

دورة في

# بناء برنامج منتديات بسيط باستخدام PHP و MySQL

*(MySimpleBB)*

تأليف: محمد قاسم حسين

# جدول المحتويات

- معلومات حول الدورة
- [اليوم الاول] مقدمه و العمليات الاساسيه
- [اليوم الثاني] الاقسام
- [اليوم الثالث] تحسين الاقسام - الجزء الاول : طرح المشكله
- [اليوم الرابع] تحسين الاقسام - الجزء الثاني : حل المشكله
- [اليوم الخامس] الاعضاء
- [اليوم السادس] المواضيع و الردود
- [اليوم السابع] الامان

# معلومات حول دوره

- طُرحت الدورةُ في منتدى PHP (العنوان : [/http://forum.montadaphp.net](http://forum.montadaphp.net)) مُنذ 2010-2-13 إلى 2010-2-19
- تقع دوره تحت رخصة GNU FDL
- جميع الشيفرات دوره تقع تحت رخصة GNU GPL
- موقع مؤلف دوره : <http://www.maastaar.com>

# اليوم الاول : مقدمه و العمليات الاساسيه

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

نبدأ اليوم إن شاء الله بدورتنا التي من خلالها سننشئ سويًا برنامج منتديات بسيط باستخدام لغة PHP مع محرك قواعد البيانات MySQL، ليس الهدف الأساسي من هذه الدورة هو الجواب على سؤال "كيف ابرمج برنامج منتديات"، فكما تعلمون انه عندما يبدأ الشخص بكتابة برامجه بلغة PHP لن يسأل مثل هذا السؤال لأنه سيكون لديه تصوراً واضحاً حول طريقة بناء هذا النوع من البرامج، من خلال هذه الدورة سوف أُلخّص لكم من ما تعلمته مع هذا النوع من البرامج.

قبل البدء، لابد ان تكون معرفتك بـ PHP و MySQL و SQL معرفه جيده، مع الاسف لن تعلمك هذه دوره القصيره اساسيات اللغه منذ البدء وصولاً إلى بناء برنامج منتديات، لاحظوا انني في هذه دوره لن اركز على استخدام HTML كثيراً، و بالتالي قد تلاحظون بعض التجاوزات في كتابة HTML خلال دورتنا، يمكنك إصلاح هذه التجاوزات ان شئت :-)

سنبدأ اليوم بتصميم برنامجنا، كيف سيكون و ما هي الميزات الاساسيه التي سنكتبها، سوف نتبع في هذه دوره ابسط اسلوب في البرمجه مع PHP حتى اننا لن نستخدم اي محرك قوالب، بل سنضع شيفرة الـ HTML في ملفات PHP مباشره، كما اننا لن نعتمد على اي كائن يخص قواعد البيانات، سنستخدم دوال MySQL بشكل مباشر، سنسمي برنامجنا بـ MySimpleBB بالطبع انت حر بالاسم و لكن لابد من فرض اسم منذ البدايه للبرنامج :-).

المميزات التي سيحتويها MySimpleBB هي :

- الاقسام.
- المواضيع و الردود بالطبع :-).
- المستخدمين.

لابد من التنويه إلى ان هذه الدروس تقع تحت رخصة GNU FDL و البرنامج النهائي يقع تحت رخصة GNU GPL، و بالتالي محتويات هذه دوره حره و وفقاً لهاتين الرخصتين.

لنبدأ ببعض المهمات البرمجيّه البسيطة و المتكرره والتي بالطبع لا تفوتكم :-)، سنكتب ملف يحتوي على الاعمال الروتينيّه والتي يجب ان تتم في كل صفحه نفتحها عند تصفح منتدانا، احد اهم هذه الاعمال هو الاتصال بقاعدة البيانات، لنسمي الملف بـ common.php و نبدأ به كالتالي :

```
//////////
// DB information
$db_host      =      'localhost';
$db_user      =      '';
$db_password  =      '';
$db_name      =      'mysimplebb';
```

هذه المتغيرات تحتوي على معلومات قواعد البيانات، سوف نستخدمها للاتصال بقواعد البيانات، وضعناها في متغيرات منفصله حتى نتمكن من تغيير المعلومات بسهولة، غير المعلومات بما يناسبك :-)

خطوتنا التاليه هي الاتصال بقواعد البيانات باستخدام الداله mysql\_connect و الداله mysql\_select\_db لإختيار قاعدة البيانات المطلوبه :

//////////

// Connect and select database

```
$connect = mysql_connect($db_host,$db_user,$db_password) or die(mysql_error());  
$select = mysql_select_db($db_name) or die(mysql_error());
```

حتى الآن هذا كل شيء، كل ما نحتاجه هو الاتصال بقاعدة البيانات في كل ملف. نلتقي غداً إن شاء الله :-)

# اليوم الثاني : الاقسام

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

سنتحدث اليوم عن احد الميزات الهامه في برامج المنتديات الحديثه وهي الاقسام، يمكننا إضافة الاقسام مع MySimpleBB ويكون لكل قسم مواضيعه الخاصه به، هناك نوعان من الاقسام، النوع الاول : الاقسام العاديه وهي الاقسام التي تُكتب المواضيع فيها، النوع الثاني : الاقسام الرئيسييه وهي الاقسام التي تقع تحتها الاقسام العاديه و للتسهيل سنطلق عليها اسم "التصنيفات"، مثل المنتديات الحديثه بالضبط يمكنني من خلال هذه الميزه وضع تصنيف ليكن اسمه "البرمجه الآمنه" وتحت اقسام عاديه يمكن كتابة المواضيع فيها مثل "المنتدى العام"، "البرمجه الآمنه مع MySQL"، "البرمجه الآمنه مع SQLite" و هكذا، هل الامور واضحه الآن؟ :-)

لنبدأ في قواعد البيانات، سيكون لدينا جدولاً واحداً للنوعين من الاقسام اسمه sections و حقله كالتالي :

id : وهو رقم مميز لكل قسم.

title : اسم القسم.

parent : القسم الاب، في حال كان القسم رئيسياً (تصنيف) ستكون قيمة هذا الحقل هي 0، اما اذا كان القسم عادياً ستكون قيمة هذا الحقل = الرقم المميز للتصنيف الذي يقع القسم العادي تحته

لتوضيح الامور اكثر بخصوص الحقل parent لنضرب مثلاً، لنفرض ان لدينا تصنيفاً رقمه المميز (id) يساوي 3 و اسمه (title) "نظم التشغيل"، و لأنه قسم رئيسي (تصنيف) فإن الحقل parent سيكون قيمته صفراً، و نفرض ان لدينا قسماً عادياً متفرعاً من التصنيف "نظم التشغيل" ليكن اسمه "غنوا\لينكس" ورقمه المميز (id) يساوي 5، و لأنه قسم عادي يقع تحت التصنيف "نظم التشغيل" فيجب ان تكون هناك قيمة في الحقل parent غير قيمه 0، يا ترى ما هي؟ نعم بالضبط :-). كما توقعتم تماماً ستكون 3 لأنه الرقم المميز للتصنيف "نظم التشغيل" والذي يقع القسم "غنوا\لينكس" تحته، اعتقد ان الامور باتت واضحه :-).

لماذا نتبع هذا الاسلوب، ببساطه حتى نتمكن من تحقيق ميزة "التصانيف و الاقسام العاديه"، ماذا لو كان لدينا اقسام عاديه فقط ولا تختلف في تصانيفها بالطبع لن نحتاج إلى الآليه المذكوره في الاعلى، وبالطبع من خلال هذا الاسلوب يمكننا تحقيق ميزة "الاقسام الفرعيه" وهي الاقسام التي تتفرع من الاقسام العاديه.

بعدها وضحنا الامور لنبني الجدول الآن كالتالي :

```
CREATE TABLE `sections` (  
  `id` INT( 9 ) NOT NULL AUTO_INCREMENT ,  
  `title` VARCHAR( 255 ) NOT NULL ,  
  `parent` INT( 9 ) NOT NULL ,  
  PRIMARY KEY ( `id` )  
) ENGINE = MYISAM
```

لندخل بعض المعلومات في الجدول حتى تكون هناك اقساماً حقيقيه يمكننا التعامل معها :-). نريد اضافة تصنيف اسمه "قسم PHP" و قسم ليكن اسمه "منتدى البرمجه العام" يقع تحت التصنيف السابق "قسم PHP"، و المعلومات كالتالي :

```
id = 1
PHP قسم = title
parent = 0
```

```
id = 2
منتدى البرمجه العام = title
parent = 1
```

اوامر SQL لإدخال هذين القسمين :

```
INSERT INTO `sections` (
`id`,
`title`,
`parent`
)
VALUES (
'1', 'قسم PHP', '0'
), (
'2', '1', 'منتدى البرمجه العام'
);
```

لنبدأ بالعمل الحقيقي الآن (-:), وهو كتابة الملف index.php وهو صفحة المنتدى الرئيسي، لننسى عالم البرمجه قليلاً و نتحدث معاً كمستخدمين ما رأيكم؟ (-: ) كمستخدم عادي عندما أنوي زيارة أي منتدى على شبكة الانترنت سأتوقع أن صفحته الرئيسي ستحتوي بشكل اساسي على مجموعه من الاقسام التي تقع تحت تصنيفات معينه، يمكنني اختيار القسم الذي يناسبني فأضغط على وصلته لأصل إلى صفحة اخرى تعرض قائمه بمواضيع هذا القسم.

هذه هي وظيفة الملف index.php الذي سنشرع في كتابته، و هي عرض التصنيفات و الاقسام التي تقع تحتها، قبل كل شيء نطلب تضمين الملف common.php الذي كتبناه في الدرس السابق حتى يقوم بالمهام الروتينييه و يتصل بقواعد البيانات :

```
<?php
include('common.php');
```

ما نريده الآن هو طلب التصنيفات من قواعد البيانات لعرضها، و كما نعلم التصنيفات تتميز بأن قيمة الحقل parent فيها دائماً = 0 و بالتالي نكتب استعلام SQL على هذا الاساس، و قبل كتابة الاستعلام سنفتح الوسم <ul> لترتيب عرض الصفحه :

```
echo '<ul>';

$cats_query = mysql_query("SELECT * FROM sections WHERE parent='0'");
```

كالمعاده (-: ) نستخدم الآن الداله mysql\_fetch\_array لإحضار النتائج من قواعد البيانات على شكل مصفوفه ثم نطبع اسم التصنيف :

```
while ($cat = mysql_fetch_array($cats_query))
{
    echo '<li>' . $cat['title'];
```

حتى هذا السطر قمنا بتحقيق المطلوب الاول وهو عرض التصنيفات (لا بد من الملاحظة اننا لم نغلق التكرار while بعد)، المطلوب الثاني هو عرض الاقسام التي تقع تحت هذه التصنيفات، كل قسم تحت التصنيف الذي يخصه، كما نعلم ان كل قسم عادي قيمة الحقل parent الخاص به هو الرقم المميز للتصنيف الاب و الذي يقع القسم العادي تحته، و بالتالي لإحضار قائمة الاقسام الكاملة التي تقع تحت هذا التصنيف سنكتب إستعلاماً جديداً يأخذ الرقم المميز للتصنيف الحالي الذي طبعنا اسمه و يطلب قائمة الاقسام التي تقع تحت هذا التصنيف كالتالي :

```
$sections_query = mysql_query("SELECT * FROM sections WHERE parent='" . $cat['id'] . "'");
```

نستخدم الدالة mysql\_fetch\_array لإحضار النتائج و طباعتها :

```
while ($section = mysql_fetch_array($sections_query))
{
    echo '<ul><li><a href="forum.php?id=' . $section['id'] . "'>' . $section['title'] .
'</a></li></ul>';
}
```

لاحظ اننا استخدمنا الوسم a لتحويل اسم القسم إلى وصله، ولا يخفى عليكم ان هذه الوصله ستوصلنا إلى صفحه تعرض لنا قائمة المواضيع المكتوبه في القسم المُختار.

حققنا الآن المطلوب الثاني و طبعنا الاقسام العاديه كل قسم تحت تصنيفه، نُغلق الآن الوسوم المفتوحه و نُغلق التكرار الاول و نُهي الملف :

```
    echo '</li>';
}

echo '</ul>';

?>
```

هذا كل شيء (-)، حسناً حصلنا اليوم على جزء هام من برامج المنتديات وهي الصفحه الرئيسيه التي تعرض اقسام المنتدى، و تعرفنا على تركيبه الاقسام في قواعد البيانات و الغرض منها (-)



```
<?php

include('common.php');

echo '<ul>';

$cats_query = mysql_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
    echo '<li>' . $cat['title'];

    $sections_query = mysql_query("SELECT * FROM sections WHERE parent='" . $cat['id'] . "'");

    while ($section = mysql_fetch_array($sections_query))
    {
        echo '<ul><li><a href="forum.php?id=' . $section['id'] . '">' . $section['title'] .
'</a></li></ul>';
    }

    echo '</li>';
}

echo '</ul>';

?>
```

# اليوم الثالث : تحسين الاقسام - الجزء الاول : طرح المشكله

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

في الدرس السابق تحدثنا عن تركيبية الاقسام و طريقة تخزينها في قواعد البيانات، ثم كتبنا الملف `index.php` ووظيفته عرض التصنيفات و ما يقع تحتها من اقسام، في درسنا اليوم سنتعرف على موطن ضعف هام في الشيفره التي كتبناها (بالأمس :-).

لنعود إلى شيفرتنا عند السطر 7 و 8 و 9 :

```
$cats_query = mysql_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
```

كما تعلم، هذا الجزء من الشيفره يطلب جميع البيانات من الجدول `sections` والتي قيمة الحقل `parent` فيها يساوي 0 وهي التصنيفات كما شرحنا مسبقاً، ثم نكمل شيفرتنا حتى نصل إلى السطر رقم 12 وهو :

```
    $sections_query = mysql_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
    """);
```

و كما نعلم يقوم هذا السطر بطلب الاقسام التي تقع تحت تصنيف محدد، هذا السطر عبارته عن استعلام يُرسل إلى MySQL و كما نعلم كلما زاد عدد الاستعلامات كلما قل اداء البرنامج و كلما زادت حاجته إلى الموارد، حسناً ما هي المناسبه لقول هذا الكلام الآن؟ (-) سأخبرك، لاحظ ان الاستعلام الذي ذكرته في الاعلى موجود داخل تكرار `while`، هذا يعني ان الاستعلام سيتم تنفيذه في كل مره يتحقق الشرط الموجود في `while` إلى ان ينتهي التكرار، عندما نراجع الملف `index.php` سنجد اننا كتبنا استعلامين فقط للاصناف و الثاني للاقسام و بالطبع، طلب استعلامين بهذه البساطه من قواعد البيانات عدد معقول جداً و لن يؤثر على الاداء، و لكن عندما ندرك الحقيقه التي ذكرناها منذ قليل و هي ان الاستعلام الثاني يقع داخل حلقة `while` سنعلم حتماً اننا نطلب اكثر من استعلامين من قواعد البيانات، لان كل تصنيف يتم عرض اسمه سنطلب بعدها من قواعد البيانات عرض الاقسام التي تقع تحته من خلال السطر المذكور في الاعلى، حسناً ماذا لو كان لدينا 50 تصنيفاً؟ ماهي عدد الاستعلامات التي سنطلبها من قواعد البيانات؟ نعم هي 51 استعلاماً، الاول يطلب قائمة التصنيفات الـ 50 كامله، و كل تصنيف بعد طباعة اسمه سننفذ الاستعلام الثاني الذي يأخذ الاقسام التي تقع تحت هذا التصنيف، و هكذا علاقه طرديه كلما زادت عدد التصنيفات كلما زادت الاستعلامات.

عند بناء برامج المنتديات الحقيقيه فإن الصفحه الرئيسيه ستكون لها مهام اخرى غير طباعة التصنيفات و الاقسام، ففي برنامجنا الحالي لا وظيفه لملف `index.php` سوى هذه الوظيفه و لكن برامج المنتديات الحقيقيه لديها الكثير من الوظائف

الآخري و التي تتطلب غالباً إستعلامات من قاعدة البيانات، و هكذا تزيد عدد الاستعلامات والتي تؤدي حتماً إلى أداء اسوأ، و بالطبع لتحسين الاداء لا بد من تقليل الإستعلامات بقدر الإمكان، في حالتنا الآن لا يوجد سوى إستعلامان، و لكن المشكله تكمن في الاستعلام الثاني الذي يقع داخل حلقة while و تتكرر هذه الحلقة بعدد التصنيفات الموجوده، و بالتالي لابد من طريقه للحصول على الاقسام التي تقع تحت التصنيفات بدون كتابة هذا الاستعلام داخل الحلقة while، هل يمكن ذلك؟ :-)

قبل الجواب على هذا السؤال و قبل طرح الحل، لنقوم ببعض الاختبارات، سنكتب داله نستخدمها بدلاً من mysql\_query (تعتمد عليها في الحقيقه)، ووظيفة هذه الداله تنفيذ الاستعلام المطلوب و وضع عدد عام يعد عدد الاستعلامات التي قمنا بطلبها، الموضوع بسيط أليس كذلك؟ :-). نفتح الملف common.php و نضيف به دالتنا و ليكن اسمها simple\_query، هدفنا من هذه الداله الحصول على العدد النهائي من الاستعلامات التي طلبناها من قواعد البيانات :

```
$query_counter = 0;

function simple_query($query)
{
    global $query_counter;

    $query_counter++;

    return mysql_query($query);
}
```

ولا يخفى عليكم ان المتغير \$query\_counter هو الذي سيحتوي على عدد الاستعلامات التي طلبناها في الصفحة، بعد كتابة هذه الداله نستخدمها في شيفرتنا القديمه index.php بدلاً من mysql\_query بحيث يصبح ملف index.php كالتالي :

```
<?php

include('common.php');

echo '<ul>';

$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
    echo '<li>' . $cat['title'];

    $sections_query = simple_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
    "'");

    while ($section = mysql_fetch_array($sections_query))
    {
        echo '<ul><li><a href="forum.php?id=' . $section['id'] . '>' . $section['title']
        . '</a></li></ul>';
    }
}
```

```

    }

    echo '</li>';
}

echo '</ul>';

?>

```

حسناً و لطباعة عدد الاستعلامات التي طلبناها من قواعد البيانات نضيف السطر التالي اسفل index.php :

```
echo '<br />Queries number : ' . $query_counter;
```

بالنسبه لي فهذه هي تركيبة التصنيفات و الاقسام التي اضفتها إلى قواعد البيانات :

```

* PHP Section
  o General Programming
  o Object Oriented Programming
* Operating System Kernels
  o Linux
  o HelenOS

```

تصنيفان وهما PHP Section و Operating System Kernels و أسفل كل واحد منهما قسمين، و بالتالي ما هو عدد الاستعلامات الذي سيظهر عندي في النهايه؟ (-: إستعلام واحد يطلب التصنيفات - وهما إثنان - و لكل تصنيف إستعلام يخصه يطلب الاقسام التي تقع تحته، و بالتالي ستكون 3 إستعلامات.

سنقوم الآن بإختبار بسيط و لكن قبله يجب علينا تفريغ جدول sections إذا كان يحتوي على معلومات، بعد تفريغ الجدول سنقوم بإضافة 50 تصنيفاً في قواعد البيانات و اسفل كل تصنيف قسم واحد، لتنفيذ ذلك يمكننا إستخدام الشيفره التاليه :

```

<?php

include('common.php');

$x = 0;

while ($x < 50)
{
    $insert_cat = mysql_query("INSERT INTO sections(id,title,parent) VALUES(NULL, ' " .
    rand(0,500) . "', '0')");
}

```

```

if ($insert_cat)
{
    $id = mysql_insert_id();

    echo '<strong>Cat : ' . $id . ' -> OK</strong><br />';

    $insert_section = mysql_query("INSERT INTO sections(id,title,parent)
VALUES(NULL, ' . rand(500,1000) . ',' . $id . ')"");

    if ($insert_section)
    {
        echo '<em>Section : ' . mysql_insert_id() . ' -> OK</em><br />';
    }
}

$x += 1;
}

?>

```

قم بتخزين هذه الشيفره في ملف ليكن اسمه `add_sections_test.php` كما تلاحظ تضيف هذه الشيفره 50 تصنيفاً و تعطي لكل تصنيف إسماً وهو رقم عشوائي، و اسفل كل تصنيف تضيف قسماً مع اسم عشوائي، قم بتشغيل الملف `add_sections_test.php` حتى يؤدي وظيفته، و نعود بعدها إلى الصفحه الرئيسيه لنرى التصنيفات و الاقسام الجديده، اذا لاحظنا عدد الإستعلامات سنجده 51، تماماً كما فرضنا سابقاً، نشغل الملف مره اخرى ليضيف 50 جديده، نعود للصفحه الرئيسيه و سنجد عدد الاستعلامات اصبح 101، جرب تشغيل الملف `add_sections_test.php` عدّة مرات متتاليه ليضيف المزيد من الاقسام و انتقل دائماً إلى الصفحه الرئيسيه لتلاحظ عدد الإستعلامات.

كلما زادت التصنيفات كلما زاد عدد الاستعلامات، و اذا كان عدد الاستعلامات كافياً ستلاحظ البطئ الشديد على الصفحه، لنفرغ الجدول `sections` و ننتقل إلى الملف `add_sections_test.php` و نغير السطر 7 وهو

```
while ($x < 50)
```

إلى :

```
while ($x < 2000)
```

نعم لنجرب إضافة 2000 تصنيف دفعه واحده (-)، شغلّ الملف و انتقل إلى الصفحه الرئيسيه لتلاحظ ان ادائها بدأ بالتراجع و اصبحت أبطأ، و اذا كنتم تشغلون البرنامج على سيرفر شخصي يمكنكم ملاحظة تأثير الصفحه على معالج حواسيبكم (CPU) عن طريق برامج المراقبه، لنضيف 2000 تصنيف آخر حتى يصبح الناتج 4000 تصنيف، عند الذهاب للصفحه الرئيسيه

ستلاحظون مدى البطء الشديد عن تحميل الصفحة و سيصبح عدد الإستعلامات 4001، و من يراقب العمليات (Processes) التي تعمل على حاسوبه عندما يطلب الصفحة index.php للعرض سيجد ان سيرفر MySQL هو الاعلى في إستهلاك المعالج (CPU)، و طبعاً السبب معروف، لاننا طلبنا منه 4001 استعلام :-)

System Monitor

Monitor Edit View Help

System Processes Resources File Systems Hardware

Load averages for the last 1, 5, 15 minutes: 0.98, 0.74, 0.63

Process Name	User	Status	Virtual Memory	Resident Memory	% CPU ^	ID
mysql	mysql	Sleeping	108.2 MiB	17.3 MiB	42	3186
X	root	Running	313.9 MiB	47.8 MiB	23	2699
gnome-system-mo	maastaar	Running	90.5 MiB	21.8 MiB	18	6331
firefox	maastaar	Sleeping	471.9 MiB	257.9 MiB	8	4715
gnome-screenshot	maastaar	Running	106.8 MiB	16.6 MiB	0	6403
gnome-panel	maastaar	Sleeping	107.5 MiB	29.3 MiB	0	3915
metacity	maastaar	Sleeping	22.6 MiB	11.4 MiB	0	3906
httpd2-prefork	wwwrun	Sleeping	93.0 MiB	8.0 MiB	0	6112
httpd2-prefork	wwwrun	Sleeping	93.0 MiB	7.9 MiB	0	6111
httpd2-prefork	wwwrun	Sleeping	93.2 MiB	8.1 MiB	0	6110
pickup	postfix	Sleeping	6.7 MiB	1.7 MiB	0	5998
leafpad	maastaar	Sleeping	75.0 MiB	14.4 MiB	0	5873
httpd2-prefork	wwwrun	Sleeping	95.1 MiB	11.2 MiB	0	5685
gedit	maastaar	Sleeping	128.5 MiB	31.9 MiB	0	5455
gvfsd-http	maastaar	Sleeping	8.7 MiB	2.5 MiB	0	5088

End Process

حسناً، ماهو الحل يا ترى؟ كيف يمكننا إضافة ما شئنا من التصنيفات و الاقسام مهما كان عددها بدون استهلاك موارد بشكل كبير؟ هذا ما سنطرحه غداً إن شاء الله :-)

فرغ الجدول sections و حضر لك كوباً من القهوة بعد درس اليوم الطويل و استعد للإنتلاق غداً إن شاء الله :-).

# اليوم الرابع : تحسين الاقسام - الجزء الثاني : حل المشكله

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

في الدرس السابق اكتشفنا مشكله في الاداء و فصلناها، و سنطرح اليوم حل هذه المشكله إن شاء الله :-)

كما نعلم ان مشكلتنا الاساسيه هي وجود سطر إستعلام داخل حلقة التكرار (while)، و بالتالي كلما زادت مرات التكرار زادت عدد الإستعلامات التي ستُنَفَّذ، و كلما زاد عدد الاستعلامات زاد طلب الموارد و قل الاداء، و بالتالي لابد من إيجاد طريقه نأخذ من خلالها التصنيفات و الاقسام التي تقع تحتها بدون كتابة إستعلام داخل حلقة التكرار (while).

قبل البدء بمناقشة الحل، من المفترض اننا فرغنا الجدول sections في الامس، اذا لم تكن فعلت ذلك فإفعل الآن :-). نحتاج إلى اضافة 50 قسماً فقط، عدّل الملف add\_sections\_test.php من السطر :

```
while ($x < 2000)
```

غيره إلى

```
while ($x < 50)
```

ثم شغل الملف ليقوم بالمطلوب.

## الطرح النظري للحل

نعود لمشكلتنا، كما نعلم ان المشكله تكمن في الإستعلام التالي :

```
$sections_query = simple_query("SELECT * FROM sections WHERE parent=' . $cat['id'] . '");
```

وهو الذي يأخذ قائمة الاقسام، حل مشكلتنا هو حذف هذا السطر الذي يقع داخل حلقة التكرار while، ولكن لا يخفى عليك بحذف هذا السطر لن نتمكن من اخذ معلومات الاقسام و كل ما سنحصل عليه هو معلومات التصنيفات.

حسناً :-). هناك طريقه اخرى لأخذ قائمة الاقسام بدون إستعلام و ذلك عن طريق الكاش (cache)، الفكرة من الكاش هو تخزين نسخه اخرى من معلومات معينه بهدف تحسين الاداء - لاحظوا اننا نتحدث عن فكرة الكاش هنا بالنسبه لتطبيقات الويب

- كيف سنستخدم الكاش في حالتنا هذه؟

سنضيف حقلاً إلى الجدول sections نسميه forums\_cache، هذا الحقل سيحتوي على معلومات الكاش للاقسام التي تقع تحت كل تصنيف، لنفرض ان لدينا تصنيف اسمه "قسم البرمجه" و اسفل هذا التصنيف ثلاث اقسام "البرمجه بلغة بي اتش بي"، "البرمجه بلغة سي"، "البرمجه بلغة جافا"، بالطبع سنضيفها إلى قواعد البيانات كالعاده، لكل واحد منهم مدخل في الجدول sections، الشيء الجديد الآن هو الحقل forums\_cache، هذا الحقل سيحتوي على المعلومات فقط بالنسبه للتصنيفات (قيمة parent فيها تساوي 0 كما نعلم) و المعلومات هي قائمة الاقسام التي تقع هذا التصنيف، و في فرضنا السابق سيحتوي الحقل forums\_cache بالنسبه للتصنيف "قسم البرمجه" على مصفوفة تجمع معلومات الثلاث منتديات، و بالتالي عندما نريد إحضار قائمة التصنيفات و الاقسام التي تقع تحتها، سنستخدم استعمال لإحضار قائمة التصنيفات وهو نفسه الذي كتبناه في ملف index.php و الموجود في السطر 7 :

```
$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
```

الذي تغير الآن هو وجود الحقل forums\_cache والذي يحتوي على مصفوفة الاقسام التي تقع تحت التصنيف، و بالتالي يمكننا إستخدام المصفوفة المأخذه في الحقل forums\_cache لكل تصنيف و عرض معلومات الاقسام التي تقع تحت التصنيف، و بالتالي نكون قد استغينا عن الاستعلام الذي سبب لنا مشكله في الاداء.

## تنفيذ الحل برمجياً

لا ننسى اولاً إضافة الحقل forums\_cache إلى الجدول sections ليكون من النوع text :

```
ALTER TABLE `sections` ADD `forums_cache` TEXT NOT NULL ;
```

قبل الشروع بكتابة الحل، سنقوم ببعض الاختبارات التوضيحيه، قم بإنشاء ملف جديد ليكن اسمه forums\_cache\_test.php، كما أسلفنا، ان الحقل forums\_cache يجب ان يحتوي مصفوفة الاقسام التي تقع تحت التصنيف، ما نريده الآن عملياً هو إحضار قائمة التصنيفات ثم الحصول على قائمة الاقسام ووضعها في مصفوفه، ثم تخزين المصفوفه التي تنتج في الحقل forums\_cache، لنكتب تطبيقاً بسيطاً لذلك في الملف forums\_cache\_test.php.

نطلب اولاً الملف الاساسي common.php :

```
<?php
```

```
include('common.php');
```

مثل الصفحه الرئيسي، نكتب إستعلاماً يطلب قائمة التصنيفات :



```
$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
```

ما نريده الآن الحصول على مصفوفه تحتوي على قائمة الاقسام التي تقع تحت التصنيف، و نقوم بذلك عن طريق التالي :

```
    $sections_query = simple_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
    """);
```

```
while ($sections[] = mysql_fetch_array($sections_query));
```

نلاحظ من الشيفره التي في الاعلى انه اصبح لدينا مصفوفه باسم \$sections والتي تحتوي كما لا يخفى عليكم قائمة الاقسام (-).

لنجرب الآن طباعة قيمة \$sections :

```
print_r($sections);
```

سنلاحظ اننا حققنا ما اردنا (-)، الشيفره الكامله :

```
<?php
```

```
include('common.php');
```

```
$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
```

```
while ($cat = mysql_fetch_array($cats_query))
```

```
{
```

```
    $sections_query = simple_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
    """);
```

```
while ($sections[] = mysql_fetch_array($sections_query));
```

```
print_r($sections);
```

```
}
```

```
?>
```

ما نريده الآن تخزين المصفوفه \$sections داخل الحقل forums\_cache للتصنيف الذي رقمه المميز \$cat['id']، نعم العمليه بسيطه (-) نُدمر المصفوفه إلى الداله serialize لتسهيل تخزينها، ثم نقوم بتحديث معلومات التصنيف ووضع الذي

نتج من الدالة serialize :

```
array_pop($sections);

$ready_to_store = serialize($sections);

$update = simple_query("UPDATE sections SET forums_cache='" . $ready_to_store . "' WHERE
id='" . $cat['id'] . "'");

if ($update)
{
    echo $cat['id'] . ' -> OK <br />';
}

unset($sections);
```

الشيء كله كامله :

<?php

```
include('common.php');

$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
    $sections_query = simple_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
    "'");

    while ($sections[] = mysql_fetch_array($sections_query));

    array_pop($sections);

    $ready_to_store = serialize($sections);

    $update = simple_query("UPDATE sections SET forums_cache='" . $ready_to_store . "' WHERE
id='" . $cat['id'] . "'");

    if ($update)
    {
        echo $cat['id'] . ' -> OK <br />';
    }
}
```

```

unset($sections);
}
?>

```

عند تشغيل الملف سيتم تحديث جميع التصنيفات بمعلومات الاقسام التي تحتها، قبل الإنتقال إلى index.php و تحويلها إلى نظام الكاش لايد من تنويه :-)

لا يخفى علينا انه اصبحت لدينا الآن نسختان من معلومات الاقسام، النسخه الاصليه وهي المدخلات الموجوده في جدول sections، النسخه الاخرى هي نسخة الكاش التي خزناها في الحقل forums\_cache، و بالتالي نسخة الكاش دائماً تتبع النسخه الاصليه، في حال حدوث اي تغيير في النسخه الاصليه من معلومات الاقسام يجب تحديث نسخة الكاش، مثلاً في حال قمنا بتحديث معلومات احد الاقسام او حذفنا احد الاقسام او حتى اضفنا قسماً جديداً تحت التصنيف فيجب علينا تحديث نسخة الكاش حتى تواكب النسخه الاصليه من المعلومات، هل الامور واضحه الآن؟ :-)

ننتقل إلى الصفحه index.php الآن و نتخلص اخيراً من الاستعلام الذي كتم على انفسنا و ما يتعلق به من سطور :-)، احذف السطور التاليه لنبداً بدايه جديده :

```

$sections_query = simple_query("SELECT * FROM sections WHERE parent='" . $cat['id'] .
"");

while ($section = mysql_fetch_array($sections_query))
{
    echo '<ul><li><a href="forum.php?id=' . $section['id'] . '>' . $section['title'] .
'</a></li></ul>';
}

```

عند حذف تلك السطور ستلاحظ ان عدد الاستعلامات تقلص إلى استعلام واحد فقط :-)

ما نريده الآن اخذ البيانات المُخزنه في الحقل forums\_cache، وهي كما تعلم المصفوفه التي تحتوي على اقسام هذا التصنيف ثم استخراج المعلومات من هذه المصفوفه و طباعتها، هذا كل شيء :-)

نُعيد المصفوفه إلى حالتها الطبيعيه :

```

$sections = unserialize($cat['forums_cache']);

```

ثم نستخرج المعلومات و نطبعاها :

```

foreach ($sections as $section)
{
    echo '<ul><li><a href="forum.php?id=' . $section['id'] . '"> . $section['title'] .
'</a></li></ul>';
}

```

الشيء كله كامله :

```

<?php

include('common.php');

echo '<ul>';

$cats_query = simple_query("SELECT * FROM sections WHERE parent='0'");
while ($cat = mysql_fetch_array($cats_query))
{
    echo '<li> . $cat['title'];

    $sections = unserialize($cat['forums_cache']);

    foreach ($sections as $section)
    {
        echo '<ul><li><a href="forum.php?id=' . $section['id'] . '"> . $section['title'] .
'</a></li></ul>';
    }

    echo '</li>';
}

echo '</ul>';

echo '<br />Queries number : ' . $query_counter;

?>

```

انتقل الآن إلى الصفحة الرئيسية لترى ان كل شيء يعمل كما كان يعمل تماماً و لكن الفرق الوحيد هو عدد الإستعلامات (-):  
 نعم نعم انه إستعلام واحد فقط (-):

جرب الآن تشغيل الملف add\_sections\_test.php بعد تعديل السطر :

```
while ($x < 50)
```

لنجرّب هذه المره إضافة 4000 تصنيف بالإضافة إلى الـ 50 الموجوده الآن و بالتالي يصبح السطر كالتالي :

```
while ($x < 4000)
```

بعد عمليتنا الاجراميه التي اضعنا من خلالها 4000 تصنيف يجب ان لا ننسى تشغيل الملف `forums_cache_test.php` ليكونَ معلومات الكاش، ستلاحظ البطئ الشديد عند تشغيل هذا الملف و بالطبع تعلم السبب :-)

الآن انتقل إلى الصفحه الرئيسيه، مفاجأه!! يتم عرض الـ 4000 تصنيف بسرعه شديده و عدد الاستعلامات لا زال واحداً :-)

يكفي لهذا اليوم، نلتقي غداً إن شاء الله :-)

# اليوم الخامس : الاعضاء

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

سنُكمل اليوم إن شاء الله برنامجنا و نضيف إليه ميزة الاعضاء وهي احد الميزات الاساسيه في برامج المنتديات الحديثه، بالطبع لتنفيذ هذه الميزه سنحتاج إلى جدول يخزن قائمة الاعضاء و معلوماتهم، سنحتاج إلى الثلاث حقول الاساسيه وهي id و هو الحقل الذي يخزن الرقم المميز للعضو، username و هو الحقل الذي يخزن اسم المستخدم، password و هو الحقل الذي يخزن كلمة المرور :

```
CREATE TABLE member (  
`id` INT( 9 ) NOT NULL AUTO_INCREMENT ,  
`username` VARCHAR( 100 ) NOT NULL ,  
`password` VARCHAR( 100 ) NOT NULL ,  
PRIMARY KEY ( `id` )  
) ENGINE = MYISAM
```

سنضيف بيانات عضو جديد من اجل تجاربنا (-):

```
INSERT INTO member (  
`id` ,  
`username` ,  
`password`  
)  
VALUES (  
'NULL', 'test', '123'  
);
```

## صفحة تسجيل الدخول

سنبدأ اولاً بصفحة تسجيل الدخول، من خلال هذه الصفحه يُدخل العضو اسم المستخدم و كلمة المرور، و في حال صحة المعلومات يخزن البرنامج كوكيز في حاسوب المستخدم يدل على صحّة المعلومات و ان هذا الزائر يمكنه التمتع بصلاحيات الاعضاء.

لننشئ الملف الجديد login.php، و كاعاده نبدأ بمناداة الملف common.php :

```
<?php
```

```
include('common.php');
```

ثم نعرض النموذج الذي سيُدخل العضو معلوماته فيه :

```
if (empty($_GET['login']))
{
    echo '<form method="post" action="login.php?login=1">';
    echo '<input type="text" name="username">';
    echo '<input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل دخول">';
    echo '</form>';
}
```

نُكمل باقي الملف وهو الجزء الذي سيختبر صحّة المعلومات و تسجيل الكوكيز في حال نجاح الاختبار، اولاً نتحقق هل قام العضو بتعبئة الحقول المطلوبه ام انه تركها فارغه :

```
else
{
    if (empty($_POST['username'])
        or empty($_POST['pass']))
    {
        die('يرجى تعبئة المعلومات');
    }
}
```

بعدها نتأكد من صحة المعلومات عن طريق إرسالها إلى قواعد البيانات و التأكد هل هي موجوده ام لا :

```
$check_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] .
'' AND password='" . $_POST['pass'] . "'");
$correct = mysql_num_rows($check_query);
```

في حال وجود الحقل ستكون قيمة correct تساوي 1 و بالتالي سنختبر هذا المتغير و في حال صحة المعلومات نقوم بتخزين ملفات كوكيز تحتوي على اسم المستخدم و كلمة المرور :

```
if ($correct)
{
    setcookie('mysimplebb_username', $_POST['username']);
    setcookie('mysimplebb_password', $_POST['pass']);

    echo 'تم تسجيل الدخول';
}
```

و إلا نخبر المستخدم ان المعلومات غير صحيحه :

```
else
{
    echo 'خطأ في تسجيل الدخول';
}
?>
```

الشيقره كامله :

```
<?php

include('common.php');

if (empty($_GET['login']))
{
    echo '<form method="post" action="login.php?login=1">';
    echo '<input type="text" name="username">';
    echo '<input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل دخول">';
    echo '</form>';
}
else
{
    if (empty($_POST['username'])
        or empty($_POST['pass']))
    {
        die('يرجى تعبئة المعلومات');
    }

    $check_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] . "
```



```

'' AND password=''' . $_POST['pass'] . ''");
    $correct = mysql_num_rows($check_query);

    if ($correct)
    {
        setcookie('mysimplebb_username', $_POST['username']);
        setcookie('mysimplebb_password', $_POST['pass']);

        echo 'تم تسجيل الدخول';
    }
    else
    {
        echo 'خطأ في تسجيل الدخول';
    }
}

?>

```

كما تلاحظ اننا نخزن كلمة المرور في قواعد البيانات كما هي، من الناحية الامنيه يعتبر ذلك خطراً، و بالتالي من الافضل تشفير كلمات المرور عند تخزينها في قواعد البيانات، سنستخدم خوارزمية التشفير MD5، لنحرق معلومات العضو الذي اضفناه منذ قليل إلى الجدول member، من خلال phpMyAdmin يمكننا تحرير معلومات العضو و اختيار MD5 من قائمة Function بجانب الحقل password :

```
UPDATE `member` SET `password` = MD5( '123' ) WHERE `member`.`id` =1 LIMIT 1 ;
```

الآن لابد من إجراء تعديل على الملف login.php حتى يتوافق مع الطريقة الجديده في تخزين كلمة المرور، ما نحتاجه هو تشفير كلمة المرور التي يدخلها العضو في صفحة تسجيل الدخول باستخدام خوارزمية التشفير MD5 ثم تمريرها إلى قواعد البيانات للتحقق من صحّة المعلومات، هذا كل شيء (-) و بالتالي نغير سطر الاستعلام :

```

$check_query = mysql_query("SELECT * FROM member WHERE username=''' . $_POST['username'] . ''
AND password=''' . $_POST['pass'] . ''");

```

إلى :

```

$check_query = mysql_query("SELECT * FROM member WHERE username=''' . $_POST['username'] . ''
AND password=''' . md5($_POST['pass']) . ''");

```

لاحظ مررنا المتغير ['POST['pass\_\$ إلى الداهه md5. كذلك بالنسبه للسطر الذي يُخزن كلمة المرور في الكوكيز وهو :

```
setcookie('mysimplebb_password', $_POST['pass']);
```

نقوم بتمرير المتغير ['POST['pass\_\$ إلى الدالة md5 حتى تُخزّن كلمة المرور بشكل مشفّر :

```
setcookie('mysimplebb_password',md5($_POST['pass']));
```

إذا جربنا الصفحة الآن سنجدها تعمل كما كانت بلا فرق :- الفرق الوحيد اننا زدنا من الامان.

## صفحة التسجيل

سنشرع في كتابة صفحة التسجيل الآن، و لا يخفى عليكم انه من خلالها يمكننا إضافة المزيد من الاعضاء إلى قاعدة البيانات، تطلب هذه الصفحة من الزائر الذي يريد التسجيل المعلومات الاساسيات و هما اسم المستخدم و كلمة المرور، ثم تقوم بإضافة العضو إلى قواعد البيانات.

نبدأ بالملف register.php، كالعاده نبدأ بمناداة الملف common.php :

```
<?php
```

```
include('common.php');
```

سيتم عرض النموذج لمن اراد التسجيل، سيطلب هذا النموذج كما اسلفنا اسم المستخدم و كلمة المرور :

```
if (empty($_GET['register']))
{
    echo '<form method="post" action="register.php?register=1">';
    echo 'اسم المستخدم : <input type="text" name="username"><br />';
    echo 'كلمة المرور : <input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل">';
    echo '</form>';
}
```

ننتقل الآن لكتابة الشيفره الاساسيه و التي تتعامل مع قواعد البيانات، اولاً لا بد من التحقق اذا كان الزائر قد كتب المعلومات المطلوبه ام لا :

```
else
{
    ///////////

    if (empty($_POST['username'])
        or empty($_POST['pass']))
```

```
{
    die('يرجى تعبئة كافة المعلومات');
}
```

```
//////////
```

هناك نقطة هامة لا بد من الالتفات لها، عندما يطلب الزائر التسجيل و يضع اسم المستخدم المطلوب لا بد من فحص اسم المستخدم هذا، لأنه قد يكون موجوداً في قواعد البيانات، اي انه محجوز من خلال شخص آخر قام بتسجيله قبل الزائر الذي يحاول التسجيل حالياً، نقوم بذلك كالتالي :

```
$exist_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] .
"");
$exist = mysql_num_rows($exist_query);

if ($exist > 0)
{
    die('اسم المستخدم المطلوب مُسجّل مسبقاً');
}
```

```
//////////
```

انتهينا الآن من الاختبارات و في حال نجحت جميعها هذا يعني انه لا توجد مشاكل و بالتالي نضيف المستخدم إلى قواعد البيانات، لا ننسى نقطة هامة ان كلمات المرور عندما نخزنها فإننا نخزنها كما تعلم مشفرة باستخدام الخوارزمية MD5، و بالتالي لا بد من تمرير قيمة ['POST['pass\_ \$ إلى الدالة md5 و الناتج هو الذي سيُخزّن في قواعد البيانات.

```
$password = md5($_POST['pass']);
```

```
$insert = mysql_query("INSERT INTO member(id,username,password) VALUES('NULL','" .
$_POST['username'] . "','" . $password . "')");
```

```
if ($insert)
{
    echo 'تم تسجيل العضو ' . $_POST['username'];
}
```

```
//////////
```

```
}
```

```
?>
```

```

<?php

include('common.php');

if (empty($_GET['register']))
{
    echo '<form method="post" action="register.php?register=1">';
    echo 'اسم المستخدم : <input type="text" name="username"><br />';
    echo 'كلمة المرور : <input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل">';
    echo '</form>';
}
else
{
    //////////

    if (empty($_POST['username'])
        or empty($_POST['pass']))
    {
        die('يرجى تعبئة كافة المعلومات');
    }

    //////////

    $exist_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] .
    """);
    $exist = mysql_num_rows($exist_query);

    if ($exist > 0)
    {
        die('اسم المستخدم المطلوب مُسجّل مسبقاً');
    }

    //////////

    $password = md5($_POST['pass']);

    $insert = mysql_query("INSERT INTO member(id,username,password) VALUES('NULL','" .
    $_POST['username'] . "','" . $password . "')");

    if ($insert)

```

```

{
    echo 'تم تسجيل العضو ' . $_POST['username'];
}

//////////
}

?>

```

## التحقق إذا كان الزائر عضواً

عندما يُسجّل العضو دخوله بمعلومات صحيحة سيتم تخزين معلوماته في الكوكيز، من المفترض ان يكون للعضو مجموعه من الميزات التي يتمتع بها، مثلاً إمكانية كتابة المواضيع و الردود، و بالتالي نحن بحاجة إلى التأكد دائماً : الزائر الذي يتصفح المنتدى حالياً هل هو عضو ام لا، اذا كان عضواً هذا يعني انه مُخزّن في حاسوبه كل من الكوكيز `mysimplebb_username` و `mysimplybb_password` و اللذان سجلناهما عندما نجح المستخدم في تسجيل دخوله، حسناً ليس كل زائر يحتوي حاسوبه على الكوكيز `mysimplebb_username` و `mysimplybb_password` يعني انه عضو فعلي لانه كما تعلمون يمكن التلاعب بقيم الكوكيز، و بالتالي قد يكون الزائر تلاعب بالكوكيز حتى يتمكن من الوصول إلى حساب احد الاعضاء بدون معرفة كلمة المرور، و بالتالي عندما نجد ان العضو قام بتسجيل دخوله فعلاً لا بد من التحقق من معلوماته دائماً و هل هو عضو فعلي يستحق الوصول إلى مميزات الاعضاء ام لا.

نكتب شيفرتنا هذه المره في الملف `common.php` لانها يجب ان تُنفذ في كل الصفحات، نخص اولاً الكوكيز `mysimplebb_username` و `mysimplybb_password` هل هما موجودان، في حال وجودهما نتحقق من المعلومات المُخزّنه داخلهما، في حال صحة هذه المعلومات هذا يعني ان الزائر الحالي عضو فعلي و المعلومات التي زودنا بها صحيحة و بالتالي يمكنه الوصول إلى مميزات الاعضاء :

```

//////////

$is_member = false;

if (!empty($_COOKIE['mysimplebb_username'])
    and !empty($_COOKIE['mysimplebb_password']))
{
    $check_query = mysql_query("SELECT * FROM member WHERE username='" .
$_COOKIE['mysimplebb_username'] . "' AND password='" . $_COOKIE['mysimplebb_password'] . "'");
    $correct = mysql_num_rows($check_query);

    if ($correct)
    {
        $is_member = true;

        echo ' السلام عليكم يا ' . $_COOKIE['mysimplebb_username'] . '<br />';
    }
}

```

بملاحظة المتغير `is_member$` سنجد ان قيمته ستكون `false` عندما يكون الزائر عادياً و لم يسجل دخوله كعضو، و قيمته ستكون `true` عندما يكون الزائر قد سجل دخوله كعضو بمعلومات صحيحة، سنستخدم هذا المتغير في الملفات الاخرى عندما نحتاج إلى معرفة هل الزائر الحالي عضو ام انه زائر عادي، و بالتالي يمكننا منع الزوار العاديين من الولوج إلى صفحات معينه و مميزات معينه.

حالياً لدينا صفتين يجب ان لا يصل لهما العضو، الاولى هي صفحة تسجيل الدخول `login.php`، و الثانيه هي صفحة التسجيل `register.php`، لتنفيذ ذلك سنفحص المتغير `is_member$` في حال كانت قيمته `true` هذا يعني ان الزائر الذي يحاول الوصول إلى الصفحة عضو فعلي و قام بتسجيل دخوله، و بالتالي نمنعه من الوصول :

```
if ($is_member)
{
    die('لا يمكن للاعضاء الوصول لهذه الصفحة');
}
```

نعم، اعلم ان درس اليوم طويل (-) لا بأس لقد انتهينا، نلتقي غداً إن شاء الله (-):

الشفيرات النهائية للملفات `login.php` و `register.php` :

login.php :

```
<?php
```

```
include('common.php');
```

```
if ($is_member)
{
    die('لا يمكن للاعضاء الوصول لهذه الصفحة');
}
```

```
if (empty($_GET['login']))
{
    echo '<form method="post" action="login.php?login=1">';
    echo '<input type="text" name="username">';
    echo '<input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل دخول">';
    echo '</form>';
}
else
{
    if (empty($_POST['username'])
        or empty($_POST['pass']))
```

```

{
    die('يرجى تعبئة المعلومات');
}

$check_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] .
" AND password='" . md5($_POST['pass']) . "'");
$correct = mysql_num_rows($check_query);

if ($correct)
{
    setcookie('mysimplebb_username', $_POST['username']);
    setcookie('mysimplebb_password', md5($_POST['pass']));

    echo 'تم تسجيل الدخول';
}
else
{
    echo 'خطأ في تسجيل الدخول';
}
}

?>

```

register.php :

```

<?php

include('common.php');

if ($is_member)
{
    die('لا يمكن للاعضاء الوصول لهذه الصفحة');
}

if (empty($_GET['register']))
{
    echo '<form method="post" action="register.php?register=1">';
    echo 'اسم المستخدم : <input type="text" name="username"><br />';
    echo 'كلمة المرور : <input type="password" name="pass">';
    echo '<input type="submit" name="send" value="تسجيل">';
    echo '</form>';
}
else

```

```

{
    //////////

    if (empty($_POST['username'])
        or empty($_POST['pass']))
    {
        die('يرجى تعبئة كافة المعلومات');
    }

    //////////

    $exist_query = mysql_query("SELECT * FROM member WHERE username='" . $_POST['username'] .
    """);
    $exist = mysql_num_rows($exist_query);

    if ($exist > 0)
    {
        die('اسم المستخدم المطلوب مُسجلاً مسبقاً');
    }

    //////////

    $password = md5($_POST['pass']);

    $insert = mysql_query("INSERT INTO member(id,username,password) VALUES('NULL','" .
    $_POST['username'] . "',''" . $password . "')");

    if ($insert)
    {
        echo 'تم تسجيل العضو ' . $_POST['username'];
    }

    //////////
}

?>

```



# اليوم السادس : المواضيع و الردود

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

سنتحدث اليوم عن الميزه الرئيسيه في برامج المنتديات و هي المواضيع و الردود، سنضيف جدولين احدهما للمواضيع و الآخر للردود، جدول المواضيع يحتوي على 7 حقول و هي :

id : الرقم المميز

title : عنوان الموضوع

subject\_text : نص الموضوع

writer : كاتب الموضوع

section : القسم الذي وُضع فيه هذا الموضوع

time : الوقت الذي كُتب فيه الموضوع

order\_time : وقت آخر رد على الموضوع

امر SQL لإنشاء الجدول :

```
CREATE TABLE subjects (
`id` INT( 9 ) NOT NULL AUTO_INCREMENT ,
`title` VARCHAR( 150 ) NOT NULL ,
`subject_text` TEXT NOT NULL ,
`writer` VARCHAR( 100 ) NOT NULL ,
`section` INT( 9 ) NOT NULL ,
`time` VARCHAR( 100 ) NOT NULL ,
`order_time` VARCHAR( 100 ) NOT NULL ,
PRIMARY KEY ( `id` )
) ENGINE = MYISAM
```

حسناً، بعض الحقول بحاجه إلى مزيد من التوضيح، الحقل section سيحتوي على الرقم المميز للقسم الذي كُتب به، لنفرض ان لدينا قسم باسم "منتدى PHP" و رقمه المميز 5 و كتبنا موضوعاً بعنوان "مقدمه حول البرمجه الآمنه" في هذا القسم، هذا يعني ان قيمة الحقل section لهذا الموضوع هي 5.

الحقل time سيحتوي على تاريخ و وقت كتابة الموضوع، سنخزن الوقت بصيغة Unix time، يمكننا الحصول على الوقت

الحالي بهذه الصيغة من خلال الدالة time(). و يمكننا فيما بعد تحويل صيغة Unix time إلى صيغته قابله للقراءة.

بالنسبه للحقل order\_time فإنه شبيه بالحقل time من ناحية صيغة التخزين، اي اننا سنستخدم Unix time مع هذا الحقل كذلك، سنرى فائدة هذا الحقل عندما نريد عرض قائمة المواضيع وفقاً لأحدث الردود، تعودنا دائماً في المنتديات ان المواضيع التي تحتوي على احدث الردود تكون في الاعلى، و التي لا تحتوي على الردود او الردود فيها قديمه فإنها تكون بالاسفل حسب التاريخ، هذا الحقل سيساعدنا على تحقيق هذه الميزة.

نعرّج على جدول الردود الذي سيحتوي على 6 حقول و هي :

id : الرقم المميز

title : عنوان الرد

reply\_text : نص الرد

writer : كاتب الرد

subject : الرقم المميز للموضوع الذي كُتب الرد فيه

time : الوقت الذي كُتب فيه الرد

امر SQL لإنشاء الجدول :

```
CREATE TABLE reply (
`id` INT( 9 ) NOT NULL AUTO_INCREMENT ,
`title` VARCHAR( 150 ) NOT NULL ,
`reply_text` TEXT NOT NULL ,
`writer` VARCHAR( 150 ) NOT NULL ,
`subject` INT( 9 ) NOT NULL ,
`time` VARCHAR( 100 ) NOT NULL ,
PRIMARY KEY ( `id` )
) ENGINE = MYISAM
```

الحقل time في هذا الجدول شبيه بجدول المواضيع.

## صفحة إضافة الموضوع

مثل برامج المنتديات الاخرى، عندما نريد الوصول إلى صفحة إضافة الموضوع لا بد اولاً من الدخول في صفحة القسم الذي نريد كتابة الموضوع فيه، سنجد رابطاً يؤدي إلى صفحة إضافة الموضوع في هذا القسم، هذا يعني اننا الآن سننشئ الملف forum.php و هو كما تعلمون الملف الذي سيصل إليه الزائر عندما يختار احد الاقسام من الصفحة الرئيسية، و بالتالي سيعرض هذا الملف قائمه بمواضيع القسم كما انه سيعرض الرابط الذي يؤدي إلى صفحة إضافة الموضوع، سنكتب شيفرة

عرض قائمة المواضيع فيما بعد كل ما نود فعله الآن هو تجهيز الملف ووضع الاساسيات فيه، كالعاده نبدأ بتضمين الملف  
: common.php

```
<?php
```

```
include('common.php');
```

كما تعلمون اننا سنُرسل الرقم المميز للقسم المطلوب من خلال المصفوفة GET\_ إلى الملف forum.php حتى نتمكن من  
إحضار معلومات القسم المطلوب، كما نلاحظ في الملف index.php السطر 16 :

```
echo '<ul><li><a href="forum.php?id=' . $section['id'] . '"> . $section['title'] .  
'</a></li></ul>';
```

اننا نمرر الرقم المميز للقسم إلى الرابط، و بالتالي يجب ان لا يكون المتغير id فارغاً بل يحتوي على رقم صحيح يحدد لنا  
القسم المطلوب، و بالتالي نضحص المتغير لنتأكد منه :

```
if (empty($_GET['id']))  
{  
    die('المسار المُتبع غير صحيح');  
}
```

نعرض الآن الوصلة التي تحدثنا عنها، الجدير بالذكر هنا ان هذه الوصلة يجب ان لا تظهر لغير الاعضاء، و بالتالي سنستخدم  
المتغير \$is\_member الذي عرفناه مسبقاً و الذي يحدد اذا كان الزائر عضواً ام لا :

```
if ($is_member)  
{  
    echo '<a href="new_topic.php?id=' . $_GET['id'] . '">أضف موضوعاً</a><br />';  
}  
  
>
```

إنتهينا الآن من تجهيز الملف forum.php، سنبدأ بكتابة الملف new\_topic.php و كما هو واضح يقوم هذا الملف  
بإضافة المواضيع الجديده إلى القسم الذي رقمه المميز موجود في ['GET['id\_\$، نبدأ بتضمين الملف common.php :

```
<?php
```

```
include('common.php');
```

```
if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}
```

نتحقق هل الزائر الحالي عضو فعلاً أم لا، في حال لم يكن عضواً لآبداً من منعه من الولوج إلى الصفحة :

```
if (!$is_member)
{
    die('لا يمكن لغير الاعضاء الوصول إلى هذه الصفحة');
}
```

نبدأ بعرض النموذج الذي سيستخدمه العضو لكتابة عنوان و محتوى موضوعه :

```
if (empty($_GET['start']))
{
    echo '<form method="post" action="new_topic.php?start=1&id=' . $_GET['id'] . '">';
    echo 'عنوان الموضوع : <input type="text" name="title"><br />';
    echo 'محتوى الموضوع : <textarea name="text"></textarea><br />';
    echo '<input type="submit" name="send" value="أضف">';
    echo '</form>';
}
```

نبدأ الآن بالشفيره الفعليه و التي تضيف الموضوع إلى قاعدة البيانات، لآبداً من التحقق هل قام العضو فعلاً بكتابة عنوان الموضوع و محتواه :

```
else
{
    if (empty($_POST['title'])
        or empty($_POST['text']))
    {
        die('يُرجى كتابة المعلومات المطلوبه');
    }
}
```

كما ذكرنا عند إنشاء جدول المواضيع اننا سنستخدم صيغة Unix time لتخزين الوقت، يمكننا الحصول على الوقت الحالي باستخدام الداله time كما أسلفنا :

```
$unixtime = time();
```

نكتب أخيراً الاستعلام الذي يضيف الموضوع إلى قاعدة البيانات، هذا كل شيء :

```
$insert = mysql_query("INSERT INTO
subjects(id,title,subject_text,writer,section,time,order_time) VALUES('NULL','" .
$_POST['title'] . "','" . $_POST['text'] . "','" . $_COOKIE['mysimplebb_username'] . "','" .
$_GET['id'] . "','" . $unixtime . "','" . $unixtime . "')");
```

```
if ($insert)
{
    echo 'تم كتابة الموضوع';
}
```

```
}
```

```
?>
```

الشيء الكامله :

```
<?php
```

```
include('common.php');
```

```
if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}
```

```
if (!$is_member)
{
    die('لا يمكن لغير الاعضاء الوصول إلى هذه الصفحة');
}
```

```
if (empty($_GET['start']))
{
    echo '<form method="post" action="new_topic.php?start=1&id=' . $_GET['id'] . "'>';
    echo 'عنوان الموضوع : <input type="text" name="title"><br />';
    echo 'محتوى الموضوع : <textarea name="text"></textarea><br />';
    echo '<input type="submit" name="send" value="أضف">';
    echo '</form>';
}
```

```
else
```

```
{
    if (empty($_POST['title']))
```

```

        or empty($_POST['text']))
    {
        die('يُرجى كتابة المعلومات المطلوبة');
    }

    $unixtime = time();

    $insert = mysql_query("INSERT INTO
subjects(id,title,subject_text,writer,section,time,order_time) VALUES('NULL','" .
$_POST['title'] . "','" . $_POST['text'] . "','" . $_COOKIE['mysimplebb_username'] . "','" .
$_GET['id'] . "','" . $unixtime . "','" . $unixtime . "')");

    if ($insert)
    {
        echo 'تم كتابة الموضوع';
    }
}

?>

```

بعدما انشأنا صفحة إضافة المواضيع، لنجرب إضافة بعض المواضيع في قسم معين.

## عرض قائمة المواضيع

نعود للملف forum.php لنضيف له قطعة جديدة من الشيفره وظيفتها عرض قائمة المواضيع الخاصه بهذا القسم، سنستخدم الحقل section الموجود في جدول subjects لناخذ قائمة المواضيع التي تكون قيمة حقلها تساوي ['GET[id\_\$ و سناخذ البيانات بترتيبها وفقاً للحقل order\_time للسبب الذي ذكرناه، و بالتالي سيكون استعلامنا كالتالي :

```
SELECT * FROM subjects WHERE section='" . $_GET['id'] . "' ORDER BY order_time DESC
```

نكتب الآن شيفرة الـ PHP التي تحقق المطلوب :

```

echo '<br /><strong>قائمة المواضيع</strong><br />';
$get_subjects_query = mysql_query("SELECT * FROM subjects WHERE section='" . $_GET['id'] . "'
ORDER BY order_time DESC");
while ($row = mysql_fetch_array($get_subjects_query))
{
    نطبع قائمة المواضيع :

    echo '<a href="topic.php?id=' . $row['id'] . '>' . $row['title'] . '</a><br />';
}

```

هذا كل شيء (-):

الشيء الكامل للملف forum.php بعد التعديل :

```
<?php

include('common.php');

if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}

//////////

if ($is_member)
{
    echo '<a href="new_topic.php?id=' . $_GET['id'] . '">أضف موضوعاً</a><br />';
}

//////////

echo '<br /><strong>قائمة المواضيع</strong><br />';
$get_subjects_query = mysql_query("SELECT * FROM subjects WHERE section='" . $_GET['id'] . "'
ORDER BY order_time DESC");
while ($row = mysql_fetch_array($get_subjects_query))
{
    echo '<a href="topic.php?id=' . $row['id'] . '">' . $row['title'] . '</a><br />';
}

//////////

?>
```

### صفحة عرض الموضوع

نبدأ بالملف topic.php و الذي يعرض المواضيع و الردود التي تتبعه، نبدأ بالتضمين و التحقق من المتغير ['GET']['id\_'] والذي يحتوي على الرقم المميز للموضوع :

```
<?php

include('common.php');

if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}
```

نأخذ معلومات الموضوع من قواعد البيانات :

```
$subject_query = mysql_query("SELECT * FROM subjects WHERE id='" . $_GET['id'] . "'");
$subject = mysql_fetch_array($subject_query);
```

في حال كان الموضوع غير موجود في قواعد البيانات نخبر الزائر بذلك و نوقف تنفيذ الصفحة :

```
if ($subject === false)
{
    die('الموضوع غير موجود');
}
```

اخيراً نطبع معلومات الموضوع :

```
echo '<br />عنوان الموضوع : <strong>' . $subject['title'] . '</strong><hr />';
echo $subject['subject_text'] . '<hr />';
```

```
$write_date = date('d-m-Y g:i:s A', $subject['time']);
```

```
echo 'وقت كتابة الموضوع : ' . $write_date;
```

```
?>
```

الشيفره الكامله :

```
<?php

include('common.php');
```



```

if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}

$subject_query = mysql_query("SELECT * FROM subjects WHERE id='" . $_GET['id'] . "'");
$subject = mysql_fetch_array($subject_query);

if ($subject === false)
{
    die('الموضوع غير موجود');
}

echo '<br />عنوان الموضوع : <strong>' . $subject['title'] . '</strong><hr />';
echo $subject['subject_text'] . '<hr />';

$write_date = date('d-m-Y g:i:s A', $subject['time']);

echo 'وقت كتابة الموضوع : ' . $write_date;

?>

```

### صفحة إضافة الرد

الملف البرمجي المسؤول عن إضافة الردود new\_reply.php يشبه بشكل كبير ملف إضافة المواضيع new\_topic.php مع بعض الاختلافات البسيطة، سنبدأ أولاً بإضافة وصله في صفحة عرض الموضوع، تظهر هذه الوصلة للأعضاء فقط و التي تربط إلى صفحة إضافة الرد، فمن اراد الرد على الموضوع الذي يعرضه حالياً يجب عليه الضغط على هذه الوصلة لتوجهه إلى صفحة إضافة الرد، في نهاية الملف topic.php نضيف الوصلة :

```

if ($is_member)
{
    echo '<br /><a href="new_reply.php?id=' . $subject['id'] . '">أضف رداً</a>';
}

```

نبدأ الآن بإنشاء الملف new\_reply.php، كعادته نبدأ بمناداة الملف common.php :

```
<?php
```

```
include('common.php');
```

كما تلاحظ يتم تمرير رقم الموضوع الذي نريد الرد عليه إلى 'id\_' [GET] للصفحة new\_reply.php، و بالتالي لا بد من فحص حالة هذا المتغير كما فعلنا سابقاً :

```
if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}
```

نتأكد هل الزائر الحالي لصفحة إضافة الردود عضو و إلا نمنعه من الوصول إلى الصفحة :

```
if (!$is_member)
{
    die('لا يمكن لغير الاعضاء الوصول إلى هذه الصفحة');
}
```

نعرض الآن النموذج الذي من خلاله يكتب العضو الرد و يضيفه إلى قواعد البيانات :

```
if (empty($_GET['start']))
{
    echo 'الرد على الموضوع رقم ' . $_GET['id'];
    echo '<form method="post" action="new_reply.php?start=1&id=' . $_GET['id'] . "'>';
    echo 'عنوان الرد : <input type="text" name="title"><br />';
    echo 'محتوى الرد : <textarea name="text"></textarea><br />';
    echo '<input type="submit" name="send" value="أضف">';
    echo '</form>';
}
```

نكمل بقية الشيفره و المسؤوله عن إدراج المعلومات في قواعد البيانات، نتحقق أولاً هل قام العضو بكتابة محتوى الرد :

```
else
{
    if (empty($_POST['text']))
    {
        die('يُرجى كتابة محتوى الرد');
    }
}
```

نأخذ الوقت بصيغة Unix و نُدرج الرد في قواعد البيانات :

```
$unixtime = time();
```

```
$insert = mysql_query("INSERT INTO reply(id,title,reply_text,writer,subject,time)
VALUES('NULL','" . $_POST['title'] . "','" . $_POST['text'] . "','" .
$_COOKIE['mysimplebb_username'] . "','" . $_GET['id'] . "','" . $unixtime . "'");
```

في حال نجاح إدراج الرد في قواعد البيانات سنقوم بعملية هامه جداً، هل تتذكرون الحقل `order_time` الذي اضفناه لجدول المواضيع `subjects` قلنا ان هذا الحقل مسؤول عن ترتيب الموضوع ضمن قائمة المواضيع، كما نعلم ان في اغلب برامج المنتديات تُرتب قائمة المواضيع حسب تاريخ آخر رد، مثلاً الموضوع الذي كُتب فيه رداً بتاريخ اليوم سيظهر في اعلى الموضوع الذي كُتب فيه رداً يوم امس في قائمة المواضيع، و كما قلنا اننا سننفذ هذه الميزة من خلال الحقل `order_time` و الذي سيحتوي على تاريخ آخر رد كُتب في الموضوع و بالتالي سنتمكن من معرفة احدث موضوع تم الرد عليه و ترتيب قائمة المواضيع وفقاً لذلك كما فعلنا في الملف `forum.php` تحديداً في الاستعلام :

```
$get_subjects_query = mysql_query("SELECT * FROM subjects WHERE section='" . $_GET['id'] . "'
ORDER BY order_time DESC");
```

لاحظ :

```
ORDER BY order_time DESC
```

نعود للملف `new_reply.php` المطلوب الآن هو بعد نجاح إضافة الرد في قواعد البيانات تحديث الحقل `order_time` الخاص بالموضوع الذي كتبنا الرد عليه بقيمة جديدته وهي التاريخ الحالي :

```
if ($insert)
{
    $update = mysql_query("UPDATE subjects SET order_time='" . $unixtime . "' WHERE
id='" . $_GET['id'] . "'");

    if ($update)
    {
        echo 'تم كتابة الرد';
    }
}
}
?>
```

عندما نجرّب الآن إضافة رد على احد المواضيع سنلاحظ ان الموضوع سيصعد إلى اعلى قائمة المواضيع لانه صاحب احدث رد.

الشيء كامله للملف new\_reply.php :

```
<?php

include('common.php');

if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}

if (!$is_member)
{
    die('لا يمكن لغير الاعضاء الوصول إلى هذه الصفحه');
}

if (empty($_GET['start']))
{
    echo 'الرد على الموضوع رقم : ' . $_GET['id'];
    echo '<form method="post" action="new_reply.php?start=1&id=' . $_GET['id'] . '">';
    echo 'عنوان الرد : <input type="text" name="title"><br />';
    echo 'محتوى الرد : <textarea name="text"></textarea><br />';
    echo '<input type="submit" name="send" value="أضف">';
    echo '</form>';
}
else
{
    if (empty($_POST['text']))
    {
        die('يُرجى كتابة محتوى الرد');
    }

    $unixtime = time();

    $insert = mysql_query("INSERT INTO reply(id,title,reply_text,writer,subject,time)
VALUES('NULL','" . $_POST['title'] . "','" . $_POST['text'] . "','" .
$_COOKIE['mysimplebb_username'] . "','" . $_GET['id'] . "','" . $unixtime . '"");

    if ($insert)
    {
```

```

        $update = mysql_query("UPDATE subjects SET order_time='" . $unixtime . "' WHERE
id='" . $_GET['id'] . "'");

        if ($update)
        {
            echo 'تم كتابة الرد';
        }
    }
}

?>

```

### تتمة صفحة عرض الموضوع

بقي لدينا شيء واحد و هو عرض قائمة الردود على موضوع معين، نعود للملف topic.php لكتابة الشيفره التي تعرض الردود، كما نعلم ان في جدول الردود حقل بإسم subject يحتوي على الرقم المميز للموضوع الذي كُتب الرد من أجله، و بالتالي يمكننا الحصول بسهولة على قائمة الردود الخاصه بموضوع معين، بالطبع نريد عرض الردود من الاحداث إلى الاقدم :

```

echo '<div align="center"><strong>قائمة الردود</strong></div>';

```

```

$reply_query = mysql_query("SELECT * FROM reply WHERE subject='" . $_GET['id'] . "' ORDER BY
id DESC");
while ($reply = mysql_fetch_array($reply_query))
{

```

تعرض المعلومات الخاصه بالرد :

```

    echo '<br />عنوان الرد : <strong>' . $reply['title'] . '</strong><hr />';
    echo $reply['reply_text'] . '<hr />';

    $reply_date = date('d-m-Y g:i:s A',$reply['time']);

    echo 'وقت كتابة الرد : ' . $reply_date;

    echo '<br /><br /><br /><hr />';
}

?>

```

(هذا كل شيء :-)

الشيء كله للملف topic.php :

```
<?php

include('common.php');

if (empty($_GET['id']))
{
    die('المسار المُتبع غير صحيح');
}

$subject_query = mysql_query("SELECT * FROM subjects WHERE id='" . $_GET['id'] . "'");
$subject = mysql_fetch_array($subject_query);

if ($subject === false)
{
    die('الموضوع غير موجود');
}

echo '<br />عنوان الموضوع : <strong>' . $subject['title'] . '</strong><hr />';
echo $subject['subject_text'] . '<hr />';

$write_date = date('d-m-Y g:i:s A', $subject['time']);

echo 'وقت كتابة الموضوع : ' . $write_date;

if ($is_member)
{
    echo '<br /><a href="new_reply.php?id=' . $subject['id'] . '">أضف رداً</a>';
}

echo '<div align="center"><strong>قائمة الردود</strong></div>';

$reply_query = mysql_query("SELECT * FROM reply WHERE subject='" . $_GET['id'] . "' ORDER BY id DESC");
while ($reply = mysql_fetch_array($reply_query))
{
    echo '<br />عنوان الرد : <strong>' . $reply['title'] . '</strong><hr />';
    echo $reply['reply_text'] . '<hr />';

    $reply_date = date('d-m-Y g:i:s A', $reply['time']);

    echo 'وقت كتابة الرد : ' . $reply_date;
```

```
echo '<br /><br /><br /><hr />';  
}  
?>
```

يا إلهي! هل انتهينا؟ نعم لقد انتهينا اليوم، نلتقي غداً إن شاء الله :-)

# اليوم السابع : الامان

بسم الله الرحمن الرحيم

الصلاة و السلام على أشرف المرسلين سيدنا محمد و على آله الطيبين الطاهرين.

انتهينا الآن من كتابة برنامج منتديات يحتوي على الميزات الاساسيه لبرامج المنتديات، تبقى أن نتحدث عن الأمان، سنتحدث عن اهم ثغرتين SQL Injection و XSS (-:

## SQL Injection

او حُقن SQL، احد اشهر انواع الثغرات التي تظهر في البرامج التي تعتمد على قواعد البيانات، لنفرض الشيفره التاليه :

```
$username = 'test';
```

```
$query = mysql_query("SELECT * FROM member WHERE username='" . $username . "'") or  
die(mysql_error());
```

كما هو واضح تقوم هذه الشيفره بالإستعلام عن المدخلات التي قيمة الحقل username فيها تساوي test، كما تلاحظ قمنا بتخزين اسم المستخدم المطلوب في المتغير \$username، مع الاسطر السابقه فإن الاستعلام الذي يُرسل إلى MySQL هو :

```
SELECT * FROM member WHERE username='test'
```

صحيح؟ (-)، حسناً ماذا لو غيرنا قيمة \$username و جعلناها كالتالي :

```
$username = "test' AND password='" . md5('123');
```

```
$query = mysql_query("SELECT * FROM member WHERE username='" . $username . "'") or  
die(mysql_error());
```

دقق قليلاً، ستجد ان الاستعلام الذي سيُرسل إلى MySQL هو :

```
SELECT * FROM member WHERE username='test' AND password='md5(123)'
```



لاحظ الجزء الزائد وهو :

```
AND password='md5(123)'
```

هذا الجزء غير موجود اصلاً ضمن الاستعلام الأصلي الذي مررناه إلى الدالة `mysql_query`، بل قمنا بإضافته إلى المتغير `$username` و الذي بدوره يمر على الاستعلام الأصلي، هذا يعني أننا حقننا هذا الجزء في الاستعلام الأصلي.

حسناً لنفترض الأسوأ، المتغير `$username` الآن متغير محلي لا يمكن لأحد تغيير قيمته إلا عن طريق تعديل ملفات PHP، حسناً ماذا لو كان المتغير `username` متغير عام يمكن للزائر العادي تغيير قيمته؟ أعني ماذا لو كان المتغير `username` موجود ضمن المصفوفه `_GET` او `_POST` او حتى `_COOKIE` ؟ لاحظ الشيفره التاليه :

```
$username = $_GET['username'];
```

```
$query = mysql_query("SELECT * FROM member WHERE username='" . $username . "'") or die(mysql_error());
```

هذه الشيفره شبيهه بالسابقه و لكن غيرنا قيمة المتغير `username` لتأخذ القيمه الموجوده في `$_GET['username']`، هذه الشيفره ستقوم بالاستعلام عن المستخدم الذي أرسل الزائر إسمه إلى `$_GET['username']`، مثال :

```
http://maastaar/MySimpleBB/sql_injection_test.php?username=test
```

عند زيارة الصفحه بهذا الشكل سيكون الاستعلام المُمرر إلى MySQL كالتالي :

```
SELECT * FROM member WHERE username='test'
```

حسناً، ماذا لو زار الزائر هذه الصفحه :

```
http://maastaar/MySimpleBB/sql_injection_test.php?username=test' AND password='123
```

ما هو الاستعلام الذي سيُمرر إلى MySQL؟ نعم انه التالي :

```
SELECT * FROM member WHERE username='test' AND password='123'
```

هل لاحظت المشكله؟ تمكن الزائر العادي من التلاعب باستعلامات البرنامج! و اضاف إلى الاستعلام الأصلي جزء خارجياً كته بنفسه، هذه هي حقن SQL، حيث تم حقن جزء جديد إلى استعلام موجود اساساً، في حالتنا المشروحه في الاعلى لا توجد خطوره تقريباً، و لكن هناك مجموعه من الحالات الخطيره التي اذا أُحسن إستغلالها سيتمكن المخترق من الحصول على

معلومات هامه مثل كلمة المرور او حتى التعديل على المعلومات في قواعد البيانات، حتى و ان كانت ثغرات SQL Injection الموجوده في برامجنا ليست بهذه الخطوره لا بد ان نتخلص منها و نجعل برامجنا خاليه منها، فمن غير المرغوب ان يتمكن اي شخص من التلاعب في الاستعلامات الداخليه للبرنامج :-)

لنتسائل الآن، اين بدأت المشكله بالضبط؟ بقليل من التركيز سنجد ان المشكله بدأت مع علامه الاقتباس الفرديه (')، نعلم جميعنا مدى اهمية هذه العلامه في لغة SQL، نستخدم هذه العلامه غالباً عندما نريد تمرير معلومات لإضافتها إلى قواعد البيانات او لتحديثها او حتى تمرير المعلومات من اجل استخدامها مع WHERE مثلاً كما في حالتنا، ثغرات SQL Injection تعتمد على علامه الاقتباس الفرديه بشكل اساسي، لاحظ الصفحه الاخيره التي اشرنا لها :

```
http://maastaar/MySimpleBB/sql_injection_test.php?username=test' AND password='123
```

لاحظ الجزء الذي حقهناه في الاستعلام الاصلي :

```
test' AND password='123
```

لاحظ اول علامه اقتباس فرديه بعد test، هنا قمنا بإغلاق علامه الاقتباس التي فتحناها عن هذا الجزء من الاستعلام الاصلي :

```
username='
```

و بعد اغلاقنا لعلامه الاقتباس السابقه اكملنا الاستعلام و اضفنا الجزء المطلوب حقنه في الاستعلام الاصلي، هذا يعني ببساطه ان الحل يكمن في تهريب اي علامه اقتباس فرديه يضيفها الزائر إلى القيم التي تُمرر إلى قاعدة البيانات، الداله addslashes تقوم بتهريب علامات الاقتباس عن طريق إضافة \ إليها، و بالتالي حتى نتخلص من ثغرة SQL Injection الموجوده في هذه الشيفره :

```
$username = $_GET['username'];
```

```
$query = mysql_query("SELECT * FROM member WHERE username='" . $username . "'") or die(mysql_error());
```

سنقوم بتمرير المتغير \$username إلى الداله addslashes كالتالي :

```
$username = $_GET['username'];
```

```
$query = mysql_query("SELECT * FROM member WHERE username='" . addslashes($username) . "'") or die(mysql_error());
```

في هذه الحالة عندما نزرور الصفحة التاليه :

```
http://maastaar/MySimpleBB/sql_injection_test.php?username=test' AND password='123
```

فإن الاستعلام الذي سيُمرر إلى MySQL كالتالي :

```
SELECT * FROM member WHERE username='test\' AND password='\123'
```

لاحظ الـ \ قبل علامتي الاقتباس، عندما يرى MySQL الـ \ لن يحسب علامة الاقتباس التي تقع وراءها على إنها جزء من جملة SQL بس سيحسبها على انها قيمه عاديه، و بالتالي وفقاً لهذا الاستعلام فإننا نبحث عن المستخدم الذي اسمه يساوي test' AND password='123، إذاً الحل بتهريب علامات الاقتباس.

أما المقياس في تحديد هل هذا الاستعلام يحتوي على ثغره SQL Injection ام لا فإنه كالتالي، اي استعلام يستقبل معلومات من الزائر ولا يتم معالجة هذه المعلومات عن طريق تهريب علامات الاقتباس فيها فإن هذا الاستعلام يحتوي على ثغره SQL Injection.

الآن، من الاسطر التي كتبناها سابقاً في برنامجنا، هل يمكنكم إستخراج استعلامات تحتوي على ثغرات من نوع SQL Injection؟ نعم هناك الكثير منها (-) سأعطيكم مثلاً واحداً مع التعديل عليه و سأترك لكم بقية الاستعلامات لتكتشفوها بأنفسكم و تخبروني عنها (-):

في الملف common.php السطر :

```
$check_query = mysql_query("SELECT * FROM member WHERE username='".  
$_COOKIE['mysimplebb_username'] . "' AND password='". $_COOKIE['mysimplebb_password'] . "'");
```

يمكن للزائر العادي ان يتلاعب بقيم الكوكيز mysimplebb\_username و mysimplebb\_password المٌخزنه في حاسوبه، و يمكنه أن يحقن الاستعلام بجزء جديد، اذاً يحتوي هذا الاستعلام على ثغرة SQL Injection، كيف نتخلص منها؟ ببساطه addslashes (-):

```
$check_query = mysql_query("SELECT * FROM member WHERE username='".  
addslashes($_COOKIE['mysimplebb_username']) . "' AND password='".  
addslashes($_COOKIE['mysimplebb_password']) . "'");
```

## XSS

ننتقل للنوع الثاني من الثغرات وهو XSS او Cross-site Scripting، هذا النوع من الثغرات يشبه SQL Injection من حيث المبدأ، للتوضيح نعرض الشيفره التاليه :

```
$id = 1;
```

```
echo '<a href="topic.php?id=' . $id . '">قراءة الموضوع</a>';
```

واضح ان هذه الشيفره تُظهر لنا وصله في الصفحه بعنوان "قراءة الموضوع" و تنقلنا هذه الوصله إلى الصفحه التي تعرض معلومات الموضوع الذي رقمه المميز = 1، لاحظ اننا خزنا رقم الموضوع في المتغير \$id و مررنا هذا المتغير إلى وسم HTML، حالياً وسم HTML الكامل الذي سيُرسل للمتصفح هو :

```
<a href="topic.php?id=1">قراءة الموضوع</a>
```

حسناً ماذا لو غيرنا قيمة \$id كالتالي :

```
$id = '1"><strong class=""
```

```
echo '<a href="topic.php?id=' . $id . '">قراءة الموضوع</a>';
```

ماذا ستكون النتيجة؟ لاحظ ان النص "قراءة الموضوع" اصبح بالخط العريض لاننا حقناً وسم HTML بجزء من عندنا وهو `<strong class=""` ووظيفة هذا الوسم كما لا يخفى عليكم تضخيم النص، وسم HTML الكامل الذي سيُرسل للمتصفح هو :

```
<a href="topic.php?id=1"><strong class="">قراءة الموضوع</a>
```

مثلما افترضنا الأسوأ مع SQL Injection، سنفترضه هنا كذلك، ماذا لو كان المتغير \$id يمكن للمتصفح العادي تغييره و التلاعب به؟

```
$id = $_GET['id'];
```

```
echo '<a href="topic.php?id=' . $id . '">قراءة الموضوع</a>';
```

اعتقد ان الامور اوضح بالنسبه لك الآن :-)، و لكن قد تتساءل ما المشكله الامنيه الحقيقيه من وراء هذه الثغره؟ قد يتمكن المخرب من التلاعب في طريقة عرض صفحات البرنامج عن طريق حقن وسوم HTML زائده و لكن هل يعتبر هذا ضرر امني فعلاً؟

في الحقيقه الخطر من وراء XSS يكمن في امكانية استخدام لغة جافاسكربت، في المثال السابق قمنا بحقن الوسم `strong` في شيفرة HTML الاصليه و لم يكن الضرر بالغاً حيث غيرنا طريقة عرض الصفحه الاصليه فقط، حسناً ماذا لو حقناً شيفرة جافاسكربت بدلاً من الوسم `strong`؟ و وظيفة هذه الشيفره هي اخذ قيمة `document.cookie` ثم الانتقال إلى صفحه خارجيه ووظيفة هذه الصفحه هي إرسال قيمة `document.cookie` إلى البريد الالكتروني للمخرب؟ ألا تعتبر هذه الثغره

خطيره في هذه الحالة حيث سيتمكن من المخرب من سرقة البيانات الموجوده في الكوكيز :-)

تبدأ هذه الثغرة بالظهور عندما يُرسل الزائر علامة اقتباس مزدوج (") و بالتالي لا بد من تهريبها عندما يُرسل المتصفح معلومات تُستخدم داخل وسم HTML، الدالة htmlspecialchars تقوم بهذه الوظيفة مع تهريب مجموعه اخرى من الاحرف التي قد تُستخدم في ثغرات XSS.

حسناً، لنبحث في الشيفرات التي كتبناها سابقا عن ثغره من نوع XSS، كالعاده سأعرض لكم مثلاً واحداً و اترك لكم البقيه تكتشفونها بأنفسكم، الملف forum.php السطر التالي :

```
echo '<a href="topic.php?id=' . $row['id'] . '"> . $row['title'] . '</a><br />';
```

هذا السطر يعرض عنوان الموضوع على شكل وصله، يمكن للعضو كتابة موضوع و يكون عنوان الموضوع شيفرة جافاسكربت يحقنها ضمن شيفرة HTML هذه، كيف يمكننا منع المخرب من تحقيق هذا الشيء؟ ببساطه نستخدم الداله htmlspecialchars.

```
echo '<a href="topic.php?id=' . htmlspecialchars($row['id']) . '"> .  
htmlspecialchars($row['title']) . '</a><br />';
```

هذا و الحمد لله رب العالمين.

و الله من وراء القصد.