

## الفصل السابع : التحقق من بيانات النماذج ، Validating form data

عند التعامل مع نماذج HTML بغرض السماح للمستخدمين بإدخال بيانات محددة فيجب علينا التحقق من تلك البيانات قبل إجراء أي عملية عليها. و تكمن أهمية إجراء عملية التحقق لأننا لا نضمن أن تكون البيانات المُدخلة صحيحة، وبدون إجراء عملية التحقق فقد تتسبب تلك البيانات بحدوث نتائج غير متوقعة قد يصعب معالجتها لاحقاً. لذلك يجب أن لا نثق في المستخدم لأن الأخطاء قد تحدث عمداً أو عن غير قصد. عملية التحقق من البيانات يمكن أن تتم بواسطة طريقتين:

✓ عن طريق جهة العميل client-side باستخدام JavaScript مثلاً.

✓ عن طريق جهة الخادم server-side باستخدام php.

ما هي الأشياء التي يجب أن نتحقق منها

✓ التحقق من الحقول الفارغة.

✓ التحقق من البريد الإلكتروني.

✓ طول البيانات المُدخلة.

✓ حقول الأرقام فقط.

✓ حقول الحروف فقط.

✓ حقول الحروف و الأرقام.

✓ إزالة وسوم HTML.

و لكي تكون الصورة واضحة أكثر سنقوم بإنشاء نموذج HTML يحتوي على الحقول التالية:

✓ حقل الاسم.

✓ حقل البريد الإلكتروني.

✓ حقل الهاتف.

✓ حقل موضوع الرسالة.

✓ حقل نص الرسالة.

الآن سنقوم بإنشاء الصفحة contact.html و التي ستحتوي على النموذج المطلوب كما يلي:

```
<html>
<head>
<title>اتصل بنا</title>
</head>
<body>
<form method='post' action='sendMessage.php'>
<table dir='rtl' bgcolor='lightgray' align='center' >
<tr><td><b>الاسم:</b></td><td><input type='text' name='name'/></td></tr>
<tr><td><b>الهاتف:</b></td><td><input type='text' name='phone'/></td></tr>
<tr><td><b>البريد الإلكتروني:</b></td><td><input type='text'
name='email'/></td></tr>
<tr><td><b>موضوع الرسالة:</b></td><td><input type='text'
name='subject'/></td></tr>
```

```

<tr><td><b>نص الرسالة:</b></td><td><textarea cols='30' rows='5'
name='message'>
</textarea></td></tr>
<tr><td colspan='2'><input type='submit' value='ارسل الرسالة'
name='doContact'></td></tr>
</table>
</form>
</body>

```

في حال قمنا بتنفيذ الصفحة السابقة فسيكون الناتج، النموذج التالي:

- الآن، عندما يقوم المستخدم بإرسال النموذج سنقوم بالتحقق مما يلي:
- ١) الحقول كلها ليست فارغة.
  - ٢) حقل الاسم يحتوي على حروف فقط.
  - ٣) حقل الهاتف يحتوي على أرقام فقط و بطول عشرة أرقام.
  - ٤) حقل البريد الالكتروني يحتوي على عنوان بريدي صالح.
  - ٥) حقل موضوع الرسالة يحتوي على حروف و أرقام فقط.
  - ٦) حقل نص الرسالة يحتوي على حروف و أرقام فقط.

و لتحقيق ذلك سنقوم باستخدام الدوال الموضحة في الجدول التالي:

الدالة	الوظيفة
<code>empty()</code>	تتحقق ما إذا كان المتغير فارغاً.
<code>ctype_alnum()</code>	تتحقق ما إذا كان المتغير يحتوي على حروف و أرقام.
<code>ctype_alpha()</code>	تتحقق ما إذا كان المتغير يحتوي على حروف فقط.
<code>ctype_digit()</code>	تتحقق ما إذا كان المتغير يحتوي على أرقام فقط.
<code>preg_match()</code>	تقوم بإجراء مقارنة بتعبير عادي Regular Expression.
<code>strlen()</code>	للتحق من طول السلسلة النصية.

بعد ما قمنا باستعراض الدوال التي سنقوم باستخدامها سنقوم الآن بكتابة الشفرة البرمجية للصفحة `sendMessage.php` و التي ستتعامل مع البيانات المرسله من قبل النموذج في الصفحة السابقة:

```

<?php

if(!isset($_POST['doContact']))
{
    header('Location:login.html');
}
else
{
    $errors = array();
    if(empty($_POST['name']) || empty($_POST['phone'])
    || empty($_POST['email']) || empty($_POST['subject']) || empty($_POST['message']))
    {
        $errors[]='الرجاء تعبئة جميع الحقول';
    }
    if(!ctype_alpha($_POST['name']))
    {
        $errors[]='حقن الاسم يجب أن يحتوي على حروف فقط';
    }
    if(!ctype_digit($_POST['phone']))
    {
        $errors[]='حقن الهاتف يجب أن يكون رقما';
    }
    elseif(strlen($_POST['phone'])>10)
    {
        $errors[]='رقم الهاتف يجب أن لا يزيد عن عشرة ارقام';
    }

    if (!preg_match("/^[a-z0-9-]+(\.[a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)+$/",
    $_POST['email']))
    {
        $errors[]='البريد الالكتروني المدخل غير صالح';
    }
    if(!ctype_alnum($_POST['subject']))
    {
        $errors[]='حقن الموضوع يجب أن يحتوي على أحرف و أرقام فقط';
    }
    if(!ctype_alnum($_POST['message']))
    {
        $errors[]='حقن نص الرسالة يجب أن يحتوي على أحرف و أرقام فقط';
    }
    if(!empty($errors))
    {
        foreach($errors as $val)
        {
            echo"<p align='center'>$val</p>";
        }
    }
    else{
        echo'تمت عملية إرسال الرسالة بنجاح';
    }
}
?>

```

## الشرح:

١. في هذا الجزء من البرنامج قمنا بالتحقق من أن المستخدم وصل لهذه الصفحة من خلال الصفحة login.html بعد الضغط على زر الإرسال و هنا افترضنا أن المستخدم يصل لهذه الصفحة من خلال الصفحة المذكورة. لذلك قمنا بإعادة توجيه المستخدم للصفحة login.html عن طريق استخدام الدالة header ( سنتطرق لطرق إعادة توجيه المستخدم في الفصل الثاني عشر ).
٢. في هذا الجزء على افتراض أن المستخدم قام بإرسال نموذج الاتصال في الصفحة login.html قمنا بما يلي:

- ✓ تعريف مصفوفة فارغة لتخزين الأخطاء التي سنقوم بعرضها للمستخدم.
  - ✓ التحقق من أن جميع الحقول ليست فارغة عن طريق استخدام الدالة ( empty ) و تمرير قيمة الحقل لها. و في حال نسي المستخدم أي من حقول النموذج قمنا بإنشاء رسالة خطأ مناسبة و إضافتها للمصفوفة السابقة.
  - ✓ التحقق من أن حقل الاسم يحتوي على حروف فقط عن طريق استخدام الدالة ctype\_alpha ( ) و تمرير قيمة حقل الاسم . و في حال احتوى على قيم غير حرفية قمنا بإنشاء رسالة مناسبة و إضافتها للمصفوفة السابقة.
٣. في هذا الجزء قمنا بالتحقق من أن حقل الهاتف يحتوي على أرقام فقط باستخدام الدالة ctype\_digit ( ) و تمرير قيمة الحقل لها. بعد ذلك قمنا بالتحقق من أن طول حقل الهاتف لا يزيد عن ١٠ أرقام عن طريق استخدام الدالة ( strlen ) و تمرير قيمة الحقل لها. و في كلا الحالتين في حال احتوى حقل الهاتف على قيم غير رقمية أو كان الرقم أطول من ١٠ أرقام قمنا بإنشاء رسالة مناسبة و إضافتها للمصفوفة السابقة.
  ٤. في هذا الجزء سنقوم بالتحقق من صحة البريد الإلكتروني المدخل عن طريق تمرير القيمة المدخلة من قبل المستخدم و مطابقتها بقيم البريد الإلكتروني الصالحة عن طريق استدعاء الدالة ( preg\_match ). هذه الدالة تأخذ معطيين الأول النموذج pattern و الثاني القيمة التي سنقوم بمطابقتها بالنموذج. النماذج المستخدمة في عملية المقارنة تسمى Regular Expressions. و للمزيد عنها يمكن زيارة الرابط التالي:  
[http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression)
- و أخيراً في حال عدم تطابق البريد الإلكتروني المدخل مع النموذج pattern المحدد، قمنا بإنشاء رسالة مناسبة و إضافتها للمصفوفة السابقة.
٥. في هذا الجزء قمنا بالتحقق من أن حقل الموضوع و الرسالة يحتوي فقط على حروف و أرقام فقط عن طريق استخدام الدالة ( ctype\_alnum ) التي تأخذ معطى واحد هو قيمة الحقل. في حال احتوى أي من الحقول على قيم غير الأحرف أو الأرقام ، قمنا بإنشاء رسالة مناسبة لكل حقل و إضافتها للمصفوفة الخاصة بالأخطاء.
  ٦. بعد الانتهاء من التحقق من قيم الحقول قمنا باختبار قيمة المصفوفة السابقة. في حال لم تكن فارغة سنقوم بطباعة محتوياتها باستخدام حلقة التكرار foreach بخلاف ذلك قمنا بطباعة رسالة للمستخدم تفيد بأن الرسالة قد تم إرسالها ( و هو ما لم يحدث بشكل فعلي لأننا لم نقوم بإرسال الرسالة، لكن يمكننا عمل ذلك عن طريق استخدام دالة البريد الإلكتروني أو إضافة الرسالة في قاعدة بيانات و سوف نتطرق لذلك لاحقاً ).

أخيراً، في حال أردنا إزالة وسوم HTML من القيمة المدخلة في أي من الحقول السابقة كحقل الرسالة النصية مثلاً، نستطيع استخدام الدالة ( strip\_tags ) و التي تأخذ معطى واحد( بالإضافة إلى معطى اختياري ) هو القيمة المراد إزالة وسوم HTML منها ، مثال:

strip\_tags( \$\_POST['message'] )

## أسئلة الفصل

١. ما أهمية التحقق من البيانات المُدخلة في النموذج؟
٢. أيهما أفضل التحقق من البيانات بواسطة جهة العميل أو جهة الخادم مع ذكر السبب؟
٣. هل هناك طريقة أخرى غير استخدام الدالة `empty()` للتحقق من الحقول الفارغة ؟
٤. هل يمكن استخدام وسوم HTML بهدف الإضرار بتطبيق الإنترنت و كيف يمكن منع المستخدم من إدخالها في النموذج؟
٥. مستخدماً شبكة الإنترنت كيف يمكن التغلب على مشكلة SPAM مع النماذج؟