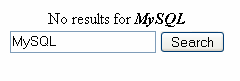
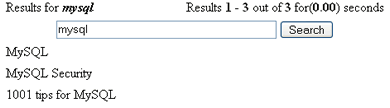
http://www.dmxzone.com/go/14246/free-how-to-build-a-php-search-engine/



However, if results are found, the page should look like this:



TABLE

CREATE TABLE `news` (  
        `id` tinyint(4) NOT NULL auto\_increment,  
        `title` varchar(200) NOT NULL default '',  
        `article` text NOT NULL,  
        UNIQUE KEY `id` (`id`),  
        FULLTEXT KEY `title` (`title`,`article`)  
             ) ENGINE=MyISAM AUTO\_INCREMENT=1 DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1;

Save the following page as **Search.php.** You may want to use the POST method instead of the GET method for your form since it is more secure. For this tutorial I chose using the GET method since this allows me to check if the search engine is working properly.

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
< html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
< meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>Untitled Document</title>  
</head>  
<body>  
<h1>Search Engine with PHP</h1>  
< form method="**get**" action=" ">  
    <input type="text" name="**search\_term**" title="Search…" value="Search...">  
    <input type="submit" name="search" title="Search Now!  
    "value="Search" class="searchbutton" />  
</form>  
</body>  
</html>

As you can see this is just simple code for a form with a text field and a button.

Now let's make the connection string. Open a new PHP document, name it **connectionstring.php** and save it in the same directory in which **Search.php** is located.

<?php  
      $hostname\_MyConnectionPHP = "localhost";  
      $database\_MyConnectionPHP = "database";  
      $username\_MyConnectionPHP = "usrname";  
      $password\_MyConnectionPHP = "mypassword";  
      $connections = mysql\_connect($hostname\_MyConnectionPHP, $username\_MyConnectionPHP, $password\_MyConnectionPHP) or die ( "Unabale to connect to the database" );  
?>

Replace "database", "usrname" and "mypassword" with the respective data that you use for the MySQL database.

The following code is the core of our search engine. We will calculate the search time using a function, make a database connection with another function, and finally we will check for matches with an appropriate algorithm running a query to select a table to look for matches. **Add the following code to your Search.php page**:

<?PHP

function getmicrotime()  
{  
    list($usec, $sec) = explode(" ", microtime());  
    return ((float)$usec + (float)$sec);  
}

$connection\_string = dirname(\_\_FILE\_\_) . "/**connectionstring.php**";  
require\_once($connection\_string);  
mysql\_select\_db("test") or die ( 'Unable to select database.' );

$RESULTS\_LIMIT=10;

if(isset($\_GET[**'search\_term'**]) && isset($\_GET[**'search\_button'**]))  
{  
    $search\_term = $\_GET[**'search\_term'**];

if(!isset($first\_pos))  
    {  
        $first\_pos = "0";  
    }

$start\_search = getmicrotime();

$sql\_query = mysql\_query("SELECT \* FROM news WHERE MATCH(title,article) AGAINST('$**search\_term'**)");

// many matches (too many matches cause returning of 0 results)

    if($results = mysql\_num\_rows($sql\_query) != 0)  
            {  
                $sql =  "SELECT \* FROM news

WHERE MATCH(title,article) AGAINST('$search\_term') LIMIT $first\_pos, $RESULTS\_LIMIT";  
                  $sql\_result\_query = mysql\_query($sql);     
            }  
    else  
            {  
                  $sql = "SELECT \* FROM news

WHERE (title LIKE '%".mysql\_real\_escape\_string($search\_term).

"%' OR article LIKE '%".$search\_term."%') ";

                  $sql\_query = mysql\_query($sql);  
                  $results = mysql\_num\_rows($sql\_query);  
                  $sql\_result\_query = mysql\_query("SELECT \* FROM news

WHERE (title LIKE '%".$search\_term.

"%' OR article LIKE '%".$search\_term."%') LIMIT $first\_pos, $RESULTS\_LIMIT ");  
            }  
    $stop\_search = getmicrotime();  
    $time\_search = ($stop\_search - $start\_search);  
}  
?>

On the second line we initialize function to calculate the time for the search.

The **require\_once** function initializes a connection to the database – we call the **connectionstring.php** file that we created earlier.

$RESULTS\_LIMIT variable is used to specify how many results to display per page.

Next we use an "if clause" to check if any phrase or word is set in the search field form. This is done with **$search\_query**. Then we check if our marker for page listing exists with **$first\_pos** variable – if not we start from the first page.

Then we need to initialize a SQL query – simply with **$sql\_query**. mysql\_query is a keyword in MySQL. Unfortunately, using that keyword has a major drawback. MySQL does not display results of the phrases that are found in the database too often as it would display too many results. That is why we do a second check. The algorithm counts the number of results with the **$results** variable and if it is zero we run second query with a standard field search of the table with LIKE %...%.

Back to Search.php where we have inserted our form. Modify the code as follows:

<?PHP  
if($results != 0)  
{  
?>     
   <table border="0" cellspacing="2" cellpadding="2">  
  <tr>  
    <td width="47%">Results for <?PHP echo "<i><b><font color=#000000>".$search\_term."</font></b></i> "; ?></td>  
    <td width="53%" align="right" height="22">Results <b>  
      <?PHP echo ($first\_pos+1)." - ";  
      if(($RESULTS\_LIMIT + $first\_pos) < $results) echo ($RESULTS\_LIMIT + $first\_pos);  
      else echo $results ; ?>  
    </b>  
      out of <b><?PHP echo $results; ?></b>  
      for(<b><?PHP echo sprintf("%01.2f", $time\_search); ?></b>)  
      seconds </td>  
  </tr>  
  <tr>  
    <form action="" method="GET">  
      <td colspan="2" align="center"> <input name="search\_term" type="text" value="<?PHP echo $search\_term; ?>" size="40">  
        <input name="search\_button" type="submit" value="Search"> </td>  
    </form>  
  </tr>  
  <?PHP     
    while($row = mysql\_fetch\_array($sql\_result\_query))  
    {  
    ?>  
      <tr align="left">  
        <td colspan="2"><?PHP echo $row['title']; ?></td>  
      </tr>  
  <?PHP  
    }  
    ?>  
< /table>  
< ?PHP  
}  
elseif($sql\_query)  
{  
?>  
< table border="0" cellspacing="2" cellpadding="0">  
    <tr>  
        <td align="center">No results for   <?PHP echo "<i><b><font color=#000000>".$search\_term."</font></b></i> "; ?></td>  
    </tr>  
    <tr>  
        <form action="" method="GET">  
        <td colspan="2" align="center">  
            <input name="search\_term" type="text" value="<?PHP echo $search\_term; ?>">  
            <input name="search\_button" type="submit" value="Search">  
        </td>  
        </form>  
    </tr>  
< /table>  
< ?PHP  
}  
?>

Sorting into pages

Our final step is to show the number of result pages. We are calculating how many results the SQL query has returned. We start by calculating the first position in order to place it in relevance. If the maximum number of results is exceeded the algorithm creates a forward button for the user to switch to the next page. We already have defined the maximum number per page via $RESULTS\_LIMIT. If the user goes to the second page of the results he can go back and check previous results with a back button. Well, these so called buttons are not really buttons – they are simple ">>" and "<<" signs. Of course feel free to add your own buttons, or even some flash images.

I am also using a PHP function stripslashes in order to escape some quotes. Why? I use it to clean the results from the quotes in the MySQL query. That way the algorithm will not fail if any quotes are inserted.

We can do all this with the following code:

<?PHP  
if($first\_pos > 0)  
{  
  $back=$first\_pos-$RESULTS\_LIMIT;  
  if($back < 0)  
  {  
    $back = 0;  
  }  
  echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$back' ></a>";  
}  
if($results>$RESULTS\_LIMIT)  
{  
  $sites=intval($results/$RESULTS\_LIMIT);  
  if($results%$RESULTS\_LIMIT)  
  {  
    $sites++;  
  }  
}  
for ($i=1;$i<=$sites;$i++)  
{  
  $fwd=($i-1)\*$RESULTS\_LIMIT;  
  if($fwd == $first\_pos)  
  {  
      echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd '><b>$i</b></a> | ";  
  }  
  else  
  {  
      echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd '>$i</a> | ";     
  }  
}  
if(isset($first\_pos) && $first\_pos < $results-$RESULTS\_LIMIT)  
{  
  $fwd=$first\_pos+$RESULTS\_LIMIT;  
  echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd ' > >></a>";  
  $fwd=$results-$RESULTS\_LIMIT;  
}  
?>  
    </td>  
  </tr>  
< /table>

**The next code block contains the entire code for the search engine. I've inserted comments so that you can track the process:**

<?PHP  
function getmicrotime()  
{  
    list($usec, $sec) = explode(" ", microtime());  
    return ((float)$usec + (float)$sec);  
}

//initializing connection to the database   
$connection\_string = dirname(\_\_FILE\_\_) . "/connectionstring.php";  
require\_once($connection\_string);

//selecting table  
mysql\_select\_db("test") or die ( 'Unable to select database.' );

//max number of results on the page  
$RESULTS\_LIMIT=10;  
if(isset($\_GET['search\_term']) && isset($\_GET['search\_button']))  
{  
      $search\_term = $\_GET['search\_term'];  
    if(!isset($first\_pos))  
    {  
        $first\_pos = "0";  
    }  
    $start\_search = getmicrotime();

      //initializing MySQL Quary    
    $sql\_query = mysql\_query("SELECT \* FROM news WHERE MATCH(title,article) AGAINST('$search\_term')");

    //additional check. Insurance method to re-search the database again in case of too many matches (too many matches cause returning of 0 results)  
    if($results = mysql\_num\_rows($sql\_query) != 0)  
            {  
                $sql =  "SELECT \* FROM news WHERE MATCH(title,article) AGAINST('$search\_term') LIMIT $first\_pos, $RESULTS\_LIMIT";  
                  $sql\_result\_query = mysql\_query($sql);           
            }  
    else  
            {  
                  $sql = "SELECT \* FROM news WHERE (title LIKE '%".mysql\_real\_escape\_string($search\_term)."%' OR article LIKE '%".$search\_term."%') ";  
                  $sql\_query = mysql\_query($sql);  
                  $results = mysql\_num\_rows($sql\_query);  
                  $sql\_result\_query = mysql\_query("SELECT \* FROM news WHERE (title LIKE '%".$search\_term."%' OR article LIKE '%".$search\_term."%') LIMIT $first\_pos, $RESULTS\_LIMIT ");  
            }  
    $stop\_search = getmicrotime();

      //calculating the search time  
    $time\_search = ($stop\_search - $start\_search);  
}  
?>  
< ?PHP  
if($results != 0)  
{  
?>     
   <!-- Displaying of the results -->  
< table border="0" cellspacing="2" cellpadding="2">  
  <tr>  
    <td width="47%">Results for <?PHP echo "<i><b><font color=#000000>".$search\_term."</font></b></i> "; ?></td>  
    <td width="53%" align="right" height="22">Results <b>  
      <?PHP echo ($first\_pos+1)." - ";  
      if(($RESULTS\_LIMIT + $first\_pos) < $results) echo ($RESULTS\_LIMIT + $first\_pos);  
      else echo $results ; ?>  
    </b>  
      out of <b><?PHP echo $results; ?></b>  
      for(<b><?PHP echo sprintf("%01.2f", $time\_search); ?></b>)  
      seconds </td>  
  </tr>  
  <tr>

    <**form action="" method="GET"**>  
      <td colspan="2" align="center">

<input name="search\_term" type="text" value="<?PHP echo $search\_term; ?>" size="40">  
        <input name="search\_button" type="submit" value="Search"> </td>  
    </form>  
  </tr>  
  <?PHP     
    while($row = mysql\_fetch\_array($sql\_result\_query))  
    {  
    ?>  
      <tr align="left">  
        <td colspan="2"><?PHP echo $row['title']; ?></td>  
      </tr>  
  <?PHP  
    }  
    ?>  
< /table>  
< ?PHP  
}  
//if nothing is found displays a form and a message: there are nor results

elseif($sql\_query)  
{  
?>  
< table border="0" cellspacing="2" cellpadding="0">  
    <tr>  
        <td align="center">No results for   <?PHP echo "<i><b><font color=#000000>".$search\_term."</font></b></i> "; ?></td>  
    </tr>  
    <tr>  
        <form action="" method="GET">  
        <td colspan="2" align="center">  
            <input name="search\_term" type="text" value="<?PHP echo $search\_term; ?>">  
            <input name="search\_button" type="submit" value="Search">  
        </td>  
        </form>  
    </tr>  
< /table>  
< ?PHP  
}  
?>  
< table width="300" border="0" cellspacing="0" cellpadding="0">  
      <?php  
      if (!isset($\_GET['search\_term'])) { ?>  
    <tr>  
        <form action="" method="GET">  
        <td colspan="2" align="center">  
            <input name="search\_term" type="text" value="<?PHP echo $search\_term; ?>">  
            <input name="search\_button" type="submit" value="Search">  
        </td>  
        </form>  
    </tr>  
    <?php  
      }  
      ?>  
  <tr>  
    <td align="right">  
< ?PHP  
//displaying the number of pages where the results are sittuated  
if($first\_pos > 0)  
{  
  $back=$first\_pos-$RESULTS\_LIMIT;  
  if($back < 0)  
  {  
    $back = 0;  
  }  
  echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$back' ></a>";  
}  
if($results>$RESULTS\_LIMIT)  
{  
  $sites=intval($results/$RESULTS\_LIMIT);  
  if($results%$RESULTS\_LIMIT)  
  {  
    $sites++;  
  }  
}  
for ($i=1;$i<=$sites;$i++)  
{  
  $fwd=($i-1)\*$RESULTS\_LIMIT;  
  if($fwd == $first\_pos)  
  {  
      echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd '><b>$i</b></a> | ";  
  }  
  else  
  {  
      echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd '>$i</a> | ";     
  }  
}  
if(isset($first\_pos) && $first\_pos < $results-$RESULTS\_LIMIT)  
{  
  $fwd=$first\_pos+$RESULTS\_LIMIT;  
  echo "<a href='search.php?search\_term=".stripslashes($search\_term)."&first\_pos=$fwd ' > >></a>";  
  $fwd=$results-$RESULTS\_LIMIT;  
}  
?>  
    </td>  
  </tr>  
< /table>

Do not forget to upload the files connectionstring.php and Search.php in the same directory on your server