

Проверяем на сайте пользователь или нет

Когда я столкнулся с этой проблемой, решение пришлось искать довольно долго. В статье я постараюсь, по возможности подробно, описать, каким образом можно проверить находится ли авторизованный пользователь на сайте или нет.

Для начала, немного о проблеме. Есть сайт. На сайте есть система авторизации. Данные пользователей хранятся в таблице users (id, username, password – самый простой вариант). Когда пользователь входит на сайт, он вводит имя пользователя и пароль, программа сопоставляет введенные пользователем в форме авторизации данные с данными в таблице пользователей и открывает для пользователя возможности, доступные только зарегистрированным пользователям. [Как сделать авторизацию на PHP](#) я уже описывал, поэтому сразу перейдем к тому, как можно проверить, на сайте ли авторизованный пользователь, или уже ушел.

Для решения данной проблемы нам понадобится: немного знаний в области работы с базой данных, PHP и JQuery (для написания AJAX-запросов). Тем кто не знаком с JQuery, хочу сразу сказать – ничего страшного там нет, если вы читали статьи о том, как сделать добавление товара в корзину интернет-магазина без перезагрузки страницы, то вы уже немного познакомились с тем, как писать простые AJAX запросы.

Я буду считать, что система авторизации у вас уже налажена, поэтому подробно на ней останавливаться не буду. Далее нам понадобится еще одна таблица, назовем ее online_users. В ней создаем всего два поля user_id – идентификатор пользователя из таблицы users, и last_time – в котором будет храниться метка времени. Зачем она нужна, объясню немного позже, но уже можно догадаться. Формат поля last_time – VARCHAR(255):

```
CREATE TABLE IF NOT EXISTS `online_users` ( `user_id` smallint(6) NOT NULL, `last_time` varchar(255) NOT NULL, ) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
```

Итак таблица у нас есть, теперь нужно написать PHP функцию которая будет добавлять записи в эту таблицу:

```
function user_online($user_id) {  
//сначала удаляем уже существующую запись пользователя:  
mysql_query("DELETE FROM online_users WHERE user_id='$user_id'");  
//теперь добавляем данные:  
mysql_query("INSERT INTO online_users SET user_id='$user_id', last_time='".mktime()."'");  
}
```

Во втором запросе мы используем функцию mktime(), которая возвращает метку времени и очень удобна для операций с датами, поскольку дальше нам нужно будет считать разницу времени между последней записью в базу и текущим временем.

Дальше нам понадобится JavaScript функция, назовем ее также user_online(), которая будет с некоторой периодичностью обращаться к PHP файлу, который и будет вызывать функцию user_online(). Для этого, нам понадобится AJAX и JQuery:

```
function user_online(user_id) {  
$.post( "/online_user.php", {user_id:user_id}, function(data){});  
window.setTimeout(user_online, 10000);  
}
```

Первая строка в данной функции производит AJAX запрос к файлу online_user.php, в котором будет вызываться PHP-функция user_online(). А вот вторая строка – специальная функция setTimeout, которая, как бы правильно выразиться, производит задержку времени, то есть вызывает функцию, имя которой стоит первым аргументом, через количество миллисекунд, которые указаны во втором аргументе. Если проще, JavaScript функция user_online() вызывает сама себя через 10 секунд, и так, пока пользователь не закроет окно браузера, то есть браузер каждые 10 секунд сообщает, что пользователь все еще на сайте.

Ну и на последок, нам нужна еще одна php функция, которая будет определять на сайте пользователь, или нет, назовем ее is_online():

```
function is_online($user_id) {
//извлекаем записи из таблицы онлайн-пользователей:
$result=mysql_query("SELECT * FROM online_users WHERE user_id='$user_id'");
//переводим запрос в ассоциативный массив:
$online_user=mysql_fetch_assoc($result);
//а теперь самое интересное. Функция будет возвращать TRUE если пользователь онлайн, или FALSE в
противном случае:
//для начала проверяем не пустой ли массив $online_user:
if (empty($online_user)) {
return FALSE;
}
else {
//создаем метку времени:
$current_timestamp = mktime();
//проверяем разницу между временем в данный момент, и последней записью в базе данных:
if (($mtime-$online_user['last_time'])>=10) {
//если последняя запись была более 10 секунд назад
return FALSE;
}
else {
return TRUE;
}
}
}
```

Вот и все. Теперь для того, чтобы определить на сайте пользователь или нет, нужно вызвать функцию is_online с идентификатором нужного пользователя. Если она вернула FALSE (is_online(\$user_id)) - то пользователь уже ушел с сайта, если TRUE, то пользователь находится на сайте. Ну и конечно не забыть на всех страницах сайта добавить JavaScript функцию user_online() и передать в нее идентификатор пользователя.