الفصل السابع
التعامل مع العميل

كما رأينا في الفصول السابقة، فإن الـPHP يوفر رقماً عظيماً من المميزات عن الـhtml لبناء مواقع الويب، من الأشياء الأساسية التي لم نتكلم عنها حتى الآن هي الموثوقية (أو الاستقرار) وهو بالمعني الصحيح والصريح:

القابيلة على الاحتفاظ بالمعلومات بين صفحتين منفردتين أو مختلفيتن في المستعرض...

بدون أي إضافات، HTTP لا يوفر أي ميكانيكيه للحفاظ على البيانات وجعلها مستقرة لمعالجة تتم بين صفحتين، كل طلب لصفحة في الإنترنت (request) ليس له أي علاقة بأي طلب آخر... مثلاً عندما تطلب موقع المطور العربي ومن ثم منتدى المطور العربي فإن كلا الطلبين ليس لهما علاقة ببعضهما...

بمصطلح آخر يمكننا أن نقول أن الـHTTP فاقد لحالته (stateless) أي أنه لا يعرف، أي أن أمر طلب الصفحة ينتهي عند انتهاء الطلب، فهي عندما تقوم بنقل بيانات صفحة من السيرفر إلى المستخدم فهي تعرف من هو المستخدم الذي يطلب البيانات وعلى أي نافذة سيتم نقل البيانات وعند انتهاء ذلك فإن كل هذا الموضوع ينتهي وإذا عاد المستخدم فطلب صفحة أخرى فإنه لا يعرف إن كان هو نفس المستخدم أو لا !

إن القدرة على الحفاظ على وجود البيانات ليست وسيلة أو ميزة أو قوة مقتصرة على الـPHP فقط.

obeikandi.com

فلقد رأيت كيف استطعنا إرسال معلومات من صفحة إلى صفحة بدون خسران أي معلومات وذلك عن طريق الـhtml وبالرغم من ذلك فإن المستخدم عندما يقوم بإغلاق الصفحة عند استقبالها للبيانات فإن ذلك يعني فقدانها للأبد، عن طريق استخدام الـPHP يمكننا إخبار السيرفر بأن يقوم بإرجاع البيانات بطريقة تمكننا من الحفاظ عليها، مثلما سنري في هذا الفصل، هناك ثلاث طرق لعمل ذلك....

التميز الحقيقي في قوة الفهم للـPHP، يتطلب منا مفهومية جيدة في كيفية استعمال الـPHP في التفاعل مع المستخدم والمتصفح الذي يستخدمه لكي نتغلب على نقاط الضعف التي في الـhttp.

هذا هو موضوعنا لهذا الفصل والذي سنتكلم فيه عن:

١ - الـHTTP والـ html ومحدودية قدراتهم، وكيف يستطيع الـPHP التغلب على القصور فيهم.
٢ - الاحتفاظ بالمعلومات التي نريد أن نستخدمها بين طلب لصفحتين مختلفتين.
٣ - مكنكة الحفاظ على البيانات.
٤ - الكعكات (cookies) وكيفية استخدامها.
٥ - PHP4 والـ native session – المكنكة الداخلية للحفاظ على وجودية البيانات.

هذا الفصل مفيد بشكل ظاهري لمن هو جديد على إنشاء مواقع متفاعلة متوسطة – كبيرة الحجم بواسطة الـPHP..
إنه يحتوي على الكثير من بعض الأمثلة التي تفيد.

obeikandl.com

الهدف من هذا الفصل هو أن تتعرف على كيفية الحفاظ على معلومات المستخدم عبر متغير أو أكثر بين أكثر من صفحة، مثل أن تجعل اسم المستخدم ظاهراً في كل صفحة يقوم بالولوج إليها... مما يؤكد استمرارية وجود البيانات.

لنفرض أن لدينا موقعاً على الإنترنت هذا الموقع يهتم ببيع وتسويق مواد غذائية أو أن هذا الموقع يقدم مسابقات ثقافية، في العادة عندما يقوم المستخدم بطلب شراء سلعة معينة أو عندما يختار الدخول في مسابقة من المسابقات الثقافية فإنه يقوم بدخول أكثر من صفحة بالتتابع.

يختار السلعة في الصفحة الأولى وبعد ذلك يقوم برؤية معلومات السلعة في الصفحة الثانية والصفحة الثالثة يقوم فيها بتعبئة معلوماته للشراء أو غير ذلك إلى أن ينتهي من كافة المعلومات وبعد ذلك تنتج له في النهاية صفحة فيها معلوماته والسلعة التي قام باختيارها وفاتورة الشراء !!

أو يقوم باختيار نوع المسابقة الثقافية في الصفحة الأولى وبعد ذلك يقوم بالحصول على عدة أسئلة مقسمة على عدة صفحات إلى أن ينتهي من المسابقة فتخرج له في النهاية مجموع الدرجات للأسئلة ومعلوماته وهل هو فائز أم خاسر!!

في الواقع هذا ما يسمونه بالمحافظة على الجلسة (maintain session) وأقصد بذلك دخول المستخدم إلى صفحة وانتقاله من صفحة إلى صفحة مع المحافظة على معلوماته وغير ذلك من البيانات، لكي نستطيع متابعته أولاً بأول.

في بروتوكول الـhtml والـhttp لا نستطيع معرفة إذا ما كان الشخص عندما يطلب صفحة ما هو نفسه عندما يذهب إلى الصفحة الثانية إذ إن المستخدم عندما يطلب صفحة ما (request) من السيرفر فإن السيرفر يقوم بمعرفة من أي مكان بالعالم يتكلم هذا الشخص ويقوم بإرسال استجابة إليه باعطائه الصفحة التي كان

obeikandi.com

يطلبها (response) ولكن بعد ذلك فإن السيرفر لا يعرف إذا كان هذا الشخص هو نفسه الذي يقوم بطلب الصفحة الثانية أو الثالثة في السيرفر.

هنا تأتي ميزة الـPHP وغيره من لغات برمجة الإنترنت لصناعة ميكانيكية إبقاء تفاعل مستمر بين المستخدم والسيرفر عن طريق الـsession و الـcookie، ولكي لا نعقد الموضوع دعونا نتكلم عن ذلك عملياً فذلك أفضل لفهم الموضوع من الثرثرة التي لا فائدة منها.

استخدام الحقول المخفية

سنقوم الآن بإنشاء ثلاث صفحات، الصفحة الأولى تطلب من المستخدم إدخال اسمه، والصفحة الثانية تقوم بالترحيب به واعطائه ثلاثة أسئلة، والصفحة الثالثة تقوم باعطائه النتيجة.

افتح محرر نصوص لديك واكتب الكود التالي:

```
</p>ادخل اسمك الكريم<p dir="rtl" align="center">
<form method="POST" action="quiz2.php">
<hr>
<input type="text" name="name" size="20"><br>
" ></p>إرسال<input type="submit" value="
</form>
```

احفظها باسم quiz.php

قم بفتح محرر النصوص واكتب الكود التالي:

```
<html dir ="rtl">
<?
If  (isset($name)) {
". $name ;مرحبا بك يا Echo "
Echo '
<br>
<form method="POST" action="quiz3.php" dir="rtl">
```

```
<input type="hidden" name = "thename" value =
"'.$name.'">"
</p>من هو أول الخلفاء الراشدين ؟
أبو بكر  <p dir="rtl"><input type="radio" value="
أبو بكر" name="khlifa">الصديق
</p>الصديق.
عمر بن  <p dir="rtl"><input type="radio" value="
عمر" checked name="khlifa">الخطاب
</p>بن الخطاب
</p>من هو الفاروق ؟  <p dir="rtl">
<p dir="rtl"><input type="radio" name="faroq"
عمر بن الخطاب">عمر بن الخطابvalue="
</p>الخطاب
<p dir="rtl"><input type="radio" name="faroq"
سالم بن" checked>سالمvalue="
</p>عامر
" dir="rtl">إرسال<input type ="submit" value = "
</form>' ;
}
else
{
" ;غير مصرح لك بدخول هذه الصفحةecho "
}
?>
```

احفظها باسم quiz2.php

قم بفتح المفكرة واكتب الكود التالي:

```
<?
If  ((isset($thename)) && (isset($khlifa)) &&
(isset($faroq)))
{
'. $thename ;لقد انتهت المسابقه يا echo '
$range=0;
$co = 0;
```

obeikandl.com

```
") {أبو بكر الصديق            if ($khlifa ==""
$range=$range+10;
$co = $co +1;
}
")عمر بن الخطاب    if ($faroq =="
{
$range=$range+10;
$co=$co+1;
}
if ($range < 10)
{
";ليس هناك أي إجابة صحيحة                  echo "
}
else
{
=". $co ;عدد الأسئلة التي أجبت عليها              echo "<br>"."
". $range ;الدرجة التي حصلت عليها                 echo "<br>". "
          }
}
?>
```

قم بوضع الملفات الثلاثة السابقة في مجلد السيرفر ثم قم بتشغيلها.

الشرح

قمت في هذا المثال بمحاولة صنع مكنكة تواصل للبيانات، بمعنى أنني أحاول أن أقوم بالاحتفاظ بالبيانات عبر الثلاث صفحات بشكل متواصل، لاحظ أنني كنت اختبر في quiz2 و quiz3 باختبار المتغيرات قبل طباعة أي شيء فقد يقوم المستخدم مثلاً بالاحتفاظ بالصفحة التي وصل إليها في المفضلة ثم يقوم بإكمال المسابقة في وقت آخر ولكني لا أريد ذلك بل أريد ان أجعل وقتها محدوداً (طبعاً هذا الكلام سيحصل إذا كانت المسابقة طويلة) لذلك فإنني عند الانتقال من صفحة إلى صفحة أقوم بإختبار إن كانت جميع هذه القيم موجودة ولاحظ أنني كنت احتفظ دوماً بقيم المتغيرات في متغيرات جديدة في حقول مخفية وكلما كان عدد المعلومات أكبر في كل مرة كان عدد الحقول المخفية أكثر، إن لهذه الطريقة أيضاً مشاكلها فقد

obeikandl.com

يفتح المستخدم كود الـhtml ويقوم بتفحص كيفية ملاحقته عبر المسابقة وقد يصنع هو الكود في وقت لاحق لكي يستطيع إكمال المسابقة بهذه الخدعة الماكرة... لذلك يفضل أن لا تقوم بذلك وتقوم بجعل المسألة السابقة أكثر تعقيداً باستخدام الـ regular expression بمحاولة تلغيم البيانات بواسطته ومن ثم فك هذا التلغيم في الصفحات التي تصل إليها البيانات.

إرسال بيانات بواسطة query strings

نستطيع إرسال بيانات بسيطة بواسطة الاستعلامات التي نقوم بإضافتها إلى اسم الصفحة في الأعلى متبوعة بـ(؟) علامة استفهام ثم اسم متغير وقيمته وإذا كان هناك أكثر من متغير يتم الربط بينهم بعلامة & وراجع فصل النماذج لمزيد من المعلومات.

قم بعمل صفحة وسمها ask.php وقم بكتابة الكود التالي: فيها:

```
<?
If (isset($ask)) {
If ($ask == login) {
";تم تسجيل الدخول إلى الصفحة  Echo "
}
}
if (!isset($ask)) {
". “<br>”;لم يتم تسجيل الدخول إلى الصفحةecho "
Echo "<A HREF=$PHP_SELF?ask=login>      اضغط هنا ليتم تسجيل دخولك
</a><br>";
}
?>
```

قم بتجربة هذا المثال على موقع يدعم PHP على نظام تشغيل لينوكس إذا لم يعمل بشكل جيد على الويندوز

obeikandl.com

لاحظ أننا في أول الولوج إلى الصفحة لم نستخدم أي استعلامات وعند الضغط على الرابط قام الرابط بإرسال قيمة المتغير الذي يقوم الPHP باختبارها فإذا وجد أنه قد تم إرسالها (بواسطة الرابط الذي تم الضغط عليه) قام بطباعة (تم تسجيل الدخول) وإذا لم يجدها قام بطباعة (لم يتم تسجيل الدخول) بالإضافة إلى طباعة الرابط الذي يحتوي على المتغير في طياته.

obeikandl.com