



Ratingsystem i PHP og MySQL

Lær at lave et system til at vise rating/bedømmelse på artikler og nyheder.

Skrevet den **03. Feb 2009** af **virtual1ty** i kategorien **Programmering / PHP** | ★★★★★

I denne artikel vil du lære at lave et rating-system.
Artiklen kræver grundlæggende kendskab til PHP og MySQL.

Hvad er et rating-system?

Et rating-system er en funktion, der gør det muligt for brugere at bedømme ("rate") artikler, nyheder etc. Brugeren vil kunne se den gennemsnitlige rating, der ligner noget i stil med:
"Denne artikel har en rating på: 4,7".

Hvordan fungerer det?

Systemet fungerer ved, at hver gang en bruger bedømmer en artikel, vil hans/hendes bedømmelse blive lagt oveni den nuværende, der er i databasen.
Eksempel:

Den totale rating i databasen er 100.
En bruger bedømmer en artikel med en 5'er ("Du får en stor femtal af mig!").
Dette tal (5) vil så blive lagt oveni den totale rating i databasen:

$$100 + 5 = 105$$

Så nu er den totale rating altså 105.

Samtidig med, at dette sker, bliver et andet felt i databasen inkrementeret med 1. Dette vil sige, at der bliver lagt 1 til tallet, så hvis feltet, som vi kalder "count", er 16, vil dette tal efter bedømmelsen være 17. Dette tal bruger systemet, når det regner den gennemsnitlige rating ud.
Eksempel:

Total rating: 40
Count: 11

$$\text{Gennemsnit: } 40 / 11 = 3,6$$

Har du styr på det nu? Så lad os komme igang med at kode!

Databasen

Lad os starte med at få styr på vores database.

Opret en database som du kalder rating.
Derefter udfører du følgende SQL-forespørgsel i den netop oprettede database:

```
CREATE TABLE `ratings` (  
  `id` int(11) NOT NULL auto_increment,  
  `count` INT (4) NOT NULL,  
  `totalrating` INT (9) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

Feltet "id" angiver id'et på artiklen eller nyheden, som rating'en hører til.

Feltet "count" indeholder antal stemmer, som vi skal bruge for at regne gennemsnittet ud.

Feltet "totalrating" indeholder den totale bedømmelse (alle brugernes bedømmelse lagt sammen).

Nu har vi vores database og tabel klar, lad os komme igang med PHP-koden.

Rate.php

Dette er den fil, hvor brugeren kan bedømme en artikel, og det er også den samme

fil, der tilføjer bedømmelsen til databasen. Du kan naturligvis dele koden op, hvis du ikke vil have det hele i én fil.

Vi springer direkte ud i koden:

```
<?php  
  
// opret forbindelse til databasen (husk at udfylde brugernavn + password)  
$con = mysql_connect("localhost", "", "");  
mysql_select_db("rating", $con);
```

Her opretter vi forbindelse til databasen og vælger databasen "rating", som vi oprettede lige før.

```
// tjek om variabelen 'rating' er sat  
if (isset($_POST['rating'])) {  
  
  $rating = $_POST['rating'];  
  
  if ($rating > 5) { // tjek om rating er mere end 5  
    print "Du kan ikke rate mere end 5!";  
  } else if ($rating < 1) { // tjek om rating er mindre end 1  
    print "Du kan ikke rate mindre end 1!";  
  } else {  
    // læg rating'en oveni den nuværende totale rating, og inkrementér count med 1  
    // artiklens/nyhedens id henter vi fra URL'en med $_GET (kan selvfølgelig ændres)  
    mysql_query("UPDATE `ratings` SET `totalrating` = `totalrating`+$rating, `count` = `count` +1  
WHERE `id` = 'mysql_real_escape_string($_GET[id])'");  
  }  
}
```

Denne kode siger forhåbenligt lidt sig selv med kommentarerne. Det vi gør, er - kort fortalt:

Vi tjekker om variabelen rating er sat. Denne sendes med POST når brugeren bedømmer en artikel.

Derefter tjekker vi om rating er større end 5 eller mindre end 1. Man kan spørge sig selv:

"Hvorfor dog det? Man kan jo bare undlade at give brugeren mulighed for at stemme sådan".

Det er korrekt, men der findes visse programmer, som kan opfange POST-data når de sendes, og give brugeren mulighed for at ændre i dem - hvilket gør at de f.eks. kan rate en artikel med 999 eller lignende.

Når vi så har tjekket om bedømmelsen er "lovlig", opdaterer vi databasen med den nye bedømmelse. Lad os gå videre.

```
// vælg alle rækker fra `ratings`  
// id'et henter vi igen med $_GET  
$q = mysql_query("SELECT * FROM `ratings` WHERE `id` = 'mysql_real_escape_string($_GET[id])'");  
  
$res = mysql_fetch_array($q);  
  
$total = $res['totalrating'];  
$count = $res['count'];
```

Her trækker vi værdien af count og totalrating ud af databasen og gemmer dem i variablerne \$total og \$count.

Disse skal vi bruge, når vi skal regne gennemsnittet ud som vi kommer til nu.

```
// tjek om $total eller $count ikke er lig med 0  
if ($totalrating != "0" AND $count != "0") {
```

Her tjekker vi om \$totalrating eller \$count IKKE er lig med 0. Dette er for at undgå at vi får en grim "Division by zero"-fejl fra PHP, når vi regner gennemsnittet ud.

```
$avg = round($total / $count, 1); // regn gennemsnittet ud ved at dividere $total med $count
```

Her regner vi gennemsnittet ud ved at dividere \$total med \$count.

Resultatet bliver afrundet og gemt i variabelen \$avg.

Og nu kommer vi til en overraskelse! Vi skal da selvfølgelig også have en grafisk rating-viser!

En grafisk rating-viser er en lille søjle, der bliver vist inde i en kasse (typisk en tabel), og jo højere rating'en er, jo længere vil søjlen være. Hvordan laver vi den? Se her:

```
$width = round(100 / 5 * $avg); // her beregner vi, hvor langt billedet i den grafiske rating-viser skal være
```

Vi regner ud hvor mange procent billedet skal fylde af tabellen. Tabellen er 100 pixels bred, og den maksimale

bedømmelse er 5. $100 / 5 = 20$. Derefter ganger vi dette tal (20) med gennemsnittet.

Lad os sige, at gennemsnittet er 4,5. Regnestykket bliver så: $20 * 4,5 = 90$.

Dette vil sige, at søjlen skal være 90% bred i forhold til tabellens bredde. Bredden sætter vi således:

```

```

Vi indsætter simpelthen billedet "bar.gif", og sætter bredden til \$width %.
Så hvis \$width er lig med 50, vil koden komme til at se sådan ud, når siden vises:

```

```

Dette kommer jeg lidt mere ind på lige om lidt.

Lad os komme videre med noget mere kode.

```
} else {  
    $avg = 0; // sæt $avg til 0 hvis både $total og $n er lig med 0  
    $width = 0; // det samme med $width  
}
```

Her sætter vi \$avg og \$width til 0 hvis \$total og \$n er lig med 0.
Nu skal vi skrive den gennemsnitlige bedømmelse og vise den grafiske rating-viser til brugeren.

```
// skriv den gennemsnitlige rating til brugeren  
print "Denne artikel har en gennemsnitlig rating på $avg";  
  
// opret tabellen, hvor søjlen skal vises i  
echo "<table width='100' style='border:solid 1px;' cellpadding='0'>  
    <tr>  
        <td><img src='bar.gif' width='\$width%' height='5'></td>  
    </tr>  
</table>";  
?>
```

Nu mangler vi blot at oprette den form, hvor brugeren kan bedømme en artikel.
Skriv nedenstående kode nedenunder PHP-koden.

```
<html>  
<head>  
<title>Rating-system</title>  
</head>  
  
<body>  
  
<form action="" method="POST">  
  
<select name="rating">  
<option>Vælg en bedømmelse</option>  
<option value="1">1</option>  
<option value="2">2</option>  
<option value="3">3</option>  
<option value="4">4</option>  
<option value="5">5</option>  
</select>
```

```
<input type="submit" value="Rate!">
</form>

</body>
</html>
```

Resultatet bliver en lille dropdown-box, hvor brugeren har mulighed for at vælge imellem 1-5. Nu er systemet faktisk færdigt! Håber det kan bruges til noget.

BEMÆRK:

Husk at ændre dine filer til at tilføje nyheder og artikler, så følgende SQL-forespørgsel udføres når en nyhed eller artikel tilføjes:

```
INSERT INTO `ratings` (count, totalrating) VALUES ('0', '0')
```

Dette er for at oprette en række til bedømmelsen for den nye artikel eller nyhed.

OBS: Dette system tjekker ikke, om brugeren har stemt før eller er logget ind. Denne artikel viser blot princippet i sådan et system.

En dag hvor jeg har tid, skal jeg nok tilføje en funktion, der tjekker om personen har stemt før.

Kommentar af qtax87 (nedlagt brugerprofil) d. 04. May 2008 | 1

Okay rating system.

Som zhx> siger mangler der et tje på om brugeren og/el. ipen har stemt.

Kommentar af jih d. 06. Jun 2008 | 2

samme som det førnævnte, mangler lige et tje om brugeren har stemt, men giver godt viden om hvordan det hele fungerer, og det er nemt nok at smide et ip tje ind, så 4/5 fra mig :-)

Kommentar af inctor d. 09. Sep 2008 | 3

Sagde du ikke total-rating var 100? - Hvorfor angiver du så totalrating som "0" i den sql-forespørgsel i slutningen af artiklen?

Ellers udmærket..

Kommentar af cronaldo d. 24. Dec 2008 | 4

Tjoe - det er såmænd meget fint, men hvorfor ikke lave det sådan, at man som admin kan smække ind i ens adminsistem og søge på nøjagtigt hvilken bruger, der har givet hvilken karakter til hvilken fil ? :) .. Jeg ved godt det er meget simpelt og sådan, men det er altid en fordel at kunne gå ind og søge - og evt. finde - folk, der altid giver min. karakter for at spolere systemet .. :) ..

Kommentar af zhx d. 04. May 2008 | 5

Ellers okay artikel, jeg syntes dog du mangler en del på det, en bruger kan fx. stemme på den samme ting 100 gange hvis han ville det... :=)

Kommentar af sammys d. 11. Jul 2008 | 6

Har smemt meget god.

Var lige hvad jeg ledte efter, et sted hvor jeg kunne få en ide om hvordan jeg skal lave mit eget rating system.

Jeg synes ikke det gør så meget, at den ikke tjekker om brugeren har stemt før, for så kan man jo lærer noget samtidig :)

Kommentar af superfisker d. 07. Jul 2008 | 7

En super artikel for nybegyndere, du beskriver enkelt, men klart hvad du gør.

Dog bliver det kun "God" da du ikke viser hvordan man kun kan stemme engang. Godt nok er det meget enkelt at tilføje selv, men det kan godt være svært at regne ud hvordan man gør når man er nybegynder.

Kommentar af ep_13 d. 19. Jul 2008 | 8