

Bazaar – php example code – part 16 – administration of users

Article focus on bulding blocks for enabling user administration by page admin. How user list is introduced in page admin management page and way how to change user data in this view.



























Introduction of user management



Administrator of the page must be able display info about user, change their role or delete them. User password is sensitive information and can not be save as plain text. Our database store only secure hashes of them.

Basic functionality for management of the user can be described this way. Page admin on their own admin page has user administration section with list of all users. These information are displayed in a table. At the end of appropriate row is link pointing to second context page with option for change category of the user, edit of user data or enable to remove a user.

Only editing of user data lead to another page very similiar to the users profile page, where through the form can be data reedit.

Whole process depict following sequence of screenshots.

III. Manage List of active bazaar users and roles						
users_id	username	nickname	write_date	user_role	manage	
 	 	 	 	 		
1	test	test	2020-10-28 15:26:12	user		
2	test2		2020-10-24 17:42:45	user		
9	test3	test3	2020-10-24 18:20:02	user		
10	admin	admin	2020-10-30 18:22:02	admin		
11	user	user	2020-10-31 11:02:23	admin		
12	user2	user2	2020-11-09 14:15:31	user		
13	user3	user3	2020-11-09 14:51:40	user		
14	user5	user5	2020-11-09 15:29:14	user		

 1 2 

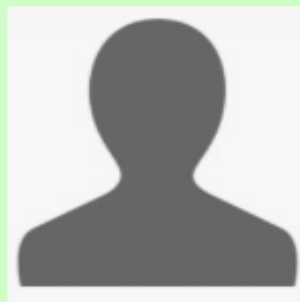
Section for display user info on administrators page



bazaar

[Bazaar admin page](#)[Return to main shop page](#)


Please select what you will do.



Are you sure perform selected operation with bazaar user?

User ID: 1

Name: test

score image

Please select your operation

click to select option ▼

make_admin

make_user

change_user_data

delete_user

[Back to admin page.](#)


Visit us on CDesigner.eu

Subscribe to newsfeed




Content on linked page for appropriate user management



 (1) 55 €



 [Back to admin page.](#)

User_role of succesfully logged user with name **test** is **user** .

No records found.

User ID is only one filed that can not be changed, all other can admin of page edit:

User ID:

1

User name:

test

E-mail:

test@test.sk

Further user data data:

*Nickname:

test

First name:

test

Last name:

Adress in form - Street Nr.:

test address

City:

city

ZIP code in form XXXXX:

03852

Part of user data editing page available from invocation on user management page – change_user_data option

Code implementation part

Our solution consist from part generating user list in form a table in admin.php script:

```
/*
*****
*****
*/
/*      III. Showing registered user table with option for
adminster them      */
/*
*****
*****
*/
$results_per_page_3 = 8;
//calculate pagination information
$cur_page = isset($_GET[,page']) ? $_GET[,page'] : 1;
// results per page default declater as 5 on top of page and c
hanged in submitt part after reset button handling $results_pe
r_page = 5;
$skip = (($cur_page -1) * $results_per_page_3);
// first question to database table for obtaining number of p
ublished items in a database – obtain value for $total
$sql = "SELECT * FROM bazaar_user"; // read in reverse order o
f score – highest score first
$output_for_number_rows_count_3 = mysqli_query($dbc, $sql); //
query database
$total_3 = mysqli_num_rows($output_for_number_rows_count_3);
//get number of rows in databse
// querying bazaar_category for listed category items

//older approach without SORT functionality

// $sql = „SELECT * FROM bazaar_item ORDER BY item_add_date DE
SC LIMIT $skip, $results_per_page“; // read in reverse order
of score – highest score first
/**
```

* SORTING – PART II. Here is into sql request implemented along which field and how ascend or desc is output ordered

*/

```
if(isset($_GET['sort_by']) && isset($_GET['order'])) {  
    // take a data from GET link generated by adminscript  
    $sort_by = htmlspecialchars($_GET['sort_by']);  
    $order = htmlspecialchars($_GET['order']);  
    // debug echo „sort_by“.$sort_by;  
    // debug echo „order“.$order;  
    if(($sort_by == „users_id“) && ($order == „1“)) { // along  
name and ASC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY users_id ASC  
LIMIT $skip, $results_per_page_3“;  
    };  
    if(($sort_by == „users_id“) && ($order == „-1“)) { // along  
name and DESC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY users_id DESC  
LIMIT $skip, $results_per_page_3“;  
    };  
    if(($sort_by == „username“) && ($order == „1“)) { // along  
name and ASC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY username ASC  
LIMIT $skip, $results_per_page_3“;  
    };  
    if(($sort_by == „username“) && ($order == „-1“)) { // along  
name and DESC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY username DESC  
LIMIT $skip, $results_per_page_3“;  
    };  
  
    if(($sort_by == „nickname“) && ($order == „1“)) { // along  
name and ASC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY nickname ASC  
LIMIT $skip, $results_per_page_3“;  
    };  
    if(($sort_by == „nickname“) && ($order == „-1“)) { // along  
name and DESC order  
        $sql = „SELECT * FROM bazaar_user ORDER BY nickname DESC  
LIMIT $skip, $results_per_page_3“;  
    };  
};
```



```

        if(($sort_by == „write_date“) && ($order == „1“)) { // along name and ASC order
            $sql = „SELECT * FROM bazaar_user ORDER BY write_date ASC LIMIT $skip, $results_per_page_3“;
        };
        if(($sort_by == „write_date“) && ($order == „-1“)) { // along name and DESC order
            $sql = „SELECT * FROM bazaar_user ORDER BY write_date DESC LIMIT $skip, $results_per_page_3“;
        };
        if(($sort_by == „user_role“) && ($order == „1“)) { // along price and ASC order
            $sql = „SELECT * FROM bazaar_user ORDER BY user_role ASC LIMIT $skip, $results_per_page_3“;
        };
        if(($sort_by == „user_role“) && ($order == „-1“)) { // along price and DESC order
            $sql = „SELECT * FROM bazaar_user ORDER BY user_role DESC LIMIT $skip, $results_per_page_3“;
        };
        if(($sort_by == „default“)) { // along category and DESC order

            $sql = „SELECT * FROM bazaar_user ORDER BY users_id ASC LIMIT $skip, $results_per_page_3“;
        };
    } else { // first run without ordering – no get link generated
        $sql = „SELECT * FROM bazaar_user ORDER BY users_id ASC LIMIT $skip, $results_per_page_3“; // read in reverse order of score – highest score first
    }
    // processing table output form bazaar_category
    echo „<h4>III. Manage List of active bazaar users and roles</h4>“;
    //echo „<br>“;
    //echo „    <button class=“btn btn-secondary btn-lg “ onclick=“location.href=\\'managecategory.php\\'“ type=“button“> Create new category-subcategory -> </button>“;
    echo „<br>“; echo „<br>“;
    if($output = mysqli_query($dbc, $sql)){

```

```

        if(mysqli_num_rows($output) > 0){ // if any record ob
tained from SELECT query
            // create table output
            echo „<table>“; //head of table
                echo „<tr>“;
                    echo „<th>users_id <br /><a id=“SORT“ href
=“admin.php?sort_by=users_id&order=1“> <img id=“arrow“ src
=“./images/arrowup.png“> </a>
                        <a id=“SORT“ href=“admin.php?sort_by=users
_id&order=-1“> <img id=“arrow“ src=“./images/arrowdown.png
“> </a></th>“;
                            echo „<th>username <br /><a id=“SORT“ href
=“admin.php?sort_by=username&order=1“> <img id=“arrow“ src
=“./images/arrowup.png“> </a>
                                <a id=“SORT“ href=“admin.php?sort_by=usern
ame&order=-1“> <img id=“arrow“ src=“./images/arrowdown.png
“> </a></th>“;
                                    echo „<th>nickname <br /><a id=“SORT“ href
=“admin.php?sort_by=nickname&order=1“> <img id=“arrow“ src
=“./images/arrowup.png“> </a>
                                        <a id=“SORT“ href=“admin.php?sort_by=nickn
ame&order=-1“> <img id=“arrow“ src=“./images/arrowdown.png
“> </a></th>“;
                                            echo „<th>write_date <br /><a id=“SORT“ hr
ef=“admin.php?sort_by=write_date&order=1“> <img id=“arrow“
src=“./images/arrowup.png“> </a>
                                                <a id=“SORT“ href=“admin.php?sort_by=write
_date&order=-1“> <img id=“arrow“ src=“./images/arrowdown.p
ng“> </a></th>“;
                                                    echo „<th>user_role <br /><a id=“SORT“ hre
f=“admin.php?sort_by=user_role&order=1“> <img id=“arrow“ s
rc=“./images/arrowup.png“> </a>
                                                        <a id=“SORT“ href=“admin.php?sort_by=user_
role&order=-1“> <img id=“arrow“ src=“./images/arrowdown.pn
g“> </a></th>“;
                                                            echo „<th>manage</th>“;

echo „</tr>“;

```

```

        while($row = mysqli_fetch_array($output)){ //next
rows outputed in while loop
        echo " <div class=\"mailinglist\"> " ;
        echo „<tr>“;
            echo „<td>“ . $row[,users_id'] . „</td>“;
            echo „<td>“ . $row[,username'] . „</td>“;
            echo „<td>“ . $row[,nickname'] . „</td>“;
            echo „<td>“ . $row[,write_date'] . „</td>“
;

            echo „<td>“ . $row[,user_role'] . „</td>“;
            // removal line with removing link line

            // echo „<td colspan=\"1\"> Manage entry:
</td>“; // description on first line
            echo „<td colspan=\"1\"><a id=\"DEL\" href=
f=\"manageuser.php?users_id='.$row[,users_id'] . ,&username
='
            . $row[,username'] . ,&user_role=
'. $row[,user_role'] .'><center><img id=\"next\" src=\"./images/
manageuser.png\"></center></a></td></tr>“; //construction of GE
Table link

            // for removecategory.php input

        echo „</tr>“;
        echo " </div> " ;
    }
    echo „</table>“;
    //count nuber of pages total
    $num_pages_3 = ceil($total_3 / $results_per_page_
3);

    //generate navigational page links if we have mor
e than one page

    if($num_pages_3 > 1) {
        $user_search = „“; // not implemented yet, th
en set as clear values
        if(empty($sort_by)) { // if not obtained by g
et then default order is applied

```

```

        $sort_by="default";
    };
    if(empty($order)) { // if not obtained by get
then default order is applied
        $order="1";
    };

    // included function for pagination generatio
n function stored in functions.php page
    echo generate_page_links($user_search, $sort_
by, $order, $cur_page, $num_pages_3);
    echo „<br><br>“;
}
echo „<br />“;
// Free result set
mysqli_free_result($output);
} else{
    echo „There is no benchmark result in chart. Pleas
e wirite one.“; // if no records in table
}
} else{
    echo „ERROR: Could not able to execute $sql. “ . mysql
i_error($dbc); // if database query problem
}
// Close connection
mysqli_close($dbc);
?>

```

Example now contains ordering functionality, this part will be explained with details in later published articles.

All functionality for specific user management is located in script manageuser.php. This script is invoked through GET link from admin.php page.

Script obtain data from GET link after invocation, but second submitting mechanism with post enable selection from other specific operation.

Changing role from user to admin and vice versa, deletion of user are created with appropriate SQL statement executed at SQL database.

Interesting part of script is responsible for page redirecting:

```
switch ($operation) {
    case „make_admin„:
        $sql = „UPDATE bazaar_user SET user_role =
,admin‘ WHERE users_id = $users_id LIMIT 1“;
        // execute SQL
        mysqli_query($dbc, $sql);
        // confirm executed command
        echo „<p> User <strong>‘ . $username . „</
strong> with id <strong>‘ . $users_id . „</strong> was succesf
ully promoted as page <b>admin<b>. </p>“;
        break;
    case „make_user„:
        $sql = „UPDATE bazaar_user SET user_role = ,
user‘ WHERE users_id = $users_id LIMIT 1“;
        // execute SQL
        mysqli_query($dbc, $sql);
        // confirm executed command
        echo „<p> User <strong>‘ . $username . „</st
rong> with id <strong>‘ . $users_id . „</strong> was succesful
ly promoted as page <b>user<b>. </p>“;
        break;
    /* to do */ case „change_user_data„: // -> make solution fo
r update other users data – link profile page for another user
        // next part of code redirect to manageuse
rdata.php page for administration of appropriate users data by
admin

?>
<script type = „text/javascript“>
```

```
function Redirect() {  
    window.location = „./manageuse  
rdata.php?users_id=<?php echo $users_id ?>&username=<?php echo  
    $username ?>“; // send users_id of user to edit must send to  
manageuserdata.php to be able edit appropriate user and not us  
er extracted from session informatin  
}  
    document.write(„You will be redi  
rected to manageuserdata.php page.“);  
    setTimeout(,Redirect()’, 0);  
  
</script>
```

After submitting request for change_user_data, javascript redirect currently logged admin on manageuserdata.php and generate GET link with ID and Username.

Content of manageuserdata.php or whole script of project can be obtained from github [here](#).

Conclusion

Previous text show example of way how to add user management functionality to our bazaarap. This approach demonstrate classical way for user management. Real world application need further security hardening with eliminating GET data exchange. POST method with SESSION verification provide further level of validation.