

Лабораторная работа № 3

Загрузка файлов на сервер

Тема работы: Изучение приемов работы с multipart-формами.

Цель работы: Изучить приемы загрузки файла на сервер через пользовательскую форму.

Продолжительность работы: 4 часа.

Ход работы:

1. Создайте файл, содержащий следующий код:

```
1.  <?
2.  $path=GetCWD()."/loadfiles";
3.
4.  if(!file_exists($path))
5.
6.
7.  die("<b>Пожалуйста, создайте папку <font
8.  color=red>".$path."</font> и <a href=##63;>
9.  повторите попытку загрузить файл</a>.</b>");
10.
11.
12. if(empty($_FILES['UserFile']['tmp_name']))
13.
14. echo
15. "<form method=post enctype=multipart/form-data>
16. Выберите файл: <input type=file name=UserFile>
17.
18. <input type=submit value=Отправить>
19. </form>";
20.
21. elseif(!is_uploaded_file($_FILES['UserFile']['tmp_
22. name']))
23. die("<b><font color=red>Файл не был загружен!
24. Попробуйте <a href=##63;>повторить
25. попытку</a>!</font></b>");
26. else
27. {
28. if(@!copy($_FILES['UserFile']['tmp_name'],
29. $path.chr(47).$_FILES['UserFile']['name']))
```

Путь к каталогу для
закачки файлов.
Проверка на
существование каталога
для загрузки.
При отсутствии папки,
выводится
соответствующее
сообщение.

Проверка, был ли выбран
файл.
Вывод формы для отправки
файла.
Приглашение к выбору
имени файла.

Если не удалось
загрузить файл.
Выводим сообщение.

Копируем файл на сервер.

30. die("Файл не был загружен!	Сообщение при ошибке
31. Попробуйте повторить	загрузки.
32. попытку!");	
33. else	Если все прошло без
34. echo	ошибок.
35. "<center>Файл \"<font	Вывод сообщения об
36. color=red>".\$_FILES['UserFile']['name']."\"	успешной загрузке.
37. успешно загружен на сервер!</center>".	
38. "<hr>".	
39. "Тип файла:	Выводим тип загруженного
40. ".\$_FILES['UserFile']['type']." ".	файла.
41. "Размер файла:	Размер файла в Кб.
42. ".round(\$_FILES['UserFile']['size']/1024,2)."	
43. кб.".	
44. "<hr><center>Загрузить ещё один	Предложение загрузить
45. файл! </center>";	еще один файл.
46. }	
47. ?>	
48.	

Разберем принцип его работы.

Файлы будут загружаться в каталог files, который должен находиться в том же каталоге, что и исполняемый сценарий. Для того чтобы узнать полный путь к каталогу, в котором выполняется сценарий, используется функция GetCWD (). После этого добавляется к этому значению строку /loadfiles и полученный путь заносится в переменную \$path, чтобы с ним было удобно работать. Если каталог loadfiles отсутствует в текущей директории, то выводится сообщение, что его необходимо создать и только после этого загружать файл на сервер. Для того чтобы повторить попытку загрузки файла, введена следующая ссылка:

href=?

Это код символа "?" (знак вопроса), в результате чего выполняется перезагрузка страницы и указывается, что параметров у нее нет.

Затем нужно вывести форму, но перед тем как это сделать, проверяется, не загружает ли пользователь в данный момент файл. Для этого проверяется

элемент `UserFile` в массиве `$_FILES`. Проверка этого элемента производится с помощью функции `empty()`. Если элемент `UserFile`, который передается через форму, пустой, то, следовательно, нужно вывести пользователю форму для выбора файлов. В противном случае имя файла уже указано и следует выполнять загрузку файла на сервер.

После прохождения всех проверок файл копируется на сервер с помощью функции:

```
copy($_FILES['UserFile']['tmp_name'],  
$path.chr(47).$_FILES['UserFile']['name'])
```

Данная функция используется в условном операторе, благодаря чему здесь выполняется проверка условия, был ли файл скопирован без ошибок:

```
if(!copy($_FILES['UserFile']['tmp_name'],  
$path.chr(47).$_FILES['UserFile']['name']))
```

При копировании используется случайное временное имя, которое генерирует сам PHP в переменной `tmp_name`, а реальное имя загружаемого файла хранится в переменной `name`. Функция `chr()` возвращает символ, который соответствует переданному в качестве аргумента коду из таблицы ASCII. В данном случае передается код 47, соответствующий символу косой черты `"/`. Таким образом, складывается полный путь к файлу, который копируется на сервер. Если при выполнении этой операции никаких ошибок не произошло, то выводится сообщение об ее успешном выполнении, а также информация о загруженном файле, такая, например, как тип файла и его размер.

2. Добиться безошибочного исполнения кода.
3. Измените скрипт так, чтобы осуществлялась проверка на существование на сервере файла, имеющего имя, совпадающее с именем копируемого файла, и предлагалось подтвердить или отказаться от копирования.

Требования к отчету: Отчет должен содержать детально прокомментированные тексты программ, скриншоты результатов работы программ и отдельно описания функций, примененных в скриптах.