороться с очень быстро развивающимся аппаратным рынком, где потребители ежегодно переходят на новые устройства.

Адаптивный интерфейс — это результат объединения концепций веб - дизайна с несколькими устройствами для обеспечения потрясающего пользовательского опыта на любой платформе [18].

Основой концепции адаптивного интерфейса является отзывчивый дизайн. Веб - разработчики быстро узнали, что им нужно пересмотреть свое мышление с точки зрения дизайна сайта из - за взрывной популярности мобильных устройств. В то время как мышь и клавиатура делают просмотр веб - сайта на настольном компьютере легким для любого человека, навигацию по сайту с помощью пальцев и прикосновений представляет новый вызов. Это помогает гарантировать, что пользователи разных устройств могут найти портал, получить доступ к контенту и взаимодействовать с удобством на разных устройствах. Хороший отзывчивый дизайн означает, что пользователи сайта могут ожидать неизменно высококачественных взаимодействий на любом устройстве.

Хотя веб - сайты с адаптивным дизайном настраиваются на основе устройства, на котором они просматриваются, и генерируют индивидуальные макеты для страниц на основе устройства, обращающегося к страницам, адаптивные интерфейсы направлены на создание полностью индивидуализированных интерфейсов для каждого уникального пользователя.

Полный код вёрстки главной страницы разрабатываемого портала приводится в приложении А, на основании которого в последствии был построен шаблон для Drupal. В Drupal 8 используется новый шаблонизатор Twig. Он разделяет логику сайта на PHP и оформление сайта на HTML. Он заставляет

верстальщика не использовать PHP-код в шаблоне, а это значит нельзя написать запрос к базе данных, нельзя подгрузить пользователя, ноду или термин через PHP код. Нужно передавать уже готовые переменные в шаблон и выводить их через Twig. Собственный шаблон, создаётся на основе стандартного шаблона Drupal 8 по умолчанию.

Интерфейс одной из внутренних страниц портала выглядит в соответствии с рисунком 2.10.



Рисунок 2.10 — Внутренняя страница портала

Помимо основной версии сайта, разрабатывается также и мобильная версия в соответствии с рисунком 2.11.

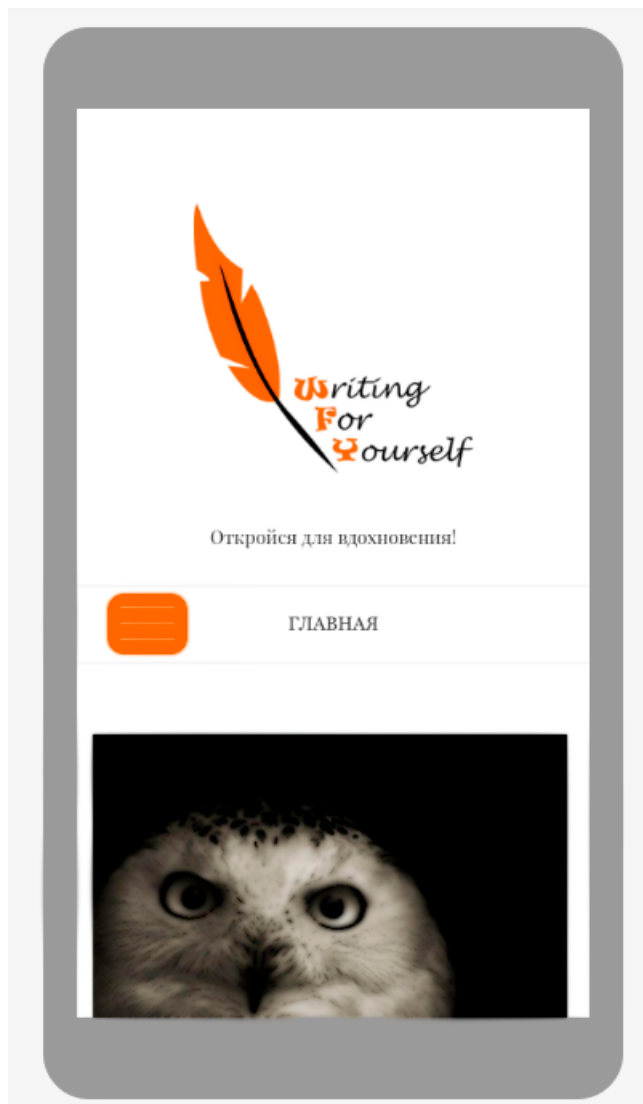


Рисунок 2.11 — Mobile version

Была проведена проверка оптимизации портала для мобильных устройств с помощью сервиса search. Получен положительный результат проверки.

2.5 Модули системы

Модули являются важными компонентами Drupal, которые добавляют расширенную функциональность на сайт [19, 20]. В разрабатываемой системе используются многие модули, разработанные для Drupal 8. Рассмотрим некоторые из них.

Admin Toolbar. Этот модуль заменяет исходное меню администратора отзывчивой версией, которая значительно улучшает исходное меню, добавляя раскрывающиеся меню. Это ускоряет доступ ко всем страницам администрирования. Модуль был доработан, чтобы выводить дополнительные ссылки для очистки кэша.

Fieldgroup. Модуль группы позволяет организовать контент в виде вертикальных или горизонтальных объединений.

Linkit. Позволяет автозаполнять все внутренние и внешние ссылки.

Pathauto. Этот модуль автоматически генерирует релевантные и легко запоминаемые URL-адреса в соответствии с веб - страницей.

Blog. Создание блога. Используются настройки полей.

IMCE. Удобство работы с картинками. При вставке изображений в текстовую область она попадает в специальную папку. Используя модуль, можно создавать подпапки в папке картинок и грузить файлы в нужные папки, а затем удобно использовать повторно.

Meta tags quick. Для добавления к полям ввода ключевых слов и описания или создания их автоматически из содержимого материалов. Были определены базовые ключевые слова для каждого жанра, которые будут добавляться к материалам.

ImageMagick. Набор эффектов и возможности управления различными параметрами изображений.

Avatar Kit. Используется для предоставления уникальных аватаров для пользователей, когда они не загружают их сами.

Back To Top. Добавляет кнопку, которая находится в нижней части экрана и позволяет пользователям плавно прокручивать страницу с помощью jQuery. Только на внутренних страницах сайта.

Calendar. В этом модуле создаются поля дата в форматах календаря.

Disqus. Платформа с широким функционалом для создания комментариев.

В системе используются несколько готовых модулей Drupal 8, которые необходимы для настройки и управления различными аспектами веб - сайта. Drupal 8 все еще находится в процессе разработки с новыми модулями, которые реализуются на регулярной основе, а старые модули Drupal 7 переносятся

на Drupal 8. Но часть модулей нуждается в доработке и настройке для конкретных потребностей разработки. Возможности редактирования и создания своих модулей на основе существующих и были использованы в проекте.

ЗАКЛЮЧЕНИЕ

Большое количество инвестиций осуществляется компаниями по всему миру в области веб - разработки. Они инвестируют в творческие идеи, которые помогают уникально расширить свой бизнес. Недавние исследования показали, что количество транзакций, которые компании получают из онлайн-источников, намного больше, чем те, которые они делают физически.

Разработка веб - сайта в основном охватывает работу, связанную со всеми этапами создания сайта. Компоненты, участвующие в этом процессе, могут варьироваться от разработки единой статической страницы для рекламы контактной информации или места компании и могут распространяться на самые сложные интернет - приложения и сетевые сервисы в Интернете.

В ходе работы были рассмотрены и разобраны основы веб - разработки. Для понимания всех технологий, используемых для реализации проекта, были изучены следующие средства разработки: PHP, HTML, CSS, MySQL. Также были рассмотрены особенности создания приложения на основе CMS Drupal. В итоге были выявлены принципы и возможности использования данных средств.

В процессе изучения была приведена реализация веб - приложения для начинающих писателей. Для этого было проведено тщательное изучение всех этапов разработки. После этого были приведены практические примеры, как используются те или иные принципы и средства при создании приложения.

Созданная система может использоваться по назначению, а также быть модернизирована в дальнейшем до новой версии с целью улучшения функционала.

В итоге, в ходе рассмотрения были изучены все аспекты темы и выполнены все поставленные задачи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Newark, Quentin. What is Graphic Design? — RotoVision, 2002. — P. 256. — ISBN: 9782880465391.
2. Ryan, Hembree. The Complete Graphic Designer: A Guide to Understanding Graphics and Visual. — Beverly, 2008. — P. 192. — ISBN: 9781592535026.
3. Harvey, Wilson. 1,000 Graphic Elements: Special Details for Distinctive Designs. — Rockport Publishers, 2004. — P. 320. — ISBN: 9781592530779.
4. McNeil, Patrick. The Web Designer's Idea Book, Volume 3. — HOW Books, 2013. — P. 288. — ISBN: 9781440323966.
5. Frain, Ben. Responsive Web Design with HTML5 and CSS3 - Second Edition. — Packt Publishing, 2015. — P. 350. — ISBN: 9781784398934.
6. Hanington, Bruce. Universal Methods of Design. — Rockport Publishers, 2012. — P. 208. — ISBN: 9781592537563.
7. Craig, James. Designing with Type, 5th Edition: The Essential Guide to Typography. — Watson-Guptill, 2006. — P. 176. — ISBN: 9780823014132.
8. Felici, James. The Complete Manual of Typography: A Guide to Setting Perfect Type. — Peachpit Press, 2002. — P. 384. — ISBN: 9780321127303.
9. Nixon, Robin. Learning PHP, MySQL JavaScript: With jQuery, CSS HTML5. — O'Reilly Media, 2014. — P. 812. — ISBN: 9781491918661.
10. Ullman, Larry. PHP for the Web: Visual QuickStart Guide (5th Edition). — Peachpit Press, 2016. — P. 528. — ISBN: 9780134291253.
11. Burge, Stephen. Drupal 8 Explained: Your Step-by-Step Guide to Drupal 8. — Independently published, 2017. — P. 436. — ISBN: 9781520990262.
12. Tomlinson, Todd. Enterprise Drupal 8 Development: For Advanced Projects and Large Development Teams. — Apress, 2017. — P. 309. — ISBN: 9781484202548.

13. Barker, Deane. Web Content Management: Systems, Features, and Best Practices. — O'Reilly Media, 2016. — P. 378. — ISBN: 9781491908129.
14. PHP: Documentation. — URL: <http://php.net/docs.php> (online; accessed: 21.04.2017).
15. Справочник по HTML. — URL: <http://htmlbook.ru/html> (дата обращения: 13.12.2016).
16. Справочник по CSS. — URL: <http://htmlbook.ru/css> (дата обращения: 16.12.2016).
17. Documentation Drupal. — URL: <https://www.drupal.org/documentation> (online; accessed: 19.05.2017).
18. Drupal: Modules. — URL: https://www.drupal.org/project/project_module (online; accessed: 15.02.2017).
19. MySQL: Documentation. — URL: <https://dev.mysql.com/doc/> (online; accessed: 14.04.2017).
20. Hostinger. — URL: <https://www.hostinger.ru> (online; accessed: 18.10.2016).

ПРИЛОЖЕНИЕ А

Вёрстка главной страницы

```
1 index.html
2 ++++++
3
4 <!doctype html>
5 <html>
6 <head>
7   <meta charset="utf-8">
8   <title>Main Page</title>
9   <link rel="stylesheet" href="css/style.css">
10 </head>
11 <body>
12   <div id="background">
13     <div id="head_menu">
14       <p id="head_menu_main" class="text">ГЛАВНАЯ</p>
15       <p id="about_project" class="text">О ПРОЕКТЕ</p>
16       <p id="help" class="text">ПОДДЕРЖКА</p>
17       
18       
19       
20       
21     </div>
22     <div id="logo">
23       
24       <p class="text">Создай собственный мир!</p>
25     </div>
26     <div id="menu">
27       <div id="menu1">
28         <p class="text text-5">Фэнтези</p>
29       </div>
30       <div id="menu2">
31         <p class="text">Детектив</p>
32       </div>
33       <div id="menu3">
34         <p class="text">Фантастика</p>
35       </div>
36       <div id="menu4">
37         <p class="text">Антиутопия</p>
38       </div>
39       <div id="menu5">
40         <p class="text">Романтика</p>
41       </div>
42       <div id="menu6">
43         <p class="text">Ужасы</p>
44       </div>
45     </div>
46     <p class="text">Популярное</p>
47     <div id="content">
```



```

48 <div id="block_most">
49   
50   <div id="block_head" class="element">
51     <p id="most_name" class="text">Фантастика</p>
52   </div>
53   <p id="most_title" class="text">На клыльях ночи</p>
54   <div id="most_popular_six">
55     <div id="block_most1">
56       
57       <div class="element"></div>
58       <p class="text">Романтика</p>
59       <p class="text">Сказание о Рождестве</p>
60       <button class="_button">---</button>
61     </div>
62     <div id="block_most2">
63       
64       <div class="element"></div>
65       <p class="text">Фэнтези</p>
66       <p class="text">Лесной житель</p>
67       <button class="_button">---</button>
68     </div>
69     <div id="block_most3">
70       
71       <div class="element"></div>
72       <p class="text">Мистика</p>
73       <p class="text">Укрытие волка</p>
74       <button class="_button">---</button>
75     </div>
76     <div id="block_most4">
77       
78       <div class="element"></div>
79       <p class="text">Фэнтези</p>
80       <p class="text">Пещера за водопадом</p>
81       <button class="_button">---</button>
82     </div>
83     <div id="block_most5">
84       
85       <div class="element"></div>
86       <p class="text">Фантастика</p>
87       <p class="text">Крыло дракона</p>
88       <button class="_button">---</button>
89     </div>
90     <div id="block_most6">
91       
92       <div class="element"></div>
93       <p class="text">Детектив</p>
94       <p class="text">Корона и меч</p>
95       <button class="_button">---</button>
96     </div>
97   </div>

```

```

98     </div>
99 </div>
100 <div id="new_slide">
101     <div>
102         <button id="left_button_slide" class="_button">&lt;</button>
103         <button id="right_button_slide" class="_button">&gt;</button>
104     </div>
105     <div>
106         <div>
107             <div id="new_name">
108                 <p class="text">Приключения</p>
109             </div>
110             <p class="text">Знак смерти, или как одолеть судьбу</p>
111             <p class="text">Автор: admin 23/01/2017 - 15:07</p>
112             <div class="element"></div>
113             <p class="text">Для селения Айсрок наступают темные времена.
114 Страшная болезнь окутала улицы и дома. Но это ещё не самое страшное:
115 соседнее поселение хочет воспользоваться моментом
116 и захватить новые территории. Среди этого мрака и множества смертей
117 только у наследницы вождя есть шанс что-то изменить...</p>
118             <p class="text">Читать дальше</p>
119         </div>
120         <p id="new" class="text">new!</p>
121         
122     </div>
123 </div>
124 <div id="footer">
125     
126     <div id="footer_menu_container">
127         <p id="main_footer" class="text">ГЛАВНАЯ</p>
128         <p id="about_project_footer" class="text">О ПРОЕКТЕ</p>
129         <p id="help_footer" class="text">ПОДДЕРЖКА</p>
130     </div>
131     <p class="text">COPYRIGHT © 2017 Writing for Yourself.
132         ALL RIGHTS RESERVED.</p>
133 </div>
134 </div>
135 </body>
136 </html>
137
138 style.css
139 ++++++
140
141 body {
142     background-color: rgb(255, 255, 255);
143     font: 700 16px/1.38 Georgia;
144     color: rgb(0, 0, 0);
145 }
146
147 .element {

```

```

148     float: left;
149     clear: both;
150 }
151
152 .element .text {
153     width: 80%;
154     min-height: 13px;
155     margin-left: 3%;
156     text-align: center;
157     color: rgb(0, 0, 0);
158 }
159
160 .image {
161     display: block;
162     height: auto;
163     overflow: hidden;
164 }
165
166 .text {
167     float: left;
168 }
169
170 ._button {
171     display: block;
172     float: left;
173     background-color: rgb(255, 102, 0);
174     text-align: center;
175 }
176
177 #block_head {
178     position: relative;
179     z-index: 64;
180     width: 50%;
181     height: 24px;
182     margin: -119px 0 0 6%;
183     background-color: rgb(255, 255, 255);
184 }
185
186 #image_most {
187     position: relative;
188     float: left;
189     z-index: 62;
190     width: 226px;
191     margin-left: -1%;
192 }
193
194 #head_menu_main {
195     position: relative;
196     z-index: 0;
197     width: 10%;

```

```

198     min-height: 29px;
199     margin: 9px 0 0 10%;
200     font-weight: 400;
201     text-align: center;
202     color: rgb(0, 0, 0);
203 }
204
205 #most_title {
206     position: relative;
207     z-index: 65;
208     margin: -75px 0 0 5%;
209     color: rgb(255, 255, 255);
210 }
211
212 #about_project {
213     position: relative;
214     clear: both;
215     z-index: 1;
216     width: 10%;
217     min-height: 29px;
218     margin: -29px 0 0 20%;
219     font-weight: 400;
220     text-align: center;
221     color: rgb(0, 0, 0);
222 }
223
224 #help {
225     width: 15%;
226     min-height: 29px;
227     margin: -29px 0 0 30%;
228     font-weight: 400;
229     text-align: center;
230     color: rgb(0, 0, 0);
231 }
232
233 #main_footer {
234     position: relative;
235     z-index: 83;
236     width: 19%;
237     min-height: 29px;
238     font-weight: 400;
239     text-align: center;
240     color: rgb(255, 255, 255);
241 }
242
243 #about_project_footer {
244     position: relative;
245     clear: both;
246     z-index: 84;
247     width: 18%;

```

```

248     min-height: 29px;
249     margin: -29px 0 0 18%;
250     font-weight: 400;
251     text-align: center;
252     color: rgb(255, 255, 255);
253 }
254
255 #help_footer {
256     width: 25%;
257     min-height: 29px;
258     margin: -29px 0 0 40%;
259     font-weight: 400;
260     text-align: center;
261     color: rgb(255, 255, 255);
262 }
263
264 #left_button_slide {
265     width: 7%;
266     height: 52px;
267     border-radius: 5px;
268     color: rgb(255, 255, 255);
269 }
270
271 #right_button_slide {
272     width: 7%;
273     height: 52px;
274     margin-left: 1%;
275     border-radius: 5px;
276     color: rgb(255, 255, 255);
277 }
278
279 #background {
280     float: left;
281     width: 100%;
282     height: 2000px;
283     background-color: rgba(255, 255, 255, 0.97);
284 }
285
286 #menu6,
287 #menu2,
288 #menu3 {
289     float: left;
290     width: 12%;
291     height: 50px;
292     margin: -2px 0 0 2%;
293     background-color: rgb(255, 102, 0);
294 }
295
296 #content {
297     position: relative;

```

```

298     float: left;
299     clear: both;
300     z-index: 67;
301     width: 100%;
302     height: 500px;
303 }
304
305 #block_most {
306     float: left;
307     width: 20%;
308     height: 500px;
309     margin-left: 10%;
310     background-color: rgba(222, 222, 222, 0);
311 }
312
313 #most_popular_six {
314     float: left;
315     clear: both;
316     width: 320%;
317     height: 500px;
318     margin: -374px 0 0 110%;
319     background-color: rgba(222, 222, 222, 0);
320 }
321
322 #block_most1 {
323     position: relative;
324     float: left;
325     z-index: 30;
326     width: 30%;
327     height: 186px;
328     margin-top: -1px;
329     background-color: rgba(222, 222, 222, 0);
330 }
331
332 #block_most2 {
333     position: relative;
334     float: left;
335     z-index: 36;
336     width: 30%;
337     height: 184px;
338     margin: -1px 0 0 4%;
339     background-color: rgba(222, 222, 222, 0);
340 }
341
342 #block_most3 {
343     position: relative;
344     float: left;
345     z-index: 42;
346     width: 30%;
347     height: 190px;

```

```

348     margin-left: 7%;
349     background-color: rgba(222, 222, 222, 0);
350 }
351
352 #block_most4 {
353     position: relative;
354     float: left;
355     clear: both;
356     z-index: 48;
357     width: 30%;
358     height: 179px;
359     margin-top: 7px;
360     background-color: rgba(222, 222, 222, 0);
361 }
362
363 #block_most5 {
364     position: relative;
365     float: left;
366     z-index: 54;
367     width: 30%;
368     height: 198px;
369     margin: 6px 0 0 7%;
370     background-color: rgba(222, 222, 222, 0);
371 }
372
373 #head_menu {
374     position: relative;
375     float: left;
376     width: 100%;
377     height: 42px;
378     background-color: rgba(218, 218, 218, 0.52);
379 }
380
381 #block_most6 {
382     position: relative;
383     float: left;
384     z-index: 60;
385     width: 30%;
386     height: 181px;
387     margin: 7px 0 0 8%;
388     background-color: rgba(222, 222, 222, 0);
389 }
390
391 #new_slide {
392     position: relative;
393     float: left;
394     clear: both;
395     z-index: 82;
396     width: 100%;
397     height: 410px;

```

```

398     margin-top: -41px;
399     background-color: rgba(222, 222, 222, 0.75);
400 }
401
402 #footer {
403     float: left;
404     clear: both;
405     width: 100%;
406     height: 244px;
407     background-color: rgb(38, 38, 38);
408 }
409
410 #footer_menu_container {
411     position: relative;
412     float: left;
413     z-index: 86;
414     width: 53%;
415     height: 29px;
416     margin: 32px 0 0 3%;
417     background-color: rgba(222, 222, 222, 0);
418 }
419
420 #logo {
421     float: left;
422     clear: both;
423     width: 100%;
424     height: 323px;
425 }
426
427 #menu {
428     float: left;
429     clear: both;
430     width: 100%;
431     height: 50px;
432     margin-top: 2px;
433     border: 2px solid rgb(0, 0, 0);
434     background-color: rgb(0, 0, 0);
435 }
436
437 #menu1 {
438     float: left;
439     width: 12%;
440     height: 50px;
441     margin: -2px 0 0 10%;
442     background-color: rgb(255, 102, 0);
443 }
444
445 #menu4 {
446     float: left;
447     width: 12%;

```



```
448     height: 50px;
449     margin: -2px 0 0 2%;
450     background-color: rgb(255, 102, 0);
451 }
452
453 #menu5 {
454     float: left;
455     width: 12%;
456     height: 50px;
457     margin: -2px 0 0 2%;
458     background-color: rgb(255, 102, 0);
459 }
```