



Dynamic PHP Contact Form

Documentation.

- **About**
- **Installation of the Form**
- **Modification of the Form**
- **Terms and Conditions**

About

This is a full documentation for the Dynamic PHP form, which can be easily embedded to any website and is fully customizable. There are numerous occasions where you could use this form for your website.

The main feature of the form is dynamically controlled strings with entry fields by the user, whom can add/remove those as much as needed. This feature makes the form look compact without the need to include many static strings of entry fields, which will make the form look unprofessional and bulky. You will be able to modify the amount of entry fields and the possible amount of dynamic strings added/removed by the customer which makes it very flexible. The form includes dynamic Captcha, which can be reloaded on the actual page, without the need to refresh the whole page, 3 file attachment option with files been saved on your server, organized by the email of each submitter and sent to a specified email with direct link to each attachment.

The process of submitting the form is also fully dynamic, all happening on one page, including “Success” and “Error” messages. The fields which are required and not required can be customized. The view of the form is fully customizable as well. What makes this form great is that it’s very simple, without any unneeded garbage in the code – YOU can fully customize the view, look and functionality of the form according to your needs!

When you purchase the form you can use it on any amount of sites, there are no limitations. Feel free to contact us if you have any questions, concerns or issues with the form!

Installation of the Form

In order to install the form, you will need to copy the “form-files” folder to the root of your server and either use the form.html file – which is the form itself, or embed it to your site. In case you will choose to embed it, you will need to:

- Copy everything in the `<HEAD></HEAD>` section found in the form.html file to the `<HEAD></HEAD>` section of your web page, which will contain the form (please note: there is no `<TITLE>` tag in the form.html file, be careful not to delete your `<TITLE>` tag when copying `<HEAD>` contents.
- Insert the body of the form within your `<BODY>`. The body of the form code starts with: “`<form action="form-files/post_form.php...."`” and ends with the: “`<iframe id="successfull_submit"`”.
- That’s it!

Modification of the Form (Main Info)

Many parts of the form are self-explanatory, however some require more in-depth explanation. You will only need to modify `style_form.css`, `post_form.php` and the `form.html` (or the code embedded to your site) to customize the form.

The `style_form.css` file contains all the attributes of how the form will look and each attribute includes explanation of what it modifies.

The `post_form.php` is where the form is getting processed to be sent to specified email. Within `post_form.php`, on lines 11 and 12, on line 11 – specify the email address where the form will be sent (put within the quotes, do not remove the quotes). On line 12, you can specify the subject of the email, that way you will know where the form is coming from and which form was submitted, in case you have several forms installed on your website.

```
11     $to = "EMAIL WHERE FORM CONTENTS WILL BE SENT";
12     $subject = "SUBJECT OF THE EMAIL";
```

Also, make sure to adjust the URL of your website, so that attachments within the email sent by the form could be opened as a direct link

```
46     ${'img_url'}.${file_num} = 'http://YOURSITEURL.COM/form-files/uploads/' . $dirname . '/' . $_FILES[$input_name]["name"];
```

If you will be installing several forms within your website you will need to have a separate folder for each form, such as: `forms/equipment-disposal-form`, `forms/contact-form`, `forms/passenger-listing-form`, and so on. In such case you will need to modify the paths to various modules in the form.

In `post_form.php` you will need to adjust the following parts:

```
44     if(!empty($_FILES[$input_name]["name"] )){
45         ${'target'}.${file_num} = $_SERVER['DOCUMENT_ROOT'] . '/form-files/uploads/' . $dirname . '/' . $_FILES[$input_name]["name"];
46         ${'img_url'}.${file_num} = 'http://YOURSITEURL.COM/form-files/uploads/' . $dirname . '/' . $_FILES[$input_name]["name"];
```

In the `form.html` file or the web page where you embedded the form you will need to adjust those parts:

```
170 <form action="form-files/post form.php" method="post" name="contact_form" id="cont;
```

```
225     
226     <a href="#" onclick="
227     document.getElementById('captcha_img').src = 'form-files/get_captcha.php?' + Math.random();
228     document.getElementById('captcha').value = '';
229     return false;">
230     </a>
```

That way each form will have its own “nest” and everything will be organized.

Modification of the Form (Minor Adjustments)

In order to adjust the maximum amount of dynamic strings allowed to be added by user:

In form.html, on line 113, adjust the number next to “var limit =” to the desired number. Then, in post_form.php, on line 112, in the “for(i\$=1; \$i<=20; \$i++)” – next to “\$i<=” put the same number.

form.html	post_form.php
<pre>103 document.getElementById('error_message').style.display= "none"; 104 return true; 105 } 106 } 107 } 108 </script> 109 110 <script> 111 var i = 3; 112 var total = 3; 113 var limit = 20; 114 var del_lines = new Array(); 115 var cur_num; 116 var array_el = 0; 117 118 function add_new_line(){ 119 if (total==limit) { 120 alert("You have reached the limit of adding lines"); 121 }else{</pre>	<pre>104 if ((\$upload_error == 0) && (\$error_size==0) && (\$error_ext==0)){ 105 \$body .= "Name: ".\$name."\n"; 106 \$body .= "Email: ".\$email."\n"; 107 \$body .= "Phone: ".\$phone."\n"; 108 \$body .= "Address: ".\$address."\n"; 109 \$body .= "Message: ".\$message."\n\n"; 110 111 for(\$i=1; \$i<=20; \$i++){ 112 113 \$dynamicentryvaluefirst = 'dynamicentryvaluefirst'.\$i; 114 \$dynamicentryvaluesecond = 'dynamicentryvaluesecond'.\$i; 115 \$dynamicentryvaluethird = 'dynamicentryvaluethird'.\$i; 116 \$dynamicentryvaluefourth = 'dynamicentryvaluefourth'.\$i; 117 \$dynamicentryvaluefifth = 'dynamicentryvaluefifth'.\$i; 118 119 if (isset(\$_POST[\$dynamicentryvaluefirst])){ 120 \$cur_dynamicentryvaluefirst = \$_POST[\$dynamicentryvaluefirst]; 121 }else{</pre>

Modification of the Form (Complex Adjustments - Static)

How to add a new input field(s) in the "Contact Information" section

1. In the form.html file, add the new input field as shown below:

```
175 <div class="form_section">Contact Information</div>
176 <label>Contact Name:*</label><input name="name" id="name" type="text" style="width:390px;" /><br/>
177 <label>Email:*</label><input name="email" id="email" type="text" style="width:390px;" /><br/>
178 <label>Address:*</label><input name="address" id="address" type="text" style="width:390px;" /><br/>
179 <label>Phone:</label><input name="phone" id="phone" type="text" style="width:390px;" /><br/>
180 <label>New field:</label><input name="new_field" id="new_field" type="text" /><br/>
181 <label>Message:*</label><textarea name="message" id="message" rows="5" /><br/>
182 </div>
```

2. In the post_form.php file you have to add two lines of code:

Around line 19 of the file you will find the below piece of code. Meanings:

`$new_field` → variable name (you can give any name you want after the `$` symbol)

`$_POST['new_field']` → with this you read the contents of the new input field (what the user wrote in it).

```
14 $name = $_POST['name'];
15 $email = $_POST['email'];
16 $phone = $_POST['phone'];
17 $address = $_POST['address'];
18 $message = $_POST['message'];
19 $new_field = $_POST['new_field'];
20 $total_lines = $_POST['total_lines'];
21
```

Use the same name you gave to the new input field in the form.html file.

Around line 108 in the same file you will find the above piece of code, where the body of the email is constructed.

```
104 if (($upload_error == 0) && ($error_size == 0) && ($error_ext == 0)) {
105     $body .= "Name: ".$name."\n";
106     $body .= "Email: ".$email."\n";
107     $body .= "Phone: ".$phone ."\n";
108     $body .= "New Field Value: " . $new_field . "\n";
109     $body .= "Address: ".$address."\n";
110     $body .= "Message: ".$message ."\n\n";

```

Modification of the Form (Complex Adjustments - Dynamic)

How to add a new input field(s) to the dynamic strings section.

1. In form.html file, around line 188 you have to add the new input field in all of the three static lines and in the dynamically created lines as well, as shown below (move the
 to the last static line):

```
183 <div class="field-set clr" id="field-set1">
184 <label>Description 1</label>
185 <label>Description 2</label>
186 <label>Description 3</label>
187 <label>Description 4</label>
188 <label>Description 5</label>
189 <label>New Description Field</label><br/>
190 <input name="dynamicentryvaluefirst1" id="dynamicentryvaluefirst1" type="text" />
191 <input name="dynamicentryvaluesecond1" id="dynamicentryvaluesecond1" type="text" />
192 <input name="dynamicentryvaluethird1" id="dynamicentryvaluethird1" type="text" />
193 <input name="dynamicentryvaluefourth1" id="dynamicentryvaluefourth1" type="text" />
194 <input name="dynamicentryvaluefifth1" id="dynamicentryvaluefifth1" type="text" />
195 <input name="new_field1" id="new_field1" type="text" /><br/>
196 </div>
197 <div class="field-set clr" id="field-set2">
198 <input name="dynamicentryvaluefirst2" id="dynamicentryvaluefirst2" type="text" />
199 <input name="dynamicentryvaluesecond2" id="dynamicentryvaluesecond2" type="text" />
200 <input name="dynamicentryvaluethird2" id="dynamicentryvaluethird2" type="text" />
201 <input name="dynamicentryvaluefourth2" id="dynamicentryvaluefourth2" type="text" />
202 <input name="dynamicentryvaluefifth2" id="dynamicentryvaluefifth2" type="text" />
203 <input name="new_field2" id="new_field2" type="text" /><br/>
204 </div>
205 <div class="field-set clr" id="field-set3">
206 <input name="dynamicentryvaluefirst3" id="dynamicentryvaluefirst3" type="text" />
207 <input name="dynamicentryvaluesecond3" id="dynamicentryvaluesecond3" type="text" />
208 <input name="dynamicentryvaluethird3" id="dynamicentryvaluethird3" type="text" />
209 <input name="dynamicentryvaluefourth3" id="dynamicentryvaluefourth3" type="text" />
210 <input name="dynamicentryvaluefifth3" id="dynamicentryvaluefifth3" type="text" />
211 <input name="new_field3" id="new_field3" type="text" /><br/>
212 </div>
213 <div id="extra_lines" class="clr"></div>
```

IMPORTANT: Pay much attention to the numbering of the fields. The extra field in the 1st line must end to "1", the extra field in the 2nd line must end to "2" and so on.

Now it's time to add the extra field in the dynamically created lines. To do this, you must add a line in the script that generates them. In the head section of the form.html page find the below piece of code around line 148 and add the lines as shown below:

```
140 var newdiv = document.createElement('div');
141 newdiv.id = 'field-set' + cur_num;
142 newdiv.className = 'field-set clr';
143
144 newdiv.innerHTML = '<input name="dynamicentryvaluefirst' + cur_num + '" id="dynamicentryvaluefirst' + cur_num + '" type="text" />' +
145 '<input name="dynamicentryvaluesecond' + cur_num + '" id="dynamicentryvaluesecond' + cur_num + '" type="text" />' +
146 '<input name="dynamicentryvaluethird' + cur_num + '" id="dynamicentryvaluethird' + cur_num + '" type="text" />' +
147 '<input name="dynamicentryvaluefourth' + cur_num + '" id="dynamicentryvaluefourth' + cur_num + '" type="text" />' +
148 '<input name="dynamicentryvaluefifth' + cur_num + '" id="dynamicentryvaluefifth' + cur_num + '" type="text" />' +
149 '<input name="new_field' + cur_num + '" id="new_field' + cur_num + '" type="text" />' +
150 '<a href="#" class="del" title="Delete this row" onclick="remove_line(' + cur_num + ')>X</a>';
151 document.getElementById('extra_lines').appendChild(newdiv);
152 total++;
153 document.getElementById('total_lines').value = total;
```

Now open the post_form.php file and add the required lines, as shown below (around lines 119 – 150 of the file):

```
for($i=1; $i<=10; $i++){
```

```

$type = 'type'.$i;
$model = 'model'.$i;
$condition = 'condition'.$i;
$comments = 'comments'.$i;
$cost = 'cost'.$i;
$new_field = 'new field'.$i;

```

This is the new field's name you gave on the contacts.html file **WITHOUT NUMBERING.** Numbering is implemented dynamically by the \$i counter.

112
113
114
115
116
117
118
119
120

```
for($i=1; $i<=20; $i++){
```

```

$dynamicentryvaluefirst = 'dynamicentryvaluefirst'.$i;
$dynamicentryvaluesecond = 'dynamicentryvaluesecond'.$i;
$dynamicentryvaluethird = 'dynamicentryvaluethird'.$i;
$dynamicentryvaluefourth = 'dynamicentryvaluefourth'.$i;
$dynamicentryvaluefifth = 'dynamicentryvaluefifth'.$i;
$new_field = 'new field'.$i;

```

145
146
147
148
149
150
151
152
153
154

```

if (isset($_POST[$dynamicentryvaluefifth])){
    $cur_dynamicentryvaluefifth = $_POST[$dynamicentryvaluefifth];
}else{
    $cur_dynamicentryvaluefifth = "";
}
if (isset($_POST[$new_field])){
    $cur_new_field = $_POST[$new_field];
}else{
    $cur_new_field = "";
}

```

You are almost done... One more addition is needed, as shown below (around line 151)

```

151 ${'line'.$i} = $cur_dynamicentryvaluefirst . " | " . $cur_dynamicentryvaluesec . " | " . $cur_dynamicentryvaluefifth . " | " . $cur_new_field;
152 if({'$line'.$i} == (" | | | | |")){
153
154 }else{
155     $body .= "Line".$i.": ".({'$line'.$i})."\n";
156 }

```

The last symbol must have 2 spaces on the left (before the previous one)

one space to the left and one to the right

Terms and Conditions

**You can use the form on as many websites you want,
there are no restrictions at all!**

Please note, this product is not intended for re-selling.

**If you find someone is selling it outside of <http://www.dynamicphpform.com>,
please report to us immediately: info@techstoreon.com! Thanks!**