Bazaar – php example code – part 17 – ordering tables by category

Article focus on way how to display content of list inserted in table order by activating visual elements in header of appropriate table. Our focus is enable reorder listening ascending or descending way along selected column.

Expectations from ordering system

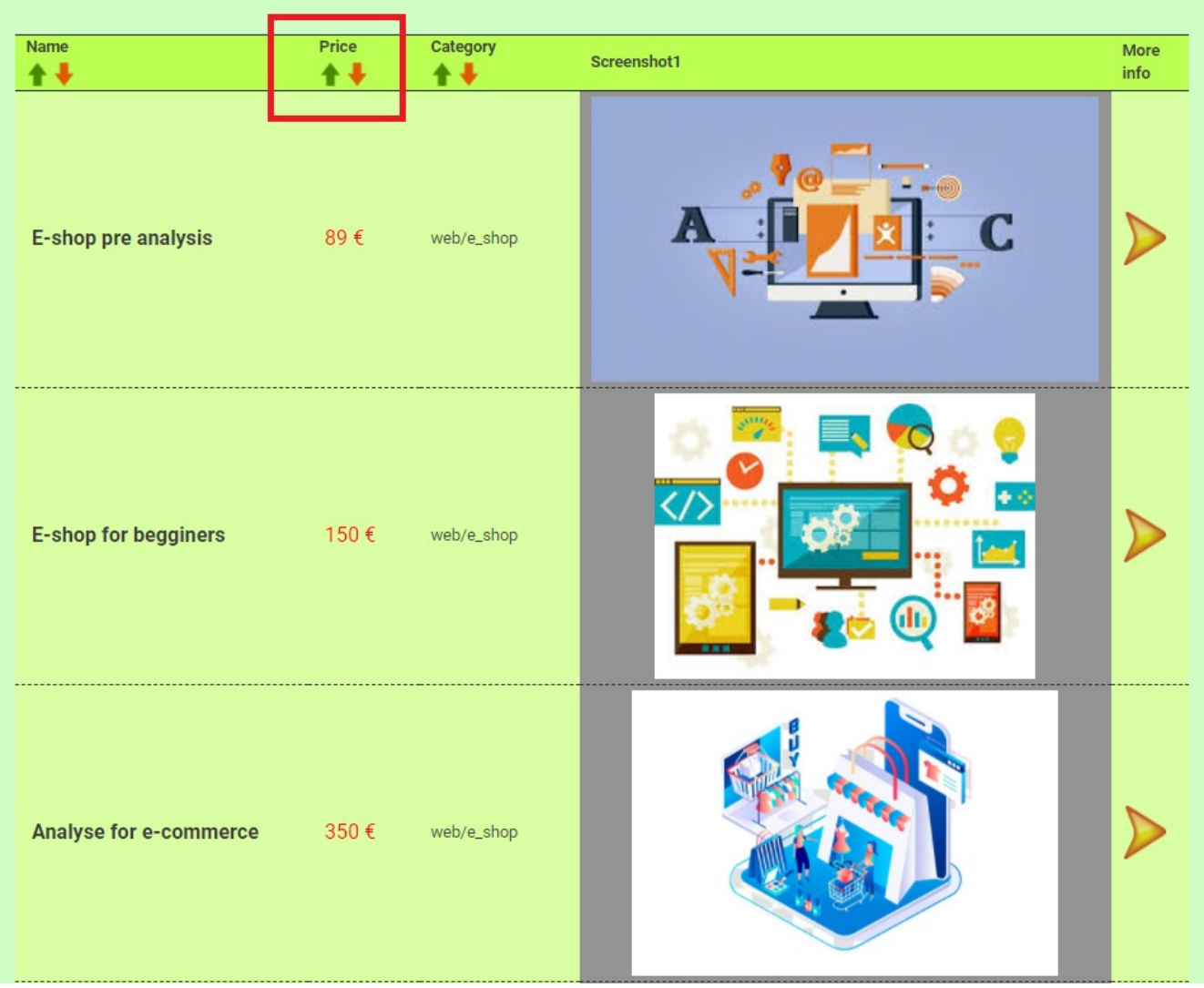
Content of the lists display usefull information for page users. Quick way how to obtain relevant data is ordering them ascending or descending by interesting category. For implementation of ordering ability in our tables we must:

- add links (best way is to associate them with nice graphics that is intuitive for the user) for ascending or descending ordering in header of our table
- generate GET links for submitting necessary data for ordering functionality
- prepare appropriate scripts handling calculations for number of displayed data, use LIMIT functionality in SQL queries
- generate pagination links at the left bottom part of our lists for browsing functionality through our results

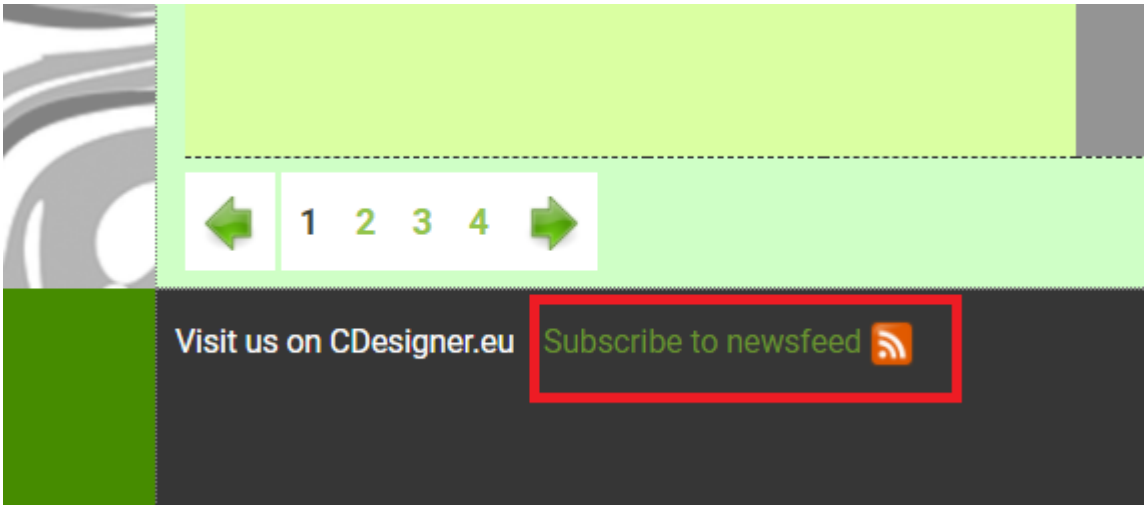
How ordering works, brief introduction – our code must take in mind number of items for display and number of list items for

display. After calculations number of browsable page, function for generating pagination links must generate appropriate links. User select appropriate page or next/ previous page. Data from user are send to themeself as GET links. From obtained data outr code prepare SQL request containing LIMIT keyword fro slecting only rows from appropriate calculated position and with expected number of selected items. In table output ist next displayed selected data and links on page must be displayed with respect of actually selected page and ordering.

Next pictures visualy display our expectations/ implementations of this functionality.



Links for ordering output data along selected category



Pagination on the bottom of the page

Code implementing pagination on page

First discused code is responsible for generation of links at the bottom part of a page.

Our code page named functions.php contains function invoked in other pages handling pagination functionality. Contentn of this function is:

```
<?php
/**
 * Changelog
 * v 1.0 – first working version, update 6.11.2020
 * v 1.1 – for ordering option was updated header of function by adding $sort –> $sort_by and new $order
 *              $odrder variable can by 1 for ASC order and -1 for DESC order, update 28.11.2020
 */
function generate_page_links($user_search, $sort_by, $order, $cur_page, $num_pages) { //($user_search, $sort_by, $order, $cur_page, $num_pages);
```

```
    $page_links = „“;
    echo „<br>“;

    // if this is not first in row, we need generate the „prev
ious“ link
    if  ($cur_page > 1) {
        $page_links .= ‚<a id=“pagination“ href=“, . $_SERVER[
‚,PHP_SELF‘] . ‚?usersearch=‘
        .$user_search . ‚&sort_by=‘ . $sort_by . ‚&order=‘ . $
order .’&page=‘ . ($cur_page – 1) . ‚“><img src=“./images/prev
ious_icon.png“ alt=“previous image“ width=“30″ height=“30″></a
>‘;
    } else {
        $page_links .= ‚<span id=“pagination“><img src=“./imag
es/previous_icon.png“ alt=“previous image“ width=“30″ height=“
30″></span> ‚;
    }
    // Loop through the pages generating the page numbered lin
ks
    for($i = 1; $i <= $num_pages; $i++) {
        if  ($cur_page == $i) {
        $page_links .= ‚<span id=“pagination“>‘ . $i. ‚</span>
‘; // span inline element mark non a tag (unlinked number) as
pagination for further formating by css
        } else {
        $page_links .= ‚<a id=“pagination“ href=“, . $_SERVER[
‚,PHP_SELF‘] . ‚?usersearch=‘
        .$user_search . ‚&sort_by=‘ . $sort_by . ‚&order=‘ . $
order .’&page=‘ . $i . ‚“>‘ . $i . ‚</a>‘;
        }
    }
    // If this page is not last in row, generate „next“ link
    if  ($cur_page < $num_pages) {
        $page_links .= ‚<a id=“pagination“ href=“, . $_SERVER[
‚,PHP_SELF‘] . ‚?usersearch=‘
        .$user_search . ‚&sort_by=‘ . $sort_by . ‚&order=‘ . $
order .’&page=‘ . ($cur_page + 1) . ‚“><img src=“./images/next
_icon.png“ alt=“next image“ width=“30″ height=“30″></a>‘;
    } else {
        $page_links .= , <span id=“pagination“><img src=“./ima
ges/next_icon.png“ alt=“next image“ width=“30″ height=“30″></s
```

```
pan>‘;
    }

    return $page_links;
}
?>
```

In our code are at current time not implemented all expeted functionality. User serarch is not passed but all other arguments are used for passing necessary data for genration of pagination links.

Oru code must handle situations when there are no more page next or previous, and must be able to generate all other numbered page links. All links are selectable on focus by user.

Real implementation on pages

Next part of index page is responsible for ordering data and iterating through all available pages with content. Our code relevant for this functions is highlighted by blue color.

```
– output  omited –
/*********************************************************/
/*             Output in paginated form
                          */
/*********************************************************/
 /***
  *  Display pagination on the page – part included to listeni
ng in this area
  */
/* Attempt MySQL server connection. Assuming you are running M
ySQL
server with default setting (user ‚root‘ with no password) */
$dbc = mysqli_connect(DB_HOST, DB_USER, DB_PW, DB_NAME);
//GET data for pagination send to page herself
```

```
//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) * $results_per_page);
// Check connection
if($dbc === false){
    die(„ERROR: Could not connect to database – stage of artic
le listing. “ . mysqli_connect_error());
}
// first  question to database table for obtaining number of p
ublished items in a database – obtain value for $total
$sql = „SELECT * FROM bazaar_item WHERE published=“.“,1´“.“ AN
D cart_number=“.“,0´“.“ ORDER BY item_id DESC „;  // read in r
everse order of score – highest score first
$output_for_number_rows_count = mysqli_query($dbc, $sql); // q
uery database
$total = mysqli_num_rows($output_for_number_rows_count);    //
get number of rows in databse


//older approach without SORT functionality read all rows (dat
a) from guestbook table in „test“ database
// $sql = „SELECT * FROM bazaar_item WHERE published=“.“,1´“.“
 AND cart_number=“.“,0´“.“ ORDER BY item_id DESC LIMIT $skip,
$results_per_page“;  // read in reverse order of score – highe
st score first
/**
 * SORTING – PART II. Here is into sql request implemented alo
ng which filed and how ascend or desc is output ordered
 */
if(isset($_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(($sort_by == „name“) && ($order == „1“)) { // along nam
e and ASC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
```

```
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY name_of_item ASC LIMI
T $skip, $results_per_page“;
    };
    if(($sort_by == „name“) && ($order == „-1“)) { // along na
me and DESC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY name_of_item DESC LIM
IT $skip, $results_per_page“;
    };
    if(($sort_by == „price“) && ($order == „1“)) { // along pr
ice and ASC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY price_eur ASC LIMIT $
skip, $results_per_page“;
    };
    if(($sort_by == „price“) && ($order == „-1“)) { // along p
rice and DESC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY price_eur DESC LIMIT
$skip, $results_per_page“;
    };
    if(($sort_by == „category“) && ($order == „1“)) { // along
 category and ASC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY subcategory_id ASC LI
MIT $skip, $results_per_page“;
    };
    if(($sort_by == „category“) && ($order == „-1“)) { // alon
g category and DESC order
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY subcategory_id DESC L
IMIT $skip, $results_per_page“;
    };
    if(($sort_by == „default“)) { // along category and DESC o
rder
        $sql = „SELECT * FROM bazaar_item WHERE published=“.“,
1′“.“ AND cart_number=“.“,0′“.“ ORDER BY item_id DESC LIMIT $s
kip, $results_per_page“;
    };
} else {  // first run without ordering – no get link generate
d
```

```
    $sql = „SELECT * FROM bazaar_item WHERE published=“.“,1´“.
“ AND cart_number=“.“,0´“.“ ORDER BY item_id DESC LIMIT $skip,
 $results_per_page“;  // read in reverse order of score – high
est score first
}
/*************************************************************
*********/
/*         Output in Table – solution 1 – for debuging data
from database         */
/*************************************************************
*********/
// if data properly selected from guestbook database tabele
echo „<br><br>“;
echo „<h4>Latest added items for you! </h4>“;
echo „<br>“;
/***
 *  Obtaining wished number of item per page – option for sele
ct
 */
?>
<form  method=“post“ action=“<?php echo $_SERVER[,PHP_SELF‘];
?>“>

<div class=“form-group“>
<label> Set expected number of items per page -5 is default:</
label>
<input list=“number_per_page“ name=“number_per_page“ placehold
er=“please select or write nr.“>
      <datalist id=“number_per_page“> <!– must be converted in
 subcategory_id in script – marked with (*) –>
          <option value=“5″>
          <option value=“10″>
          <option value=“15″>
          <option value=“20″>
          <option value=“50″>
          <option value=“100″>
        </datalist>



      <button type=“submit“ name=“nr_of_pages“ class=“btn btn-
```

```
warning“> Use selected number of pages! </button>
</div>
</form>
<?php
echo „<br>“; echo „<br>“;
if($output = mysqli_query($dbc, $sql)){
    if(mysqli_num_rows($output) > 0){  // if any record obtain
ed from SELECT query
        // create table output
        echo „<table>“; //head of table
            echo „<tr>“;
                //echo „<th>id</th>“;
                // functionality for ordering result
                /**
                 * SORTING – PART I. Here are generated GET li
nks for UP/DOWN ordering by appropriate category
                 */
                echo ,<th>Name  <br /><a id=“SORT“ href=“index
.php?sort_by=name&amp;order=1″> <img id=“arrow“ src=“./images/
arrowup.png“> </a>
                                    <a id=“SORT“ href=“index.php?s
ort_by=name&amp;order=-1″> <img id=“arrow“ src=“./images/arrow
down.png“> </a> </th>‘; //order 1 up -1 down
                echo ,<th>Price <br /><a id=“SORT“ href=“index
.php?sort_by=price&amp;order=1″> <img id=“arrow“ src=“./images
/arrowup.png“> </a>
                                    <a id=“SORT“ href=“index.php?s
ort_by=price&amp;order=-1″> <img id=“arrow“ src=“./images/arro
wdown.png“> </a></th>‘;
                echo ,<th>Category <br /><a id=“SORT“ href=“in
dex.php?sort_by=category&amp;order=1″> <img id=“arrow“ src=“./
images/arrowup.png“> </a>
                                    <a id=“SORT“ href=“index.php?s
ort_by=category&amp;order=-1″> <img id=“arrow“ src=“./images/a
rrowdown.png“> </a> </th>‘;
                echo „<th>Screenshot1</th>“;
                echo „<th>More info</th>“;


            echo „</tr>“;
        while($row = mysqli_fetch_array($output)){ //next rows
```

```
 outputed in while loop
            echo “ <div class=\“mailinglist\“> “ ;
            echo „<tr>“;
                //echo „<td>“ . $row[,item_id‘] . „</td>“;
                echo „<td class=\“item_name\“>“ . $row[,name_o
f_item‘] . „</td>“;
                echo „<td class=\“price\“>“ . $row[,price_eur‘
] . “ € </td>“;
                                /* convert category_id in to categ
ory and subcategory */
                                $subcategory_id = $row[,subcategor
y_id‘];
                               $category_idsupl    = „“ ;
                               $subcategory_idsupl = „“ ;
                               // (*) – conversion of category an
d subcategory into category%id

                                    //create SELECT query for cate
gory and subcategory names from database
                                     $sql_supl = „SELECT category,
 subcategory FROM bazaar_category WHERE subcategory_id = „.“,$
subcategory_id'“ ;
                                    /*$output_supl = mysqli_query(
$dbc, $sql_supl);
                                    $row_supl = mysqli_fetch_array
($output_supl);
                                    $category_id    = $row_supl[,c
ategory‘] ;
                                    $subcategory_id = $row_supl[,s
ubcategory‘] ;
                                    echo „<td>“ . $category_id.“/“
.$subcategory_id.“</td>“;*/
                                    // execute sql and populate da
ta list with existing category in database
                                    if($output_supl = mysqli_query
($dbc, $sql_supl)){
                                        if(mysqli_num_rows($output
_supl) > 0){  // if any record obtained from SELECT query
                                            while($row_supl = mysq
li_fetch_array($output_supl)){ //next rows outputed in while l
oop
```

```
                                                $category_idsupl
  = $row_supl[,category‘] ;
                                                $subcategory_idsup
l = $row_supl[,subcategory‘] ;


                                            }

                                              // Free result set
                                            mysqli_free_result($ou
tput_supl);
                                        } else {
                                            echo „There is no souc
h                                                       category-
subcategory in category table. Please correct your error.“; //
 if no records in table
                                        }
                                    } else{
                                        echo „ERROR: Could not abl
e to execute $sql. “ . mysqli_error($dbc); // if database quer
y problem
                                    }
                echo „<td>“ . $category_idsupl.“/“.$subcategor
y_idsupl.“</td>“;

                    $image_location = IMAGE_PATH.$row[,screens
hot1‘];
                echo „<td id=\“gray_under_picture\“> <img  src
=\“$image_location\“ alt=\“ screenshot of product primary \“
height=\“250\“> </td>“;
                echo ,<td colspan=“1″><a id=“DEL“ href=“item.p
hp?item_id=‘.$row[,item_id‘]. ,“><img id=“next“ src=“./images/
next.png“>   </a></td></tr>‘; //construction of GETable link
            echo „</tr>“;
            echo “ </div> “ ;
        }
        echo „</table>“;
        //count nuber of pages total
        $num_pages = ceil($total / $results_per_page);
```

```
        //generate navigational page links if we have more tha
n one page

        if($num_pages > 1) {
            $user_search = „“; // not implemented yet, then se
t as clear values
            if(empty($sort_by)) { // if not obtained by get th
en 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 generation fun
ction stored in functions.php page
            echo generate_page_links($user_search, $sort_by, $
order, $cur_page, $num_pages);
            echo „<br><br>“;
        }
        // Free result set
        mysqli_free_result($output);
    } else{
        echo „There is no item for sell. Please add one.“; //
if no records in table
    }
} else{
    echo „ERROR: Could not able to execute $sql. “ . mysqli_er
ror($dbc); // if database query problem
}
// Close connection
mysqli_close($dbc);
```

From our code excerpt can be visible these parts: 1) parts responsible for calculating number of necessary pages for display, 2) part handling ordering logic responsible for creation of SQL queries, 3) table part responsible for display

of data and generating links and part 4) invoking function for generating pagination information at the bottom part of page.

Conclusion

Our code fulfil our expectation. Because we used functional approach, maintainability is hard. One big problem that we gained is visible if we work with two or more pagination/ ordering functionality. information about current page is passed by GET link and is one for all ordered tables. When we order along one table, also other tables are iterated by the same page. But this is acceptable solution. because focus of user is only for actually handled part of page.

For futher improvement is way how to maintaing ordering along two or more pagination links best part of focus. Also further abstraction this functionality with OOP implementation can lead to a more reusable part of code.

Full code for further study can be obtained from github here.