Московский Государственный Технический Университет

им. Н.Э. Баумана

Калужский филиал

Кафедра ЭИУ5-КФ

Системы Автоматизированного Проектирования

**Курсовая работа на тему:**

**«Файлообменник с веб-интерфейсом»**

Калуга, 2007

**Содержание**

Введение

Исследовательская часть

Инструменты построения страниц

Языки веб-программирования

Методы общения клиента и сервера

Базы данных.

Веб-сервер

Редакторы кода

Инструкция программиста.

Системные требования

Инструкция по установке

Структура проекта

Описание исходных кодов

Инструкция пользователя.

Заключение.

Список литературы.

# Введение

В курсовой работе была поставлена задача создания файлообменника с веб-интерфейсом, с помощью которого пользователи могут обмениваться файлами в сети.

Цель работы: показать знание языка веб-программирования PHP, а также средств HTML, CSS и базы данных MySQL.

Основные функции сайта: загрузка, изменение, удаление файлов с сайта (через веб-интерфейс), отображение загруженных файлов в удобном виде, контролируемое скачивание загруженных файлов. Курсовой проект представляет собой сайт, готовый к выкладыванию на подходящем хостинге. Для создания курсового проекта использовался текстовый редактор EditPlus.

## Актуальность поставленной задачи

При появлении интернета у людей возникло множество проблем. Одна из таких проблем – обмен файлами в сети. Именно ей я и решил заняться.

Для обмена файла между пользователями по сети существует несколько способов. Во-первых, e-mail (почта). Для пересылки файлов по почте необходимо знать ящик человека, которому отправляется письмо, при этом размер пересылаемых файлов увеличивается на 37%. Это не всегда удобно. Во-вторых, можно пересылать файлы при помощи сервиса ICQ. Это опять же не удобно, т.к. требуется регистрация обоих пользователей в системе и наличие у них клиентов, поддерживающих передачу файлов. Кроме того, при помощи первых 2х способов передать файлы можно лишь ограниченному числу людей.

Если же требуется передать файл большему числу людей, используют:

1. FTP (File Transfer Protocol). Для этого необходим FTP-сервер

2. Веб-сайт. Для этого требуется хостинг, не у всех он есть.

3. Файлообменник. Самый доступный способ массового распространения файлов. Файлообменники призваны решить нарастающую потребность в обмене файлами для тех, кто по тем или иным причинам не хочет использовать предыдущие способы.

## Структура пояснительной записки

Пояснительная записка состоит из вступления, трёх глав и заключения.

Глава 1 – исследовательская часть, которая содержит целесообразность выбора инструментов построения веб-страниц, языка веб-программирования, среды разработки.

Глава 2 – инструкция программиста, в состав которой входят системные требования, инструкция по установке файлообменника, описание структуры проекта и подробное описание методов.

Глава 3 – инструкция пользователя, которая содержит описание способов и приёмов работы с файлообменником с точки зрения пользователя, модератора и администратора.

# Исследовательская часть

Файлообменник предназначен для размещения в интернете, поэтому он должен иметь вид сайта, состоящего из отдельных страниц. Для построения страниц сайта можно использовать различные инструменты.

## 

## Инструменты построения страниц

Традиционно для отображения содержимого сайтов используется HTML. Аббревиатура HTML расшифровывается как Hyper Text Markup Language – язык гипертекстовой разметки. То есть HTML – не язык программирования, а просто одно из средств для рисования страничек. Существует несколько спецификаций языка. В написании своей работы я придерживался наиболее распространённой в настоящее время спецификации HTML 4.01 Transitional. Про эту спецификацию можно посмотреть на w3c.org

Для улучшения внешнего вида HTML-страничек применяют CSS (Cascading Style Sheets) – листы стилей. Существует несколько спецификаций CSS. Разные браузеры поддерживают разные спецификации CSS.

|  |  |  |  |
| --- | --- | --- | --- |
| Браузеры | CSS | CSS2 | CSS3 |
| MSIE 6.0 SP2 | + | - | - |
| Opera 8.x | + | + | - |
| Konqueror | + | + | + |
| Gecko 1.8.x | + | + | + |

Я для работы использовал CSS2, т.к. его возможностей мне достаточно для создания полноценных сайтов, а CSS3 пока что поддерживается не всеми браузерами.

Для тестирования сайта я использовал браузеры Опера 8.52 и MSIE. Сравнение характеристик браузеров – это отдельная тема, не буду заострять на ней внимание. Теоретически курсовойпроект совместим и с другими браузерами.

Широко используются скрипты, написанные на JS (JavaScript) и VBS (Visual Basic Script) для выполнения некоторых простых действий на клиентском компьютере. Они помогают изменять информацию на странице (чаще связанную с оформлением), не перезагружая страницу. Также иногда используют технологию ActiveX – чаще всего для тех же целей, что и JS. Встречаются и другие применения ActiveX – пример тому webmoney. Иногда создатели сайтов делают использование многих наворотов обязательным (картинки, скрипты, ActiveX). Это мешает нормальному восприятию содержимого сайта. В своём проекте я счёл нецелесообразным применение таких скриптов.

С помощью таких средств принято разрабатывать страницы сайтов. Но статические страницы, написанные с помощью этих средств, не позволяют осуществлять общение сайта и посетителя. Это весьма удобно реализуется с помощью языков веб-программирования.

## 

## Языки веб-программирования

Разнообразие сайтов в сети практически невозможно сделать, используя лишь статические страницы. Чтобы сделать информацию, отображаемую на сайте, более актуальной, интерактивной, и т.п. нужно динамически генерировать содержимое сайта, или хотя бы отдельных его страниц. Для этого были придуманы серверные приложения (CGI) и скрипты, обрабатывающиеся определённым интерпретатором. Наиболее распространены Perl, PHP и ASP.

На следующие особенности следует обратить внимание при сравнении (достоинства отмечены «+», недостатки «-»):

**Perl**

+ считается самым быстрым языком

+ старейший язык, следовательно, проверен временем, приёмы программирования давно отработаны и задокументированы.

- сложность вывода HTML-кода

**PHP** (4)

+ простота интеграции с HTML и ему подобными форматами

+ простота в освоении новичком и исключительная масштабируемость

+ популярность – полно документации как официальной (php.net и книги) так и сторонних курсов и форумов

+ популярность – всегда можно получить помощь в решении любой проблемы

+- бесплатность – никаких гарантий и служб поддержки

- популярность – все дыры в программировании (новичков) известны

**ASP**

+ надёжность,

+ служба поддержки,

+ интеграция с платформой .Net,

- строгая типизация,

- платный продукт,

- требует Windows-хостинг.

После такого сравнения для изучения я выбрал язык PHP, из-за таких преимуществ:

1. Наличие понятной и доступной документации
2. Простота в установке и настройке.
3. Поддержка самой популярной open source БД – MySQL
4. Поддержка веб-сервера Apache
5. Свобода в выборе хостинга, т.к. подавляющее большинство хостингов используют платформу FreeBSD+Apache+PHP+MySQL

Аббревиатура PHP сейчас расшифровывается как Hypertext Preprocessor. Подробнее про язык PHP написано на php.net

Но как происходит «общение» посетителя с сайтом? Для этого уже довольно давно были придуманы действенные методы.

## 

## Методы общения клиента и сервера

Для передачи данных от браузера клиента приложению или скрипту на сервере используются методы GET и POST протокола HTTP (HyperText Transfer Protocol), а также через куки (cookie), реферреры и другие данные, передаваемые браузером.

При передаче данных методом GET данные отображаются в адресной строке браузера, при использовании метода POST передаваемые данные не видно. Также есть возможность загрузки файлов методом POST, однако для использования этого метода необходимо сделать submit формы.

Куки – это переменные, предлагаемые к установке браузеру. Правильный браузер должен запоминать куки и при каждом посещении страницы сайта отправлять веб-серверу значения этих переменных.

Реферрер – это указатель перехода. Реферреры придуманы для отслеживания перемещения пользователей по сети. При нажатии на ссылку правильный браузер должен отправить (вместе с куками) адрес страницы, на которой находилась эта ссылка.

Также следует отметить такой важный передаваемый параметр, как сигнатура браузера. По ней однозначно определяется, кто зашёл на сайт.

Данным, переданным этими способами (HTTP GET/POST, cookie, referrers, user\_agent) нельзя доверять.

Переданную информацию нужно где-то хранить. Информацию можно сохранять разными способами. Наиболее распространены базы данных, т.к. многие операции они делают за программистов, облегчая им труд.

## 

## Базы данных.

Среди многообразия баз данных я выбрал MySQL 4.0,и вот почему. На их сайте читаем: «MySQL предлагает такую комбинацию производительности, цены и возможностей, которую вряд ли смогут предложить другие».

И действительно, основные достоинства MySQL – это:

1. Бесплатность (на 1 месте)

2. Небольшой размер и дистрибутива, и установленной программы.

3. Простота использования.

4. Скорость и стабильность.

5. Популярность.

Следует добавить, что возможности MySQL уже стали избыточными для большинства веб-приложений (да, такое часто происходит со многими хорошими программами). Более того, новая версия 5.0, которая была объявлена стабильной в феврале, несовместима с предыдущими версиями, и многие сайты стали неработоспособными после обновления серверов. Для работы моего сайта вполне достаточно возможностей MySQL 3.23, но я использовал MySQL 4.0, т.к. 3.23 уже морально устарела.

Доступ к сайту осуществляется при помощи веб-сервера. Веб-сервер – это программный продукт, осуществляющий функционирование сайтов по протоколу HTTP.

## 

## Веб-сервер

Лучшим веб-сервером для моих целей является Apache (Апач).

«The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Apache has been the most popular web server on the Internet since April 1996. The November 2005 Netcraft Web Server Survey found that more than 70% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined»[[1]](#footnote-1).

Кроме того, по Апачу много вполне понятной документации, он прост в установке и настройке, очень мало весит (меньше 5М), работает исключительно стабильно.

## Редакторы кода

Для написания кода я использовал 2 текстовых редактора.

Первый – AkelPad – это бесплатный текстовый редактор для Windows, призванный заменить Блокнот, входящий в состав этой системы. Разработан Алексеем Кузнецовым, студентом 5го курса физтеха.

Его преимущества:

* маленький размер (менее 20 Кб в архиве)
* работа с файлами произвольного размера (тогда как у Блокнота есть ограничение 64 Кб)
* работа с кодировками: Windows, DOS, КОИ8-Р, различные варианты Unicode (UCS-2LE, UCS-2BE, UTF-8), а также с любой кодовой страницей, установленной в системе
* корректное автоопределение кодировки
* поиск и замена как обычных строк текста, так и последовательностей специальных символов
* продуманные "горячие" клавиши назначены почти каждой команде (выход по ESC, сохранить по F2 и т.п.)
* простота и отсутствие перегруженности ненужными и усложняющими работу функциями

Подробнее - akelpad.net.ru

Второй – EditPlus. Это уже коммерческий (платный) редактор. Цитата:

EditPlus is an Internet-ready 32-bit text editor, HTML editor and programmers editor for Windows. While it can serve as a good replacement for Notepad, it also offers many powerful features for Web page authors and programmers.

* Syntax highlighting for HTML, CSS, PHP, ASP, Perl, C/C++/C#, Java, JavaScript and VBScript. Also, it can be extended for other programming languages based on custom syntax files.
* Seamless Web browser for previewing HTML pages, and FTP commands for uploading local files to FTP server.
* Other features include HTML toolbar, user tools, line number, ruler, URL highlighting, auto-completion, cliptext, column selection, powerful search and replace, multiple undo/redo, spell checker, customizable keyboard shortcuts, and more.[[2]](#footnote-2)

Уже не так быстр, как AkelPad, но довольно удобен в настройке и работе. Рекомендован 440hz.ru.

# Инструкция программиста

## 

## Системные требования

Для работы этого проекта необходим сервер и клиент. Сервер должен состоять из операционной системы с установленным веб-сервером, базой данных MySQL и интерпретатором PHP(4) версии не менее 4.3.2.

Клиент должен иметь графический браузер, поддерживающий протокол HTTP, стандарты HTML4 и CSS2.

Я использовал следующую конфигурацию:

Сервер: Win32, Apache 2.0.55, PHP 4.4.2, MySQL 4.0.26

Клиент: Win32, Opera 8.52

## Инструкция по установке

Для установки проекта необходимо в файл init.class прописать необходимые значения – путь к проекту, e-mail тех.поддержки и реквизиты БД (логин, пароль, имя БД).

Веб-сервер настраивается следующим образом (на примере Apache):

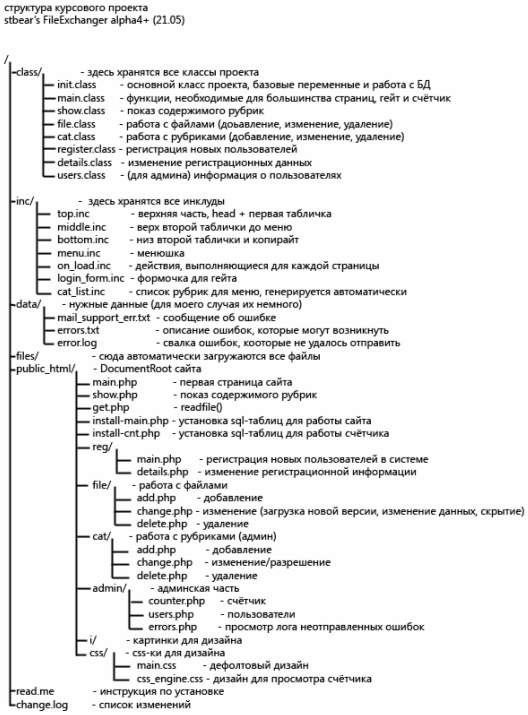
DocumentRoot "путь:/к/проекту/public\_html"

DirectoryIndex main.php

Ну и ещё такая важная вещь – в php.ini параметр register\_globals должен быть установлен в Off (сейчас это делают по умолчанию).

Этого должно хватить для успешной работы проекта.

Структура проекта



## Описание исходных кодов

Для построения этого проекта, как видно из схемы на предыдущей странице, был взят принцип организации сайтов Руслана Курепина.

Чтобы не объяснять всё с самого начала, приведу здесь сразу код главной страницы сайта и объясню построчно. Cтрочки пронумерованы.

файл: **public\_html/main.php**

1. Открываем скобку – входим в режим PHP.

<?

2. Подключаем класс. Такая форма записи позволяет переносить проект по папкам, не исправляя каждый раз путь к файлу.

require(str\_replace('/public\_html','/class',$\_SERVER['DOCUMENT\_ROOT']).'/main.class');

3. Создаём экземпляр класса.

$my=new class\_main;

4. Подключаем on\_load.inc – действия, выполняющиеся для многих страниц.

include($my->path\_inc.'/on\_load.inc');

5. Заголовок страницы – отображается в двух местах.

$title='Main Page';

6. Навигатор.

$navigator='<a href=/>Главная</a>';

7. Все критические функции выполнены, мы знаем что страница загрузится – отправляем HTTP-заголовки (хидеры).

$my->send\_headers();

8-9. Подключаем инклуды (подключаемые куски кода).

include($my->path\_inc.'/top.inc');

include($my->path\_inc.'/middle.inc');

10. Выходим из режима PHP

?>

11-12. Выдаём HTML-код без обработки, а именно 2 строчки приветствия

<p>какой-то текст на главной странице.<br>например, такой:

<p>Здравствуйте!<br>Это файлообменник, моя курсовая за 4 семестр.<br>

13-15. Подключаем нижнюю часть таблички.

<?

include($my->path\_inc.'/bottom.inc');

?>

Следует заметить, что после 8 строчки я не использую flush()… наверное, это нецелесообразно в таком маленьком проекте.

Теперь подробно рассмотрим инклуды.

**inc/top.inc**

1. Говорим браузеру, что это HTML 4.0

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

2. Начало HTML-кода.

<html>

3. Открываем head.

<head>

4. Подключаем иконку

<link rel="shortcut icon" href="/i/logo.gif">

5. Строка добавлена для поддержки просмотрщика счётчика, чтобы меньше модифицировать код. Суть: если вызван файл counter.php, то подключается нужный CSS-файл

<? if(preg\_match("/counter.php/",$\_SERVER['REQUEST\_URI'])) {?> <link rel="stylesheet" href="/css/css\_engine.css" type="text/css"><?}?>

6. Подключаем основную css-ку. В случае счётчика она выше по приоритету, чем предыдущая, поэтому затирает лишние стили.

<link rel="stylesheet" href="/css/main.css" type="text/css">

7. Говорим браузеру, что эта страница не предназначена для размещения во фреймах.

<meta http-equiv="Window-target" content="\_top">

8. Сведения об авторе.

<meta name="Author" content="<?=$my->copy\_name?>">

9. Мой любимый текстовый редактор

<meta name="Generator" content="<?=$my->generator?>">

10. Версия сайта.

<meta name="Version" content="<?=$my->version?>">

11. Заголовок – показывается вверху окна.

<title><?=$title?> :: <?=$my->site\_name?></title>

12,13. Закрываем head, открываем body – начинаем рисовать страничку.

</head>

<body>

14. Таблица во всю ширину экрана

<table width="100%" border=0>

15,16. Надпись: File Exchanger.

<tr>

<td valign="bottom"><nobr><h2 id="top\_h3"><?=$my->site\_name?></h3></nobr></td>

17. Заголовок: Main Page (или какой там прописан в $title)

<td rowspan=2 id="top\_title"><?=$title?></td>

18-20. Форма для гейта (вход-выход)

<td rowspan=2>

<? include($my->path\_inc.'/login\_form.inc'); ?>

</td>

21-23. пишем второй строчкой версию и закрываем таблицу

</tr>

<tr><td valign=top><p id="top\_version"><?=$my->version?></td></tr>

</table>

24. навигатор, который мы заранее прописали в $navigator.

<hr><p id="navv">навигатор :: <?=$navigator?><hr>

**Небольшое теоретическое отступление по поводу CSS.**

Вообще, существует три способа использования CSS.

1. Включение CSS непосредственно в HTML-код. Например, так:

<p style="text-align:center;color:#d00">Красный цвет по центру

Может использоваться для одноразовых включений в HTML (наряду с <font>) и в случаях, когда нецелесообразно менять существующие CSS-файлы или подключать новые.

2. Включение CSS в блоке <head> страницы. Вот так:

<html><head>

...

<style>

red {color:#f00}

</style>

</head><body>

...

<p class=red>Красный текст

...

</body></html>

Такой способ не рекомендуется применять, однако это предпочтительно в одном случае – если дизайн сайта, настраивается под каждого пользователя индивидуально. Кодеру меньше работы – переменные дизайна банально вставляются на нужные места и нет смысла писать несколько CSS-ок.

3. Подключение внешних CSS-файлов. Реализуется так:

<link rel=stylesheet href=css/mycss.css type=text/css>

И в нужную папку кладётся нужный файл.

Как видно, в этом проекте я использую третий способ.

**inc/middle.inc**

1. основная таблица во весь экран.

<table cellspacing=5 cellpadding=5 width="100%" border=0>

2. она состоит из одной строчки.

<tr><td width=200 valign="top">

3. в первой ячейке – содержимое menu.inc.

<? include($my->path\_inc.'/menu.inc'); ?>

4-5. во второй – результат гейта (вы зашли успешно либо вы не зашли вообще). Потом начинается основной текст.

</td><td valign="top">

<?=$my->gate\_result?>

**inc/bottom.inc**

1-2. закрываем основную таблицу

</td></tr>

</table>

3. копирайт

<p id="copy">&copy; 2006 <a href="<?=$my->copy\_url?>"><?=$my->copy\_name?></a>.

4-5. конец HTML

</body>

</html>

Теперь объясняю что за переменные типа $path\_inc, $copy\_name, $site\_name, $generator, $version и др. Встречались в предыдущих инклудах. Эти глобальные переменные прописаны в init.class. Вот он:

<?

Называем класс и открываем скобку

class class\_init

{

Пути к проекту на диске – при переносе должны быть изменены

var $path\_data='d:\Work\FileExchanger alpha\data';

var $path\_inc='d:\Work\FileExchanger alpha\inc';

var $path\_files='d:\Work\FileExchanger alpha\files';

var $path\_http='http://localhost/';

копирайт

var $copy\_name='stbear-l2';

var $copy\_url='http://copi.ru/68614';

мой любимый редактор

var $generator='AkelPad 2.1.6';

имя сайта – отображается в top.inc

var $site\_name='File Exchanger';

адреса e-mail

var $email\_support='stbear@centrum.cz';

var $email\_robot='robot@localhost';

версия

var $version='alpha4+';

время кэширования страниц (функция send\_headers())

var $cache\_time=180;

максимальный номер ошибки, которая отправляется админу.

var $debug\_level=50;

указатель на соединение с базой

var $sql\_link;

реквизиты доступа к БД

var $sql\_host='localhost';

var $sql\_login='root';

var $sql\_pass='';

var $sql\_db='db\_files';

запрос формируется в этой переменной

var $sql\_task;

ошибки SQL (для дебага)

var $sql\_error;

var $sql\_errno;

результат выполнения запроса

var $sql\_res;

Стартует соединение с БД, если этого не было сделано ранее

function sql\_start()

{

Устанавливаем соединение с SQL-сервером. Использовать **p**connect предпочтительнее

if(!$this->sql\_link=@mysql\_pconnect($this->sql\_host,$this->sql\_login,$this->sql\_pass)) return(11);

Выбираем нашу БД.

if(!@mysql\_select\_db($this->sql\_db)) return(12);

return(0);

}

Основная функция для взаимодействия с БД

function sql\_run()

{

Если нет соединения с БД, вызывается sql\_start()

if(!$this->sql\_link) if($err=$this->sql\_start()) return($err);

собственно выполняется запрос

$this->sql\_res=@mysql\_query($this->sql\_task,$this->sql\_link);

Проверяем, были ли ошибки

if($this->sql\_errno=mysql\_errno($this->sql\_link))

{

Если была ошибка, запоминаем её описание и возвращаем ошибку 13 – «ошибка выполнения запроса»

$this->sql\_error=@mysql\_error($this->sql\_link);

return(13);

}

return(0);

}

}

конец

?>

Main.class нет смысла описывать из-за его размера.

Распишу оставшийся инклуд.

**inc/on\_load.inc**

<?

Запускаем счётчик

if($err=$my->counter()) $my->get\_err($err);

если есть переменная, полученная от формы входа

if(@$\_POST['gate\_submit'])

{

Проверяем логин и пароль

if($err=$my->gate\_open()) $my->gate\_result=$my->get\_err($err);

else $my->gate\_result='<p class=ok>Вы успешно вошли в систему.';

}

Если пользователь решил выйти, удаляем сессию

if(isset($\_GET['out'])) $my->gate\_close();

восстановление сессии по кукам

elseif($err=$my->recognize\_cookie()) $my->get\_err($err);

?>

Для примера приведу функцию добавления файла.

Ппеременные с говорящими именами – объявляются в начале класса

var $in\_file\_ext;

var $in\_file\_cat;

var $in\_file\_desc;

var $in\_file\_name;

var $in\_file\_user;

var $in\_file\_id;

var $in\_file\_enable;

var $in\_file\_dt;

var $in\_file\_size;

объявление функции

function in\_file\_add()

{

Если пользователь не зарегистрирован, то что он тут делает?

if(!$\_SESSION['status']) return(6);

проверка файла, теперь переменные «наполнились»

if($err=$this->in\_file\_check()) return($err);

файл и данные в порядке, составляем запрос

$this->sql\_task='insert into tbl\_files(f\_ext,f\_name,f\_desc,f\_dt,f\_size,f\_user) values(\''.$this->in\_file\_ext.'\',\''.$this->in\_file\_name.'\',\''.$this->in\_file\_desc.'\',\''.$this->in\_file\_dt.'\',\''.$this->in\_file\_size.'\',\''.$this->in\_file\_user.'\')';а теперь выполняем запрос

if($err=$this->sql\_run()) return($err);

данные добавились, получаем ID

$this->sql\_task='select last\_insert\_id()';

if($err=$this->sql\_run()) return($err);

if(!list($this->in\_file\_id)=mysql\_fetch\_row($this->sql\_res)) return(14);

перемещаем файл

if(!move\_uploaded\_file($\_FILES[$this->in\_file\_user]['tmp\_name'],$this->path\_files.'/'.$this->in\_file\_id.'.'.$this->in\_file\_ext)) {

если он не перемещён, удаляем запись из базы...

$this->sql\_task='delete from tbl\_files where f\_id='.$this->in\_file\_id;

if($err=$this->sql\_run()) return($err);

...и возвращаем ошибку

return(23);

}

Файл удачно перемещён, делаем связку с рубрикой

$this->sql\_task='insert into tbl\_cats\_files(cf\_cat,cf\_file) values(\''.$this->in\_file\_cat.'\',\''.$this->in\_file\_id.'\')';

if($err=$this->sql\_run()) return($err);

если не стоит галка «Скрыть файл», разрешаем:

if(!@$\_POST['file\_enable'])

{

if($err=$this->in\_file\_enable()) return($err);

if($err=$this->in\_cat\_file\_enable()) return($err);

}

$this->result='<p class=ok>Файл успешно добавлен.Теперь он всегда доступен по ссылке :<br> <a href=http://'.$\_SERVER['HTTP\_HOST'].'/get.php?id='.$this->in\_file\_id.'>http://'.$\_SERVER['HTTP\_HOST'].'/get.php?id='.$this->in\_file\_id.'</a>.';

Всё в порядке, возвращаем ноль.

return(0);

}

Фрагмент **public\_html/file/add.php** (форма)

Enctype добавляется, тк мы используем POST-аплоад файлов

<form enctype="multipart/form-data" action="<?=$\_SERVER['REQUEST\_URI']?>" method=post>

Информация для пользователей

Параметры добавляемого файла:<br><br>author: <?=$\_SESSION['login']?>, date: <?=$my->get\_dt()."\n"?>, рубрика:

Тут выводится список рубрик

<? echo $my->generate\_option\_list(); ?>

<br>Название файла: <input type=text name=file\_name value="<?=@$\_POST['file\_name']?>" size=80 maxlength=50>

<br>Описание:<br><textarea name=file\_desc cols=80 rows=5><?=@$\_POST['file\_desc']?></textarea>

Имя файловой переменной – логин пользователя (так проще)

<br><br>Сам файл: <input type=file name="<?=$\_SESSION['login']?>">

<br>Скрыть файл? <input type=checkbox name=file\_enable value="тра-ля-ля!!!"<? if(@$\_POST['file\_enable']) echo 'checked'; ?>><font size=1>при включённой опции файл не будет отображаться в рубрике, но будет доступен по по сгенерированному URL-адресу</font>

<br><input type=submit name=submit value="Добавить!">

</form>

Принцип работы остальных функциональных частей аналогичен приведённым примерам. Кроме того, текст скриптов понятен и без комментариев.

# Инструкция пользователя

Для запуска программы необходимо запустить любой браузер и в строке URL ввести localhost или 127.0.0.1. На вашем экране появится следующая страница (рис.1):

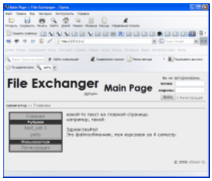


Рис.1 Главная страница файлообменника.

Теперь вам следует зарегистрироваться.

Если у Вас еще нет своего имени (логина) и пароля, щелкните мышкой на пункте меню «Регистрация». На открывшейся странице (рис.2) заполните поля «логин», «пароль» , повторите Ваш пароль в поле «Исчо разок». Если данные введены верно, нажмите кнопку «Регистрировать».

По умолчанию длина логина пользователя не может быть меньше 2 символов, а длина пароля – меньше 4.

Если Вы зарегистрировались успешно, сообщение об этом появится на страничке (см.рис.3).

Теперь Вы сможете войти в систему как зарегистрированный пользователь. Для этого введите логин и пароль в окошке справа вверху и нажмите кнопку «Войти».

Если логин и пароль введены правильно, появится сообщение «Вы успешно вошли в систему» и следующая страничка (см.рис.4).

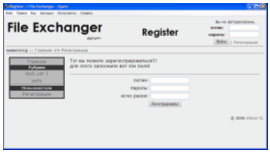


Рис.2 Страница регистрации.

Введите логин и пароль в окошке справа вверху и нажмите кнопку «Войти».

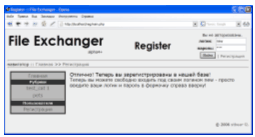


Рис.3 Страница с сообщением об успешной регистрации.

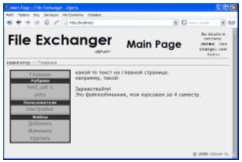


Рис.4. Страничка зарегистрированного пользователя.

Слева странички расположено меню. Для простого пользователя оно состоит из трех разделов:

- Рубрики;

- Пользователи;

- Файлы.

Для пользователя со статусом «Администратор» добавлены разделы

- Рубрики \редактирование\;

- Администрирование

Рассмотрим разделы меню подробнее

1. **Рубрики.**

Здесь перечислены темы, по которым группируются предлагаемые файлы. Щелчок левой кнопкой мыши по названию рубрики открывает перечень файлов выбранной тематики.

В примере на рис.5 на странице одновременно отображается перечень из 5 файлов. Если в рубрике больше 5 файлов – вверху и внизу странице появляется список листов. Выбрав следующую страницу Вы можете прочитать продолжение списка файлов.

Для каждого файла приводится краткая информация, автор, размер, дата последнего обновления, рейтинг популярности (количество скачиваний).

Нажав на ссылку Download, относящуюся к выбранному Вами файлу, вы можете скачать его для дальнейшего использования.

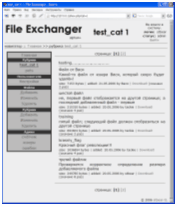


Рис.5. Перечень файлов рубрики TEST\_CAT 1.

**Пользователи**

В этом разделе только один пункт – «Настройка». Выбрав этот пункт вы можете изменить свой логин и пароль, количество файлов, одновременно отображаемых на странице, а также данные для связи с вами, такие как URL и ICQ (см.рис.6).

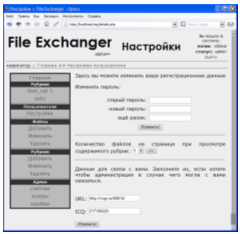


Рис.6. Страница настройки характеристик работы пользователя.

1. **Файлы**

Этот раздел содержит пункты, позволяющие изменять файлы предлагаемых рубрик:

**- Добавить**

В этом пункте вы можете выбрать файл для передачи в файлообменник, ввести его название, краткое описание. Файл можно поместить в одну из существующих рубрик (рис.7).

Если Вы не хотите, чтобы имя Вашего файла появлялось в перечне файлов рубрики, его можно пометить как скрытый. При этом доступ к файлу можно будет осуществлять по ссылке. В любом случае этот параметр всегда можно изменить через страницу изменения файлов.

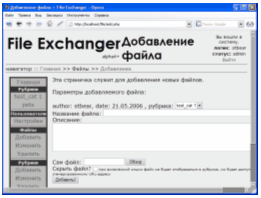


Рис.7 Страница добавления файла.

**- Изменить**

Выбрав пункт **Изменить**, Вы открываете страничку с перечнем файлов по рубрикам(рис.8). У вас есть возможность выбрать рубрику и файл для внесения изменений.

Если Вы простой пользователь, Вы можете редактировать только свои файлы. Если Вы администратор или модератор – Вам доступны файлы всех пользователей.

Для внесения изменений в файл открывается страница как на рис.9. Вы можете изменить название файла, его описание, поменять рубрику, открыть/скрыть файл, загрузить иную версию файла.

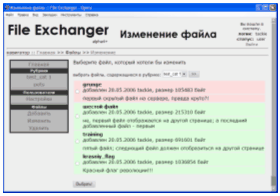


Рис.8. Страница со списком файлов для изменения.

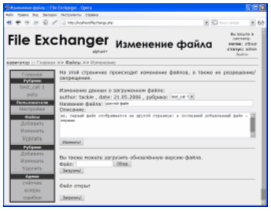


Рис.9 Страница изменения файла.

**- Удалить**

В перечне файлов (рис.10) Вы можете выбрать файл для удаления.

Если Вы простой пользователь, Вы можете удалять только свои файлы. Если Вы администратор или модератор – Вам доступны файлы всех пользователей.

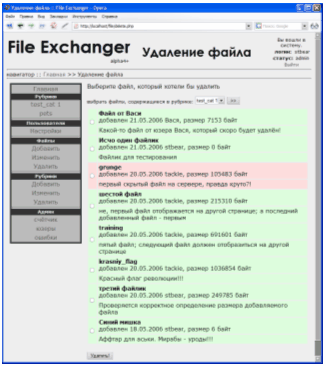


Рис.10. Страничка удаления файла.

**Рубрики**

Этот раздел доступен только администратору и содержит пункты, позволяющие изменять рубрики:

**- Добавить**

Здесь вы можете создать новую рубрику. Для этого нужно ввести её название – оно будет отображаться в списке рубрик – и описание – оно появляется при просмотре содержимого рубрики (рис. 11).

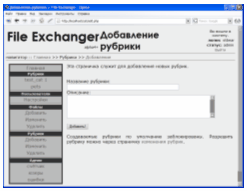


Рис.11 Страница добавления рубрики.

**- Изменить**

Для изменения рубрики вы выбираете её из списка (рис.12).

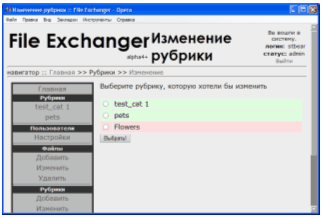


Рис.12 Страница со списком рубрик для изменения.

Открывается страница (рис.13), в которой вы сможете изменить название и описание рубрики, а также запретить или разрешить её.

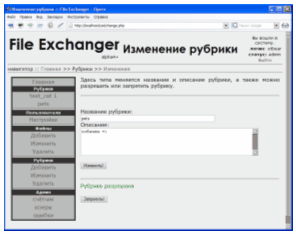


Рис.13 Страница изменения рубрики.

**- Удалить**

Для удаления рубрики её нужно выбрать из списка (рис.14).

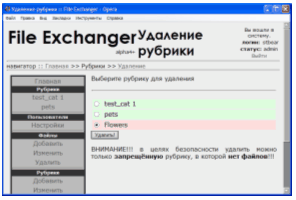


Рис.14 Страница удаления рубрик.

1. **Админ**

Раздел «Администрирование» создан для контроля эффективности работы файлообменника.

- Счетчик

Эта страничка позволяет просмотреть результаты работы счётчика. Состоит из 2х частей: в первой части (рис. 15) подсчитывается число обращений к отдельным страницам сайта. Во второй части (рис. 16) записываются IP-адреса зашедших на сайт.

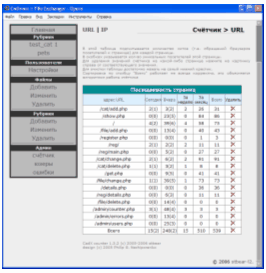


Рис.15 Просмотр числа зашедших на различные URL сайта.

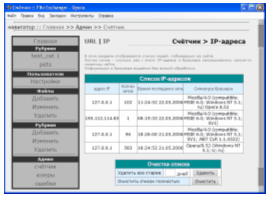


Рис.16 Страница для просмотра данных о зашедших на сайт посетителях.

- юзеры (пользователи)

На этой страничке (рис.17) можно просмотреть список пользователей и совершить над ними простые действия: повысить до модератора (синяя стрелка вверх), снять статус модератора (жёлтая стрелка вниз) или удалить (красный крестик). Также можно просмотреть дополнительную информацию о пользователе, щёлкнув на его логине (рис.18). Щёлкнув по IP-адресу (если это не 127.0.0.1), вы увидите данные о провайдере.

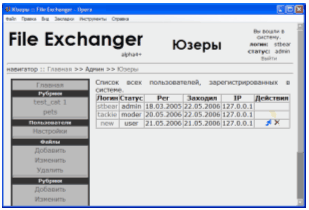


Рис.17 Страница для просмотра списка пользователей.

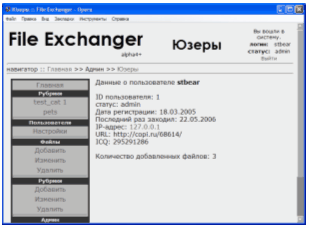


Рис.18 Страница для просмотра информации о конкретном пользователе.

- ошибки

На этой страничке (рис. 19) вы можете просмотреть лог ошибок, которые не удалось оправить тех.поддержке посредством SMTP-сервера, чтобы вы могли принять меры для их устранения.

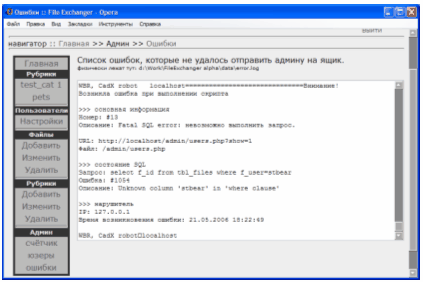


Рис.19 Страница для просмотра лог-файла ошибок.

# Заключение

В ходе курсовой работы был создан файлообменник, работающий под управлением PHP и MySQL.

В следующей версии файлообменника мне хотелось бы добавить:

1. Более совершенный счётчик, осуществляющий автоопределение пользователей и ботов, зашедших на сайт с выводом удобной статистики.

2. Работа с e-mail (регистрация, подтверждение, рассылка, аттачи)

3. Сокрытие переменных в адресной строке (как на 440hz.ru)

Задание на курсовой работе было выполнено полностью и в соответствии с техническим заданием.

# Список литературы

1. Люк Веллинг, Лора Томсон «MySQL. Учебное пособие.» Москва, Вильямс, 2005
2. http://php.net – официальный сайт PHP
3. http://phpclub.ru – сайт русских PHP-программистов.
4. http://php.ru – новый русскоязычный сайт о PHP
5. http://440hz.ru – персональный сайт Андрея Лугового, одного из создателей php.ru
6. http://kurepin.ru – персональный сайт Руслана Курепина
7. http://akelpad.net.ru – сайт редактора AkelPad
8. http://editplus.com – сайт редактора EditPlus
9. http://httpd.apache.org – официальная страница веб-сервера Apache
10. http://phpwars.net – PHP Wars, сайт о практических применениях PHP
11. http://pterodactyl.l2p.net – сайт Птеродактиля о PHP
12. http://webdeveloper.net.ru – ещё один форум для веб-разработчиков

1. Проект HTTP-сервера Apache – это попытка разработки и установки HTTP-сервера с открытыми исходными кодами для современных операционных систем, включая UNIX и Windows NT. Цель этого проекта – обеспечить безопасный, эффективный и масштабируемый сервер, который обеспечит HTTP-сервисы в соответствии с текущими стандартами HTTP.

   С апреля 1996 Apache стал наиболее популярным сервером интернета. В ноябре 2005 года обзор веб-серверов Netcraft обнаружил, что 70% сайтов интернета используют Apache, делая его таким образом более широко используемым, чем все другие веб-сервера вместе взятые. [↑](#footnote-ref-1)
2. EditPlus – это 32-разрядный текстовый редактор в среде Интернет, редактор HTML-кода и редактор программистов для Windows. В то время, как он служит отличной заменой Блокноту, он также предлагает ряд мощных средств для авторов веб-страниц и программистов.

   Подсветка синтаксиса для HTML, CSS, PHP, ASP, Perl, C/C++/C#, Java, JavaScript и VBScript. Кроме того, он может быть расширен для других языков программирования, основываясь на файлах со стандартным синтаксисом.

   Безупречный веб-браузер для предварительного просмотра HTML-страниц и команд FTP для загрузки локальных данных на FTP-сервер.

   Другие средства, включая панель HTML-инструментов, инструменты пользователя, номера строк, линейка, подсветка URL, автозавершение вырезок текста, выбор колонок, мощный механизм поиска/замены, возможность многократной отмены/повтора, проверка орфографии, переназначаемые «горячие клавиши» и др. [↑](#footnote-ref-2)