ALLOWING VISITOR TO SET WHERE VALUES IN A VIEWER - Sep 5th, 2022

In this example you're trying to limit the records shown to those that match the value of a field called project_title, so first you'll need to create a list field called project_title in your table.

Then, we'll assume that in any record the values for that field can be either Test event 1 or Test event 2.

At the top of your page in the load records calls use the code:

```
$where = "";
if (@$FORM['where'] == 'a') { $where = 'Test Event 1'; }
if (@$FORM['where'] == 'b') { $where = 'Test Event 2'; }

 list($$your_tableRecords, $$your_tableMetaData) = getRecords(array(
    'tableName'   => '$your_table',
    'where'       => " project_title = '$where' ",
  ));
```

And in the body, the form code would be:

```
<form method="POST" action="<?php echo $_SERVER['PHP_SELF'] ?>">
<select name="where">
<option value="">Select</option>
<option value="a">Event 1</option>
<option value="b">Event 2</option>
</select>

<input type="submit" name="submit" value="Choose an exhibition to View">
</form>
```

And Here's Another example of limiting the records shown on a viewer page.
This one has three possible options and incorporates check boxes in a single record editor to determine which options are available to the visitor:

At the top of the viewer page in the list records calls

```
$where = "";
if (@$FORM['where'] == 'a') { $where = "want_this_book = '1'"; }
if (@$FORM['where'] == 'b') { $where = "get_better_copy = '1'"; }
if (@$FORM['where'] == 'c') { $where = "get_better_copy = '1' OR want_this_book = '1'"; }

 list($booksRecords, $booksMetaData) = getRecords(array(
    'tableName'   => 'books',
```

```
    'loadUploads' => true,
    'allowSearch' => true,
    'where'       => $where,

    // 'debugSql' => true,
  ));

?>
```

In The Body:

```
<form method="POST" action="">
<select name="where">
<option value="">Make a selection from this pulldown menu </option>
<?php if ($common_informationRecord['sort_by_want_these_books'] == '1'):?>
<option value="a">Search For Books I Want But
Don't Have</option><?php endif ?>
<?php if ($common_informationRecord['sort_by_better_copy_of_these_books'] == '1'):?>
<option value="b">Search For Books I
Want A Better Copy Of</option><?php endif ?>
<?php if ($common_informationRecord['sort_by_want_these_books'] == '1' &&
$common_informationRecord['sort_by_better_copy_of_these_books'] == '1' ):?>
<option value="c">Search For All The Books I
Want To Add To My Collection</option><?php endif ?>
</select>

<input type="submit" name="submit" value="Sort By Your Selection">
</form>
```

NOTE: According to Dave Edis from Interactive Tools, the reason to do it that way by passing a letter (or word or code, it doesn't matter) and testing for that instead of just specifying the order by in the option value directly is because you don't want users to be able to pass MySQL directly into your program or it's a security risk.

You can expand this idea to create as complex a set of criteria s required.

ANOTHER NOTE: Jason Sauchuk offered this mini tutorial on the use of single and double quotes. He said:

In PHP, a string that is set with double quotes can have variables inserted directly into it without concatenation.

example:

```
<?php
  $myName = "Jason";
  $greeting  = "Hello, my name is $myName";
?>
```

The value of $greeting would be:
Hello, my name is Jason

If we took this same piece of code and used single quotes:

```
<?php
  $myName = "Jason";
  $greeting  = 'Hello, my name is $myName';
?>
```

The value of $greeting would be:
Hello, my name is $myName

PHP will not put the value of $myName into the string.

The materials on this web site have been created for use with CMS Builder content management software. CMS Builder software is published and licensed for use by InteractiveTools.com. Please contact Interactive Tools for information on the downloading of the software or the purchasing of licenses.

Terms of Service