

DMIT2025 – PHP/MYSQL

PHP/MYSQL BLOG PROJECT

Note: Many characters may be misinterpreted (such as the single quote ') if you just copy/paste from a PDF. Please recognize this and change any characters necessary.

This week, class will build a basic blog utilizing PHP. As opposed to the Simple Blog where we used a flatfile to store content, this will utilize a MySQL database.

The blog will be a basic CRUDS application once again, but we will also explore some new techniques:

- Pagination: dividing blog posts into separate pages. This is a common technique seen in catalogs, classified ads, online magazines, etc. Examples:
 - Most forums: <http://www.webdesignforums.net/forum/general-web-design-discussion/>
 - Google results pages.
- Emoticons: We will store special characters in our message field that will be converted to emoticon graphics (aka. "smilies") on output. This is a common technique seen in online forums, social networking sites, text messaging, etc.
- MySQL TimeStamp: MySQL has its own date format we can use.
- <select> navigation for edit.

Why build another blog? The blog is simply dynamic storage of text content and teaches us the same techniques that can be used on many other types of application such as content management.

DATABASE TABLE:

In **phpMyAdmin** >> you MUST select your database first! >> open **SQL tab** >> paste in the code below >> click "Go". Feel free to change the table name if you desire.

Paste this in phpMyAdmin's SQL tab after selecting your database.

```
CREATE TABLE IF NOT EXISTS `blog` (  
  `title` varchar(50) NOT NULL,  
  `message` text NOT NULL,  
  `timedate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `bid` int(11) NOT NULL AUTO_INCREMENT,  
  PRIMARY KEY (`bid`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Viola.....you have a new table.

LOCAL DIRECTORY STRUCTURE:

DMIT2025 – PHP/MYSQL

It's up to the student how the directory will be organized, but here is an example.



- /admin/ will include any secured files (insert, edit, login).
- /includes/ will be for a common header, footer, and mysql_connect files.
- Similar to past projects, please start off by creating your common /includes/ files such as header, footer, mysql_connect, and then proceed to create your admin/insert.php file.
- Please use the Bootstrap app template we have gone through to create your common header/footer files.

EXTERNAL FUNCTIONS FILE

Since we have a few different functions we will be using, let's create an external file for that.

In your /includes/ directory, please create a file called `_functions.php`. Then in your index.php, you can include that using this:

```
<?php
include ("includes/header.php"); // should already be here
include ("includes/_functions.php");//here we path to the _functions file.
?>
```

INSERT



DMIT2025 – PHP/MYSQL

- *Note: The insert page relies on knowledge gained in previous modules, so please refer to Validation and MySQL basics if necessary.*
- Build a basic form with 2 elements for blog Title and Message.
- Create some basic validation and error messages (min and max length).
- Upon successful submission, the content entered into the form is inserted into the database table using a MySQL INSERT statement.
 - Show a success message if validation has passed.

EMOTICON EDITOR:

The emoticon editor is a feature of several emoticons that will put a unique set of text characters into the message field. These characters will then be converted back to emoticon graphics on output.

Necessary javascript will be added to <head> of your document (probably in your includes/header.php).

```
<script type="text/javascript">
function emoticon(text) {
    var txtarea = document.name_of_form.name_of_textarea; // must change these
    text = ' ' + text + ' ';
    if (txtarea.createTextRange && txtarea.caretPos) {
        var caretPos = txtarea.caretPos;
        caretPos.text = caretPos.text.charAt(caretPos.text.length - 1) == ' ' ? text + ' ' : text;
        txtarea.focus();
    } else {
        txtarea.value += text;
        txtarea.focus();
    }
}
</script>
```

Make sure you change the name_of_form.name_of_textarea to your form element names!

Then, in your insert.php file, please add the emoticon graphics and link them to the javascript function and pass the necessary text characters to the function.

This code is HTML.

```
<a href="javascript:emoticon('D')"></a>
```

DMIT2025 – PHP/MYSQL

If this works, then clicking the emoticon icon will then enter the text characters into the message field. You can research the appropriate characters sets for the emoticons. Feel free to use a different emoticon set if you like as well. Please make sure you don't try any character sets that include single quotes.

OUTPUT

The output will be either the public *index.php* file in the project root, or else a file called *output.php* in the root which can be included in the index page.

TIMESTAMP

As you may see in phpMyAdmin, the MySQL timestamp should put the time/date in automatically everytime a blog entry is added. This can easily be formatted on output using something like the following:

```
<?php
    $thisDate = strtotime($row["timedate"]); // retrieve data from DB
    $thisDate = date("F j, Y g:i a", $thisDate); // modify using date() function
    // Please edit the format you want and write it to the browser how you like.
    // output $thisDate later in your markup
?>
```

EMOTICON OUTPUT

Please create your own function that will use `str_replace` to search for the characters in the message output and replace with the emoticon images.

```
<?php

//move this to includes/_functions.php

function addEmoticons($thisMessage){
    $thisEmoticon = "<img src=\"emoticons/icon_biggrin.gif\" />";
    $thisMessage = str_replace(":D", $thisEmoticon, $thisMessage);

    $thisEmoticon = "<img src=\"emoticons/icon_smile.gif\" />";
    $thisMessage = str_replace(":)", $thisEmoticon, $thisMessage);

    $thisEmoticon = "<img src=\"emoticons/icon_sad.gif\" />";
    $thisMessage = str_replace(":(", $thisEmoticon, $thisMessage);

    return $thisMessage;
} // end addEmoticons
?>
```

DMIT2025 – PHP/MYSQL

Thus can then be called from your MySQL SELECT query as you retrieve results:

```
<?php
// change your output where you retrieve the blog message to this
$thisMessage = htmlspecialchars ($row['message']);
$thisMessage = addEmoticons($thisMessage);
?>
```

// you can then write \$thisMessage within any formatting you like.

CLICKABLE LINKS

This feature will automatically create a link from any URL or email address you enter into your message field. It will be called and have a parameter passed to it the same way as the emoticons were (please figure it out).

```
<?php

//move this to includes/_functions.php

function makeClickableLinks($text)
{
    $text = " " . $text; // fixes problem of not linking if no chars before the link

    $text = preg_replace('((((f|ht){1}tps?:\\V\\V)[-a-zA-Z0-9@:%_\\+.~#?&\\V\\V=]+)/i',
        '\\1<a href=\\1">\\1</a>', $text);
    $text = preg_replace('([[:space:]]()([{}])(www.[-a-zA-Z0-9@:%_\\+.~#?&\\V\\V=]+)/i',
        '\\1<a href="http://\\2">\\2</a>', $text);
    $text = preg_replace('([_\\0-9a-z-]+@([0-9a-z-]+[0-9a-z-]+\\.)+[a-z]{2,3})/i',
        '\\1<a href="mailto:\\1">\\1</a>', $text);

    return trim($text);
} // end makeClickableLinks

?>
```

Now you can call this function the same way we did with the addEmoticons. Do this after we have retrieved the message field from the DB but before we write it to the browser.

PAGINATION

DMIT2025 – PHP/MYSQL

Most pagination scripts are fairly laborious and involve another MySQL table. This is one I found in a forum that works well for most instances and is simple to setup.

There are 3 sections of code:

1. One goes BEFORE your main query and is used to create a variable called \$limstring.
2. Your main query that outputs the results will then include the \$limstring variable at the end.
3. The 3rd section will go wherever you want the page links; probably just below your main results.

Please copy/paste this code, make the necessary changes, and see the results. You **MUST** have more blog entries than the \$limit variable in order to see results, so add enough entries or change that variable (or both).

```
<?php

////////// pagination
$getcount = mysqli_query ($con,"SELECT COUNT(*) FROM blog_prep");
$postnum = mysqli_result($getcount,0);// this needs a fix for MySQLi upgrade; see custom function below
$limit = 5;
if($postnum > $limit){
    $tagend = round($postnum % $limit,0);
    $splits = round(($postnum - $tagend)/$limit,0);

    if($tagend == 0){
        $num_pages = $splits;
    }else{
        $num_pages = $splits + 1;
    }

    if(isset($_GET['pg'])){
        $pg = $_GET['pg'];
    }else{
        $pg = 1;
    }
    $startpos = ($pg*$limit)-$limit;
    $limstring = "LIMIT $startpos,$limit";
}else{
    $limstring = "LIMIT 0,$limit";
}

// MySQLi upgrade: we need this for mysqli_result() equivalent
function mysqli_result($res, $row, $field=0) {
    $res->data_seek($row);
    $datarow = $res->fetch_array();
    return $datarow[$field];
}

?>
```

DMIT2025 – PHP/MYSQL

then in your existing main query, just add the LIMIT string (\$limstring) like:

```
// this is pseudocode. Figure out how to change your main query to add the $limstring variable.  
  
SELECT * FROM yourtable  
ORDER BY whatever  
$limstring
```

Then wherever you want the page navigation links, do this:

```
<?php  
////////// pagination links: perhaps put these BELOW where your results are echo'd out.  
if($postnum > $limit){  
echo "<strong>Pages:</strong> &nbsp;&nbsp;&nbsp;";  
$n = $pg + 1;  
$p = $pg - 1;  
$thisroot = $_SERVER['PHP_SELF'];  
if($pg > 1){  
echo "<a href=\"\$thisroot?pg=$p\"><< prev</a>&nbsp;&nbsp;&nbsp;";  
}  
for($i=1; $i<=$num_pages; $i++){  
if($i!= $pg){  
echo "<a href=\"\$thisroot?pg=$i\">$i</a>&nbsp;&nbsp;&nbsp;";  
}else{  
echo "$i&nbsp;&nbsp;&nbsp;";  
}  
}  
if($pg < $num_pages){  
echo "<a href=\"\$thisroot?pg=$n\">next >></a>";  
}  
echo "&nbsp;&nbsp;&nbsp;";  
}  
// ambitious students may want to reformat this. Perhaps use Bootstraps pagination markup.  
////////// end pagination  
  
?>
```

DMIT2025 – PHP/MYSQL

ADMIN SECTION CONTINUED

The student will then finish all other necessary functions of the blog such as edit, delete, and a login that secures all admin pages.

EDIT

An emoticon editor should also exist here (same as insert).

One unique feature will be a <select> being used instead of links to select which entry to edit. The following javascript should be added to the header.

```
<script>
function go()
{
    box = document.getElementById('entryselect'); // gets form element by the id.
    destination = box.options[box.selectedIndex].value;
    if (destination) location.href = destination;
}
</script>
```

The <options> of the <select> should each have a value that is the current page with its ID (*which will have to be written with a PHP/MySQL query loop*).

The HTML output:

```
<select name="entryselect" id="entryselect" onchange="go()" >
    <option value="/~username/blog/admin/edit.php?id=8">Waiting for a
    meeting</option>
    <option value="/~username/blog/admin/edit.php?id=9">Wednesday afternoon
    class</option>
</select>
```

PHP pseudocode:

```
<select name="entryselect" class="form-control" onchange="go()" >

    <?php
    //here you can rewrite a MySQL query and loop that echo's out <option> tags
    to populate the <select>
    // the value of the <option> is the URL with the query string (?id={number})
    which corresponds to the primary key value of that blog entry. The text the
    user sees should be the blog entry title.
    ?>
</select>
```


DMIT2025 – PHP/MYSQL

MySQLBlog Admin

[Blog](#) | [Insert](#) | [Edit](#) |

Edit Blog Entries

    

Testing

Title:

Entry:

[Logout](#)