

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра Компьютерных технологий
«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой Куралбаев З.К. д.ф.н., проф.
(ученая степень, звание, Ф.И.О.)

« » 2014 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Создание интернет-портала на основе
Технологии Yii РНР Framework

Специальность 5В070400 - Вычислительная техника и программное обеспечение

Выполнил Кеммбергдиев Р.А. Группа ВТ-10-3
(Ф.И.О.)

Научный руководитель Ержан А.А. д.ф.н. (к.т.н.)
(ученая степень, звание, Ф.И.О.)

Консультанты:

по экономической части: Еркешева З.Д. ст. пр.
(ученая степень, звание, Ф.И.О.)

Еркешева «28» 04 2014 г.
(подпись)

по безопасности

жизнедеятельности: Приходько И.Г. д.х.н., проф.
(ученая степень, звание, Ф.И.О.)

И.Г. Приходько «28» 04 2014 г.
(подпись)

по применению

вычислительной техники: Ержан А.А. ст. пр.
(ученая степень, звание, Ф.И.О.)

Ержан «28» 05 2014 г.
(подпись)

Нормоконтролер: Тусупов Д.М.
(ученая степень, звание, Ф.И.О.)

Д.М. Тусупов «30» май 2014 г.
(подпись)

Рецензент: Архабаев А.Н. АО GSM инженер
(ученая степень, звание, Ф.И.О.)

« » 2014 г.
(подпись)

Алматы 2014

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Факультет Информационных технологий

Кафедра компьютерных технологий

Специальность 5В070400 Вычислительная техника
и программное обеспечение

ЗАДАНИЕ

на выполнение дипломной работы

Студенту Кемелбеку Рустаму Азаматовичу
(Ф.И.О.)

Тема работы создание интернет-портала на
основе технологий Yii PHP Framework

Утверждена приказом по университету № 115 от «24» сентября 201 9 г.

Срок сдачи законченной работы _____ «_____» _____ 201 9 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): _____

1. При проектировании веб-портала необходимо
написать основополагающие принципы на языке
мус.

2.

Перечень вопросов, подлежащих разработке в дипломной работе, или краткое содержание дипломной работы: _____

1. Исследование существующих веб-порталов с поль-
зующих технологию Yii

Перечень графического материала (с точным указанием обязательных чертёжей)

- 1 Схема работы серверов Apache и Nginx
- 2 Структура модели данных MVC
- 3 Структура базы данных

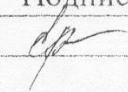

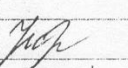

Рекомендуемая основная литература

1 Матросов А., Сергеев А., Чаушан М. HTML 4.0 - СПб 2000 - 672с

2 Л. Томсон, Л. Веллинг Разработка веб-приложений на PHP и MySQL - М. Диасофт 2003 672с

3 Рос П., Карачел К. Система баз данных проектирование, реализации и управление, 5-е изд. СПб 2004 - 1040с

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Однорядная заставка	Ержан А.А.	06 апреля 2014 г.	
Экономический расчет		01 мая 2014 -	
составности проекта	Еркешев	23 мая 2014	
Безопасность жизнедеятельности на рабочем месте	Прихорько Н.Г.	01 мая 2014 -	
Нормоконтроль	Тусулов Д.М.	23 мая 2014	
		30 мая 2014	

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Подбор литературы	16 мая	
2	исследование предметной области	8 апреля	
3	Разработка базовых данных	32 апреля	
4	Разработка алгоритмов системы	3 мая	
5	отладка системы	6 мая	
6	Техника защиты информации	20 мая	
7	Безопасность жизнедеятельности	20 мая	

Дата выдачи задания « 3 » _____ 2017 г.

Заведующий кафедрой _____ (Куранбаев З.К.)
(подпись) (Ф.И.О.)

Научный руководитель
работы _____ (Ержан А.А.)
(подпись) (Ф.И.О.)

Задание принял к
исполнению студент _____ (Камбардиев Р.А.)
(подпись) (Ф.И.О.)

Аннотация

В данном дипломном проекте проводится разработка web-интерфейса на платформе Yii Framework, использующий технологию модель-вид контроллер.

Рассмотрены существующие web-порталы, использующие данную технологию, произведен анализ функционала порталов, преимущества и недостатки.

Web-интерфейс разработан с использованием новейших web-технологий и отвечает всем современным стандартам к программному обеспечению.

Кроме этого, в дипломной работе проводится анализ потенциально опасных производственных факторов, рассматриваются вопросы безопасности труда, а также проводится технико-экономическое обоснование программного продукта.

Андатпа

Осы дипломдық жобада Yii Framework платформасында модель-бейне контроллер технологиясын қолданатын web-интерфейс әзірлемесі жүргізілді.

Бұған дейін бұл технологияны қолданған аналогтардың функционалдық талдауы жүргізілді.

Web-интерфейсі жаңа web-технологияларын қолдана отырып жасалды және бағдарламалық қамтамасыз етуде қазіргі барлық талаптарға жауап береді.

Бұдан басқа дипломдық жұмыста еңбек қауіпсіздігі мәселелері қарастырылды және жобаны әзірлеудің техникалық-экономикалық негіздемесі мен бағасы есептелді.

Abstract

In this project is conducted to develop web-interface platform Yii Framework, using technology model-view controller.

The existing web-portals using the technology, the analysis of functional portals, advantages and disadvantages

Web-interface is designed using the latest web-based technologies and meets all modern standards to the software.

In addition, the research paper conducted an analysis of potential safety hazards, the issues of safety, and also conducted a feasibility study for a software product.

Содержание

Введение.....	7
1 Аналитическая часть.....	9
1.1 Методика создания web-сайтов.....	9
1.2 Обзор инструментальных средств разработки web-сайтов.....	10
1.3 Анализ программно-аппаратной платформы реализации web-сайтов.....	11
1.4 Требования к пользовательскому интерфейсу.....	14
1.5 Постановка задачи.....	15
1.6 Назначение и область применения.....	16
2 Теоритическая часть.....	18
2.1 Язык гипертекстовой разметки HTML.....	18
2.2 Язык программирования PHP.....	20
2.2.1 Синтаксис PHP.....	21
2.2.2 Парадигма PHP.....	22
2.3 Расширяемый язык разметки XML.....	23
2.4 Javascript - язык создания сценариев.....	25
2.5 CSS - средство управления макетом HTML документов.....	27
2.6 СУБД.....	29
2.7 Обзор web-фреймворков.....	31
3 Разработка web-приложения.....	34
3.1 MySQL.....	34
3.1.1 Анализ предметной области.....	35
3.1.2 Проектирование базы данных.....	36
3.1.3 Обеспечение безопасности базы данных.....	41
3.2 Разработка интерфейса приложения.....	42
3.3 Перенос сайта с локального сервера на виртуальный хостинг.....	49
4 Техничко-экономическое обоснование проекта.....	53
4.1 Описание работы и обоснование необходимости.....	53
4.2 Трудовые ресурсы, используемые в работе.....	53
4.3 Расчет стоимости работы по проектированию и разработке.....	54
4.4 Расчет затрат на амортизацию.....	61
5 Техническое решение безопасности труда.....	66
5.1 Анализ потенциально опасных и вредных производственных факторов.....	66
5.1.1 Микроклимат рабочей зоны программиста.....	66
5.1.2 Воздействие шума на программиста. Защита от шума.....	67
5.1.3 Опасность уровня напряженности электромагнитного поля.....	67
5.2 Расчет искусственного освещения помещения.....	69
5.3 Расчет системы кондиционирования офиса.....	72
Заключение.....	76
Список используемой литературы.....	77
Приложение А.....	78

Введение

Так как в последние годы в Казахстане быстрыми темпами развивается Интернет, и большая доля населения имеет у себя дома или на работе выход к его ресурсам, возникает необходимость использования этого направления в рекламных целях. Интернету в Казахстане, как каналу коммуникации между людьми свойственны следующие особенности:

1 одним из главных преимуществ интернета заключается в том, что способствует распространению информации. Так, появившееся на информационном портале может мгновенно растиражировано по сотням социальных сетей, другим информационным порталам, а также газетными изданиями. Поэтому, данную особенность интернета я упомянул первым;

2 интернет дает хорошую возможность направить воздействие на любую узкопрофильную целевую аудиторию, в которой заинтересовано предприятие, выделять места в этой аудитории для составления более профессиональных обращений; порой даже конкретно учитывать индивидуальные особенности и свойства каждого посетителя, соединяя данные разных исследований по изучению интернет-аудиторий, большой процент аудитории интернета Казахстана, составляют активные слои населения, которые участвуют в процессе анализа и принятия решений.

Создание сайта - это еще не самый трудный шаг в сайтостроении. Намного сложнее веб-ресурс раскрутить и вывести в «ТОП» поисковиков, а также сделать интересным и полезным для пользователей.

Как ни что иное, анализ сайта онлайн поможет в продвижении сайта. Возможно, некоторые не до конца понимают как именно. Но на самом деле все очень просто. Для того чтобы развивать что-нибудь, причем делать это правильно, просто необходимо следить за его развитием. То же самое можно сказать и о веб-площадке. Постоянно следя за основными характеристиками Интернет-ресурса можно понять, в каком направлении двигаться для его дальнейшего развития. Именно поэтому многие вебмастера проверяют свои сайты с помощью различных сайтов, анализирующих веб-ресурсы. Если время от времени основные характеристики сайта (в том числе количество уникальных посетителей и просмотров) будет расти, тогда вы на правильном пути.

Вовсе необязательно, чтобы Интернет-ресурс продвигался огромными скачками, медленный «путь» всегда лучше (в этом случае и падение не будет резким). Начинать анализировать сайт можно с самого первого дня его создания, но делать это вовсе необязательно. Особый контроль нужно учредить после того, как Интернет-ресурс проиндексируется на основных поисковых системах: Google, Yandex, Rambler, Bing.

В разработке сайта необходимо большое внимание уделить модулям и плагинам разрабатываемого web-сайта. Правильная настройка и оптимизация всех html-тэгов таких как meta, description и т.д.

Для того чтобы ускорить этот процесс - можно вручную добавить сайт в инструменты для вебмастеров этих самых поисковиков. При правильном восприятии тенденций развития и динамики показателей вашего сайта - можно ускорить его продвижение и добиться гораздо лучших показателей в сео-оптимизации.

1 Аналитическая часть

1.1 Методика создания WEB-сайтов

На сегодняшний день существует огромное количество различных методов создания сайтов, различающихся в зависимости от назначения и типа сайта, умений разработчика или финансовых возможностей заказчика. При этом классифицировать методы можно по разным признакам: используемому программному обеспечению, архитектуре сервера, технологическим признакам либо даже маркетинговым. Однако перечисленные особенности касаются, прежде всего, разработчика, провайдера либо регистратора.

Для будущего владельца сайта главную роль скорее сыграют такие факторы, как время выполнения и стоимость. Разнообразных же методов разработки, отличающихся друг от друга, десятки. Несмотря на множество технологических различий, принципиально отличающихся подходов к разработке можно выделить лишь несколько. Если группировать методы создания сайтов, исходя из того, что признак цена-время является самым востребованным, то можно выделить три основных метода.

1 Разработка с нуля.

Данный метод наиболее длительный и трудный. При этом он требует знаний хотя бы одного языка веб-программирования (PHP, HTML, CSS), а также опыт работы с базами данных (MySQL). Можно также воспользоваться наиболее популярными программами для создания сайтов вручную, к которым относятся Dreamweaver, FrontPage, WebSite X5, а также Photoshop (для работы с изображениями).

Однако всю рутинную работу разработчику придется выполнить самому. Результаты при этом могут получиться более чем скромными, характерными для новичков и любителей в сфере SEO. Главным минусом данного метода является необходимость осуществлять все самостоятельно, затрачивая значительное количество времени и сил.

2 Использование конструктора.

Самый простой метод разработки, поскольку не требует знаний профессионального веб-программиста. Однако на качественную работу также рассчитывать не стоит. Существует множество инструментов для разработки по тем либо иным шаблонам и макетам. К самым популярным и хорошо изученным конструкторам относятся «narod.ru», «setup.ru», «umi.ru», «ucoz.ru» и подобные им.

Зачастую, воспользовавшись конструктором на начальном этапе, разработчик может впоследствии пожалеть о потерянном на изучение работы конструктора времени, поскольку данный метод таит в себе пару серьезных недостатков. Один заключается в ограниченности возможностей такого рода конструктора, невозможности реализовать более, нежели предусмотрено его

функциями, а другой - в наличии специальных фрагментов кода и сложности раскрутки созданного ресурса.

3 Использование системы управления сайтом.

Самый быстрый, а также широко используемый метод. На сегодняшний день широко используются такие системы («движки»), как WordPress, Joomla, Django, Drupal и прочие, которые идеально подойдут для сайтов либо блогов. Для форумов более подходящими станут: phpbb, ExBB, PunBB, SMF.

Использование движков (CMS) предоставляет доступ к множеству современных шаблонов и схем построения, а также значительно ускоряет и упрощает работу разработчика.

Будущий владелец сайта сам выбирает, какой из методов использовать. Конечно же, он также может заказать сайт в веб-студии, ведь разработанный с применением новейших технологий и инструментов слаженным коллективом дизайнеров, программистов, оптимизаторов, верстальщиков, копирайтеров, аниматоров профессиональной веб-студии сайт всегда предпочтителен.

1.2 Обзор инструментальных средств разработки Web-сайтов

Для верстки веб-страниц придумали немало различных инструментов. Все редакторы делятся на текстовые и визуальные. Визуальные редакторы выгодно отличаются тем, что не требуют от пользователя знаний html и css.

Вы можете создавать сайт как на листе бумаги - что нарисовали, то и получили, но есть и свои минусы - все визуальные редакторы имеют огрехи и ограниченные возможности.

Поэтому настоящие профессионалы предпочитают не отказываться от использования текстовых редакторов. В любом случае лучше иметь в своем распоряжении два редактора: визуальный и текстовой.

Рассмотрим подробнее существующие визуальные редакторы:

1 Macromedia Dreamweaver. Высокотехнологичный инструмент специально для создания web-страниц. По статистике это одна из лучших программ на сегодняшний день. Разработчики заявляют, что Macromedia Dreamweaver служит для проектирования сложных профессиональных web-сайтов. Dreamweaver - это не только визуальный редактор, это сложный и мощный инструмент, на освоение которого придется потратить время;

2 Adobe GoLive - прекрасное решение для тех, кто предпочитает работать с программами от Adobe. Знакомая среда позволяет легко и быстро разобраться, что здесь к чему. GoLive всего лишь визуальный редактор, который поддерживает Html, CSS и XML. Зато GoLive имеет чудесный встроенный редактор кода;

3 Microsoft FrontPage По словам разработчиков FrontPage создана для проектирования сложных веб-узлов, а также позволяет управлять разработанным сайтом. FrontPage прекрасно дружит как с Html, Css, так и с Dhtml и Javascript. Программа дает широкие возможности для управления изображениями и flash-технологиями;

4 Hometown - это наиболее популярный и мощный текстовый редактор. В нем довольно просто и удобно работать с Html-кодом, а также с CSS-редактором;

5 HTML Pad это самый серьезный конкурент программы Hometown, так как наряду с ней пользуется любовью и признанием пользователей. Помимо стандартного набора функций HTML Pad дружит с JavaScript, VBScript, Perl, SSI и ASP. А помимо все прочего программа содержит много полезных материалов по CSS и Html;

6 Notepad или обычный Блокнот эта самая простая программа, с которой многие начинали свои первые шаги в веб-программировании. Блокнот есть у всех без исключения в стандартных программах.

По крайней мере, я именно с него начинал свой путь в изучении HTML и CSS и вам советую. О том, как научиться создавать сайты, вы можете прочитать в отдельной записи на блоге, но хочу отметить, что каким бы визуальным редактором вы в дальнейшем не пользовались, опыт работы с простым текстовым редактором вам всегда пригодится

1.3 Анализ программно-аппаратной платформы реализации Web-сайта

Сегодняшний рынок программного обеспечения предъявляет большие требования к создаваемым проектам. Так, для современных программных средств важными требованиями являются переносимость, мультиплатформенность и масштабируемость.

Под переносимостью подразумевается возможность использовать программное средство на разных программно-аппаратных платформах без существенной переработки кода.

Масштабируемость означает возможность добавления новых функций и свойств программного средства с минимальным изменением всего кода в целом. Идеальным является вариант, который позволяет наращивать мощность ПП без изменения основного кода, лишь добавляя новые модули.

Поэтому при разработке учитывались оба этих требования. Естественно, создать достаточно сложное ПО, которое работало бы на всех известных платформах, практически невозможно, но следует стремиться обеспечить его функциональность на самых распространенных платформах [1].

Исходя из того, что платформа IBM PC является наиболее распространенной в Казахстане, было принято решение разрабатывать web-сайт именно под эту аппаратную платформу. Проанализировав системное программное обеспечение IBM PC-совместимой компьютерной техники, были получены следующие результаты: 73 % - OS семейства Windows, 16 % - Linux, 11 % - Free BSD, Open BSD, SCO, Mac OS X, Novell NetWare. Исходя из этих результатов, а так же из соображения, что программное обеспечение должно функционировать на как можно большем количестве платформ, было принято решение разрабатывать ПС с таким расчетом, чтобы обеспечить

функционирование, как его отдельных компонентов, так и всего комплекса в целом на двух основных программных платформах: Windows и Linux. Причем это требования распространяется на ту часть разрабатываемого программного обеспечения, которое должно быть размещено на web-сервере.

Веб-сервер - это сервер, принимающий HTTP запросы от клиентов. Обычно в роли клиентов выступают веб-браузеры, и выдающий им HTTP ответы, вместе с которыми передаются HTML страницы, изображения, файлы, медиа-потoki или другие данные. Веб-серверы являются основой Всемирной паутины.

Фактически веб-сервером называют как программное обеспечение обеспечивающее все вышеперечисленные действия, так и компьютер, на котором это программное обеспечение работает. Клиенты получают доступ к веб-серверу по средствам единого указателя ресурсов - URL адресу нужной им веб-страницы или другого ресурса располагаемого на сервере.

Apache HTTP-сервер (назван именем группы племён североамериканских индейцев апачей; кроме того, является сокращением от англ. a patchy server; среди русских пользователей общепринято искажённое апáч) - свободный веб-сервер [2].

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Open Server.

Open Server представляет из себя так называемый WAMP комплекс, суть которого можно понять из расшифровки этой аббревиатуры:

Windows - операционная система, для работы в которой предназначен данный локальный сервер

Apache - web-сервер, который «поднимается» при запуске программы Open Server

MySQL - очень популярная система управления базами данных, которая является обязательным условием для работы многих движков сайтов, в том числе таких популярный, как Joomla и WordPress

PHP - интерпретатор серверного языка программирования, на котором написано большинство CMS и создано множество других веб-приложений

Кроме этого «джентельменского набора» Опен Сервер имеет в своем загашнике Nginx (популярный веб-сервер, часто устанавливаемый вместе с Apache) и еще кое-что, показанное на приведенном ниже скриншоте. К тому же, этот локальный сервер позволяете в настройках выбрать один из нескольких вариантов Apache, PHP, MySQL и других компонентов (Рисунок 1.1).

- Open Server 5.0.0;
- Apache 2.2.27;
- Apache 2.4.9;
- Nginx 1.4.7;
- Nginx 1.5.12;
- MySQL 5.1.73;
- MySQL 5.5.37;
- MySQL 5.6.17;
- MariaDB 5.5.36;
- MongoDB 2.4.9;
- PostgreSQL 9.2.8;
- PostgreSQL 9.3.4;
- PHP 5.2.17 (Zend Optimizer 3.3.3, IonCube 4.0.7, Memcache 2.2.4);
- PHP 5.3.28 (Xdebug 2.2.4, IonCube 4.5.3, Memcache 2.2.7, Mongo 1.4.5, Imagick);
- PHP 5.4.26 (Xdebug 2.2.4, IonCube 4.5.3, Memcache 2.2.7, Mongo 1.4.5, Imagick);
- PHP 5.5.10 (Xdebug 2.2.4, IonCube 4.5.3, Memcache 2.2.7, Mongo 1.4.5, Imagick, OPcache, Win32ps 1.0.2, Uploadprogress 1.0.3.1, SSH2 0.12, RAR 3.0.0, HTTP 1.7.6, Excel 0.9.8, FFmpeg 0.8.3 и др.);
- ImageMagick 6.8.0-7 Q16;
- FTP FileZilla 0.9.41;
- Fake Sendmail 32;
- NNCron Lite 1.17;
- Memcached 1.2.6;
- Memcached 1.4.5;
- Unbound 1.4.22;
- Adminer 4.0.3;
- HeidiSQL 8.3;
- Webgrind 1.0;
- RockMongo 1.1.5;
- PHPMYAdmin 4.0.10;
- PHPPgAdmin 5.1;
- PHPMemcachedAdmin 1.2.2;

Рисунок 1.1 - Список компонентов сервера

Мне, как не специалисту в веб разработке, довольно трудно осознать всю грандиозность данного проекта под названием Open Server, но, судя по отзывам, разработка комплекса ведется с учетом пожелание пользователей и огромным вниманием к деталям (если сравнивать с Денвером, то получите «Вольво» вместо «Лады», хотя и то, и то ездит). Имеется удобный интерфейс управления локальным сервером, созданными на нем сайтами и приложениями, а также к тем инструментами и утилитами, которые входят в его комплект.

Ну, и даже такому нубу, как я, понятно, что простой набор слов «не требует установки» значит очень много. Можно распаковать архив с дистрибутивом Open Server на флешку, переносной жесткий диск и даже на CD/DVD (правда, с ограничениями). После этого с вашими сайтами и веб-приложениями можно будет работать на любом компьютере дома, на работе, в интерне-кафе или где-либо еще, где имеется подходящая операционная система (Windows 8 / Windows 7 / Windows Server 2008 / Windows Vista / Windows XP SP3).

Однако, просто портативного локального сервера Open Server веб разработчикам может быть не достаточно. Да, на основном компьютере у них, конечно же, установлен весь необходимый в работе софт и даже сверх того. Но вот вставив флешку с ОпенСервером в чужой компьютер, вы уже много из того, к чему привыкли и что просто необходимо в работе, не обнаружите.

1.4 Требования к пользовательскому интерфейсу

На сегодняшний день Интернет повсеместно используется в различных областях человеческой деятельности: коммуникация (электронная почта, совместная работа удаленных сотрудников, видеоконференции, электронные конференции, системы диалоговых сообщений (чаты), ICQ, Napster), электронная коммерция, дистанционное обучение, доступ к удаленной информации, электронные развлечения (on-line игры, видео по заказу, интерактивное телевидение и т.д.). И это далеко не полный список использования Интернета.

Современные специалисты, и тем более специалисты, специализирующиеся в области информационных технологий и связи, должны в достаточной мере владеть Интернет-технологиями, и в частности веб-технологиями, включая создание веб-сайтов средствами HTML и CSS. Владение такими технологиями, помимо возможности размещения собственной информации в Интернете, поможет на более высоком уровне использовать бесчисленные информационные ресурсы, расположенные во Всемирной паутине. Так, изучение веб-дизайна даст возможность лучше разбираться в структуре сайтов, размещенных в Интернете, более осмысленно использовать навигацию как в пределах сайта, так и между отдельными веб-сайтами. Оптимизация же собственного сайта для поисковых систем, а так же регистрация его в интернет-каталогах, поможет более эффективно использовать поисковые ресурсы Интернета.

Приведем ряд общих требований, предъявляемых к веб-сайтам:

- 1 легкость освоения;
- 2 запоминаемость;
- 3 эффективность использования;
- 4 надежность использования;
- 5 удовлетворение пользователя.

При создании веб-сайтов необходимо руководствоваться следующим:

- 1 не существует типичных людей, всегда пытайтесь представить в качестве посетителя вашего сайта реально существующего человека;
- 2 избегайте использования текста, графики и фона одного оттенка;
- 3 поддерживайте высокий контраст. Избегайте использования текста, фона и графики одинаковой яркости;
- 4 избегайте фонов, перегруженных графическими элементами;
- 5 не заставляйте пользователей запоминать информацию, так как узнавание намного проще, чем воспроизведение из памяти;
- 6 ссылки, уже посещенные пользователем, должны отличаться цветом и/или стилем от тех, по которым еще не был осуществлен переход;
- 7 так как запоминать зрительные образы намного проще, делайте страницу так, чтобы она оставалась в памяти у пользователя и отличалась от остальных;

8 ограничивайте размер однородных групп (например ссылок), из которых необходимо сделать выбор 5 - 9 элементами. Это связано с «магическим» числом миллера - $7 / 2$, указывающего, сколько объектов одновременно человек может удерживать во внимании;

9 помните, что время, которое пользователь согласен потратить на ожидание, пропорционально ожидаемому результату;

10 если время загрузки страницы составляет более 30 с., необходимо обеспечить пользователю уверенность в том, что процесс действительно происходит. Это можно реализовать, например, при помощи индикатора выполнения процесса;

11 затраченное время значит для пользователя намного больше, чем закаченные байты;

12 сводите к минимуму расстояние, преодолеваемое указателем мыши между двумя последовательными вариантами выбора;

13 постарайтесь свести к минимуму расстояние между наиболее часто посещаемыми областями страницы (особенно это касается главной страницы) и кнопкой «назад» браузера.

Ниже приведены некоторые наиболее распространенные традиции в веб-дизайне:

- логотип сайта в верхнем левом углу одновременно является ссылкой на главную страницу;

- ссылки часто продублированы в нижней части страницы. На длинных страницах применяются ссылки, возвращающие пользователя в верхнюю часть страницы;

- то, на чем возможен щелчок кнопкой мыши, имеет специальное выделение. Например, текст ссылок обычно имеет подчеркивание или какое-либо иное выделение. Кроме того, обычно применяется динамическое выделение, когда объект изменяет свой внешний вид в момент прохода над ним указателя мыши.

1.5 Постановка задачи

Перед автором была поставлена первая задача: разработать структурную схему проекта web-сайта для использования в среде Internet и локальной сети кафедры. По мнению руководителей практики, разрабатываемый web-сайт должен обладать следующими особенностями:

- гибкостью, удобной для администраторов системой управления структурой;

- web-сайт должен поддерживать использование звука, графических вставок, анимации, которые должны усиливать эмоционально-ценностный компонент содержания, формировать мотивацию;

- для пользователей должна быть также реализована возможность распечатать любую страницу web-сайта;

- для посетителей сайта должен быть создан форум, в котором пользователи могли бы задавать интересующие их вопросы и получать на них ответы в кратчайшие сроки.

Однако главной задачей проектирования было создание системы управления содержимым, которая бы позволяла вносить изменения web-сайт с возможностью разграничения прав доступа к содержимому и независимостью от технических специалистов.

Структура сайта должна быть понятной, не содержать логических противоречий, позволять посетителю сайта легко найти всю опубликованную информацию.

Логически связанные друг с другом страницы сайта должны иметь взаимные ссылки, что позволяет посетителю легко получать доступ к необходимой информации.

Навигация на сайте должна быть представлена в виде меню, которое должно обеспечивать простой и понятный доступ ко всем страницам сайта.

Тексты на сайте не должны содержать грамматических и стилистических ошибок.

Дизайн должен быть выполнен в единой цветовой гамме, без включения элементов, не сочетающихся с ней по цвету.

Сайт должен нормально просматриваться и функционировать в наиболее распространенных браузерах: Internet Explorer (версии 7.0 и выше), Opera (7-я и выше), Firefox, Chrome, Safari.

Сайт должен корректно отображаться для посетителей, имеющих разрешение экрана 800x600 и выше.

1.6 Назначение и область применения web-сайта

Минуло то время, когда самым распространенным способом создания сайта было создание набора статичных HTML страничек и их размещение в сети Интернет. Сейчас при создании сайта владелец должен стремиться к тому, чтобы получать не просто статичный, "мертвый сайт", а сайт, который может жить и изменяться.

Конечно, при создании сайта большое значение имеет его дизайнерское решение и визуальное оформление, но самая важная составляющая популярности сайта это его содержание, которое регулярно обновляется и всегда актуально. Последние новости, анонсы, предложения, цены, события, статьи и описания - вот что ищут пользователи в Интернете. Актуальное и регулярно обновляемое содержимое существенно повышает привлекательность и авторитетность сайта, что в свою очередь напрямую влияет на количество его посетителей.

Существует несколько способов управления информационным наполнением WEB сайта.

Наиболее распространенным ранее способом было редактирование HTML страниц сайта и пересылка файлов по протоколу FTP на сервер хостинг-

провайдера. Вам незнакомы эти термины? Действительно, использование данного способа требует наличия знаний языка гипертекстовой разметки (HTML) и технологий Интернет.

Технологии не стоят на месте, и сегодня, для того, чтобы облегчить жизнь пользователей, разрабатываются и приобретают все большую популярность системы управления содержимым сайтов (CMS - Content Management System). Они предназначены для создания динамических сайтов и облегчения обновления их содержимого. Доступ к системе управления возможен с любого компьютера, включенного в глобальную сеть Интернет.

2 Теоретическая часть

2.1 Язык гипертекстовой разметки HTML

HTML (от англ. HyperText Markup Language - «язык гипертекстовой разметки») - стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием сжатия.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <TABLE> предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении [2].

Аббревиатура эта расшифровывается как HyperText Markup Language и означает язык гипертекстовой разметки. Создал его всем известный основоположник «всемирной паутины WWW» Тим Бернерс-Ли (на основе уже имеющегося к тому времени языка SGML), который и сейчас продолжает участвовать в работе над новыми стандартами интернета в рамках консорциума W3c (его еще часто называют валидатор W3C (validator)).

Собственно говоря, первая версия языка Html появилась в начале девяностых годов прошлого века (жутковато звучит - прошлого века, однако, это именно так) и была ориентирована в первую очередь на передачу информации в научной среде, но, однако, с этого момента можно начинать отчет популярности всемирной паутины.

До этого интернет был уделом немногих знающих и интересующихся людей, но с появлением HTML и первых браузеров, способных интерпретировать его незамысловатый код в понятные и удобные пользователю вещи (веб страницу или, как еще часто говорят, веб документ), всемирная паутина начала свое победоносное шествие. Наверное, появление языка гипертекстовой разметки можно приравнять к переходу от текстовых к графическим операционным системам.

Довольно интересными на мой взгляд представляются темп и нюансы развития этого языка разметки. Итак, через несколько лет после появления первой версии, по инициативе Тима Бернерса-Ли был создан консорциум W3C (World Wide Web Consortium), призванный стать законодателем стандартов и не допустить разброда и шатаний в рядах разработчиков, которые могли привести к очень неприятным последствиям.

В 1994 году разрабатываются стандарты гипертекстовой разметки второй версии, а уже в 1995 ведутся работы над Html 3 с поддержкой CSS (таблиц каскадных стилей). Примерно в это же время появляется и набирает популярность первый браузер Мозаика, который очень быстро был переделан в Netscape Navigator.

MicroSoft хотела купить Netscape Navigator для интеграции его в Windows, но разработчики этого браузера отказались (в итоге появился Mozilla Firefox), в результате чего мы получили и имеем по сей день собственное творение MicroSoft (IE - Internet Explorer), которое они создали на базе открытых кодов Мозаики.

Что интересно, IE (в силу своей предустановленности в самой популярной операционной системе) сумел-таки выкинуть из рейтинга браузеров некогда очень популярный Netscape Navigator, но получил взамен ряд новых игроков (бесплатный браузер Opera, обозреватель от Mozilla, а еще сегодняшнего лидера Гугол Хром и др.). В этот период разработчики браузеров зачастую бежали впереди паровоза (валидатора) и вводили свои собственные стандарты, ибо работа над форматом Html в W3C шла довольно медленно.

Узрев такое дело, консорциум в течении одного 1997 года сделал огромный скачок - язык разметки претерпел сразу два изменения, перейдя от версии 3.2 до версии 4.0, а затем (в 1999) и до той версии, которую мы используем по сей день - Html 4.01. С тех пор, представляете, стандарт не менялся уже около двенадцати лет (всех все устраивало, и разработчиков браузеров и команду валидатора W3C).

Сейчас консорциум с подачи конгломерата разработчиков браузеров ведет активные работы над форматом, но ждать его появления в ближайшем будущем, наверное, не стоит. Хотя некоторые нововведения формата XHTML 5 уже поддерживаются некоторыми браузерами в той или иной степени. Скорее всего, именно так и будет происходить дальше - новый формат будет внедряться по частям, но полной его поддержки всеми браузерами придется ждать довольно долго.

Итак, с помощью XHTML мы создаем web страницы (документы). Множество web страниц, относящихся к одному и тому же доменному имени, называются сайтом. Понятно, что web страницы, входящие в состав сайта, должны где-то физически размещаться и быть доступными любым пользователям в течении 24 часов.

Для этих целей предназначены web сервера, которые, как правило, размещены в специально оборудованных для этого помещениях, и представляют из себя обычные компьютеры, заточенные и оптимизированные

под выполнение данной задачи. Web сервера имеют постоянный и широкополосный доступ в интернет для того, чтобы пользователи из любой точки земного шара и в любом количестве могли получить доступ к вашему сайту.

Услуги по предоставлению места для размещения сайтов предлагают так называемые хостеры и, естественно, делают они это за деньги, хотя в природе встречаются и приятные исключения в виде бесплатного хостинга (тут я писал уже что такое хостинг, домен, Днс и IP адреса). В зависимости от посещаемости вашего сайта, вам может понадобиться как весь web сервер целиком (услуга предоставления выделенного сервера), так и его малая часть (виртуальные хостинг или виртуальный выделенный сервер).

Любая страница (документ) вашего сайта будет иметь свой собственный уникальный адрес или, другими словами, URL (Uniform Resource Locator). Для того, чтобы пользователь увидел содержимое этой страницы у себя на экране компьютера, его браузер должен будет завязать диалог с сервером путем отправки http запросов и получения http ответов.

В результате этого диалога браузер получает Html код документа, разбирает его, подгружает все необходимые дополнительные элементы оформления страницы (изображения, ссс файлы, скрипты) и пользователь видит на экране страницу сайта.

2.2 Язык программирования PHP

PHP - это серверный язык создания сценариев (или стороны сервера), разработанный специально для Web. В HTML-страницу можно внедрить код PHP, который будет выполняться при каждом ее посещении. Код PHP интерпретируется Web-сервером и генерирует HTML или иной вывод, наблюдаемый посетителем страницы.

Разработка PHP была начата в 1994 г. и вначале выполнялась одним человеком, Расмусом Лерддорфом (Rasmus Lerdorf). Этот язык был принят рядом талантливых людей и претерпел три основных редакции, пока не стал широко используемым и зрелым продуктом, с которым мы имеем дело сегодня. К январю 2001 г. он использовался почти в пяти миллионах доменов во всем мире и их число продолжает быстро расти. Количество доменов, в которых в настоящее время используется PHP, можно выяснить на странице <http://www.php.net/usage.php>.

PHP - это продукт с открытым исходным кодом (Open Source). У пользователя имеется доступ к исходному коду. Его можно использовать, изменять и свободно распространять другим пользователям или организациям.

Первоначально PHP являлось сокращением от Personal Home Page (Персональная начальная страница), но затем это название было изменено в соответствии с рекурсивным соглашением по наименованию GNU (GNU = Gnu's Not Unix) и теперь означает PHP Hypertext Preprocessor (Препроцессор гипертекста PHP) [3].

2.2.1 Синтаксис PHP

Синтаксис PHP подобен синтаксису языка Си. Некоторые элементы, такие как ассоциативные массивы и цикл `foreach`, заимствованы из Perl. Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

Простейшая программа Hello world на PHP выглядит следующим образом:

```
<?php
    echo 'Hello, world!';
?>
```

Открывающий тег вида `<?=` используется для сокращённой записи конструкций используемых для вывода строки.

PHP исполняет код, находящийся внутри ограничителей, таких как `<?php ?>`. Всё, что находится вне ограничителей, выводится без изменений. В основном это используется для вставки PHP-кода в HTML-документ, например, так:

```
<html>
  <head>
    <title>Тестируем PHP</title>
  </head>
  <body>
    <?php echo 'Hello, world!'; ?>
  </body>
</html>
```

Помимо ограничителей `<?php ?>`, допускается использование дополнительных вариантов, таких как `<? ?>` и `<script language="php"> </script>`. Кроме того, до версии 6.0 допускается использование ограничителей языка программирования ASP `<% %>` (конструкции `<? ?>` и `<% %>` могут быть выключены в конфигурационном файле `php.ini`).

Имена переменных начинаются с символа `$`, тип переменной объявлять не нужно. Имена переменных, функций и классов чувствительны к регистру. Константы также чувствительны к регистру. Переменные обрабатываются в строках, заключённых в двойные кавычки, и heredoc-строках (строках, созданных при помощи оператора `<<<`). Переменные в строках, заключённых в одинарные кавычки, не обрабатываются.

PHP рассматривает переход на новую строку как пробел, так же как HTML и другие языки со свободным форматом. Инструкции разделяются с

помощью точки с запятой (;), за исключением некоторых случаев, после объявления конструкции if/else и циклов.

Переменные в функцию можно передавать как по значению, так и по ссылке (используется знак &).

PHP поддерживает три типа комментариев: в стиле языка Си (ограниченные /* */), C++ (начинающиеся с // и идущие до конца строки) и оболочки UNIX (с # до конца строки).

2.2.2 Парадигма PHP

Ключевое слово class было зарезервировано ещё в третьей версии языка. В четвёртой версии стало возможно создавать классы и объекты на их основе. Однако, принципы ООП поддерживались лишь частично, так например, все члены (переменные и методы) были открыты. К тому же создание объектов было дорогой операцией и работали они медленно.

Начиная с пятой версии PHP обладает полной поддержкой ООП. Работа с классами была оптимизирована и теперь такой код работает достаточно быстро.

Класс в PHP объявляется с помощью ключевого слова class. Методы и поля класса могут быть общедоступными (public, по умолчанию), защищёнными (protected) и скрытыми (private). PHP поддерживает все три основных механизма ООП - инкапсуляцию, полиморфизм и наследование (родительский класс указывается с помощью ключевого слова extends после имени класса). Поддерживаются интерфейсы (ставятся в соответствие с помощью implements). Разрешается объявление финальных, абстрактных методов и классов. Множественное наследование классов не поддерживается, однако класс может реализовывать несколько интерфейсов. Для обращения к методам родительского класса используется ключевое слово parent.

Начиная с версии 5.4.0 множественное наследование может быть реализовано с помощью механизма особенностей (англ. trait). Особенности похожи на примеси (англ. mixins), за исключением того что для них нельзя напрямую создать экземпляр. Повторное использование кода заключено в использовании кода особенности в нескольких классах. Допускается использовать в одном классе несколько особенностей. Механизм особенностей имеет средства разрешения конфликтов имён. При запуске программы код особенности будет «вкомпилирован» в код содержащего его класса.

Классы в PHP имеют ряд специальных методов (англ. magic methods), начинающихся с двух символов подчёркивания. Особо стоит отметить конструктор (__construct()), в версиях до 5.0 конструктором служил метод, одноимённый с классом) и деструктор (__destruct()), а также методы чтения (__get()) и записи (__set()), свёртывания (__sleep()) и развёртывания (__wake()), клонирования (__clone()) и др. Эти методы являются достаточно гибким инструментом: переопределяя их, можно добиться существенного изменения поведения объекта.

Все функции-члены реализованы виртуальными и потому все они являются методами.

Экземпляры класса создаются с помощью ключевого слова `new`, обращение к полям и методам объекта производится с использованием оператора `->`. Для доступа к членам класса из его методов используется переменная `$this` [4].

2.3 Расширяемый язык разметки XML

Если Вы знакомы с HTML, изучение XML не потребует от вас особых усилий. Хотя XML, безусловно, сильно отличается по своим возможностям и предназначению от языка гипертекстовой разметки, оба эти языка являются подмножествами SGML, и, следовательно, наследуют его базовые принципы.

Структура документа

Простейший XML- документ может выглядеть так, как это показано:

```
<?xml version="1.0"?>
<list_of_items>
<item id="1"><first/>Первый</item>
<item id="2">Второй <sub_item>подпункт 1</sub_item></item>
<item id="3">Третий</item>
<item id="4"><last/>Последний</item>
</list_of_items>
```

Обратите внимание на то, что этот документ очень похож на обычную HTML-страницу. Также, как и в HTML, инструкции, заключенные в угловые скобки называются тэгами и служат для разметки основного текста документа. В XML существуют открывающие, закрывающие и пустые тэги (в HTML понятие пустого тэга тоже существует, но специального его обозначения не требуется).

Тело документа XML состоит из элементов разметки (markup) и непосредственно содержимого документа - данных (content). XML - тэги предназначены для определения элементов документа, их атрибутов и других конструкций языка. Более подробно о типах применяемой в документах разметки мы поговорим чуть позже.

Любой XML- документ должен всегда начинаться с инструкции `<?xml?>`, внутри которой также можно задавать номер версии языка, номер кодовой страницы и другие параметры, необходимые программе-анализатору в процессе разбора документа.

В общем случае XML- документы должны удовлетворять следующим требованиям:

1 в заголовке документа помещается объявление XML, в котором указывается язык разметки документа, номер его версии и дополнительная информация;

2 каждый открывающий тэг, определяющий некоторую область данных в документе обязательно должен иметь своего закрывающего "напарника", т.е., в отличие от HTML, нельзя опускать закрывающие тэги;

3 в XML учитывается регистр символов;

4 все значения атрибутов, используемых в определении тэгов, должны быть заключены в кавычки;

5 вложенность тэгов в XML строго контролируется, поэтому необходимо следить за порядком следования открывающих и закрывающих тэгов;

6 вся информация, располагающаяся между начальным и конечными тэгами, рассматривается в XML как данные и поэтому учитываются все символы форматирования (т.е. пробелы, переводы строк, табуляции не игнорируются, как в HTML).

Если XML- документ не нарушает приведенные правила, то он называется формально-правильным и все анализаторы, предназначенные для разбора XML- документов, смогут работать с ним корректно.

Как уже отмечалось, в отличие от HTML, XML никак не определяет способ отображения и использования описываемых с его помощью элементов документа, т.е. программе-анализатору предоставляется возможность самой выбирать нужное оформление. Этого требует принцип независимости определения внутренней структуры документа от способов представления этой информации. Например, задавая в документе элемент `<flower>роза</flower>`, мы лишь указываем, что rose в данном случае является цветком, но информации о том, как должен выглядеть данный элемент документа на экране клиента и должен ли он отображаться вообще, в таком определении нет.

Для того, чтобы использовать данные, определяемые элементами XML, например, отображать их на экране пользователя, необходимо написать программу-анализатор, которая бы выполняла эти действия. Уже сегодня таких программ появилось достаточное количество и у разработчиков существует возможность выбора наиболее подходящей из них для решения конкретных проблем

Как уже отмечалось ранее, в общем случае, программы- анализаторы можно разделить на две группы: верифицирующие(т.е. использующие DTD-описания для определения корректности документа) и не верифицирующие. Если Вы создаете свой язык и описываете его грамматику на основе DTD, то для анализа документов, написанных на этом языке, безусловно, потребуется программа, проверяющая корректность составления документа. Но так как использование DTD в XML не является обязательным, то любой правильно оформленный документ может быть распознан и разобран программой, предназначенной для анализа XML- документов. В любом случае, используя универсальные XML- анализаторы, Вы можете быть уверенным в том, что если заданные в документе конструкции языка являются синтаксически правильными, то программа-анализатор сможет правильно извлечь определяемые ими элементы документа и передать их прикладной программе, выполняющей необходимые действия по отображению. Т.е. после разбора

документа в большинстве случаев, Вам предоставляется объектная модель, отображающая содержимое Вашего документа, и средства, необходимые для работы с ней (прохода по дереву элементов). При этом в некоторых анализаторах способ представления структуры документа основывается на спецификации DOM, описанной в. Поэтому у Вас появляется также возможность использовать строгую иерархическую модель DOM для построения собственных документов.

Если речь идет о способах отображения информации, хранящейся в XML, то необходимо упомянуть разрабатываемый в настоящее время W3C стандарт стилевых таблиц для XML, которые предназначены для описания правил отображения элементов XML. Более подробно мы поговорим об XSL чуть позже [5].

Создавая msxml- объект при помощи CreateObject, мы в дальнейшем вызываем его методы и свойства привычным нам способом. Отличается лишь способ вставки полученной информации в HTML- страницу - она генерируется не на стороне клиента, а приходит к нему в уже готовом виде.

В заключение хотелось бы отметить, что рассмотренные способы работы с XML- документами могут применяться для отображения их элементов на экране броузера. Не всегда они являются наиболее эффективными для форматирования текста - для каждого нового документа с измененной структурой требуются частично или полностью переписывать обработчик (в следующем разделе мы попробуем использовать для этих же целей стилевые таблицы XSL). Однако использование Java Script позволяет уже сегодня разрабатывать реальные Интернет- приложения, использующие встроенный в броузер клиента анализатор в качестве средства для доступа к структурированной информации XML.

2.4 JavaScript - язык создания сценариев

Назначение языка JavaScript - позволить сделать Web-страничку активной. Язык JavaScript не имеет никакого отношения к языку Java, несмотря на схожее название. Java имеет очень немного средств для связи аплетов с браузером, что так необходимо разработчикам в их приложениях, в то время как JavaScript разрабатывался именно для этого. В отличие от Java, JavaScript является интерпретируемым языком. Программа на языке JavaScript размещается прямо в HTML-документе в виде исходного текста и автоматически компилируется в байт-код при загрузке документа в браузер.

Для авторов HTML-документов JavaScript представляет важный шаг вперед на пути к созданию HTML-приложений, которые взаимодействуют с пользователем. Таким образом, главной целью языка JavaScript является обеспечение активного взаимодействия HTML-документов с пользователем. Этот язык не претендует на то, чтобы быть полномасштабным языком программирования, такими как Java и Си. Скорее, он является расширением языка HTML, облегчающим работу пользователя с конкретным браузером.

Важен тот факт, что JavaScript-программы действительно являются выполнимым содержимым документов: они физически находятся внутри HTML-документов, в отличие от Java-апплетов, которые существуют вне документов, их активизирующих.

JavaScript "вышел на сцену" в конце 1995 года и к настоящему времени стал очень популярным и поддерживается многими браузерами. Язык JavaScript был создан компанией Netscape Communication Corporation в сотрудничестве с компанией Sun Microsystems. Первоначальное название языка JavaScript было LiveScript. Затем, после того как язык Java получил всемирную известность, LiveScript переименовали в JavaScript. Возможно, это было сделано фирмой Netscape из коммерческих соображений: все, что имело в своем названии ключевое слово "Java", вызывало живой интерес.

При разработке языка программирования JavaScript предполагалось, что он будет предназначен для непрофессионалов. По простоте использования JavaScript больше всего напоминает Basic, хотя с помощью него квалифицированные программисты могут решать достаточно серьезные и полезные задачи. JavaScript не предназначен для создания автономных программ или апплетов, и в этом он сильно отличается от других языков, таких как Си или Java. Конструкции JavaScript встраиваются непосредственно в исходный текст документов HTML и интерпретируются браузером по мере загрузки этих документов. Более того, с помощью JavaScript вы можете динамически формировать и изменять текст загружаемого документа HTML.

Еще одна важная особенность языка JavaScript - это его объектная ориентированность. И хотя объектно-ориентированные возможности JavaScript заметно беднее, чем в C++ или Java, они все же есть в этом языке. Программистам доступны многочисленные объекты, встроенные в браузер и загруженные в него документы HTML, такие как документы HTML, фреймы, формы, ссылки на другие документы и так далее.

Допускается создание собственных классов, а также группирование объектов и структур данных для расширения базовых классов. С помощью конструкций JavaScript, встроенных в документы HTML, можно обрабатывать события. Эти события возникают в результате выполнения пользователем различных операций над документом HTML, загруженным в окно браузера. Это позволяет получать различные визуальные эффекты, например, изменение внешнего вида органов управления, когда над ними находится курсор мыши. Можно предварительно проверять данные, введенных пользователем в полях формы перед их отправкой для обработки на сервер Web, а также выполнять произвольные действия перед загрузкой документа HTML в окно браузера и делать другую полезную работу.

Отметим также то, что JavaScript не предназначен для создания сложных программ. Подобные задачи лучше решать с использованием других языков программирования, например, с помощью Java. Однако с помощью JavaScript вы можете быстро создавать активные страницы с достаточно сложным

поведением. Этот язык несложен в изучении, а программы, составленные с его использованием, легко поддаются отладке [6].

2.5 CSS - средство управления макетом HTML документа

После знакомства с HTML разработчики сайтов разделяются на две основные категории. Одна часть считает, что с помощью HTML на сайте можно создавать всё или практически всё, другая же понимает, что в целом средств разметки недостаточно для оформления веб-документов. Действительно, HTML лишь первый этап в процессе обучения созданию сайтов. Следующим шагом является изучение стилей или CSS (Cascading Style Sheets, каскадные таблицы стилей).

Поскольку на файл со стилем можно ссылаться из любого веб-документа, это приводит в итоге к сокращению объёма повторяющихся данных. А благодаря разделению кода и оформления повышается гибкость управления видом документа и скорость работы над сайтом.

CSS представляет собой свой собственный язык, который совпадает с HTML только некоторыми значениями, например способом определения цвета.

Различают несколько типов стилей, которые могут совместно применяться к одному документу. Это стиль браузера, стиль автора и стиль пользователя.

Оформление, которое по умолчанию применяется к элементам веб-страницы браузером. Это оформление можно увидеть в случае «голового» HTML, когда к документу не добавляется никаких стилей. Например, заголовок страницы, формируемый тегом <H1>, в большинстве браузеров выводится шрифтом с засечками размером 24 пункта.

Стиль пользователя - это стиль, который может включить пользователь сайта через настройки браузера. Такой стиль имеет более высокий приоритет и переопределяет исходное оформление документа.

Стили являются удобным, практичным и эффективным инструментом при верстке веб-страниц и оформления текста, ссылок, изображений и других элементов. Несмотря на явные плюсы применения стилей, рассмотрим все преимущества CSS, в том числе и незаметные на первый взгляд.

Разграничение кода и оформления

Идея о том, чтобы код HTML был свободен от элементов оформления вроде установки цвета, размера шрифта и других параметров, стара как мир. В идеале, веб-страница должна содержать только теги логического форматирования, а вид элементов задаётся через стили. При подобном разделении работа над дизайном и версткой сайта может вестись параллельно. Разное оформление для разных устройств.

С помощью стилей можно определить вид веб-страницы для разных устройств вывода: монитора, принтера, смартфона, планшета и др. Например, на экране монитора отображать страницу в одном оформлении, а при её печати

- в другом. Эта возможность также позволяет скрывать или показывать некоторые элементы документа при отображении на разных устройствах.

Расширенные по сравнению с HTML способы оформления элементов.

В отличие от HTML стили имеют гораздо больше возможностей по оформлению элементов веб-страниц. Простыми средствами можно изменить цвет фона элемента, добавить рамку, установить шрифт, определить размеры, положение и многое другое.

При хранении стилей в отдельном файле, он кэшируется и при повторном обращении к нему извлекается из кэша браузера. За счёт кэширования и того, что стили хранятся в отдельном файле, уменьшается код веб-страниц и снижается время загрузки документов.

Кэшем называется специальное место на локальном компьютере пользователя, куда браузер сохраняет файлы при первом обращении к сайту. При следующем обращении к сайту эти файлы уже не скачиваются по сети, а берутся с локального диска. Такой подход позволяет существенно повысить скорость загрузки веб-страниц.

Единое стилевое оформление множества документов. Сайт - это не просто набор связанных между собой документов, но и одинаковое расположение основных блоков, и их вид. Применение единообразного оформления заголовков, основного текста и других элементов создает преемственность между страницами и облегчает пользователям работу с сайтом и его восприятие в целом. Разработчикам же использование стилей существенно упрощает проектирование дизайна.

Централизованное хранение. Стили, как правило, хранятся в одном или нескольких специальных файлах, ссылка на которые указывается во всех документах сайта. Благодаря этому удобно править стиль в одном месте, при этом оформление элементов автоматически меняется на всех страницах, которые связаны с указанным файлом. Вместо того чтобы модифицировать десятки HTML-файлов, достаточно отредактировать один файл со стилем и оформление нужных документов сразу же поменяется.

Кроме приведенных методов минимизации CSS-файлов имеются также решения, позволяющие архивировать файлы утилитой gzip прямо на сервере и отдавать браузеру упакованную версию. Современные браузеры прекрасно понимают gzip и распаковывают его на месте. Весь процесс происходит автоматически и приводит к существенному сокращению объема передаваемых по сети файлов [7].

В этом случае, в головной части кода (между тегами Head) каждого Html документа (вебстраницы) вашего ресурса обязательно должен быть прописан в специальном теге (link) путь до этого внешнего CSS файла, ибо в противном случае посетители вашего проекта рискуют увидеть его в довольно неприглядном виде.

Если же строчка с кодом подключения файла CSS будет иметь место, то браузер сразу же после того, как прочтет этот путь, начнет его загрузку и подключит стилевое оформление, не дав посетителям увидеть ваш сайт

неодетым. Строчка подключения внешних стилей может выглядеть примерно так:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head profile="http://gmpg.org/xfn/11">

<link rel="stylesheet" href="http://ktonanovenkogo.ru/wp-
content/themes/Organic/style.css" type="text/css" media="screen"
/>

</head>
```

Если вы используете CMS, то головная часть Html кода будет формироваться в одном из файлов PHP из папки с используемой вами темой оформления. Для движков Joomla, WordPress и SMF можете узнать расположение этого файла, формирующего открывающие и закрывающие теги HEAD.

2.6 СУБД

База данных - представленная в объективной форме совокупность самостоятельных материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям.

Классификация БД по модели данных:

- 1 иерархические;
- 2 сетевые;
- 3 реляционные;
- 4 объектные;
- 5 объектно-ориентированные;
- 6 объектно-реляционные.

Классификация БД по среде физического хранения:

- БД во вторичной памяти (традиционные): средой постоянного хранения является периферийная энергонезависимая память (вторичная память) - как правило, жёсткий диск в оперативную память СУБД помещает лишь кеш и данные для текущей обработки;

- БД в оперативной памяти (in-memory databases): все данные находятся в оперативной памяти;

- БД в третичной памяти (tertiary databases): средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило, на основе магнитных лент или оптических дисков. Во

вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кеш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры.

Классификация БД по содержимому:

- 1 географические;
- 2 исторические;
- 3 научные;
- 4 мультимедийные.

Классификация БД по степени распределённости:

- 1 централизованные (сосредоточенные);
- 2 распределённые.

Отдельное место в теории и практике занимают пространственные (англ. spatial), временные, или темпоральные (temporal) и пространственно-временные (spatial-temporal) БД.

Система управления базами данных (СУБД) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Когда пользователь дает запрос на какую-либо страницу (переходит на неё по ссылке или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере, то есть выполняются все программы, связанные со страницей, и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, Perl, SSI, XML, DHTML, XHTML.

Работа программ уже полностью зависима от сервера, на котором расположен сайт, и от того, какая версия того или иного языка поддерживается, сайт училище программный файл модуль.

Важной стороной работы серверных языков является возможность организации непосредственного взаимодействия с системой управления базами данных (или СУБД) - сервером, на котором упорядоченно хранится информация, которая может быть вызвана в любой момент. Популярными среди систем управления базами данных являются:

- 1 Firebird;
- 2 Microsoft SQL Server;
- 3 Microsoft SQL Server Express;
- 4 mSQL;
- 5 MySQL;
- 6 Oracle;
- 7 PostgreSQL;
- 8 SQLite;
- 9 Sybase Adaptive Server Enterprise;
- 10 ЛИНТЕР;
- 11 MongoDB.

С учетом выбора веб-сервера и языка программирования, для разработки приложения больше подходит СУБД MySQL, так как в языке PHP реализованы

быстрые и удобные инструменты для работы с этой СУБД. Так же эта СУБД имеется практически у всех хостинговых компаний предоставляющие место под сайт в сети интернет [8].

2.7 Обзор веб фреймворков

Фреймворк (англ. framework - каркас, структура) - структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Употребляется также слово «каркас», а некоторые авторы используют его в качестве основного, в том числе, не базируясь вообще на англоязычном аналоге [9-11]. Можно также говорить о каркасном подходе [9] как о подходе к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть - каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнезда, в которых размещается вторая, переменная часть - сменные модули (или точки расширения).

Yii - это высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Название Yii (произносится как Yee или [ji:]) означает простой (easy), эффективный (efficient) и расширяемый (extensible).

Подобно большинству других PHP-фреймворков, Yii - это MVC-фреймворк. Используемая модель MVC (model-view-controller) дает большие возможности в построении модульного приложения. Также модульные приложения дают возможность работать над приложением целой группы людей, где каждый отвечает за свой кусок или модуль приложения. Еще одним преимуществом Yii является статичность его кода, что позволяет легко переносить большие нагрузки.

Превосходство Yii над другими фреймворками заключается в эффективности, широких возможностях и качественной документации. Yii изначально спроектирован очень тщательно для соответствия всем требованиям при разработке серьезных веб-приложений. Yii не является ни побочным продуктом какого-либо проекта, ни сборкой сторонних решений. Он является результатом большого опыта авторов в разработке веб-приложений, а также их исследований наиболее популярных веб-фреймворков и приложений [12].

ASP.NET MVC Framework - фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller. Данный фреймворк добавлен Microsoft в ASP.NET.

В апреле 2009 года исходный код ASP.NET MVC был опубликован под лицензией Microsoft Public License (MS-PL). 27 марта 2012 года лицензия была изменена на Apache License 2.0 [13].

Шаблон архитектуры Model-View-Controller (MVC) разделяет приложение на три основных компонента: модель, представление и контроллер.

Платформа ASP.NET MVC представляет собой альтернативу схеме веб-форм ASP.NET при создании веб-приложений. Платформа ASP.NET MVC является легковесной платформой отображения с широкими возможностями тестирования и, подобно приложениям на основе веб-форм, интегрирована с существующими функциями ASP.NET, например с главными страницами и проверкой подлинности на основе членства. Платформа MVC определяется в сборке System.Web.Mvc.

Платформа ASP.NET MVC предоставляет следующие возможности:

- Разделение задач приложения (логика ввода, бизнес-логика и логика пользовательского интерфейса), широкое возможности тестирования и разработки на основе тестирования. Все основные контракты платформы MVC основаны на интерфейсе и подлежат тестированию с помощью макетов объекта, которые имитируют поведение реальных объектов приложения. Приложение можно подвергать модульному тестированию без запуска контроллеров в процессе ASP.NET, что ускоряет тестирование и делает его более гибким. Для тестирования возможно использование любой платформы модульного тестирования, совместимой с .NET Framework.

- Расширяемая и дополняемая платформа. Компоненты платформы ASP.NET MVC можно легко заменить или настроить. Разработчик может подключать собственный механизм представлений, политику маршрутизации URL-адресов, сериализацию параметров методов действий и другие компоненты. Платформа ASP.NET MVC также поддерживает использование моделей контейнера внедрения зависимости (DI) и инверсии элемента управления (IOC). Модель внедрения зависимости позволяет внедрять объекты в класс, а не ожидать создания объекта самим классом. Модель инверсии элемента управления указывает на то, что если один объект требует другой объект, то первые объекты должны получить второй объект из внешнего источника (например, из файла конфигурации). Это облегчает тестирование.

- Расширенная поддержка маршрутизации ASP.NET. Этот мощный компонент сопоставления URL-адресов позволяет создавать приложения с понятными URL-адресами, которые можно использовать в поиске. URL-адреса не должны содержать расширения имен файлов и предназначены для поддержки шаблонов именования URL-адресов, обеспечивающих адресацию, оптимизированную для поисковых систем (SEO) и для передачи репрезентативного состояния (REST).

- Поддержка использования разметки в существующих файлах страниц ASP.NET (ASPX), элементов управления (ASCX) и главных страниц (MASTER) как шаблонов представлений. Вместе с платформой ASP.NET MVC можно использовать существующие функции ASP.NET, например вложенные главные страницы, встроенные выражения (`<%= %>`), декларативные серверные элементы управления, шаблоны, привязку данных, локализацию и т. д.

- Поддержка существующих функций ASP.NET. ASP.NET MVC позволяет использовать такие функции, как проверка подлинности с помощью форм и Windows, проверка подлинности по URL-адресу, членство и роли,

кэширование вывода и данных, управление состоянием сеанса и профиля, наблюдение за работоспособностью, система конфигурации и архитектура поставщика [4].

Zend Framework - это свободный каркас на PHP для разработки веб-приложений и веб-сервисов.

Zend старается следовать духу PHP, предоставляет простые интерфейсы и мощную функциональность для разработки приложений. Он предоставляет расширения для построения современных, быстрых и безопасных сайтов.

Основывается на идеях MVC. Разрабатывается компанией Zend. В Казахстане продуктами компании Zend среди фреймворков занимает высшие ступени. Не мало высокозагруженных сайтов, обслуживаемых казахстанскими хостинг компаниями. Как и во всем мире фреймворки довольно прилично уступают CMS (Content Management system).

Помимо MVC-компонентов Zend Framework содержит множество библиотек, полезных для построения приложения. Также есть компоненты для интеграции со сторонними сервисами - например, YouTube, del.icio.us и многими другими. Начиная с версии 1.6 поставляется с JavaScript фреймворком Dojo, а также включает в себя компоненты для работы с ним [2].

3 Разработка веб-приложения

3.1 MySQL

MySQL (произносится май-эс-кю-эл) - очень быстрая, надежная система управления реляционными базами данных (СУРБД). База данных позволяет эффективно хранить, искать, сортировать и получать данные. Сервер MySQL управляет доступом к данным, позволяя работать с ними одновременно нескольким пользователям, обеспечивает быстрый доступ к данным и гарантирует предоставление доступа только имеющим на это право пользователям. Следовательно, MySQL является многопользовательским, многопоточковым сервером. Он применяет SQL (Structured Query Language - язык структурированных запросов), используемый по всему миру стандартный язык запросов в базы данных. MySQL появился на рынке в 1996 г., но его разработка началась еще в 1979 г. В настоящее время, по прошествии трех лет своего существования, эта система завоевала приз читательских симпатий журнала Linux Journal.

В настоящее время пакет MySQL доступен как программное обеспечение с открытым исходным кодом, но в случае необходимости можно получить и коммерческие лицензии.

Приступая к созданию высокозагруженного сайта, можно использовать множество различных продуктов.

Потребуется выбрать аппаратное обеспечение для Web-сервера, операционную систему, программное обеспечение Web-сервера, систему управления базами данных и язык программирования или создания сценариев.

Выбор некоторых из этих компонентов будет зависеть от уже произведенных выборов. Например, не все операционные системы могут работать на любом оборудовании, не все языки создания сценариев могут обеспечить подключение ко всем базам данных и т.д.

В этой дипломной работе не уделяется особое внимание аппаратному обеспечению, операционным системам и программному обеспечению Web-сервера. Нам это не требуется. Одно из замечательных свойств PHP в том, что он доступен для Microsoft Windows, для многих версий UNIX и выполняется на любых полнофункциональных Web-серверах [14].

Система MySQL обладает такой же гибкостью.

Чтобы продемонстрировать это, разработка и тестирование высокозагруженного сайта производилась на локальном сервере под управлением операционной системы Microsoft Windows с использованием Web-сервера Apache и на серверах Казахстанской хостинг компании PS под управлением операционной системы Debian GNU/Linux и web-сервера Nginx.

Устройство репликации.

Мастер-база записывает каждый запрос, изменяющий данные или схему данных, в специальный файл - так называемый binary-log. Когда бинарный лог

достигает определенного размера, запись переходит к следующему файлу. Существует специальный индекс этих бинарных логов, а также определенный набор команд для управления ими (например, для удаления старых бинлогов).

Слейв получает по сети эти файлы в сыром виде (что упрощает реализацию мастера) и применяет к своим данным. Слейв помнит позицию, до которой он дочитал бинлоги, и поэтому при перезапусках просит мастер продолжить отправлять ему логи начиная с нужной позиции.

Существует два режима репликации - STATEMENT и ROW. В первом режиме мастер записывает в бинлог исходные запросы, которые он выполнял для изменения данных (UPDATE/INSERT/DELETE/ALTER TABLE/...). На слейве все эти запросы выполняются так же, как они выполнялись бы на мастере.

В ROW-режиме, доступном начиная с MySQL версии 5.1, в бинлог пишутся не запросы, а уже измененные этими запросами данные (впрочем, запросы, изменяющие схемы данных (DDL), все равно пишутся как есть). Событие в ROW-режиме представляет собой:

- одну строку данных - для команд INSERT и DELETE. Соответственно, для INSERT пишется вставленная строка, для DELETE - удаленная;
- две строки - BEFORE и AFTER - для UPDATE.

При каких-то массовых изменениях данных такие бинлоги получаются значительно больше, чем если бы мы записали сам запрос, но это частный случай. Для использования в своем демоне ROW-репликация нам очень удобна тем, что не надо уметь выполнять запросы и мы получаем сразу то, что нужно - измененные строки.

Кстати, совсем необязательно включать ROW-репликацию на своем самом-важном-мастер-сервере. Мастер-сервер может кормить несколько слейвов по обычной STATEMENT репликации (резервы, бэкапы), а часть из этих слейвов могут писать ROW-логи, и уже к ним будут обращаться за данными демона.

3.1.1 Анализ предметной области

База данных должна содержать всю информацию о зарегистрированных пользователях, созданных категориях галереи, дополнительных модулях, созданных темах и сообщениях, пользовательских группах, пользовательских комментариях, приложенных файлах, компонентах, виджетах, конфигурационных настройках, правах доступа на фотографиях, опросах, сессиях.

Административная часть должна включать в себя: информацию о новых пользователях, раздел настроек сайта, возможность администрирования форума, возможность вывода статистики по форуму, управление учётными записями пользователей и другие возможности.

3.1.2 Проектирование базы данных

Процесс проектирования данных можно условно разделить на два этапа: логическое моделирование и физическое проектирование. Результатом первого из них является так называемая логическая (или концептуальная) модель данных, выражаемая обычно диаграммой «сущность-связь» или ER (Entity-Relationship) диаграммой, которая представлена в одной из стандартных нотаций, принятых для отображения подобных диаграмм. Результатом второго этапа является готовая база данных либо DDL-скрипт для ее создания.

Так как в реляционной модели данных поддерживаются между отношениями только связи типа «один-ко-многим», то необходим специальный механизм преобразования, который позволит отразить множественные связи, неспецифические для реляционной модели, с помощью допустимых для нее категорий. Это делается введением специального дополнительного связующего отношения, которое связано с каждым исходным связью «один-ко-многим», атрибутами этого отношения являются первичные ключи связываемых отношений. При этом каждый из атрибутов нового отношения является внешним ключом (FORING KEY), а вместе они образуют первичный ключ (PRIMARY KEY) новой связующей сущности. Преобразование отношений производится с помощью ассоциативных таблиц [15].

Теперь приведем конкретное описание всех таблиц на языке SQL, которые отражают все сущности и их отношения, включая ассоциативные таблицы.

Сущность «Опции» - таблица options

```
DROP TABLE IF EXISTS `options`;  
CREATE TABLE IF NOT EXISTS `options` (  
  `option_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `option_name` varchar(64) NOT NULL DEFAULT '',  
  `option_value` longtext NOT NULL,  
  `autoload` varchar(20) NOT NULL DEFAULT 'yes',  
  PRIMARY KEY (`option_id`),  
  UNIQUE KEY `option_name` (`option_name`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=2226 ;
```

Сущность «Публикации» - таблица posts

```
DROP TABLE IF EXISTS `posts`;  
CREATE TABLE IF NOT EXISTS `posts` (  
  `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `post_author` bigint(20) unsigned NOT NULL DEFAULT '0',  
  `post_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `post_date_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `post_content` longtext NOT NULL,  
  `post_title` text NOT NULL,  
  `post_excerpt` text NOT NULL,
```

```

`post_status` varchar(20) NOT NULL DEFAULT 'publish',
`comment_status` varchar(20) NOT NULL DEFAULT 'open',
`ping_status` varchar(20) NOT NULL DEFAULT 'open',
`post_password` varchar(20) NOT NULL DEFAULT '',
`post_name` varchar(200) NOT NULL DEFAULT '',
`to_ping` text NOT NULL,
`pinged` text NOT NULL,
`post_modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`post_modified_gmt` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
`post_content_filtered` longtext NOT NULL,
`post_parent` bigint(20) unsigned NOT NULL DEFAULT '0',
`guid` varchar(255) NOT NULL DEFAULT '',
`menu_order` int(11) NOT NULL DEFAULT '0',
`post_type` varchar(20) NOT NULL DEFAULT 'post',
`post_mime_type` varchar(100) NOT NULL DEFAULT '',
`comment_count` bigint(20) NOT NULL DEFAULT '0',
PRIMARY KEY (`ID`),
KEY `post_name` (`post_name`),
KEY `type_status_date` (`post_type`, `post_status`, `post_date`, `ID`),
KEY `post_parent` (`post_parent`),
KEY `post_author` (`post_author`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=888;

```

Сущность «Пользователи» - таблица users

```

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `user_login` varchar(60) NOT NULL DEFAULT '',
  `user_pass` varchar(64) NOT NULL DEFAULT '',
  `user_nicename` varchar(50) NOT NULL DEFAULT '',
  `user_email` varchar(100) NOT NULL DEFAULT '',
  `user_url` varchar(100) NOT NULL DEFAULT '',
  `user_registered` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `user_activation_key` varchar(60) NOT NULL DEFAULT '',
  `user_status` int(11) NOT NULL DEFAULT '0',
  `display_name` varchar(250) NOT NULL DEFAULT '',
  PRIMARY KEY (`ID`),
  KEY `user_login_key` (`user_login`),
  KEY `user_nicename` (`user_nicename`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;

```

Сущность «Модули» - таблица modules

```

DROP TABLE IF EXISTS `modules`;
CREATE TABLE IF NOT EXISTS `modules` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(100) NOT NULL DEFAULT '',

```

```

`note` varchar(255) NOT NULL DEFAULT '',
`content` text NOT NULL,
`ordering` int(11) NOT NULL DEFAULT '0',
`position` varchar(50) NOT NULL DEFAULT '',
`checked_out` int(10) unsigned NOT NULL DEFAULT '0',
`checked_out_time` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
`publish_up` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`publish_down` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`published` tinyint(1) NOT NULL DEFAULT '0',
`module` varchar(50) DEFAULT NULL,
`access` int(10) unsigned NOT NULL DEFAULT '0',
`showtitle` tinyint(3) unsigned NOT NULL DEFAULT '1',
`params` text NOT NULL,
`client_id` tinyint(4) NOT NULL DEFAULT '0',
`language` char(7) NOT NULL,
PRIMARY KEY (`id`),
KEY `published` (`published`,`access`),
KEY `newsfeeds` (`module`,`published`),
KEY `idx_language` (`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=90 ;

```

Сущность «Сессии» - таблица sessions

```

DROP TABLE IF EXISTS `sessions`;
CREATE TABLE `sessions` (
  `id`          varchar(60) NOT NULL DEFAULT '0',
  `member_name` varchar(64),
  `member_id`   mediumint(8) NOT NULL DEFAULT '0',
  `ip_address`  varchar(16),
  `browser`     varchar(200) NOT NULL,
  `running_time` int(10),
  `login_type`  char(3),
  `location`    varchar(40),
  `member_group` smallint(3),
  `in_error`    tinyint(1) NOT NULL DEFAULT '0',
  `location_1_type` varchar(10) NOT NULL,
  `location_1_id` int(10) NOT NULL DEFAULT '0',
  `location_2_type` varchar(10) NOT NULL,
  `location_2_id` int(10) NOT NULL DEFAULT '0',
  `location_3_type` varchar(10) NOT NULL,
  `location_3_id` int(10) NOT NULL DEFAULT '0',
  /* Keys */
  PRIMARY KEY (`id`)
) ENGINE = MyISAM;
CREATE INDEX `location1`
ON `ibf_sessions`
(`location_1_type`, `location_1_id`);
CREATE INDEX `location2`
ON `ibf_sessions`
(`location_2_type`, `location_2_id`);

```

```

CREATE INDEX `location3`
  ON `ibf_sessions`
  (`location_3_type`, `location_3_id`);
CREATE INDEX `running_time`
  ON `ibf_sessions` (`running_time`).

```

Сущность «Категории» - таблица categories

```

DROP TABLE IF EXISTS `categories`;
CREATE TABLE IF NOT EXISTS `categories` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `asset_id` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'FK to
the #__assets table.',
  `parent_id` int(10) unsigned NOT NULL DEFAULT '0',
  `lft` int(11) NOT NULL DEFAULT '0',
  `rgt` int(11) NOT NULL DEFAULT '0',
  `level` int(10) unsigned NOT NULL DEFAULT '0',
  `path` varchar(255) NOT NULL DEFAULT '',
  `extension` varchar(50) NOT NULL DEFAULT '',
  `title` varchar(255) NOT NULL,
  `alias` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin NOT
NULL DEFAULT '',
  `note` varchar(255) NOT NULL DEFAULT '',
  `description` mediumtext NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT '0',
  `checked_out` int(11) unsigned NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `access` int(10) unsigned NOT NULL DEFAULT '0',
  `params` text NOT NULL,
  `metadesc` varchar(1024) NOT NULL COMMENT 'The meta description
for the page.',
  `metakey` varchar(1024) NOT NULL COMMENT 'The meta keywords for
the page.',
  `metadata` varchar(2048) NOT NULL COMMENT 'JSON encoded metadata
properties.',
  `created_user_id` int(10) unsigned NOT NULL DEFAULT '0',
  `created_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_user_id` int(10) unsigned NOT NULL DEFAULT '0',
  `modified_time` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `hits` int(10) unsigned NOT NULL DEFAULT '0',
  `language` char(7) NOT NULL,
  `version` int(10) unsigned NOT NULL DEFAULT '1',
  PRIMARY KEY (`id`),
  KEY `cat_idx` (`extension`,`published`,`access`),
  KEY `idx_access` (`access`),
  KEY `idx_checkout` (`checked_out`),
  KEY `idx_path` (`path`),
  KEY `idx_left_right` (`lft`,`rgt`),
  KEY `idx_alias` (`alias`),
  KEY `idx_language` (`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;

```

Сущность «Баннеры» - таблица banners

```
DROP TABLE IF EXISTS `banners`;
CREATE TABLE IF NOT EXISTS `banners` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `cid` int(11) NOT NULL DEFAULT '0',
  `type` int(11) NOT NULL DEFAULT '0',
  `name` varchar(255) NOT NULL DEFAULT '',
  `alias` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin NOT
NULL DEFAULT '',
  `imptotal` int(11) NOT NULL DEFAULT '0',
  `impmade` int(11) NOT NULL DEFAULT '0',
  `clicks` int(11) NOT NULL DEFAULT '0',
  `clickurl` varchar(200) NOT NULL DEFAULT '',
  `state` tinyint(3) NOT NULL DEFAULT '0',
  `catid` int(10) unsigned NOT NULL DEFAULT '0',
  `description` text NOT NULL,
  `custombannercode` varchar(2048) NOT NULL,
  `sticky` tinyint(1) unsigned NOT NULL DEFAULT '0',
  `ordering` int(11) NOT NULL DEFAULT '0',
  `metakey` text NOT NULL,
  `params` text NOT NULL,
  `own_prefix` tinyint(1) NOT NULL DEFAULT '0',
  `metakey_prefix` varchar(255) NOT NULL DEFAULT '',
  `purchase_type` tinyint(4) NOT NULL DEFAULT '-1',
  `track_clicks` tinyint(4) NOT NULL DEFAULT '-1',
  `track_impressions` tinyint(4) NOT NULL DEFAULT '-1',
  `checked_out` int(10) unsigned NOT NULL DEFAULT '0',
  `checked_out_time` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `publish_up` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `publish_down` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `reset` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `language` char(7) NOT NULL DEFAULT '',
  `created_by` int(10) unsigned NOT NULL DEFAULT '0',
  `created_by_alias` varchar(255) NOT NULL DEFAULT '',
  `reset` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `created_by` int(10) unsigned NOT NULL DEFAULT '0',
  `modified` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `modified_by` int(10) unsigned NOT NULL DEFAULT '0',
  `version` int(10) unsigned NOT NULL DEFAULT '1',
  PRIMARY KEY (`id`),
  KEY `idx_state` (`state`),
  KEY `idx_own_prefix` (`own_prefix`),
  KEY `idx_metakey_prefix` (`metakey_prefix`),
  KEY `idx_banner_catid` (`catid`),
  KEY `idx_language` (`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```


3.1.3 Обеспечение безопасности базы данных

Одной из наиболее важных частей в БД является разработка прав доступа к ней, т.к. нужна защита от несанкционированного доступа и защита от доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа. У каждой базы данных могут быть свои роли, которые определяют права доступа к объектам. Многие администраторы не любят возиться с этими правами и из-за этого устанавливают встроенный по умолчанию `public`, который разрешает практически все.

В базе данных нашего сайта существует три основные роли:

1 администратор. Обладает правами редактирования, обновления, удаления, создания нового материала и пользователей на сайте. Фактический является супер пользователем;

2 редакторы. Обладают правами создания и редактирования материала на сайте;

3 зарегистрированные пользователи. Пользователи, которые обладают правами просмотра закрытых страниц. Не имеют доступа к административной части сайта.

Резервное копирование (англ. `back copy`) - процесс создания копии данных на носителе (жёстком диске, дискете и т.д.), предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Резервное копирование необходимо для возможности быстрого и недорогого восстановления информации (документов, программ, настроек и т.д.) в случае утери рабочей копии информации по какой-либо причине [2].

3.2 Разработка интерфейса приложения

Веб-сайт должен обладать оригинальным графическим дизайном, обеспечивающим надлежащую функциональность. Для веб-сайта должен быть разработан набор уникальных элементов фирменного стиля и визуальной идентификации: знак, логотип, цветовое решение, условные значки для элементов интерфейса.

Дизайн должен обеспечивать соответствие следующим основным требованиям:

- графические элементы должны быть выполнены с учетом специфики представления графической информации на веб-страницах;

- доступ к основной информации должен быть максимально быстрым и внимание пользователя должно концентрироваться на наиболее важных навигационных элементах;

- основная информация, меню и другие элементы навигации должны быть доступны без горизонтального прокручивания полностью открытого окна браузера;

- на главной странице необходимо использовать систему блоков с частью информации из разных разделов со ссылками "подробнее". Это позволит делать акценты на ту или иную информацию, направлять пользователей в нужный раздел;

- навигация по разделам должна быть предельно интуитивно-понятной;

- следует уделить должное внимание заголовкам и титулам. Очень важно учесть систему отступов и выравниваний. Необходимо иметь определенные принципы в верстке заголовков, выделений, абзацев, чтобы текст правильно и легко воспринимался.

При разработке дизайна веб-сайта необходимо учитывать следующие принципы:

- системный подход к работ над дизайном, обобщение передовых тенденций;

- концентрация на посетителе и на информации (user centered design);

- легкость в навигации по ресурсу (easy navigation);

- легкость в восприятии информации посетителями веб-сайта;

- минимальный "вес" страниц (в килобайтном отношении);

- соблюдение повторяемости элементов внутри системы (modularity);

- качественная и детальная проработка всей графики;

- при использовании на главной и других страницах иллюстративного образа должно быть соответствие теме раздела и не высокий объем файлов картинки. Обязательно стилевое единство иллюстраций и всего оформления;

- изобразительные элементы и даже просто цвет и тон фона не должны мешать читаемости текстовых объектов;

- изобразительные объекты не должны вытеснять основную информацию за пределы первого окна [16].

Описание программы

Приложение написано на объектно-ориентированном языке программирования PHP с применением технологии фреймворков Yii, Ajax, JQuery. Также использовался XML для передачи структурированных данных и приема данных из различных веб-форм, осуществленных на сайте. Для соединения и манипуляции с базой данных сайт использует особый класс Yii фреймворка «db». В качестве базы данных используется СУБД MySQL. Хранение паролей пользователей осуществляется не в чистом виде, а в виде хэш-функции. Хэш-функции генерируются алгоритмом MD5.

Ядро сайта спроектировано на основополагающих принципах стандарта MVC. MVC (Model-View-Controller) предполагает раздельное использование модели данных, интерфейса и взаимодействия с пользователем. Этот подход дает возможности проектированию веб-приложения каркасным, модульным методом. Достоинством модульного метода проектирования веб-приложения

является то, что при изменении части приложения он не оказывает влияние на остальную часть программы.

Изначально, приложение было разработано и протестировано на локальном веб-сервере Apache. При тестировании были произведены проверки на баги, прописывались разного рода SQL-инъекции, производилось запросы на выявление ошибок. Позже сайт был перенесен на сервер Казахстанского хостинг провайдера. На хостинге используется веб-сервер Ngnix. При переносе сайта значительных проблем не наблюдалось, что говорит о правильной разработке основных моментов в программировании и стандартизированном использовании разных технологии.

Теперь, что касается внешнего вида главной страницы сайта, показанного на рисунке 3.1. В топе сайта слева расположился логотип сайта, который ведет на главную страницу сайта. Ниже слоган компании или предприятия (смотря для кого был создан сайт), который характеризует компанию и приветственно встречает каждого нового посетителя сайта. Выше логотипа расположил иконки социальных сетей, чтобы доброжелательные пользователи могли поделиться понравившейся информацией в соцсетях. Это в свою очередь привлекает новых пользователей, которые могут стать потенциальными клиентами. Далее по всей ширине сайта расположен горизонтальное меню навигации по сайту. При необходимости можно создать выпадающее меню для экономии места и группировании однотипных категории.

Далее сайт разбивается на две колонки. В левой колонке три интерактивные формы, которые при нажатии имеют свойства сворачиваться. Форма «Консультации» имеет интерфейс стандартного мессенджера, который позволяет в онлайн режиме консультировать посетителей сайта. На тот случай если менеджер не может ответить в онлайн режиме, на сайте предусмотрена форма «Задать вопрос». Методом пост вопрос приходит на сервер и сохраняется под определенным id в базе данных и при этом сразу же оповещает менеджера о новом поступившем вопросе. В правой колонке расположен динамичный фотослайдер. Слайдер использует технологию javascript.

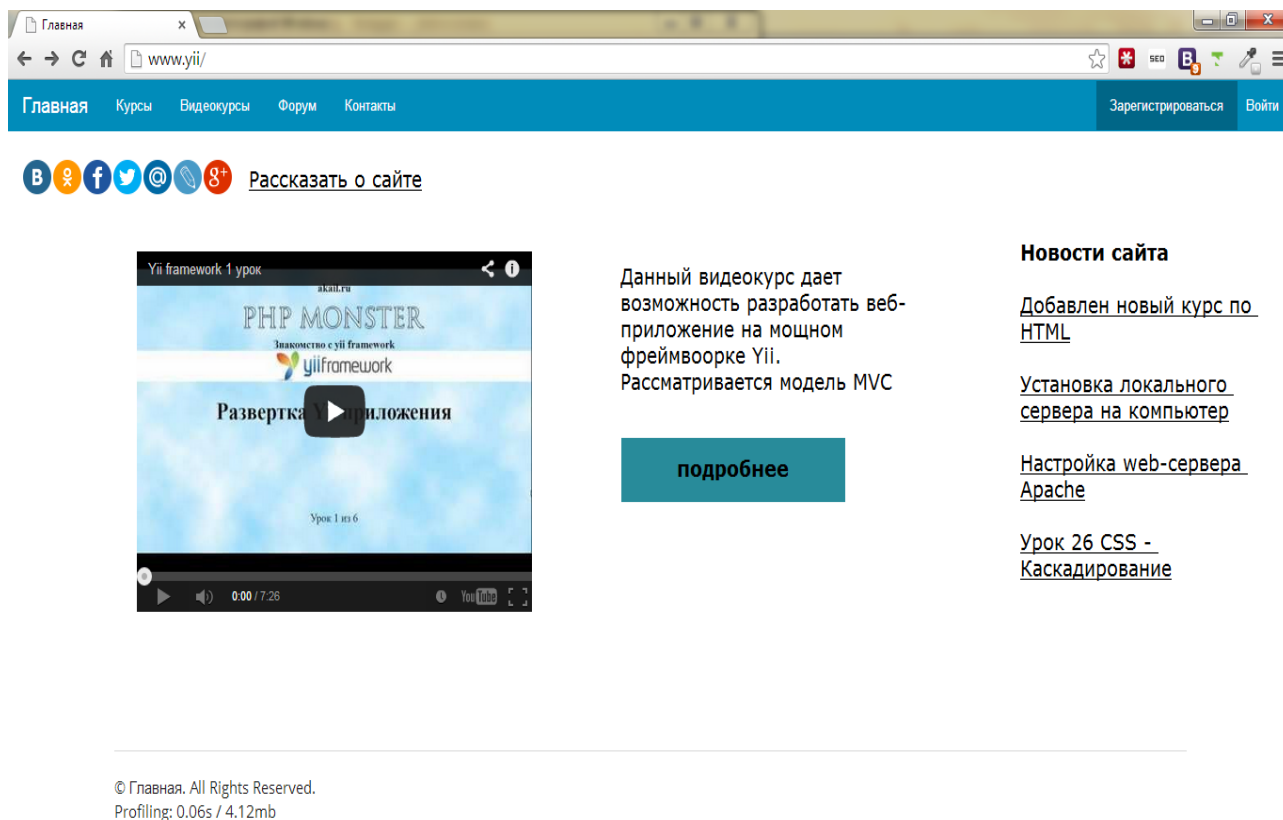


Рисунок 3.1 - Главная страница сайта

Для размещения рекламы или ссылки на важную страницу на сайте предусмотрены модули для баннера. Их на сайте три, но при увеличении спроса на баннеры его можно увеличить при этом не навредив остальному внешнему виду сайта.

Основным показателем авторитетности сайта является активность его пользователей. Исходя из этого, на сайте есть рубрика «отзывы», где посетители могут оставить свои комментарии относительно предприятия. Для фильтрации комментариев от лишнего спама и ненужных ссылок на сайте предусмотрена проверка пользователя капчей. Как показано на рисунке 3.1 футер сайта заполнен дополнительным меню (нижним меню).

Сайт спроектирован таким образом, что при переходе на другие страницы меняется только часть с контентом. Это позволяет минимизировать нагрузку на сервер и уменьшает трафик передачи через интернет. Во внутренних страницах исчезает видео с главной страницы. Однако, как показано на рисунке 3.2 во вкладке видеокурсы посетитель сайта может ознакомиться со списком доступных курсов. При нажатии кнопки «подробнее» посетитель переходит на страницу с непосредственным списком определенного видеокурса.

Специальный компонент Yii PHP Framework позволяет с легкостью интегрировать сайт с любыми сервисами. На данном примере Я показал интеграцию с сервисом youtube. Это дает возможность не переходя на сайт

сервиса проигрывать видео, при этом решается несколько важных проблем - минимизируется нагрузка на сервер, экономится место на сервере.

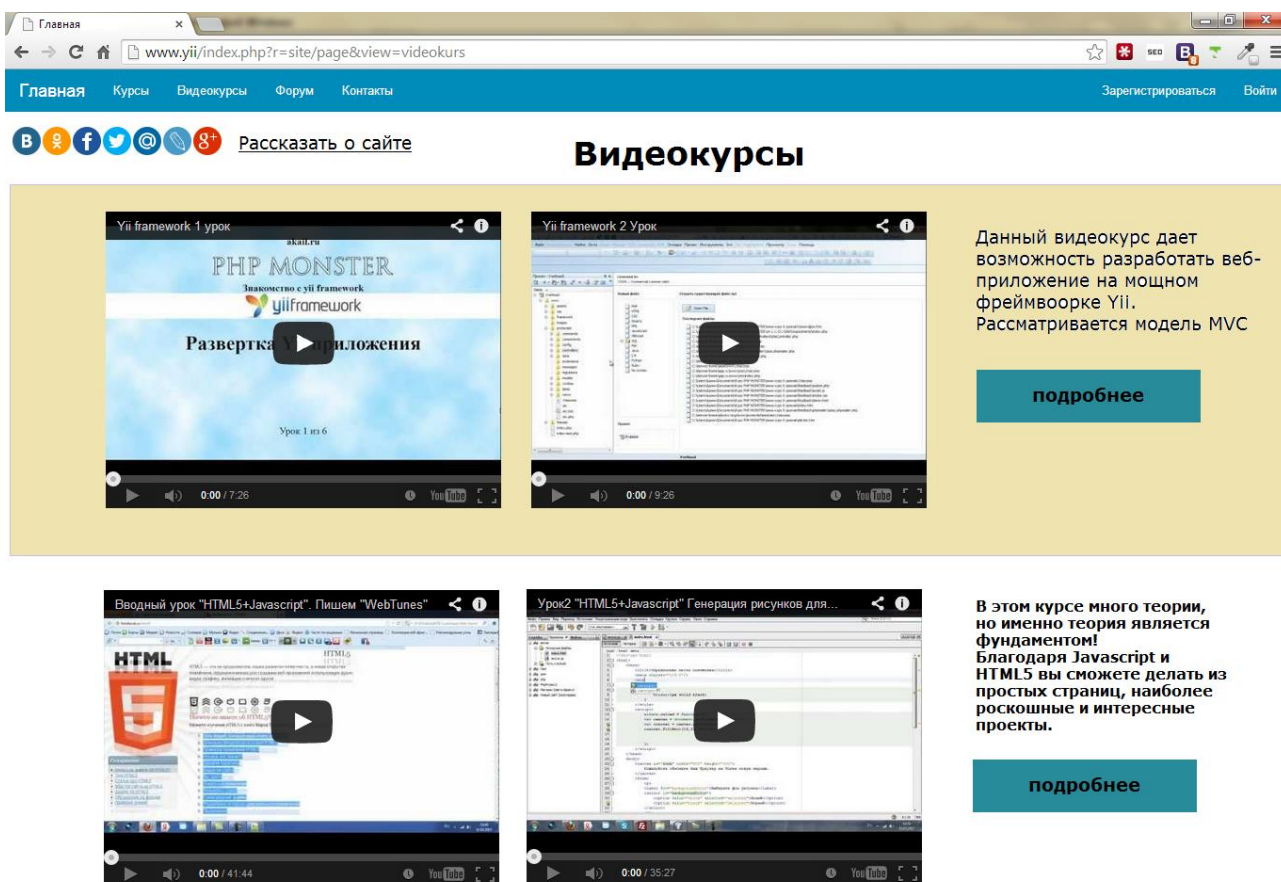


Рисунок 3.2 - Внутренняя страница «Видеокурсы»

На странице «Курсы» (Рисунок 3.3) открывается список всех доступных курсов на сайте. Краткое описание курса. На сайте представлены три курса: HTML, CSS, PHP. А также видеокурсы по администрированию Yii Framework и HTML/Javascript. Такое расположение элементов построено при принципах веб-дизайна UI, и соответствует стандартам. В каталоге еще нет таких обучающих сайтов. Предусмотрен также форум программистов, который позволяет собрать воедино всех начинающих и экспертов. На форуме можно обсуждать вопросы, проблемы. Стоит учесть и тот факт, что обсуждение или вопрос на форуме может создать любой зарегистрированный пользователь. Это дает возможность обмениваться мнениями и прорабатывать определенный подход к решению актуального вопроса. Еще одной функцией форума является то, что активность пользователей на форуме доказывает его актуальность и дает представление, что данный веб-ресурс имеет огромный спрос.

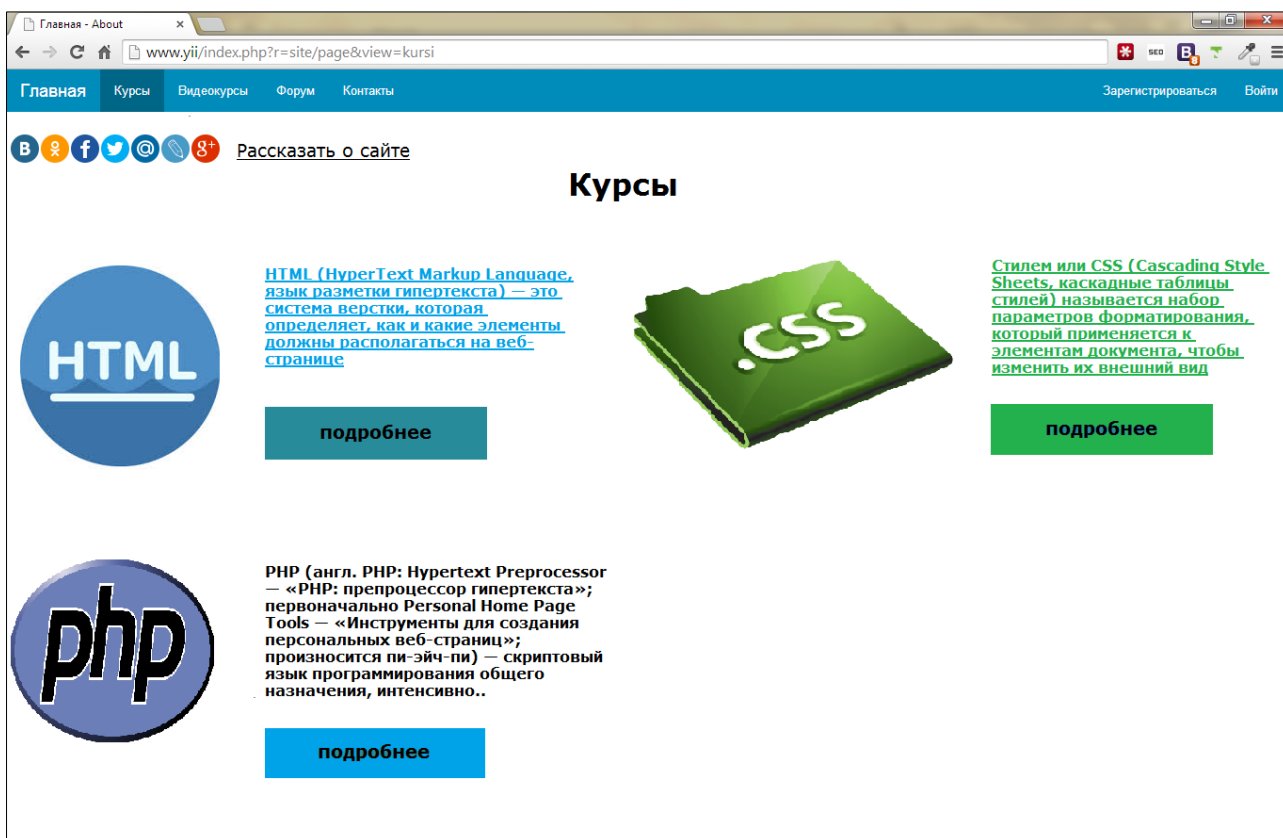


Рисунок 3.3 - Вкладка «Курсы»

В пункте меню «Контакты» представлена вся информация по которым можно связаться с администрацией сайта. Также там есть форма обратной связи для того, чтобы пользователи без проблем могли написать свои пожелания по каким-либо вопросам.

Не все страницы одинаково доступны всем пользователям на сайте. Например, чтобы ознакомиться с каким-либо курсом необходимо сначала зарегистрироваться на сайте, далее по созданному аккаунту произвести вход. Зарегистрироваться можно при переходе во вкладку «Зарегистрироваться». В этой вкладке перед пользователем открывается форма быстрой регистрации (Рисунок 3.4). Вся заполненная информация записывается в базу данных а пароль еще и шифруется. Одним из требований современного веб-ресурса является защита от спам роботов и ботов путем установления на сайте проверки капчей. Это и реализовано у нас на сайте. Капча генерирует произвольные символы, которые не в состоянии распознать не один робот.

Мощные компоненты валидации Yii PHP Framework не позволяют ввести не корректный адрес электронной почты, соответствие пароли к соображениям безопасности, наличие в имени неопознанных символов и т.д.

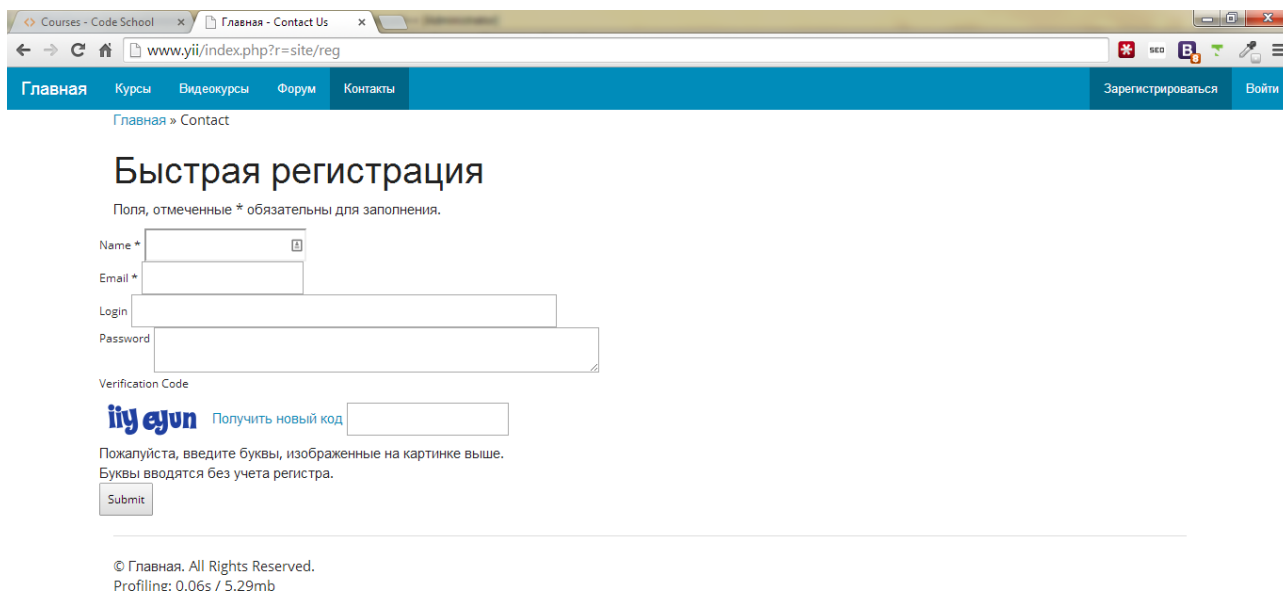


Рисунок 3.4 - Пункт меню «Зарегистрироваться»

Вход на сайт можно осуществить во вкладке «Войти» (Рисунок 3.5). Также в этой вкладке есть ссылка на страницу быстрой регистрации.

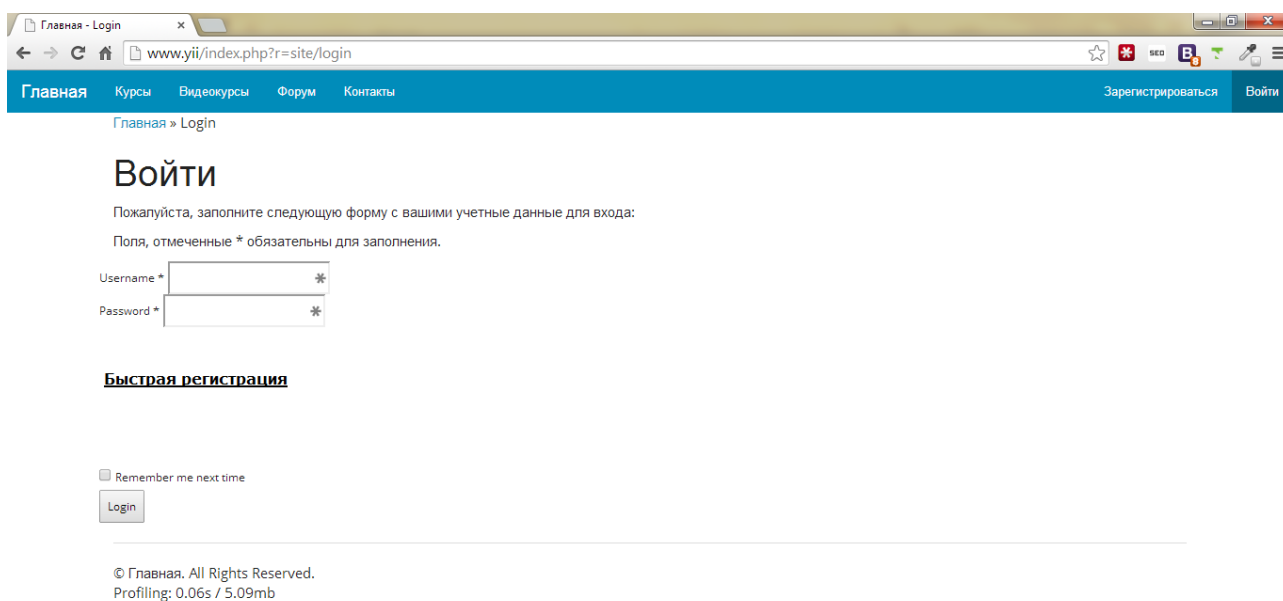
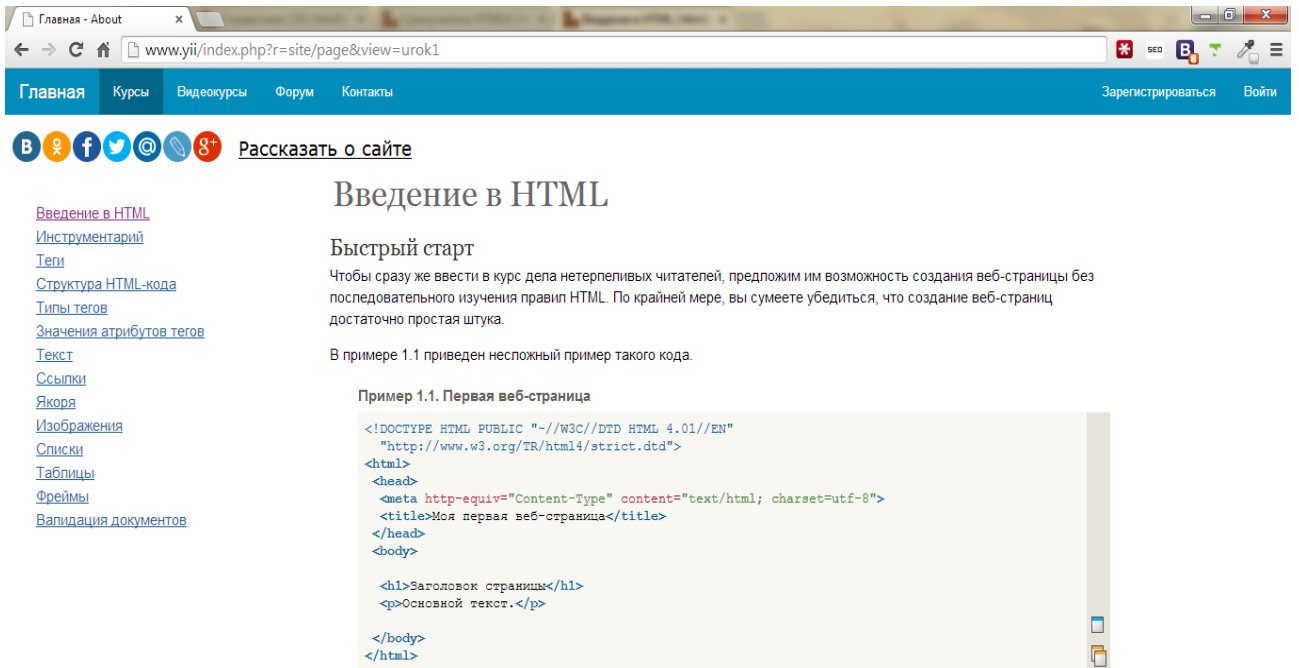


Рисунок 3.5 - Пункт меню «Войти»

На рисунках 3.6 и 3.7 представлены страницы с теоритическим материалом необходимым в процессе обучения. Наряду с теоритическим материалом там есть и практические задания. По каждому модулю предусмотрены задания на подкрепление освоенного материала.



Чтобы посмотреть результат примера в действии, проделайте следующие шаги.

1. В Windows откройте программу Блокнот (Пуск > Выполнить > набрать «notepad» или Пуск > Программы >

Рисунок 3.6 - Страница «Введение в HTML»

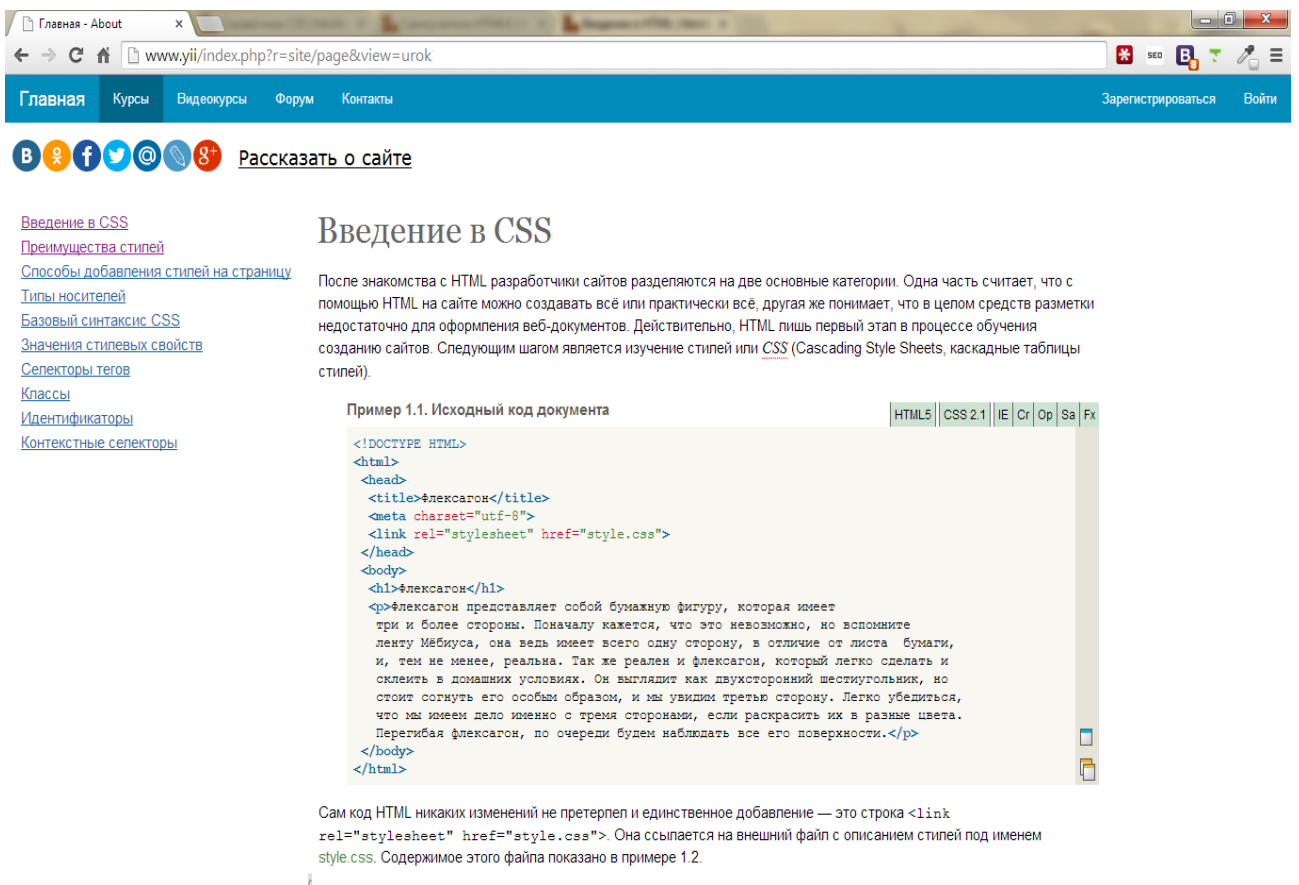


Рисунок 3.7 - «Страница введение в CSS»

3.3 Перенос сайта с локального сервера на виртуальный хостинг

Виртуальный хостинг (англ. shared hosting) - вид хостинга, при котором множество веб-сайтов расположено на одном веб-сервере. Это самый экономичный вид хостинга, подходящий для небольших проектов.

В качестве локального сервера я использовал программную оболочку «Денвер». Денвер - набор дистрибутивов предназначенных для создания и отладки веб-приложения не подключаясь к интернету. К стандартному набору денвера входит: веб-сервер Apache с поддержкой SSL, SSI; интерпретатор языков программирования PHP и PERL; СУБД MySQL; Панель phpMyAdmin для администрирования СУБД.

Процесс переноса сайта довольно трудоемкий и требует определенных знаний в современных технологиях. Во-первых, необходимо извлечь дампы базы данных. Как представлено на рисунке 3.8 в панели администрирования СУБД выбираем используемую базу данных и нажимаем вкладку «Экспорт». Далее все интуитивно понятно.

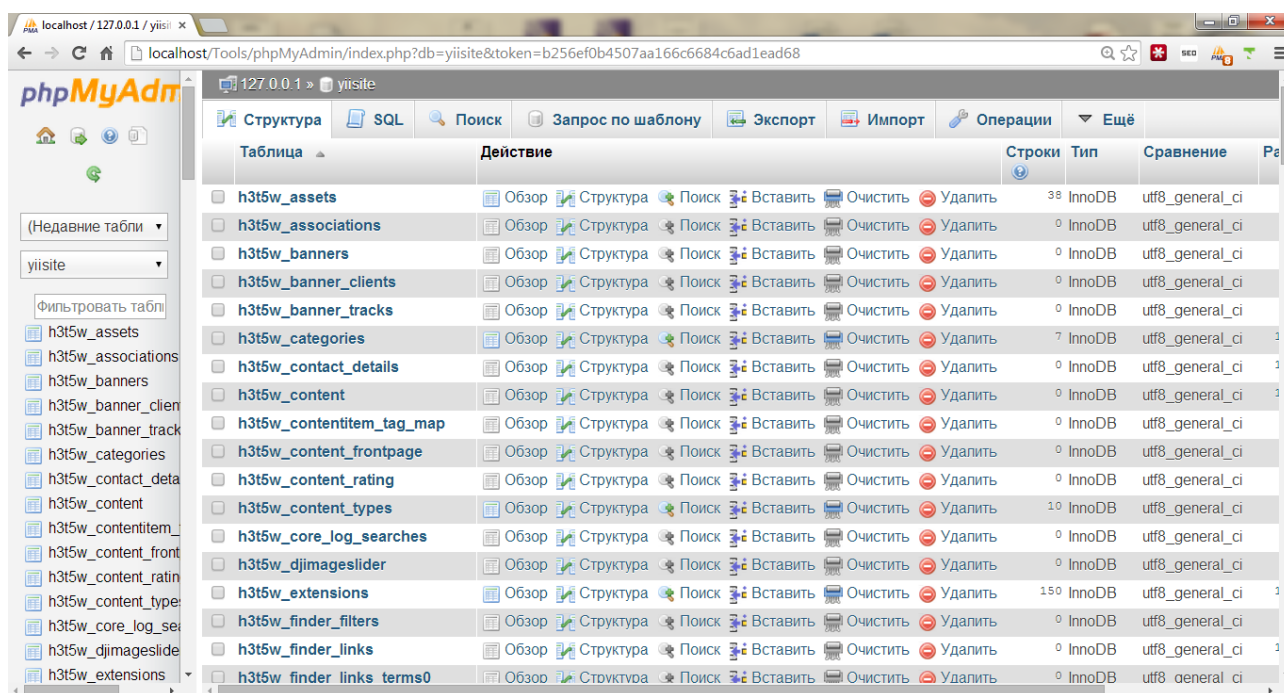


Рисунок 3.8 - Панель администрирования СУБД

В качестве хостинг провайдера мой выбор пал на Казахстанскую хостинг компанию PS (<http://www.ps.kz/>). Чтобы воспользоваться услугами компании необходимо зарегистрироваться на сайте, заполнить личную информацию в соответствии с удостоверением личности. После удачной регистрации, залогинившись нужно указать тарифный план и купить доменное имя. В панели управления хостингом (Рисунок 3.9) во вкладке «Сайты и домены» заходим в пункт «FTP доступ» и настраиваем ftp доступ к сайту.

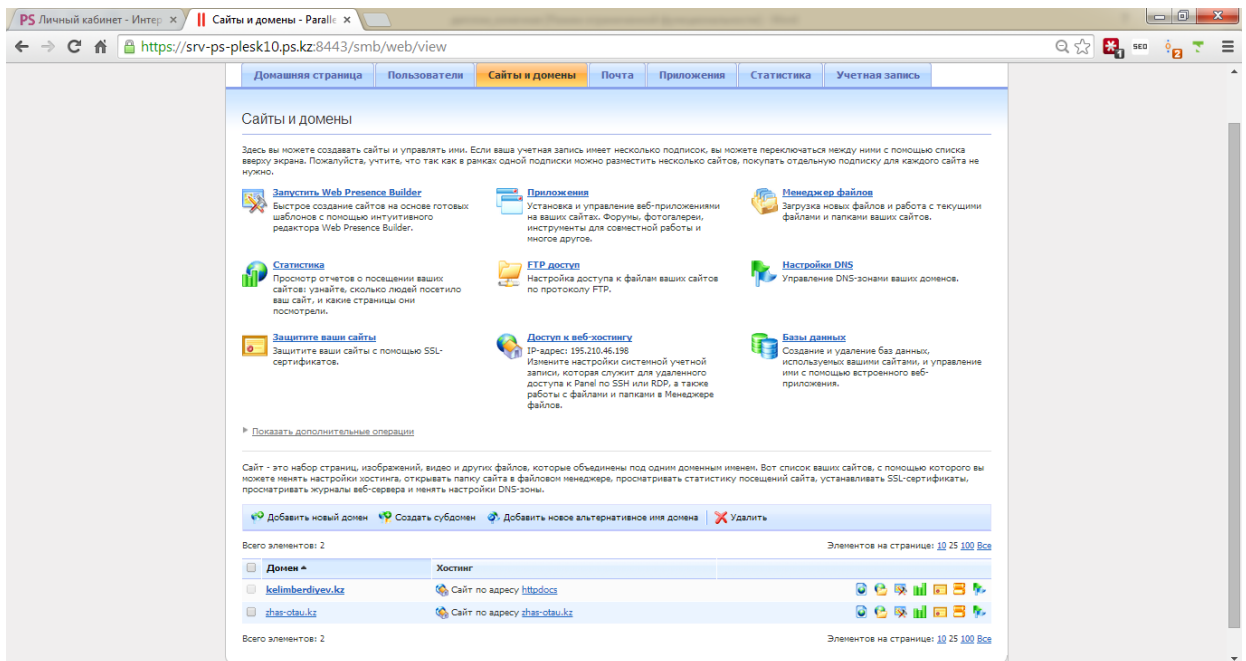


Рисунок 3.9 - Административная панель хостинга

Создаем учетную запись и логинимся через эту учетную запись с помощью программы Filezilla как показано на рисунке 3.10.

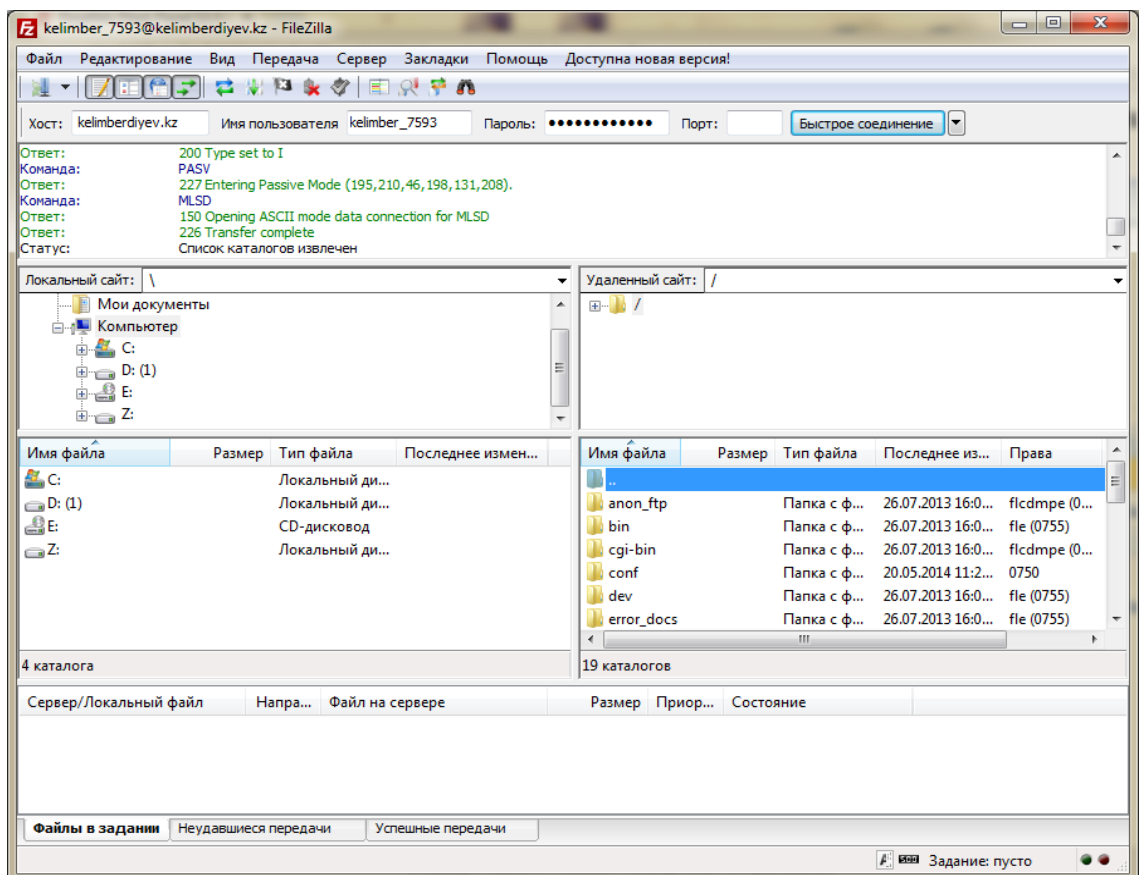


Рисунок 3.10 - Интерфейс FTP-клиента Filezilla

С помощью FTP-клиента Filezilla переносим основные файлы сайта в папку httpdocs. После переноса данных необходимо импортировать дампы базы данных на сервер баз данных на хостинге. Чтобы соединиться с базой данных необходимо в конфигурационном файле main.php поменять логин и пароль на актуальный. Настройки веб-сервера с помощью .htaccess на NGinx не действует, поэтому конфигурацию можно делать в административной панели разрешенные администратором.

Настройки DNS записи осуществляется в пункте меню «Настройки DNS» (Рисунки 3.11 и 3.12).

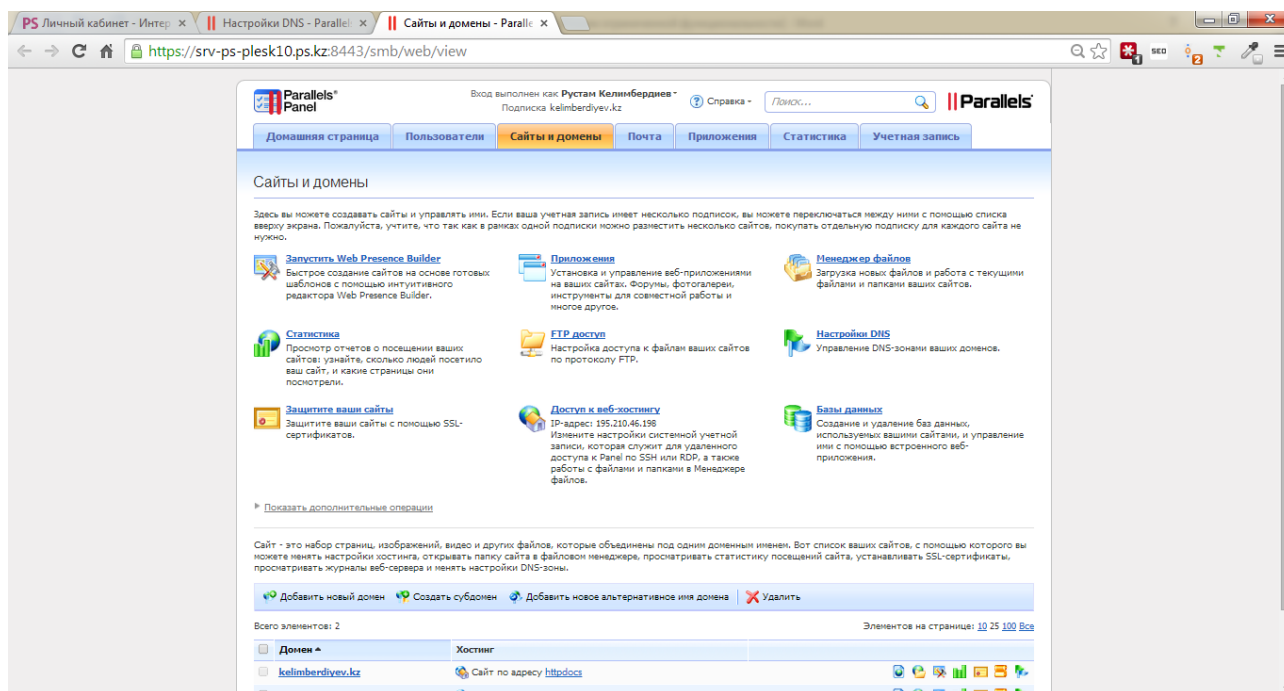


Рисунок 3.11 - Меню «Сайты и домены»

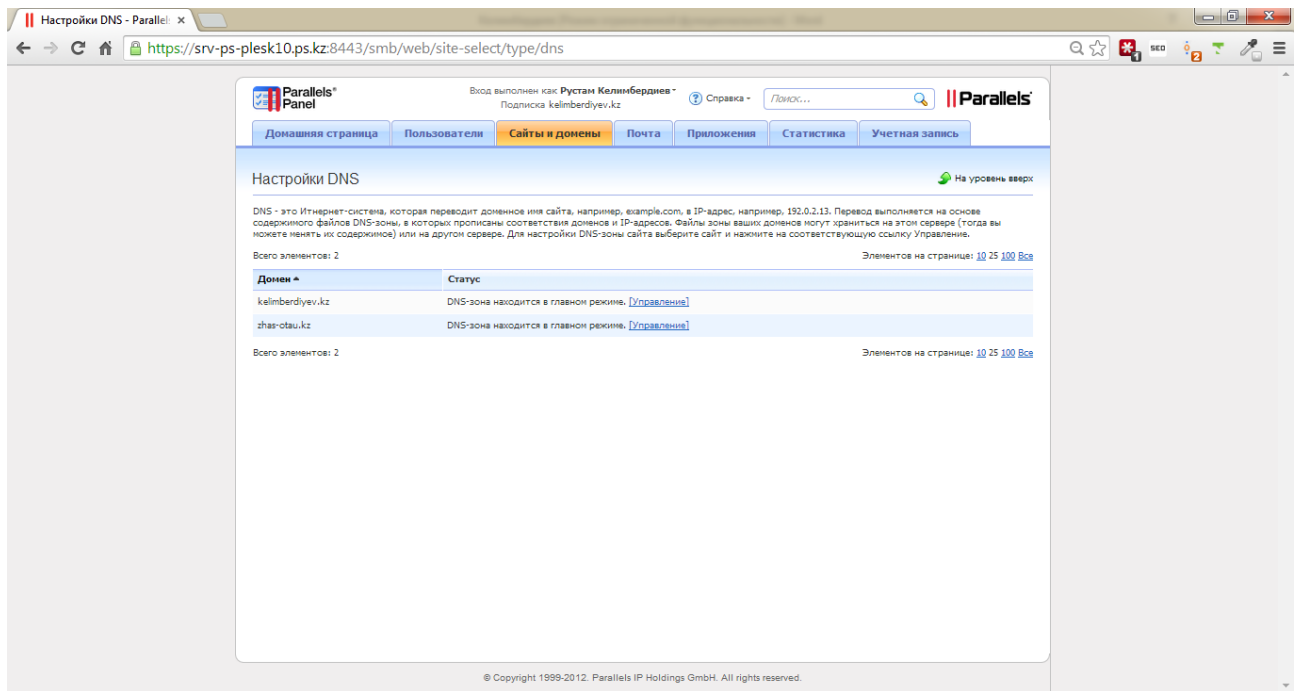


Рисунок 3.12 - Меню «Настройка DNS»

4 Технико-экономическое обоснование проекта

4.1 Описание работы и обоснование необходимости

Целью дипломного проекта является разработка Web-сайта, позволяющего эффективно и быстро увеличить продажи.

Разработка программного продукта (ПП) предусматривает проведение всех стадий проектирования (техническое задание, эскизный проект, технический проект, рабочий проект) и относится к 3-ей группе сложности. По степени новизны программный продукт относится к группе «Б».

Сегодня Интернет все больше и больше входит в нашу жизнь, становится ее необходимой составляющей. Каждый день сотни тысяч новых сайтов занимают свое место в безграничном web-пространстве. Ни печатная продукция, ни радио не могут подавать информацию в таком удобном, презентабельном виде, как это делает web-сайт. Используя гипертекст, мультимедиа и интерактивность, присущие только Интернету, можно добиться действительно эффективного воздействия на потенциальных клиентов и партнеров. Только web-сайт позволяет обрести безграничные возможности описания и рекламы продукции и услуг и разместить неограниченное количество информации.

Наличие сайта позволяет улучшить отношения с партнерами и клиентами компании. Web-сайт будет давать четкое представление о всех бизнес-процессах, протекающих в компании. Информация будет всегда актуальна и доступна круглосуточно. Вашим клиентам больше не придется звонить каждый раз, чтобы узнать, как изменились цены. Вы сможете оперативно обмениваться информацией с клиентами и партнерами и информировать их о различных акциях и проектах, например, о скидках, распродажах и других специальных предложениях. Стимулировать их к контактам по телефону и электронной почте.

В данный момент веб сайт служит не видимым связующим мостом в отношениях продавца и покупателя. Поэтому создание сайта для эффективной торговли скорее является необходимостью.

4.2 Трудовые ресурсы, используемые в работе

В работе задействованы:

- руководитель - постановка задачи, разработка основных бизнес правил для работы проекта;
- Инженер-разработчик - разработка сайта офис регистратора, проектирование базы данных, сопровождение системы;
- Консультанты по экономической части и по части ОБЖД.

Общее количество сотрудников и их заработная плата представлены в таблице 4.1.

Т а б л и ц а 4.1 - Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	80000
Консультант по части “Экономика”	1	60000
Консультант по части “Безопасность жизнедеятельности”	1	60000
Инженер-разработчик	1	120000
Итого	4	320000

4.3 Расчет стоимости работы по проектированию и разработке

Разработка многомодульного программного продукта - сложный и трудоемкий процесс, требующий наряду с интеллектуальными, техническими затратами и финансовых затрат. Поэтому необходимым является произведение расчета стоимости этой разработки. Затраты на разработку данного программного комплекса определяется по формуле

$$C = \Phi OT + C_H + A + \mathcal{E} + C_{np} + H \quad (4.1)$$

где ΦOT - фонд оплаты труда;

C_H - социальный налог;

A - амортизационные отчисления;

\mathcal{E} - затраты на электроэнергию;

C_{np} - прочие расходы;

H - накладные расходы.

Необходимый фонд оплаты труда рассчитывается по формуле

$$\Phi OT = Z_{осн} + Z_{доп} \quad (4.2)$$

где $Z_{осн}$ - основная заработная плата;

$Z_{доп}$ - дополнительная заработная плата.

Труд сотрудников института, задействованных в работе, оплачивается согласно положению о заработной плате АУЭС, труд программиста-разработчика принят условно, на договорной основе в размере 120000 тенге.

Основная заработная плата рассчитывается по формуле

$$Z_{осн} = Z_{РУК} + Z_{ИНЖ} + Z_{РУК.Э.} + Z_{РУК.БЖ}, \quad (4.3)$$

$$Z_x = T_x * H_x \quad (4.4)$$

где x - инженер, руководитель;

T - время работы;

H - почасовая оплата труда.

Базовый показатель для определения составляющих затрат труда вычисляется по формуле

$$Q = q * c \quad (4.5)$$

где Q - условное число команд;

q = 1500 - коэффициент, учитывающий условное число команд в зависимости от типа задачи;

c = 1,15 - коэффициент, учитывающий новизну и сложность программы.

$$Q = 1500 * 1,15 = 1725$$

Выбрать значение коэффициента q можно из таблицы 4.2.

Т а б л и ц а 4.2 - Условное число команд в зависимости от типа задачи

Тип задачи	Пределы изменений коэффициента
Задачи учета	От 1400 до 1500
Задачи оперативного управления	От 1500 до 1700
Задачи планирования	от 3000 до 3500
Много вариантные задачи	от 4500 до 5000
Комплексные задачи	от 5000 до 5500

Программные продукты по степени новизны может быть отнесены к одной из 4-х групп:

- группа А - разработка принципиально новых задач;
- группа Б - разработка оригинальных программ;
- группа В - разработка программ с использованием типовых решений;
- группа Г - разовая типовая задача.

Коэффициент c определяется из таблицы 4.3 на пересечении групп сложности и степени новизны. Чем сложнее задача, соответственно и больше коэффициент расчета трудоемкости. Также основным фактором расчета является степень новизны программного продукта. Таким образом определим коэффициент расчета трудоемкости.

Т а б л и ц а 4.3 - Коэффициенты расчета трудоемкости

Язык программирования	Группа сложности	Степень новизны				Коэффициент <i>B</i>
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5
Низкого уровня	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

При разработке многомодульного программного продукта используется техническое задание, согласно которому выполнение работ происходит последовательно по заданным пунктам, с учетом сроков их исполнения. График выполнения работ по разработке программного продукта представлен в таблице 4.4.

Т а б л и ц а 4.4 - График выполненных работ по разработке проекта

Код работы	Наименование работы	Ожидаемая длительность, дни	Обозначение
1	Подготовка описания задачи	2	$T_{ПЗ}$
2	Описание задачи	3	$T_{ОЗ}$
3	Разработка алгоритма	5	T_A
4	Установка и запуск локального сервера	3	$T_{ЛС}$
5	Проектирование и создание базы данных	5	$T_{БД}$
6	Разработка модулей приложения	6	T_M
7	Разработка и создание интерфейса системы	5	$T_{ИН}$
8	Разработка основной части программы	6	$T_{ОСН}$
9	Тестирование базы данных и основных модулей с целью выявления ошибок	2	$T_{ТЕСТ}$
10	Составление технической документации	3	$T_{ТЕХ}$
11	Подготовка раздела “Экономика”	9	$T_Э$
12	Подготовка раздела “Безопасность жизнедеятельности”	9	$T_{БЖ}$

Так как участники, задействованные в проекте, работают в различные промежутки времени, в течение которого реализуется проект, необходимо произвести расчет дневной и почасовой оплаты труда.

Заработная плата каждого работника за один рабочий день рассчитывается по формуле

$$D = \frac{O}{n} \quad (4.6)$$

где O - оклад работника в тенге;

n - количество дней в рабочем месяце(это 24 дня - шестидневная рабочая неделя):

- для руководителя

$$D = \frac{80000}{24} = 3333 \text{ тенге/день};$$

- для консультанта по части “Экономика”

$$D = \frac{60000}{24} = 2500 \text{ тенге/день};$$

- для консультанта по части “БЖД”

$$D = \frac{60000}{24} = 2500 \text{ тенге/день};$$

- для инженера разработчика

$$D = \frac{120000}{24} = 5000 \text{ тенге/день.}$$

Заработная плата за один час рассчитывается по формуле

$$H = \frac{D}{z} \quad (4.7)$$

где D - заработная плата работника за один рабочий день;

z - количество часов рабочего дня (8 часов)

- для руководителя

$$H = \frac{3333}{8} = 416,63 \text{ тенге/час};$$

- для консультанта по части “Экономика”

$$H = \frac{3333}{8} = 416,63 \text{ тенге/час};$$

- для консультанта по части “БЖД”

$$H = \frac{2500}{8} = 312,5 \text{ тенге/час};$$

- для инженера разработчика

$$H = \frac{5000}{8} = 625 \text{ тенге/час.}$$

Время рассчитывается в человеко-часах, причем $T_{ПЗ}$ берется по фактически отработанному времени, а время остальных этапов определяется расчётно, по условному числу команд Q .

Определяем время, затраченное на каждый этап создания программного продукта

1) $T_{ПЗ}$ (время на подготовку описания задачи), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ПЗ} = 32 \text{ чел/час};$$

2) $T_{ОЗ}$ (время на описание задачи) определяется по формуле

$$T_{ОЗ} = \frac{Q * B}{50 * K} \quad (4.8)$$

где $B=1,2$ - коэффициент учета изменений задачи, коэффициент B в зависимости от сложности задачи и числа изменений выбирается в интервале от 1,2 до 1,5 (Таблица 4.5);

$K=1,2$ - коэффициент, учитывающий квалификацию программиста.

$$T_{ОЗ} = \frac{1725 * 1,5}{50 * 1,2} = 43,125 \text{ чел/час};$$

3) T_a (время на разработку алгоритма) рассчитываем по формуле

$$T_A = \frac{Q}{50 * K} \quad (4.9)$$

$$T_A = \frac{1725}{50 * 1,2} = 28,75 \text{ чел/час};$$

Т а б л и ц а 4.5 - Коэффициенты квалификации программиста

Опыт работы	Коэффициент квалификации
До двух лет	0.8
2-3 года	1
3-5 лет	1,1 - 1,2
5-7 лет	1,3 - 1,4
более 7 лет	1,5 - 1,6

4) $T_{\text{ЛС}}$ (время на установку и запуск локального сервера) определяется аналогично T_A

$$T_{\text{ЛС}} = 28,75 \text{ чел/час};$$

5) $T_{\text{БД}}$ (время на проектирование и создание БД) определяется по формуле

$$T_{\text{БД}} = \frac{3000 * 1,5}{50 * 1,2} = 75 \text{ чел/час};$$

6) T_M (время на разработку модулей приложения) определяется по формуле

$$T_M = \frac{3000 * 1,5}{50 * 1,2} = 75 \text{ чел/час};$$

7) $T_{\text{ИН}}$ (время отладки и тестирования программы) определяется по формуле

$$T_{\text{ИН}} = \frac{1600 * 1,5}{50 * 1,2} = 40 \text{ чел/час};$$

8) $T_{\text{ОСН}}$ (время на разработку основной части), определяется по формуле

$$T_{\text{ОСН}} = \frac{4000 * 1,5}{50 * 1,2} = 100 \text{ чел/час}$$

9) $T_{ТЕСТ}$ (время на тестирование), определяется по формуле

$$T_{ТЕСТ} = \frac{3000 * 1,5}{50 * 1,2} = 75 \text{ чел/час}$$

10) $T_{ТЕХ}$ (время на составление технической документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ТЕХ} = 32 \text{ чел/час}$$

11) $T_{Э}, T_{БЖ}$ (время на подготовку разделов Экономика и БЖ), берется по факту и составляет (от 7-и до 10-ти дней по 8 часов)

$$T_{Э} = T_{БЖ} = 50 \text{ чел/час}$$

т.к. в проекте задействованы руководители, заработная плата которых отличается, то необходимо рассчитать время $T_{РВК}$. Время берется среднее и оно равно

$$T_{Р} = 80 \text{ чел/час},$$

$$T_{ЭК} = T_{БЖД} = 22 \text{ чел/час}$$

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле

$$T_{СТ} = T_{ПЗ} + T_{ОЗ} + T_{А} + T_{ЛС} + T_{БД} + T_{М} + T_{ИН} + T_{ОСН} + T_{ТЕСТ} + T_{ТЕХ} + T_{Э} + T_{БЖ} \quad (4.10)$$

$$T_{СТ} = 32 + 43,13 + 28,75 + 28,75 + 75 + 75 + 40 + 100 + 75 + 32 + 100 = 630 \text{ чел/час}$$

Таким образом, суммарная основная заработная плата по формулам 4.3 и 4.4 составит

$$З_{ОСН} = 630 * 625 + 2 * 312,5 * 22 + 80 * 416,63 = 440830 \text{ тенге}$$

Дополнительная заработная плата в среднем определяется в размере 10% от основной заработной платы и рассчитывается по формуле

$$З_{доп} = З_{осн} * 10\% \quad (4.11)$$

$$З_{доп} = 440830 * 10\% = 44083 \text{ тенге}$$

Общий фонд оплаты труда согласно формуле 4.2 составит:

$$\text{ФОТ} = 440830 + 44083 = 484913 \text{ тенге}$$

Социальный налог составляет 11% от ФОТ и рассчитывается по формуле

$$C_H = (\text{ФОТ} - \text{ПО}) * 11\% \quad (4.12)$$

где ПО (отчисления в пенсионный фонд) составляют 10% от ФОТ и рассчитываются по формуле

$$\text{ПО} = \text{ФОТ} * 10\% \quad (4.13)$$

Размер пенсионных отчислений согласно формуле 4.13 составляет

$$\text{ПО} = 484913 * 10\% = 48491 \text{ тенге}$$

Отчисления по социальному налогу согласно формуле 4.12 составят

$$C_H = (440830 - 48491) * 11\% = 43157 \text{ тенге}$$

4.4 Расчет затрат на амортизацию

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{C_{обор} * H_A * N}{100 * 12 * t} \quad (4.14)$$

где H_A - норма амортизации;

$C_{обор}$ - первоначальная стоимость оборудования;

N - количество дней на выполнение работ;

t - общее время использования персонального компьютера.

Норма амортизации H_A , рассчитывается по формуле

$$H_A = \frac{C_{\text{ОБОР}} - C_{\text{ЛИКВ}}}{T_{\text{НОРМ}} * C_{\text{ОБОР}}} * 100\% \quad (4.15)$$

где $C_{\text{ЛИКВ}} = 0,05 * 150000 = 7879,5$ - ликвидационная стоимость, составляет 5% от стоимости оборудования;

$T_{\text{НОРМ}}$ - нормативный срок службы (для персонального компьютера - 4 года).

Общее время использования персонального компьютера учитывает лишь время работы на компьютере (таблица 4.6) и рассчитывается по формуле

$$t = T_{\text{ЛС}} + T_{\text{БД}} + T_{\text{М}} + T_{\text{ИН}} + T_{\text{ОСН}} + T_{\text{ТЕСТ}} \quad (4.16)$$

$$T_{\text{СТ}} = 28,75 + 75 + 75 + 40 + 100 + 75 = 394 \text{ чел/час}$$

Т а б л и ц а 4.6 - Используемое оборудование

Наименование	Модель	Стоимость, тенге
Процессор	Intel Dual Core i5-2450M 2.5 GHz	115 000
Монитор	Sharp PC-1401	25 000
Клавиатура	Genius EgroMedia 700	2000
Компьютерная мышь	Genius XScroll	1000
Модем	TP-Link TL-WR741ND	7000
Итого		150000

Цены на ПО приведены без учета НДС.

$$H_A = \frac{150000 - 7879,5}{4 * 150000} * 100 = 23,7\%$$

$$A = \frac{150000 * 23,7 * 670}{100 * 12 * 49} = 40508 \text{ тенге}$$

Затраты на электроэнергию вычисляется по формуле

$$\mathcal{E} = M * K_3 * T * C_{\text{кВт·ч}} \quad (4.17)$$

где M - мощность ЭВМ (600 Вт=0,6 кВт);

K_3 - коэффициент загрузки (0.8);

$C_{\text{кВт·ч}}$ - 14,935 тг- стоимость 1 кВт-час электроэнергии;

T - время работы.

$$\Xi = 0,6 * 0,8 * 14,935 * 394 = 2825 \text{ тенге}$$

Расходы на материалы и комплектующие, используемые в процессе написания программного продукта ($C_{мик}$), а также затраты на техническое обслуживание и ремонт ($C_{то}$) составляют, соответственно, 2,06% и 2,266% от стоимости оборудования - формулы (4.16 - 4.17)

$$C_{мик} = 0,02 * C_{обор}, \quad (4.18)$$

$$C_{то} = 0,023 * C_{обор} \quad (4.19)$$

$$C_{мик} = 0,02 * 150000 = 3090 \text{ тенге,}$$

$$C_{то} = 0,023 * 150000 = 3399 \text{ тенге}$$

$$C_{пр} = C_{мик} * C_{то} \quad (4.20)$$

$$C_{пр} = 3090 + 3399 = 6489 \text{ тенге}$$

Накладные расходы, связанные с управлением и обслуживанием, содержанием и эксплуатацией оборудования, и прочими дополнительными затратами на обеспечение процессов производства и обращения, составляют 50% от всех затрат, вычисляются по формуле:

$$H = (\Phi OT + C_H + A + \Xi + C_{пр}) * 0,5 \quad (4.21)$$

$$H = (484913 + 43157 + 40508 + 2825 + 6489) * 0,4 = 231157 \text{ тенге}$$

Таким образом, затраты на разработку данного программного комплекса согласно формуле 4.1 равны

$$C = 484913 + 43157 + 40508 + 2825 + 6489 + 231157 = 809049 \text{ тенге}$$

Цена реализации программного продукта рассчитывается по формуле

$$ЧП = C * 15\% \quad (4.22)$$

$$ЧП = 809049 * 15\% = 121358 \text{ тенге}$$

$$P = C + ЧП \quad (4.23)$$

где Р - цена реализации продукта.

$$P = 809049 + 121358 = 930407 \text{ тенге}$$

Тогда, НДС от программного продукта составит

$$НДС = ЧП * 12\% \quad (4.24)$$

$$НДС = 121358 * 12\% = 14563 \text{ тенге}$$

Сводные результаты расчета себестоимости разработки сайта и ее структура представлены в таблице 4.7 и на рисунке 4.1.

Т а б л и ц а 4.7 - Сводные результаты расчета себестоимости веб-сайта

Статья затрат	Сумма, тенге
Фонд оплаты труда	484913
Затраты на электроэнергию	2825
Социальный налог	43157
Накладные расходы	231157
Прочие расходы	6489
Амортизационные отчисления	40508
Итого	809049

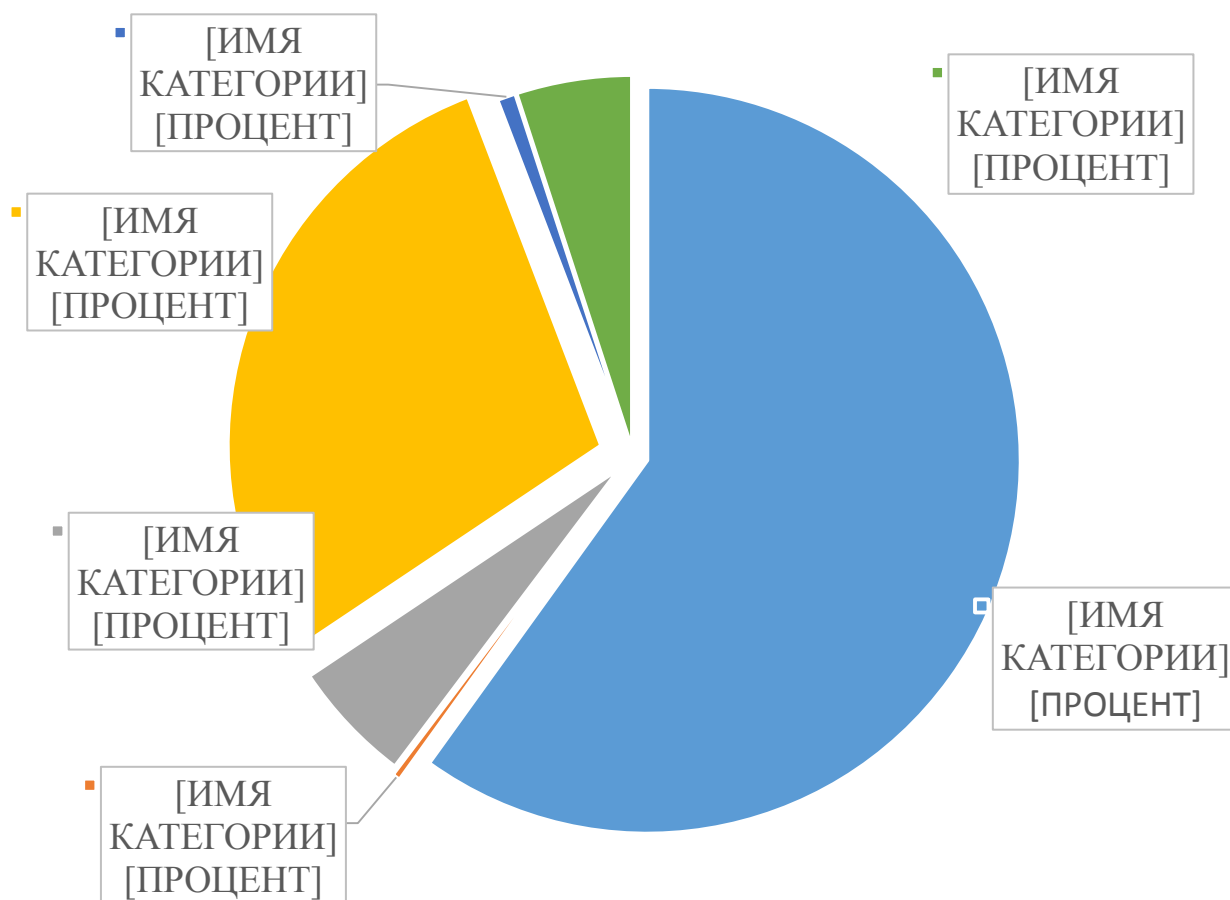


Рисунок 4.1 Диаграмма структуры себестоимости программного обеспечения

Вывод

Разработка многомодульного программного продукта - сложный и трудоемкий процесс, требующий наряду с интеллектуальными, техническими затратами и финансовых затрат.

Себестоимость разработки сайта составила 809049 тенге, наибольшую долю в себестоимости составляют затраты на оплату труда 484913 тенге (60%). Цена реализации составила 930407 тенге. Средняя цена на рынке на аналогичные продукты составляет от 550000 до 1 300000 тенге. Цена на продукт находится в допустимых пределах, что говорит об экономической эффективности продукта.

5. Технические решения безопасности труда

5.1 Анализ потенциально опасных и вредных производственных факторов

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

В помещении лаборатории на программиста могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

К химически опасным факторам, постоянно действующим на программиста относятся следующие: возникновение, в результате ионизации воздуха при работе компьютера, активных частиц.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на оператора в течение его рабочей смены можно отнести следующие:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Далее более подробно рассмотрены опасные и вредные факторы, воздействующие на программиста, возникшие в связи с разработкой данной системы.

5.1.1 Микроклимат рабочей зоны программиста

Микроклимат производственных помещений - это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Для создания и автоматического поддержания в лаборатории независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

5.1.2 Воздействие шума на программиста. Защита от шума

В помещениях с низким уровнем общего шума, каким является лаборатория где работает программист, источниками шумовых помех могут стать вентиляционные установки, кондиционеры или периферийное оборудование для ЭВМ (плоттеры, принтеры и др.). Длительное воздействие этих шумов отрицательно сказывается на эмоциональном состоянии персонала.

Согласно ГОСТ 12.1.003-76 [20] эквивалентный уровень звука не должен превышать 50 дБА. Для того, чтобы добиться этого уровня шума рекомендуется применять звукопоглощающее покрытие стен.

В качестве мер по снижению шума можно предложить следующее:

- облицовка потолка и стен звукопоглощающим материалом (снижает шум на 6 - 8 дБ);
- экранирование рабочего места (постановкой перегородок, диафрагм);
- установка в компьютерных помещениях оборудования, производящего минимальный шум;
- рациональная планировка помещения.

Поэтому я предлагаю для уменьшения шума в лаборатории использовать вместо матричного принтера, который производит много шума, более тихий - лазерный принтер.

Защиту от шума следует выполнять в соответствии с ГОСТ 12.1.003-83, а звукоизоляция ограждающих конструкций должна отвечать требованиям главы СНиП 11-12-77 [21] “Защита от шума. Нормы проектирования”.

5.1.3 Опасность повышенного уровня напряженности электромагнитного поля

Электромагнитные поля, характеризующиеся напряженностями электрических и магнитных полей, наиболее вредны для организма человек. Основным источником этих проблем, связанных с охраной здоровья людей, использующих в своей работе автоматизированные информационные системы на основе персональных компьютеров, являются дисплеи (мониторы), особенно дисплеи с электронно-лучевыми трубками. Они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье программиста.

ПЭВМ являются источниками таких излучений как:

- ультрафиолетового 200 - 400 нм;
- видимого 400 - 700 нм,
- ближнего инфракрасного 700 - 1050 нм;
- радиочастотного 3 кГц - 30 МГц;
- электростатических полей.

Может возникнуть опасность по уровням напряженности электромагнитного поля. На расстоянии 5 - 10 см от экрана и корпуса монитора уровни напряженности могут достигать 140 В/м по электрической составляющей, что значительно превышает допустимые значения СанПиН 2.2.2. 542-96 [22]. Предельно допустимые значения характеристик ЭМП указана в таблице 5.1.

Т а б л и ц а 5.1 - Предельно допустимые значения характеристик ЭМП

Наименование параметров	Допустимое Значение
Напряженность электромагнитного поля по электрической составляющей на расстоянии 50 см от поверхности видеомонитора	10 В/м
Напряженность электромагнитного поля по магнитной составляющей на расстоянии 50 см от поверхности видеомонитора	0,3 А/м
Напряженность электростатического поля не должно превышать: для взрослых пользователей	20 кВ/м
Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:	
в диапазоне частот 5 Гц - 2 кГц;	25 В/м
в диапазоне частот 2 - 400 кГц	2,5 В/м
Плотность магнитного потока должна быть не более:	
в диапазоне частот 5 Гц - 2 кГц;	250нТл
в диапазоне частот 2 - 400 кГц	25 нТл
Поверхностный электростатический потенциал не должен превышать	500 В

Ультрафиолетовое излучение полезно в небольших количествах, но в больших дозах приводит к дерматиту кожи, головной боли, рези в глазах. Инфракрасное излучение приводит к перегреву тканей человека (особенно хрусталика глаза), повышению температуры тела. Уровни напряженности электростатических полей должны составлять не более 20 кВ/м. Поверхностный электростатический потенциал не должен превышать 500 В. При повышенном уровне напряженности полей следует сократить время

работы за компьютером, делать пятнадцатиминутные перерывы в течение полутора часов работы и, конечно же, применять защитные экраны. Защитный экран, изготовляемый из мелкой сетки или стекла, собирает на себе электростатический заряд. Для снятия заряда экран монитора заземляют.

Для предупреждения внедрения опасной техники все дисплеи должны проходить испытания на соответствие требованиям безопасности (например международные стандарты MRP 2, TCO 99).

Так как работа программиста по виду трудовой деятельности относится к группе В - творческая работа в режиме диалога с ЭВМ, а по напряженности работы ко II категории тяжести (СанПиН 2.2.2.542-96), я предлагаю сократить время работы за компьютером, делать перерывы, суммарное время которых должно составлять 50 минут при 8-ми часовой смене.

5.2 Расчет искусственного освещения помещения

5.2.1 Разработка мероприятий по улучшению условий труда

Рациональное освещение в помещении, предназначенном для работы с ПЭВМ создается при наличии как естественного, так и искусственного освещения.

Недостаточное освещение приводит к сильному напряжению глаз, быстрой утомляемости, близорукости, снижению качества работы, увеличению брака. Яркое освещение раздражает сетчатку глаза, ослепляет, глаза быстро устают, растёт производственный травматизм.

Для создания нормальных условий, на рабочем месте проводят нормирование освещенности в зависимости от размеров объекта различения, контраста объекта с фоном. Определение нормированной освещенности ведется по разрядам и подразрядам выполняемых работ. Для работ, выполняемых разработчиком, отводится четвертый разряд и подразряд "Б". Минимальное значение нормированной освещенности согласно СНиП 23-05-95 $E_{\min}=200$ Лк для общей системы освещения.

Рассчитаем общее освещение офисного помещения размером 6х5.

Нормируемая освещенность - 300 лк.

В помещении принята система общего освещения люминесцентными лампами ЛБ (белого цвета), мощностью 40 Вт и световым потоком $\Phi_{л} = 3120$ лм, диаметром 40 мм. и длиной со штырьками 1213.6 мм. Высота рабочей поверхности $h_p = 0,8$ м.

Индекс помещения, определяется соотношением размеров освещаемого помещения

$$i = \frac{A \cdot B}{h(A + B)} \quad (5.1)$$

Определим необходимое расстояние между светильниками

$$L = \lambda \cdot h \quad (5.2)$$

где $\lambda = 1,2 \div 1,4$

Высота подвеса светильников над рабочей поверхностью

$$h = H - h_{\text{раб}} \quad (5.3)$$

$$h = 3 - 0,8 = 2,2 \text{ м}$$

По этим данным находим, что необходимое расстояние между светильниками равно

$$L = 2,2 \cdot 1,4 = 3,08 \approx 3 \text{ м}$$

Тогда индекс помещения определим, как

$$i = \frac{6 \cdot 5}{2,2 \cdot (6 + 5)} = 1,326$$

В качестве светильника возьмем ЛСП-02 рассчитанный на две лампы мощностью 40 Вт, диаметром 40 мм и длиной со штырьками 1213,6 мм. Световой поток лампы ЛБ 40 Вт составляет 3120 лм., световой поток, излучаемый светильником равен

$$\Phi_{\text{св}} = \Phi_{\text{л}} \cdot 2 = 3120 \cdot 2 = 6240 \text{ лм}$$

Определим число светильников по формуле

$$N = \frac{E \cdot K_3 \cdot S \cdot z}{n \cdot \Phi_{\text{л}} \cdot \eta} \quad (5.4)$$

где S - площадь помещения, $S = 30 \text{ м}^2$;

K_3 - коэффициент запаса, $K_3 = 1,5$;

E - заданная минимальная освещенность, $E = 300 \text{ лк}$;

Z - коэффициент неравномерности освещения, $Z = 1,1 \div 1,2$;

n - количество ламп в светильнике, $n = 2$;

$\Phi_{\text{л}}$ - световой поток выбранной лампы, $\Phi_{\text{л}} = 3120 \text{ лм}$;

η - коэффициент использования, $\eta = 0,61$.

$$N = \frac{300 \cdot 1,5 \cdot 30 \cdot 1,2}{2 \cdot 3120 \cdot 0,61} = \frac{16200}{3806,4} = 4,21 \approx 4 \text{ шт}$$

Размещаем 2 ряда светильников по 2 светильника в ряду с расстоянием между светильниками в ряду 3 метра (Рисунок 5.1). Всего для создания нормируемой освещенности 300 лк необходимо 4 лампы ЛБ мощностью 40 Вт.

Тогда расчетный световой поток будет определяться по формуле

$$F_p = \frac{E \cdot K_3 \cdot S \cdot z}{N \cdot \eta} \quad (5.5)$$

$$F_p = \frac{300 \cdot 1,5 \cdot 6 \cdot 5 \cdot 1,1}{4 \cdot 2 \cdot 0,61} = 3043 \text{ лм}$$

Проверка светового потока

$$\Delta F = \frac{F_{л} - F_p}{F_{л}} \cdot 100\% \quad (5.6)$$

$$\Delta F = \frac{3120 - 3043}{3120} \cdot 100\% = 0,024\%$$

Полученная величина находится в пределах $-10\% \leq \Delta F \leq +20\%$, значит перерасчет светового потока не требуется.

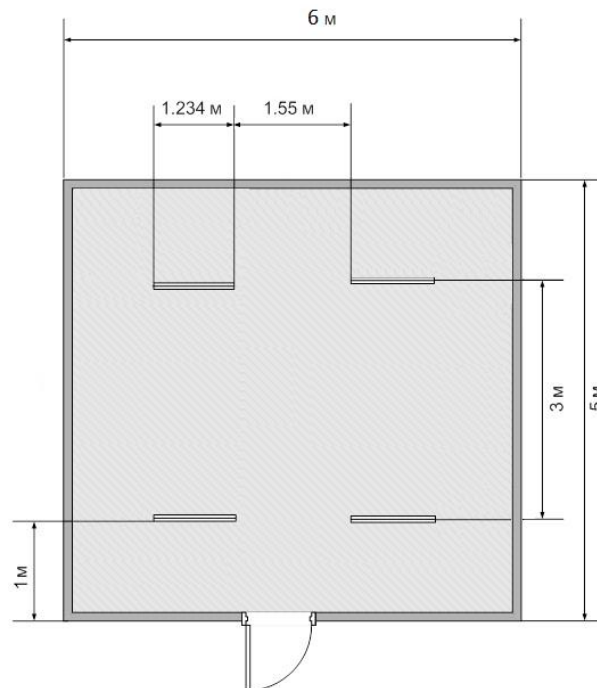


Рисунок 5.1 - Размещение светильников в рабочем помещении

В результате проделанных расчетов просчитаны необходимые меры безопасности и условия труда инженера инфокоммуникационные оборудования, которые соответствуют стандартам СНиП, и ГОСТ.

Произведена необходимая освещенность рабочей поверхности. Выбран тип источника света его светового потока с учетом нормированной освещенности. Рассчитано и показано расположение световых источников. Таким образом в данной работе мы постарались максимально минимизировать влияние вредных веществ в процессе производства. Что является основополагающим приоритетом для работодателей любых предприятия.

5.3 Расчет системы кондиционирования офиса

Приведем расчет системы кондиционирования в помещении. Кондиционирование обеспечивает наилучшее микроклимата в помещении и условия работы точной и чувствительной аппаратуры, и должно выполняться в соответствии с главой СНиП 11-33-75 [23].

Определим количество воздуха L м³/ч, которое необходимо вывести за один час из помещения, чтобы вместе с ним удалить избыток тепла $Q_{ИЗБ}$, определяется по формуле

$$L = \frac{Q_{ИЗБ}}{C_B \times t \times y_B} \quad (5.7)$$

где C_B - теплоемкость сухого воздуха, ккал/кг ($C_B=0,24$ ккал/кг град);

$t=t_{yx} - t_{ex}$ при расчетах возьмем $t=8^\circ\text{C}$;

y_B - плотность уходящего воздуха, определяемая в зависимости от температуры, кг/м³ (при расчетах принимается $y_B=1,20$ кг/м³).

Определим избыточное тепло $Q_{ИЗБ}$ ккал/ч

$$Q_{ИЗБ} = Q_{П} - Q_{ОТ} \quad (5.8)$$

где $Q_{П}$ - количества тепла поступающего в воздух помещения, ккал/ч;

$Q_{ОТ}$ - теплоотдача в окружающую среду через наружные ограждения (в теплое время года, при расчетах можно принять нулю).

Количество тепловыделений $Q_{П}$ зависит от мощности оборудования, числа работающих людей и тепла, которое вносится в помещение через оконные проемы

$$Q_{П} = Q_{ОБ} + Q_{Л} + Q_{Р} + Q_{осв} \quad (5.9)$$

где Q_{OB} - тепло, выделяемое производственным оборудованием, ккал/ч;

Q_L - тепло выделяемое людьми, ккал/ч;

Q_P - тепло, вносимое солнечной радиацией, ккал/ч.

Тепло, выделяемое производственным оборудованием в рабочем помещении, определяется из соотношения

$$Q_{OB} = 860 \cdot P_{OB} \cdot n \cdot N \quad (5.10)$$

где 860 тепловой эквивалент 1 кВт·ч, то есть тепло, эквивалентное 1 кВт·ч электрической энергии;

P_{OB} - мощность, потребляемая оборудованием $P_{OB} = 350 \text{ Вт}$;

n - коэффициент перехода тепла в помещение, $n=0,75$;

N - количество производственного оборудования.

$$Q_{OB} = 860 \cdot 0,35 \cdot 0,75 \cdot 5 = 1128,75 \text{ ккал/ч}$$

Тепло, вносимое солнечной, радиацией, определяется из соотношения

$$Q_P = m \cdot F \cdot g_{OCT} \quad (5.11)$$

где m - количество окон в помещении;

F - площадь одного окна $F=6,35 \text{ м}^2$;

g_{OCT} - солнечная радиация через остекленную поверхность, то есть количество тепла, вносимое за 1ч через остекление площадью в 1 м^2 .

$$Q_P = 1 \cdot 6,35 \cdot 145 = 920,75$$

Количество тепла, выделяемого находящимися в помещении людьми

$$Q_L = N \cdot q_L \quad (5.12)$$

где N - количество людей в помещении;

q_L , ккал/ч - тепловыделения одного человека $q_L = 60$;

$Q_L = 5 \cdot 60 = 300 \text{ ккал/ч}$.

Количество тепла, выделяемого системой освещения

$$Q_{OСВ} = 860 \cdot P_{OСВ} \cdot \alpha \cdot b \cdot \cos(\varphi) \quad (5.13)$$

где α - коэф.перевода электрической энергии в тепловую;

люминесцентные лампы $\alpha = 0,46 - 0,48$;

b - коэффициент одновременности работы (при работе всех светильников $b = 1$);

$\cos(\varphi) = 0,7 - 0,8$ - коэффициент мощности;

$P_{осв}$, кВт - мощность осветительной установки.

$$Q_{осв} = 860 \cdot 0,7 \cdot 0,48 \cdot 8 \cdot 0,04 \cdot 8 = 105,68 \text{ ккал/ч}$$

Тогда тепловыделение составит

$$Q_{изб} = Q_{п} = 1128,75 + 920,75 + 300 + 105,68 = 2455,18$$

Необходимый воздухообмен рассчитаем

$$L = \frac{2455,18}{0,24 \cdot 8 \cdot 1,20} = 1066 \text{ м}^3 / \text{ч}$$

Отношение количества воздуха, поступающего в помещение за один час, к объему помещения называется кратностью воздухообмена

$$K = \frac{L}{V_{п}} \quad (5.14)$$

где $V_{п}$ - объем помещения $V_{п} = 90 \text{ м}^3$

$$K = \frac{1066}{90} = 11 \text{ м}^3$$

Находим требуемую производительность кондиционера

$$W_K = k_3 \times L, \quad (5.15)$$

где k_3 - коэффициент запаса, $k_3 = 1,3 \div 2,0$.

$$W_K = 2 \cdot 1066 = 2132 \text{ м}^3 / \text{ч}$$

Исходя из расчетов в помещение с оборудованием, для соблюдения требуемых параметров микроклимата следует установить один кондиционер с производительностью не менее $2132 \text{ м}^3/\text{ч}$.

Выбранный кондиционер YORK МКН-МОН-25 (Рисунок 5.2). Расход воздуха (внутр./внешн.), $\text{м}^3/\text{ч}$: 950/3310.

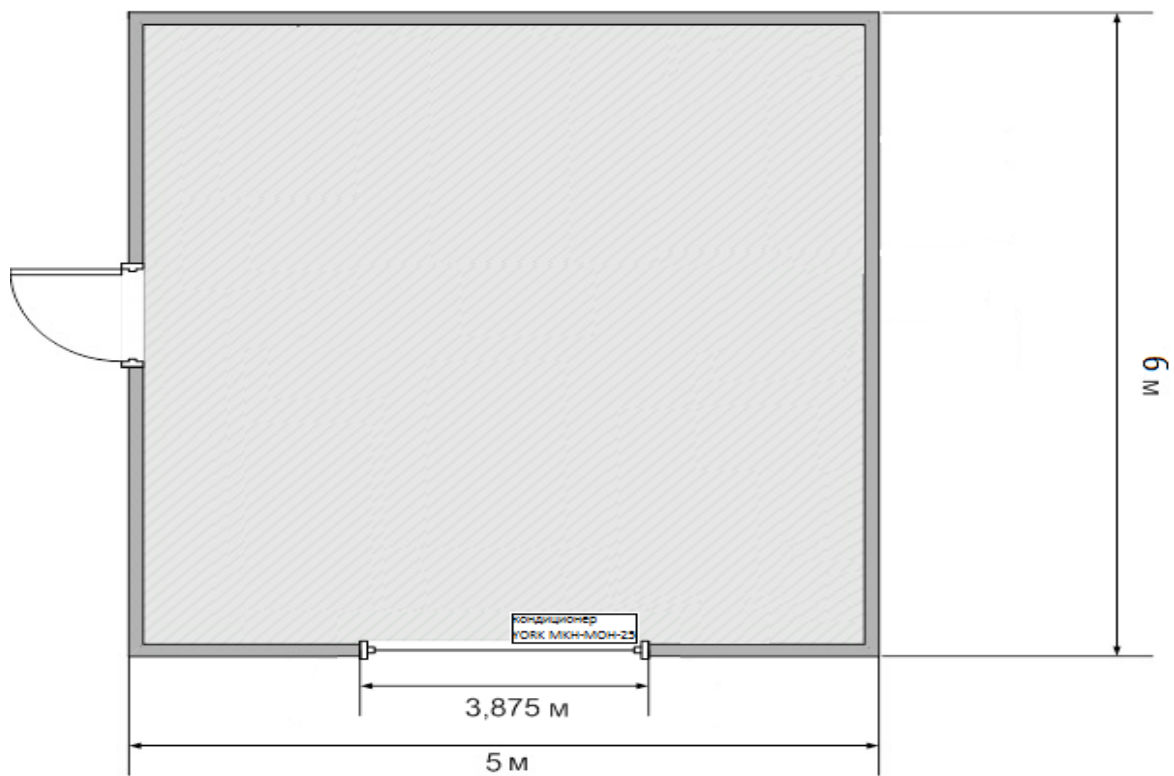


Рисунок 5.2 - Размещение кондиционера в помещении

Заключение

В данной дипломной работе мною было спроектировано и реализовано web-приложение, в котором удобный web-интерфейс сочетается с функционалом приложения. К функциональным возможностям сайта относятся: регистрация пользователей, управление сайтом через административную панель, структурирование данных используя базу данных и т.д.

В качестве серверных технологии в дипломной работе использовался web-сервер Apache, сервер баз данных MySQL, серверная технология PHP, фреймворк Yii. Была реализована схема проектирования модель-вид-контроллер, который является основным преимуществом фреймворка Yii.

В разработке интерфейса были использованы новейшие технологии twitter Bootstrap, использующий каноны CSS 3, и javascript. Самая последняя разработка язык гипертекстовой разметки HTML 5 также был использован в данном проекте. Это дает гарантию, что сайт будет одинаково правильно открываться на любом цифровом носителе - как на планшете, так и на компьютере. Стоит отметить и тот факт, что сайт находится на одном из казахстанских серверов и доступен в интернете.

Список использованной литературы

- 1 Сайт <http://webstudio2u.net/>
- 2 Сайт <http://ru.wikipedia.org>
- 3 Сайт <http://php.ru/manual/>
- 4 Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. - 5-е изд. - СПб.: БХВ-Петербург, 2004. - 1040 с.
- 5 Сайт <http://citforum.ru/internet/xml/>
- 6 Матросов А, Сергеев А, Чаунин М. HTML 4.0. - СПб.: ВHV Санкт-Петербург, 2000. - 672 с.
- 7 Сайт <http://htmlbook.ru/>
- 8 Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. - 5-е изд. - СПб.: БХВ-Петербург, 2004. - 1040 с.
- 9 В. Фаронов. Создание приложений с помощью С#. - М.: ЭКСМО, 2008. - 576 с.
- 10 К. В. Ахтырченко, Т. П. Сорокваша. - Методы и технологии реинжиниринга ИС. - Институт системного программирования РАН.
- 11 Горбунов-Посадов М.М. Расширяемые программы. - М.: Полиптих, 1999. - 336 с.
- 12 Сайт <http://www.yiiframework.ru/>
- 13 Scott Guthrie. ASP.NET MVC 1.0 650 с.
- 14 Л. Томсон, Л. Веллинг. Разработка Web-приложений на PHP и MySQL. - М.: Diasoft, 2003. - 672 с.
- 15 Сайт <http://mysql.com/>
- 16 Сайт <http://www.iteg.kg/>
- 17 З.Д. Еркешева, Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 - Вычислительная техника и программное обеспечение. - Алматы: АУЭС, 2013. - 40 с.
- 18 ГОСТ ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требование к качеству и тестирование. - М.: Издательство стандартов. - 67 с.
- 19 ГОСТ 28195-89. Оценка качества программных средств. Общие положения. - М.: Издательство стандартов. - 85 с.
- 20 СНиП 11-12-77. Защита от шума. Нормы проектирования. - М.: издательство стандартов - 87с.
- 21 ГОСТ 12.1.005-88. Санитарно-гигиенические требования к воздуху рабочей зоны. - М.: Издательство стандартов - 56с.
22. СанПиН 2.2.2. 542-96. Предельно допустимые значения характеристик ЭМП. - М.: Издательство стандартов. - 23с.
23. СНиП 11-33-75. Отопление, вентиляция и кондиционирование воздуха - М.: издательство стандартов - 88с.

Приложение А

Листинг программы

```
<?php

class EXFEEDController extends GxController
{

    public function actionView($id)
    {
        $this->render('view', array(
            'model' => $this->loadModel($id, 'EXFEED'),
        ));
    }

    public function actionCreate()
    {
        $model = new EXFEED;

        if (isset($_POST['EXFEED'])) {
            $model->setAttributes($_POST['EXFEED']);

            if ($model->save()) {
                if (Yii::app()->getRequest()->getIsAjaxRequest())
                    Yii::app()->end();
                else
                    $this->redirect(array('view', 'id' => $model->FEED_ID));
            }
        }

        $this->render('create', array('model' => $model));
    }

    public function actionCreatehidden()
    {
        $sid = $_POST['student'];
        $student = USER::model()->findByPk($sid);
        if ($_POST['ef'] != '') {
            $exfeed=EXFEED::model()->findByPk($_POST['ef']);
            $exfeed->EXFEED_STATUS=0;
        }
        else {
            $exfeed = new EXFEED();
        }
        $exfeed->users_id = $sid;
        $exfeed->EXFEED_INTEXT = $_POST['sql'];
    }
}
```

Продолжение приложения А

```
$exfeed->Level_ID = $_POST['level'];
    $exfeed->EXFEED_INCOME = (date("d-m-Y H:i:s", time()));
    if ($exfeed->save()) {
        echo("Задание отправлено на проверку!");
    } else {
        echo("Что-то пошло не так, повторите попытку.");
    }
    // print_r($exfeed);
}

public function actionUpdate($id)
{
    $model = $this->loadModel($id, 'EXFEED');

    if (isset($_POST['EXFEED'])) {
        $model->setAttributes($_POST['EXFEED']);

        if ($model->save()) {
            $this->redirect(array('view', 'id' => $model->FEED_ID));
        }
    }

    $this->render('update', array(
        'model' => $model,
    ));
}

public function actionDelete($id)
{
    if (Yii::app()->getRequest()->getIsPostRequest()) {
        $this->loadModel($id, 'EXFEED')->delete();

        if (!Yii::app()->getRequest()->getIsAjaxRequest())
            $this->redirect(array('admin'));
    } else
        throw new CHttpException(400, Yii::t('app', 'Your request is invalid.'));
}

public function actionIndex()
{
    $dataProvider = new CActiveDataProvider('EXFEED');
    $this->render('index', array(
        'dataProvider' => $dataProvider,
    ));
}
```


Продолжение приложения А

```
{
    $exfeed=EXFEED::model()-
>findByAttributes(array('FEED_ID'=>$_POST['exfeed']));
    $level=CURLEVEL::model()-
>findByAttributes(array('Level_id'=>$_POST['level']));
    $part=CURSPART::model()-
>findByAttributes(array('Part_id'=>$_POST['part']));
    $this->renderPartial("_feedcheck", array('exfeed'=>$exfeed,
'level'=>$level, 'part'=>$part));
}

public function actionFeed()
{
    $model=EXFEED::model()->findAll();
    $users=USER::model()->findAll();
    $levels=CURLEVEL::model()->findAll();
    $parts=CURSPART::model()->findAll();
    $this->render('feed', array(
        'model' => $model,
        'users'=>$users,
        'levels'=>$levels,
        'parts'=>$parts,
    ));
}

public function actionPullright()
{
    $s=$_POST['status'];
    $ef=EXFEED::model()->findByPk($_POST['exfeed']);
    $ef->EXFEED_INCOME=(date("d-m-Y H:i:s", $ef-
>EXFEED_INCOME));
    if ($s==1) {
        $ef->EXFEED_STATUS=1;

        if ($ef->save()){
            echo "Задание помечено выполненным!";
        }
    }
    if ($s==2) {
        $ef->EXFEED_STATUS=2;
        if ($ef->save()){
            echo "Решение помечено ошибочным!";
        }
    }
}

public function actionAdmin()
{
```

Продолжение приложения А

```
$model = new EXFEED('search');
    $model->unsetAttributes();

    if (isset($_GET['EXFEED']))
        $model->setAttributes($_GET['EXFEED']);

    $this->render('admin', array(
        'model' => $model,
    ));
}
```

```
}
```

```
<?php
```

```
class StudentController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD
operations
            'postOnly + delete', // we only allow deletion via
POST request
        );
    }
}
```

```
<?php
```

```
class PageController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to
     * '//layouts/column2', meaning
     * using two-column layout. See
     * 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
```

Продолжение приложения A

```
* @return array action filters
*/
public function filters()
{
    return array(
        'accessControl', // perform access control for CRUD
operations
        'postOnly + delete', // we only allow deletion via
POST request
    );
}

/**
 * Specifies the access control rules.
 * This method is used by the 'accessControl' filter.
 * @return array access control rules
 */
public function accessRules()
{
    return array(
        array('allow', // allow all users to perform
'index' and 'view' actions
            'actions'=>array('index','view'),
            'users'=>array('*'),
        ),
        array('allow', // allow authenticated user to
perform 'create' and 'update' actions
            'actions'=>array('create','update'),
            'users'=>array('@'),
        ),
        array('allow', // allow admin user to perform
'admin' and 'delete' actions
            'actions'=>array('admin','delete'),
            'users'=>array('admin'),
        ),
        array('deny', // deny all users
            'users'=>array('*'),
        ),
    );
}

/**
 * Displays a particular model.
 * @param integer $id the ID of the model to be displayed
 */
public function actionView($id)
{
    $this->render('view',array(
        'model'=>$this->loadModel($id),
    ));
}
```

Продолжение приложения А

```
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected
to the 'view' page.
 */
public function actionCreate()
{
    $model=new Page;

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Page']))
    {
        $model->attributes=$_POST['Page'];
        if($model->save())
            $this->redirect(array('view','id'=>$model-
>id));
    }

    $this->render('create',array(
        'model'=>$model,
    ));
}

/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to
the 'view' page.
 * @param integer $id the ID of the model to be updated
 */
public function actionUpdate($id)
{
    $model=$this->loadModel($id);

    // Uncomment the following line if AJAX validation is
needed
    // $this->performAjaxValidation($model);

    if(isset($_POST['Page']))
    {
        $model->attributes=$_POST['Page'];
        if($model->save())
            $this->redirect(array('view','id'=>$model-
>id));
    }
}
```

Продолжение приложения A

```
        $this->render('update', array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected
to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin grid
view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ?
$_POST['returnUrl'] : array('admin'));
    }

    /**
     * Lists all models.
     */
    public function actionIndex()
    {
        $dataProvider=new CActiveDataProvider('Page');
        $this->render('index', array(
            'dataProvider'=>$dataProvider,
        ));
    }

    /**
     * Manages all models.
     */
    public function actionAdmin()
    {
        $model=new Page('search');
        $model->unsetAttributes(); // clear any default values
        if(isset($_GET['Page']))
            $model->attributes=$_GET['Page'];

        $this->render('admin', array(
            'model'=>$model,
        ));
    }

    /**
```

Продолжение приложения А

```
* Returns the data model based on the primary key given in
the GET variable.
* If the data model is not found, an HTTP exception will be
raised.
* @param integer $id the ID of the model to be loaded
* @return Page the loaded model
* @throws CHttpException
*/
public function loadModel($id)
{
    $model=Page::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,'The requested page
does not exist.');
```

```
    return $model;
}

/**
 * Performs the AJAX validation.
 * @param Page $model the model to be validated
 */
protected function performAjaxValidation($model)
{
    if(isset($_POST['ajax']) && $_POST['ajax']==='page-form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}
}
```