

الفصل الثاني عشر : العمل مع Sessions و Cookies

كما هو معلوم أن البروتوكول المستخدم في نقل البيانات بين متصفح الإنترنت و مواقع الويب المختلفة هو بروتوكول نقل النص التشعبي (Hypertext Transfer Protocol) HTTP. لكن قد لايعرف الكثير أن هذا البروتوكول من النوع stateless بمعنى أنه يتعامل مع كل طلب Request بشكل مستقل بدون أن يكون هناك أي علاقة بين الطلب الحالي Current request و الطلب السابق Previous request. لشرح هذه العملية، لننخيل معا كيف يتم العمل في المواقع التي تتيح خدمة التسوق الإلكتروني أو ما يسمى بـ Shopping Carts. الذي يحدث أن الزائر لموقع التسوق يقوم بإستعراض المنتجات التي يقدمها الموقع و من ثم يقوم بإختيار المنتج و إضافته إلى سلة الشراء. في الوضع العادي، إذا إنتقل الزائر إلى صفحة أخرى داخل الموقع لعرض منتج آخر فإن القيمة الأولى المضافة إلى سلة الشراء و التي تمثل المنتج الأول الذي قام الزائر بإضافته إلى سلة الشراء سيتم فقدها !! و السبب كما ذكرنا أن البروتوكول HTTP من النوع الذي لا يتذكر ما تم مسبقا مع الطلبات السابقة. للتغلب على هذه المشكلة فإن كثير من مواقع الويب تقوم بإستخدام ما يسمى بـ الكعكات أو الجلسات (Sessions or Cookies). PHP تقدم دعم كامل لـ Sessions and Cookies عن طريق عدد كبير من الدوال المبنية مسبقا و الجاهزة للإستخدام من قبل مطوري الويب.

التعامل مع الـ Cookies

الكعكات عبارة عن ملفات نصية صغيرة يتم تخزينها في جهاز المستخدم بواسطة موقع إنترنت. هذه الملفات تحتوي على معلومات تسترجع من قبل موقع الإنترنت الذي قام المستخدم بزيارته. و كمثال على الكعكات المننديات الحوارية. عندما يقوم المستخدم بتسجيل الدخول إلى أي منتدى للحوار يقوم بإدخال إسم المستخدم و كلمة المرور ، و في حال قم بتحديد صح على مربع (تذكرني) في نموذج الدخول سيقوم الموقع بتخزين كعكات داخل جهاز المستخدم يتم إستخدامها لاحقا اذا قام المستخدم بزيارة الموقع لاحقا و يتم تسجيل دخوله تلقائيا عن طريق المعلومات المخزنة داخل جهازه. تعاني الكعكات Cookies من كثير من النقد لأنها تستخدم لنقل البيانات بين المستخدم و موقع الويب لذلك توصف دائما بأنها سئة و غير آمنة. لكن الحقيقة أنها تعتبر كغيرها من التقنيات يمكن إستخدامها بشكل سئ و للأسف فهي تستخدم بشكل سئ من كثير من المواقع بهدف تحسين أداء مواقعها.

أخيرا تتميز الكعكات بعض الخواص الآمنة:

- ملفات الكعكات يمكن قرائتها فقط بواسطة موقع الإنترنت الذي قام بتخزينها.
- كل موقع إنترنت لايمكنه تخزين أكثر من ٢٠ ملف Cookies.
- أقصى حجم لملف الـ Cookie الواحد هو ٤٠ كيلوبايت.
- أقصى عدد لملفات الكعكات التي يمكن تخزينها داخل جهاز المستخدم هو ٣٠٠ ملف.

لأن ملفات الكعكات سيتم تخزينها داخل جهاز المستخدم ، لذلك ستكون إمكانية التحكم فيها محدودة. فلو قام المستخدم بإبطال دعم الكعكات (من خيارات متصفح الإنترنت) فلن يتمكن أي موقع من تخزين ملفات الكعكات داخل جهاز ذلك المستخدم.

تهيئة ملفات الـ Cookies

لإعداد ملف الكعكة سنقوم بإستدعاء الدالة (setcookie) و التي تأخذ القيم التالية:

المعطى Parameters	الوصف Description
name	إسم الكعكة (إجباري)
value	قيمة الكعكة
expires	الوقت بالثواني لصلاحية الكعكات
domain	إسم موقع الإنترنت الذي خزن ملف الكعكة
path	تقوم بتحديد ما إذا كان ملف الكوكي متاحا لكامل

	الموقع أو سيكون متاح لصفحات محددة داخل مجلد يتم تحديده. الخيار (/) يعني إتاحة الملف لكامل الموقع.
secure	هذا الخيار يحدد ما إذا كان إتصال HTTP الآمن مطلوب أم لا

مثال:

```
setcookie('user_name', 'Ali ',time()+1000, '/');
```

بسبب أن ملفات الكعكات سيتم إعدادها بواسطة HTTP headers فيجب وضعها قبل أي عملية طباعة بواسطة شفرة PHP . في حال كان ملف PHP مضمنا مع وسوم HTML فيجب وضع الدالة أيضا قبل وسوم HTML. عدم أخذ هذا في الاعتبار سيؤدي إلى ظهور رسائل تحذير و سيلغي إعداد ملف الكعكة.

قراءة ملفات الـ Cookie

يتم قراءة ملفات الكعكات عن طريق الموقع الذي قام بتخزينه. و ستكون مخزنة داخل المصفوفة الترابطية \$_COOKIE و سيتم قراءتها بواسطة ملف PHP الموجود على نفس الموقع. المثال التالي يوضح كيفية قراءة ملف الكعكة:

```
<?php
if(isset($_COOKIE['user_name']))
{
    echo "Welcome:".$_COOKIE['user_name'];
}
else
{
    echo "No cookie has been set";
}
?>
```

حذف ملفات الـ Cookies

لحذف ملف كعكة نقوم بإستدعاء الدالة (setcookie()) و نقوم بكتابة نفس قيم ملف الكعكة الذي قمنا بإنشاءه مسبقا لكن مع تحديد وقت الصلاحية ملف الكعكة الى وقت سابق كالتالي:

```
<?php
$time = time()+60;
setcookie('user_name','Ali',$time-1000, '/');
?>
```

العمل مع الجلسات Working with sessions

الجلسة session تعمل نفس عمل الكعكات بتوفير طريقة للإستمرارية (تذكر الحالة) مع إختلاف بسيط في التعامل. و كما ذكرنا مسبقا أن الكعكات تقوم بتخزين ملف نصي في جهاز العميل. لذلك المشكلة مع تخزين

الملف في جهاز العميل أنه من الممكن أن يتم قراءة و تعديل البيانات المخزنة داخل ذلك الملف. لهذا السبب، كثير من مواقع الإنترنت تفضل استخدام الجلسة بدلا من الكعكات. تعمل الجلسات نفس الكعكات مع اختلاف أنه بدلا من تخزين البيانات في جهاز العميل يتم تخزين البيانات على الجهاز الخادم server لذلك السبب تعتبر الجلسات أكثر أمانا. أخير الجلسات تنتهي بمجرد إقفال المستخدم لمتصفح الإنترنت.

إنشاء الجلسة Creating session

لإنشاء جلسة كل ما علينا عمله هو استدعاء الدالة (session_start) في بداية كل صفحة نرغب بأن نستخدم في الجلسة مع ضرورة الإنتباه لما يلي:

- هذه الدالة يجب أن تستدعى مرة واحدة فقط في كل صفحة.
 - يجب أن نستدعي هذه الدالة قبل طباعة أي بيانات بواسطة الشفرة البرمجية بالإضافة إلى أنها يجب أن تسبق وسوم HTML.
- في حال تم طباعة أي مخرجات قبل استدعاء الدالة فلن تعمل الجلسة و سيظهر بدلا ذلك رسالة تحذير.

إنشاء متغيرات الجلسة Creating session variables

لإنشاء متغير الجلسة ، نستطيع إضافته إلى المصفوفة العامة \$_SESSION (ترابطية Associative) بنفس الطريقة التي نتعامل بها مع المتغيرات. فلو أردنا مثلا تخزين اسم الزائر و طباعة رسالة ترحيب له. فسوف نقوم بتخزين اسمه في الجلسة كالتالي:

```
$_SESSION['name'] = $_POST['name'];
```

كما هو واضح قمنا بتخزين اسم الزائر و المدخلة عن طريق نموذج التسجيل في المصفوفة العامة \$_SESSION. نستطيع الآن استخدام متغير الجلسة السابق في كل صفحة تبدأ ب (session_start).

حذف متغير الجلسة Destroying session variable

في حال أصبح متغير الجلسة غير مطلوب بواسطة الشفرة البرمجية فتستطيع تدميره/حذفه عن طريق استدعاء الدالة (unset) كالتالي:

```
unset($_SESSION['name']);
```

في هذه لن يصبح المتغير السابق متاحا لأي صفحة.

حذف الجلسة Destroying session

في حال لم يعد هناك أي حاجة للجلسة الحالية نستطيع تدمير الجلسة عن طريق استدعاء الدالة session_destroy () مثال: إنشاء عداد الزيارات

```
<?php
session_start();
if(isset($_SESSION['view']))
{
    $_SESSION['view'] = $_SESSION['view']+1;
}
else
{
    $_SESSION['view'] = 1;
}
echo $_SESSION['view'];
```

إعادة توجيه المستخدم Redirecting user

- في أحيان كثيرة تكون هناك حاجة لإعادة توجيه المستخدم إلى صفحة أخرى . فلو أردنا مثلاً منع وصول زوار الموقع لإحدى صفحات الموقع (بسبب سرية محتوى الصفحة و لتقييد الوصول) فنستطيع تحقيق ذلك بإعادة توجيه متصفح الزائر إلى صفحة أخرى غير الصفحة الحالية التي وصل إليها (بكتابة عنوانها مباشرة في URL). لإعادة توجيه المستخدم إلى صفحة أخرى نستطيع إحدى الطرق التالية:
- عن طريق إستدعاء الدالة (header()) و التي تقوم بإرسال متصفح الزائر إلى صفحة محددة. لكن في هذه الحالة لابد الإنتباه إلى الدالة هذه يجب أن يتم إستدعاءها قبل أي وسوم HTML و قبل مخرجات تتم عن طريق البرنامج.
 - الطريقة الثانية عن طريق إضافة شفرة جافاسكربت ضمن شفر PHP. و في هذه الحالة لدينا الحرية في وضعها في أي مكان داخل شفرة PHP.

المثال التالي يوضح كيفية التعامل مع الدالة (header() :

```
<?php
if(!isset($_POST['login']))
{
    header("Location:login.php");
}
?>
```

في حال أردنا إستخدام شفرة جافا سكربت بدلا من ذلك فنستطيع عمل ذلك كما في المثال التالي:

```
<?php
if(!isset($_POST['login']))
{
    echo "سيتم تحويلك الآن إلى صفحة تسجيل الدخول";
    echo "
        <script>
        window.location = 'login.php';
        </script>
    ";
}
?>
```

الروابط التفاعلية Dynamic links

في الفصول السابقة كنا نقوم بإرسال المعلومات عن طريق النماذج (باستخدام POST و GET) إلى صفحات PHP لكي يتم معالجتها. لكن ماذا لو أردنا بدلا من إرسال المعلومات عن طريق النماذج ، نقوم بإرسالها عن طريق الروابط التشعبية ضمن الوسم <A>. في هذه الحالة سيتم إرسال المعلومات إلى المصفوفة العامة GET (مصفوفة ترابطية Associative array). و كما هو معلوم لدينا أنه في المصفوفات الترابطية الفهارس / المفاتيح عبارة عن نصوص و ليست أرقام. لذلك لو أردنا أن نضيف عنصر داخل المصفوفة GET و له القيمة Ahmad فنقوم بما يلي :

```
$_GET[ 'name' ] = "Ahmad" ;
```

لكن كيف سيتم إرسال رابط تشعبي و يتم تخزين قيمه داخل المصفوفة GET؟
لكي نقوم بذلك نقوم أولاً بكتابة اسم الصفحة التي ستقوم باستقبال البيانات ، مثلاً login.php متبوعة
بعلامة إستفهام ؟ و من ثم إسم الفهرس/ المفتاح متبوعة بعلامة يساوي = يليها قيمة العنصر كالتالي:

القيمة اسم العنصر إسم الصفحة متبوع بـ ؟

في حال أردنا إضافة أكثر من قيمة نقوم بالفصل بين القيم كالتالي:

المثال التالي يوضح كيفية التعامل مع الروابط التفاعلية:

الصفحة login.php

```
<a href="login.php?name=Ahmad&&password=12345"> Dynamic Link </a><br />
<?php
if(isset($_GET['name']) && isset($_GET['password']))
{
    echo"Your name:".$_GET['name']."<br/>";
    echo"Your password:".$_GET['password'];
}

?>
```

أسئلة الفصل

١. ما الفرق بين : session و cookie.
٢. كيف نستطيع حفظ ملف cookie و جعله متاحا للمجلد /foo/.
٣. ما الفرق بين استخدام () unset و () session_destroy مع الجلسات.
٤. ما الطرق المستخدمة في إعادة توجيه المستخدم و ما الفرق بينها.
٥. كيف نستطيع إنشاء رابط تفاعلي.