

Глава 13

Создание доски бесплатных объявлений по продаже компьютерной техники

13.1. Постановка задачи

В этой главе мы разработаем доску бесплатных объявлений. Основные ее функции — показ уже существующих объявлений и добавлений новых объявлений. Но в этом случае в нашей доске объявлений не было бы ничего сложного, а значит — интересного. Поэтому нужно усложнить задачу. В то же время мы не будем ее очень сильно усложнять, чтобы процесс ее разработки не занял вечность. К тому же, доска бесплатных объявлений должна быть максимально комфортна в использовании и администрировании.

Начнем с просмотра. Выводить объявления мы будем не все сразу, а по пять штук — так удобнее будет их просматривать, особенно когда объявлений будет много. Не будет лишней и функция поиска — попробуйте просмотреть несколько сотен объявлений. А самое обидное заключается в том, что после просмотра 500 объявлений вы так и не найдете нужного.

Мы не будем сортировать объявления по категориям — материнские платы, процессоры и т. д. При необходимости вы сможете добавить поддержку категорий самостоятельно. Это несложно, а общую идею я вам подскажу. Пусть добавление категорий станет вашим домашним заданием.

Теперь поговорим о добавлении нового объявления. Во-первых, вместо скучного тега `<TEXTAREA>` мы будем использовать визуальный редактор SPAW2. Так наша доска объявлений будет намного привлекательнее.

При добавлении нового объявления мы должны учитывать следующее:

- ❑ Максимальная длина объявления — думаю, 1024 символов вполне хватит для текста объявления.
- ❑ Наличие кавычек в объявлении — текст сообщения мы должны обработать функцией `mysql_real_escape_string()`.
- ❑ Наличие в объявлении вредоносного кода — мы должны удалить теги, которые потенциально могут содержать вредоносный код (будем использовать функцию `strip_tags()`).

Кроме этого, дабы защитить нашу доску объявлений от спама, мы будем использовать двухуровневую системы защиты:

- ❑ Мы будем проверять, не содержит ли текст объявления слова из «черного списка». Черный список мы будем хранить в файле `black_list.txt`, и вы сможете легко добавить новые слова.
- ❑ При добавлении нового объявления текст объявления отправляется администратору по электронной почте. Администратор может или одобрить, или удалить объявление. Ссылки на одобрение или удаление объявления будут в тексте письма. Доска объявлений будет выводить только подтвержденные объявления.

13.2. Проектирование базы данных

Наша база данных будет состоять всего из одной таблицы `board`, создать которую можно следующим запросом:

```
CREATE TABLE board(  
  ID INT NOT NULL AUTO_INCREMENT ,  
  USERNAME CHAR( 40 ),  
  t TEXT,  
  APR INT DEFAULT 0,  
  PRIMARY KEY ( ID )  
)
```

Первое поле — это идентификатор объявления, данное поле будет использоваться для управления объявлением, а именно для одобрения объявления и его удаления. Второе поле — имя пользователя, добавившего объявление. Третье поле — текст объявления. Поле `APR` используется для одобрения объявления. Если это поле содержит 0, объявление не одобрено и не будет показано при выводе объявлений. При одобрении объявления это поле равно 1.

Можно также добавить два следующих поля: `DT` (не `DATE`, потому что слово `'DATE'` зарезервировано в `SQL`) и `CAT_ID`. Первое — это дата добавления объявления, что поможет при создании сценария автоматической очистки (удаления старых объявлений). Второе поле — это номер раздела, к которому относится объявление (компьютеры, ноутбуки, процессоры, материнские платы и т. д.). Таблица разделов может выглядеть так:

```
CREATE TABLE board_cats(  
  ID INT NOT NULL AUTO_INCREMENT ,  
  CNAME CHAR( 40 ),  
  PRIMARY KEY ( ID )  
)
```

Первое поле — идентификатор раздела. Идентификатор раздела используется для связи с таблицей объявлений. Второе поле — название раздела.

13.3. Структура системы

Доска объявлений будет состоять из следующих файлов:

- ❑ board.php — выводит объявления и форму добавления нового объявления;
- ❑ board_add.php — добавляет новое объявление;
- ❑ board_manager.php — сценарий управления доской объявлений;
- ❑ board_form.html — форма добавления нового объявления;
- ❑ config.php — файл конфигурации.

13.4. Разработка сценария board.php

Приступим к разработке сценария вывода доски объявлений. Напомню, что этот сценарий должен постранично выводить объявления, а также форму добавления нового объявления.

Первым делом нужно подключить файл конфигурации:

```
<?php
```

```
include_once "config.php";
```

Файл config.php содержит параметры доступа к серверу MySQL (листинг 13.1).

Листинг 13.1. Файл config.php

```
<?php
```

```
$dbhost = 'localhost';
```

```
$dbname = 'db';
```

```
$dbuser = 'root';
```

```
$dbpasswd = '';
```

```
?>
```

Далее нужно подключиться к серверу баз данных:

```
@mysql_connect($dbhost, $dbuser, $dbpasswd) or die("Ошибка подключения к БД");
```

```
@mysql_select_db($dbname) or die("Указанной базы данных не существует или  
недостаточно прав доступа");
```

Затем нужно определить количество объявлений на странице и общее количество объявлений в таблице:

```
// количество объявлений на странице
```

```
$N = 5;
```

```
echo "<h1>Доска объявлений v0.01</h1>";
```

```
// определяем общее количество записей в таблице объявлений
$r1=mysql_query("select count(*) as recs from board where apr=1");
$row = mysql_fetch_row($r1);
$rec = $row[0];           // общее количество записей в таблице
```

Далее будет следовать сценарий постраничного вывода таблицы, описанный в параграфе 8.5, но в немного измененном виде, поэтому комментариев не будет — их вы найдете в главе 8:

```
if (!isset($_GET['p'])) $p=0; else $page = $_GET['p'];
```

```
$records = $p * $N;
```

```
$q=>select * from board where apr=1 limit ".$records.", $N>;
```

```
echo "Всего записей $rec<br>";
```

```
$r=mysql_query($q);
```

```
$n = mysql_num_rows($r);
```

```
if ($p > 0) {
```

```
    $pg = $p - 1;
```

```
    echo "<a href=board.php?p=$pg>Назад</a>&nbsp;";
```

```
}
```

```
$p++;
```

```
if ($records+$N < $rec)
```

```
    echo "<a href=board.php?p=$p>Далее</a>";
```

```
// выводим записи
```

```
for ($i=0; $i<$n; $i++)
```

```
{
```

```
    $row=mysql_fetch_array($r);
```

```
    $username = $row[USERNAME];
```

```

echo "<p><table width=100%>
    <tr><td bgcolor=cccc66 width=10%>
    <font name=tahoma size=2>$username</font></td>";

echo "<td colspan=2 bgcolor=fffecb>
    <font name=tahoma size=2>$row[t]</td>
</tr></table>";
}

После вывода объявлений нужно вывести форму добавления нового объявления:
// форма добавления сообщения
echo '<p><table width=100% border=0>
<td bgcolor=black><font color=white>Новое сообщение:</td></table>';

// подключаем визуальный редактор SPAW2
include("spaw2/spaw.inc.php");

// создаем экземпляр редактора
$editor = new SpawEditor("t");

// добавлением объявления занимается сценарий board_add.php
// название формы – Main – можно использовать в JavaScript
echo '<p><form name=Main action=board_add.php method=post>
Ваше имя: <input type=text name=uname>';

// отображаем редактор
$editor->show();

// кнопка Отправить
echo '<input type=submit value=Отправить></form>';

?>

```

Вместо тега <TEXTAREA> мы используем визуальный редактор SPAW, описанный в главе 4. Наша доска объявлений будет выглядеть, как показано на рис. 13.1.

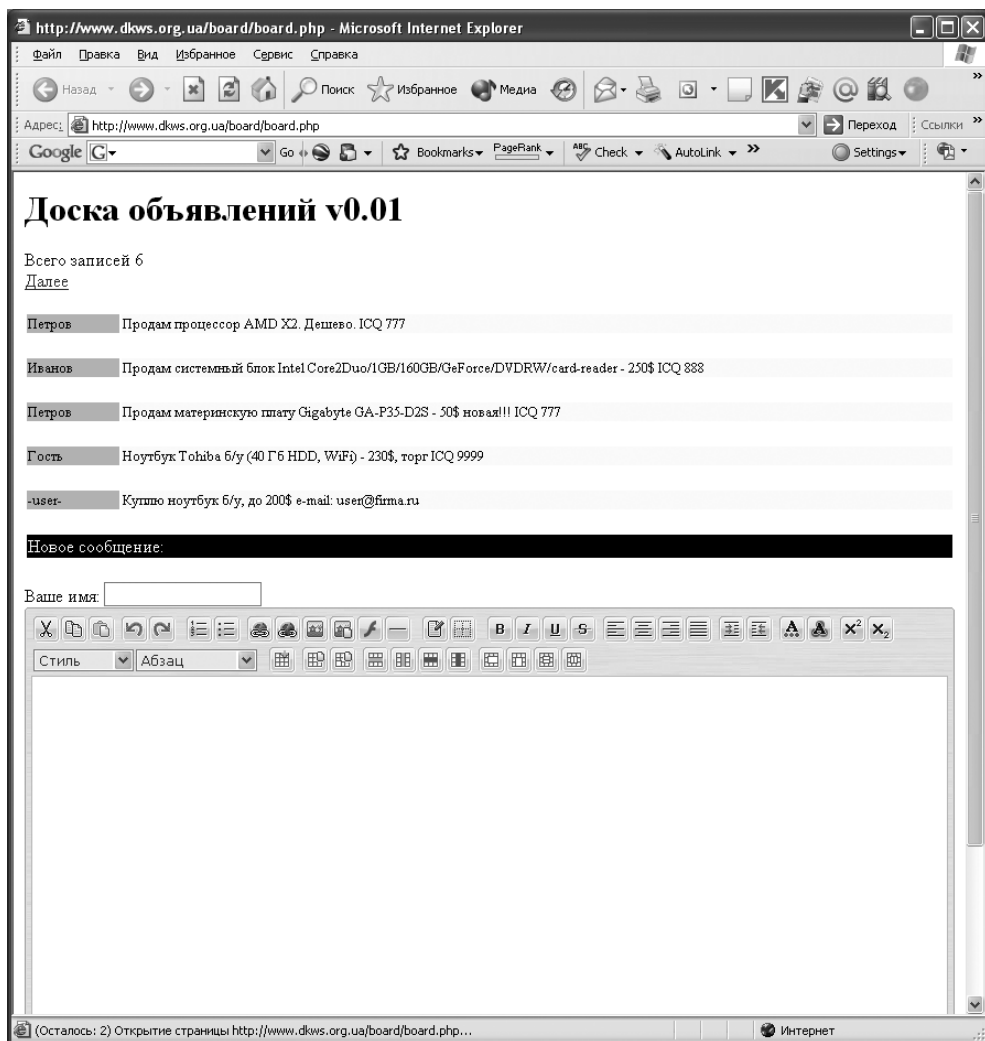


Рис. 13.1. Наша доска объявлений

При желании вы можете добавить событие `onSubmit`, проверяющее, ввел ли пользователь данные. Вот пример проверки поля имени пользователя:

```
if (this.document.Main.username.value=='') {
    alert('\Вы забыли указать имя\');
    return false;
} else { return true; }
```

Полный код сценария `board.php` представлен в листинге 13.2. Чтобы сценарий занимал не слишком много места в книге, я удалил из кода все комментарии.

Листинг 13.2. Сценарий board.php

```
<?php

include_once "config.php";

@mysql_connect($dbhost, $dbuser, $dbpasswd) or die("Ошибка подключения к БД");

@mysql_select_db($dbname) or die("Ошибка выбора БД");

$N = 5;

echo "<h1>Доска объявлений v0.01</h1>";

$r1=mysql_query("select count(*) as rec from board where apr=1");
$f = mysql_fetch_row($r1);
$rec = $f[0];

if (!isset($_GET['p'])) $p=0; else $page = $_GET['p'];

$records = $p * $N;

$q="select * from board where apr=1 limit ".$records.", $N";

echo "Всего записей $rec<br>";

$r=mysql_query($q);
$n = mysql_num_rows($r);

if ($p > 0) {
    $pg = $p - 1;
    echo "<a href=board.php?p=$pg>Назад</a>&nbsp;";
}

$p++;
```

Листинг 13.2. (продолжение)

```
if ($records+$N < $rec)
echo "<a href=board.php?p=$p>Далее</a>";

for ($i=0; $i<$n; $i++)
{
    $row=mysql_fetch_array($r);
    $username = $row[USERNAME];

    echo "<p><table width=100%>
        <tr><td bgcolor=cccc66 width=10%>
            <font name=tahoma size=2>$username</font></td>";

    echo "<td colspan=2 bgcolor=ffecb>
        <font name=tahoma size=2>$row[t]</td>
        </tr></table>";
}

echo '<p><table width=100% border=0><td bgcolor=black><font color=white>Новое
сообщение:</td></table>';

include("spaw2/spaw.inc.php");

$editor = new SpawEditor("t");

echo '<p><form name=Main action=board_add.php method=post>
Ваше имя: <input type=text name=uname>';

$editor->show();

echo '<input type=submit value=Отправить></form>';
?>
```

13.5. Добавление объявления — board_add.php

Настало время разработать сценарий добавления объявления — board_add.php. В самом добавлении записи нет ничего сложного — нужно выполнить всего два оператора:

```
$q = "insert into board values(0, \"\$uname\", \"\$t\", 0)";  
mysql_query($q) or die('Ошибка вставки');
```

Но сначала нам нужно проверить, можно ли добавлять объявление. Первым делом нужно удалить весь вредоносный код, избавиться от лишних HTML-тегов и «проэкранировать» кавычки, которые, возможно, будут в тексте сообщения или в имени пользователя:

```
// избавляемся от вредоносного кода  
$uname = strip_tags($uname);  
$uname = htmlspecialchars($uname);  
$uname = mysql_real_escape_string($uname);
```

```
$t = strip_tags($t, "<p><div><b><i><u><img><br>");  
$t = htmlspecialchars($t);  
$t = mysql_real_escape_string($t);
```

Обратите внимание, что мы удаляем из имени пользователя все HTML-теги, а из текста объявления — все теги, кроме <p>, <div>, , <i>, <u>, ,
. При необходимости вы можете добавить свои теги.

Затем нам нужно проверить, ввел ли пользователь вообще текст объявления:

```
if (strlen($uname)==0) $uname = "Гость";  
if (strlen($t)==0) die('Вы забыли указать текст сообщения');
```

Я обещал, что мы организуем проверку по черному списку. Черный список — это файл **black_list.txt**, содержащий список запрещенных слов, по одному в каждой строке. Чтобы цензура не забраковала мою книгу, приводить листинг этого файла я не стану. Чем больше этот файл, тем выше его эффективность.

Функция, проверяющая, есть ли запрещенное слово в списке, будет называться **is_in_black_list**. Ей нужно передать два параметра — имя файла черного списка и текст сообщения:

```
function is_in_black_list($f, $txt)  
{  
    $list = file($f);
```

```
// просматриваем массив запрещенных слов
```

```
// если функция strpos() возвратила значение, отличное от false,
```

```
// то слово есть в черном списке, мы прерываем выполнение функции
// и возвращаем true
foreach ($list as $k=>$v) if (strpos($txt, $v)!==false) return true;

// запрещенных слов в тексте объявления не найдено, возвращаем false
return false;

}
```

Использовать эту функцию нужно так:

```
if (is_in_black_list("black_list.txt", $t))
die("Текст объявления содержит нецензурные выражения");
```

Если текст не содержит нецензурных выражений, его можно добавить в базу данных:

```
$q = "insert into board values(0, \"\$uname\", \"\$t\", 0)";
```

```
mysql_query($q) or die('Ошибка вставки');
```

```
// получаем ID последней вставленной записи
```

```
$ID = mysql_insert_id();
```

После этого нужно вывести сообщение о том, что объявление добавлено в таблицу:

```
echo "<p>Ваше объявление было успешно добавлено в базу данных.
```

```
Пока администратор не одобрит ваше объявление, оно не появится на нашей доске";
```

Администратору нужно отправить e-mail-сообщение с текстом нового объявления. Администратор может или одобрить, или удалить объявление:

```
$msg = "Новое объявление:
```

```
$t
```

```
Вы можете одобрить его: " . $site . "board_manager.php?a=approve&id=$ID" . " или
удалить: " . $site . "board_manager.php?a=del&id=$ID";
```

```
// Отправляем сообщение
```

```
mail($admin, "Новое объявление", $msg);
```

Полный код сценария **board_add.php** представлен в листинге 13.3 (без комментариев).

Листинг 13.3. Сценарий board_add.php

```
<?php
require_once "config.php";

function is_in_black_list($f, $txt)
{
    $list = file($f);

    foreach ($list as $k=>$v) if (strpos($txt, $v)!=false) return true;

    return false;
}

$admin = "dhsilabs@mail.ru";
$site = "http://dkws.org.ua/board/";

mysql_connect($dbhost, $dbuser, $dbpasswd) or
die("Не могу подключиться к БД");

mysql_select_db($dbname) or die("Не могу выбрать БД");

$uname = $_POST['uname'];
$t = $_POST['t'];

$uname = strip_tags($uname);
$uname = htmlspecialchars($uname);
$uname = mysql_real_escape_string($uname);

$t = strip_tags($t);
$t = htmlspecialchars($t);
$t = mysql_real_escape_string($t);

if (strlen($uname)==0) $uname = 'Гость';
if (strlen($t)==0) die(Вы забыли указать текст сообщения');
```

Листинг 13.3. (продолжение)

```
if (is_in_black_list("black_list.txt", $t)) die("Текст объявления содержит  
нецензурные выражения");  
  
$q = "insert into board values(0, \"\$uname\", \"\$t\", 0)";  
mysql_query($q) or die('Ошибка вставки');  
  
$ID = mysql_insert_id();  
  
echo "<p>Ваше объявление было успешно добавлено в базу данных.  
Пока администратор не одобрит ваше объявление, оно не появится на нашей доске";  
  
$msg = "Новое объявление:  
  
$t  
  
Вы можете одобрить его: " . $site . "board_manager.php?a=approve&id=$ID" . " или  
удалить: " . $site . "board_manager.php?a=del&id=$ID";  
  
mail($admin, "Новое объявление", $msg);  
?>
```

13.6. Управление объявлениями — board_manager.php

Сценарий `board_manager.php` предельно прост. Если администратор выбрал действие **approve** (одобрить), то нужно установить поле **ARP** объявления в 1:

```
if ($action === "approve") {  
  
    $q = "update board set apr = 1 where id = $ID";  
    @mysql_query($q) or die ("MySQL error");  
    echo "Сообщение $ID было одобрено";  
  
}
```

Если же администратору не понравилось объявление, он может его удалить:

```
if ($action === "del") {  
    $q = "delete from board where id = $ID limit 1";  
    @mysql_query($q) or die ("MySQL error");  
    echo "Сообщение $ID было удалено";  
  
}
```

С помощью LIMIT мы ограничиваем удаление только 1 записи — на всякий случай. Полный код этого сценария представлен в листинге 13.4 (без комментариев).

Листинг 13.4. Сценарий board_manager.php

```
<?php  
include_once "config.php";  
  
mysql_connect($dbhost, $dbuser, $dbpasswd) or  
    die("Не могу подключиться к БД");  
  
mysql_select_db($dbname) or die("Не могу выбрать БД");  
  
$action = $_GET['a'];  
$ID = $_GET['id'];  
  
$ID = strip_tags($ID);  
$ID = htmlspecialchars($ID);  
$ID = mysql_real_escape_string($ID);  
  
$action = strip_tags($action);  
$action = htmlspecialchars($action);  
$action = mysql_real_escape_string($action);  
  
if ($action === "approve") {  
  
    $q = "update board set apr = 1 where id = $ID";  
    mysql_query($q);  
    echo "Объявление $ID было одобрено";
```

Листинг 13.4 *(продолжение)*

```
}

if ($action === "del") {

    $q = "delete from board where id = $ID limit 1";
    mysql_query($q);
    echo "Объявление $ID было удалено";

}
?>
```

Полный исходный код доски объявлений вы найдете в каталоге **glava13** на компакт-диске.