

Seminarium 2

Applikationer för internet, grundkurs ID1354

Maher Jabbar | maherj@kth.se

oct 28, 2016

Innehåll

1 Introduktion	3
2 Litterära kurser	4
3 Metod	5
4 Resultat	6
5 Diskussion	14
6 Kommentarer om kursen	14

1 Introduktion

Uppgiften för detta seminarium är att skapa format för inloggning där man ska ha användare som ska kunna logga in. Varje användare skulle ha ett användarnamn och lösenord. Man skulle även skapa en kommentars box där bara de som är har en inloggning och är inloggade hade tillgång till att kommentera t.ex. receptsidorna. De ska även kunna radera sina kommentarer då de är inloggade.

Mål för detta seminarium:

- Lära sig PHP.
- Lär sig att använda PHP för server-side programmering.

2 Litterära kurser

jag har använt mig av många tutorials om PHP från w3schools.com, föreläsningarna från 4-6 och även lite youtube klipp.

3 Metod

Jag skapade en databas för att jag ska kunna lagra användare och kommentarer från användarna. Sedan började jag göra en PHP-file för login och även en fil för logout. Jag skapade även Php file för databasen där jag inkluderade php i php filen för login med hjälp av require_once just för en koppling mellan databasen och loginsidan så att man kunde komma åt användare man skapat i databasen. Jag skapade också en php fil för att användare som är inloggade skulle kunna skriva kommentarer och sista steget skapade jag en php file deletecomment för att användarna som varit inloggad skulle kunna radera sina kommentarer .

4 Resultat

Home Kalender Köttbullar Pannkakor Logga in

Username
Enter Username

Password
Enter Password

Login

Figur 1. Login formatet som jag skapat

```
36 ▼ <form class="formstyle" method="post">
37 ▼   <div class="container">
38     <label><b>Username</b></label>
39     <input type="text" name="username" placeholder="Enter Username">
40     <label><b>Password</b></label>
41     <input type="password" name="password" placeholder="Enter Password">
42     <button type="submit">Login</button>
43   </div>
44 </form>
45 </body>
46 </html>
```

Figur 2. Hur jag designat formatet

```
151 ▼ .formstyle {
152     border: 3px solid #ca9797;
153 }
154
155 ▼ input[type=text], input[type=password] {
156     width: 100%;
157     padding: 12px 20px;
158     margin: 8px 0;
159     display: inline-block;
160     border: 1px solid #ccc;
161     box-sizing: border-box;
162 }
163
164 ▼ button {
165     background-color: #ca9797;
166     color: white;
167     padding: 14px 20px;
168     margin: 8px 0;
169     border: none;
170     cursor: pointer;
171     width: 100%;
172 }
173
174 ▼ .container {
175     padding: 16px;
176 }
177
```

Figur 3. Hur jag skapade formonet

Tasty Recipes

[Home](#)[Kalender](#)[Köttbullar](#)[Pannkakor](#)[Logga ut](#)

welcome to my own website where I put simple recipes on how to make meatballs and even pancakes.



Figur 4. När man loggar in så skickas man direkt till index.php

```
12  <?php
13  if (isset($_POST['username']) && isset($_POST['password'])) {
14      require_once("database.php");
15
16      $_POST['username'] = mysqli_real_escape_string($conn, $_POST['username']);
17      $_POST['password'] = mysqli_real_escape_string($conn, $_POST['password']);
18      $sql = "SELECT * FROM users WHERE username = '" . $_POST['username'] . "' AND password = '" . $_POST['password'] . "'";
19      $result = $conn->query($sql);
20
21  if($result->num_rows > 0){
22      $_SESSION['logged_in'] = true;
23
24      $user = $result->fetch_assoc();
25      $_SESSION['username'] = $user['username'];
26  } else {
27      echo 'Användarnamnet eller lösenordet är fel!';
28  }
29 }
30
31 if(isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true)
32     header('Location: index.php');
33
34 ?>
```

Figur 5. PHP filen för login

Det isset gör att den returnerar true ifall jag tilldelat t.ex. username till maher och password till 1234. I detta fall har jag gjort det i database.php där jag skapat användare som ska kunna logga in på min sida. Med hjälp av \$sql = SELECT * FROM users WHERE username hämtas alla skapade användare som finns i databasen. Med "if (\$result->num_rows > 0)" menas ifall jag loggar in med en användare som existerar så returneras \$_SESSION['logged_in'] = true; och då loggas man in. Ifall 0 är större än num_rows så visas 'Användarnamnet eller lösenordet är fel!' eftersom det inte användare man försökt logga in med existerar inte. Med if-satsen längst ned menas ifall username och password är true så loggas man in och skickas direkt till index.php som visas i figur 4.

\$_POST är data som skickats från formulär och sparas i en post array medan \$_SESSION är data

som lagras och finns kvar tills användaren stänger webbläsaren eller om man kör ex. session_destroy.

```
10      <?php
11  ▼    if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true) {
12    ?>
13    <li class="menu"><a href="logout.php">Logga ut</a></li>
14    <?php
15  ▼  } else {
16    ?>
17    <li class="menu"><a href="login.php">Logga in</a></li>
18    <?php
19    }
20    ?>
21
22  </ul>
23  </div>
```

Figur 6. Menyraden där login länken ligger

Med if-satsen menas ifall den är true så dyker det upp en "logga ut" länk istället för "logga in" då det visar att man lyckades logga in och ifall det inte går att logga in så händer det inget och "logga in" syns fortfarande i menyraden.

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "tastyrecipes";
6
7  $conn = new mysqli($servername, $username, $password, $dbname);
8
9  ▼ if ($conn->connect_error) {
10    die("Connection failed: " . $conn->connection_error);
11  }
12  ?>
```

Figur 7. database.php kommer åt databasen i phpadmin



Figur 8.

Så ser kommentarfältet ut som jag skapat och det går bara kommentera om man är inloggad

```

1  <?php
2  if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true) {
3      if (isset($_POST['comment']) && $_POST['comment'] != '') {
4          require_once("database.php");
5
6          $_POST['comment'] = mysqli_real_escape_string($conn, $_POST['comment']);
7
8          $sql = "INSERT INTO comments (username, comment, recipe) values ('".$_SESSION['username']."' , '".$_POST['comment']."' ,
9          '".$_recipe."')";
10         if ($conn->query($sql) == false)
11             echo $conn->error;
12     }
13 ?>
```

Figur 9. PHP filen för comment.php

Om if-satsen för isset är true då ska det gå att kommentera och database.php inkluderas då comment.php ska kunna få tillgång till skapade användare som ska kunna kommentera.

Med \$sql = "INSERT INTO comments (username, comment, recipe)... menas att användarens kommentar postas i ett av receptsidorna då man skriver en kommentar i en av sidorna.

```

<?php
$recipe = 'meatballs';
require_once('comment.php');
require_once('deletecomment.php');
require_once('recipeloader.php');
?>
<?php
if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true) {
?>
<textarea rows="4" cols="50" name="comment" form="commentform" placeholder="Skriv din kommentar här..."></textarea>
<form action="" method="post" id="commentform">
<input type="submit" name="" value="Kommentera">
</form>
<?php
}?
?>

```

Figur 10.

Här har jag skapat så att kommentarsfältet bara syns för den som är inloggad

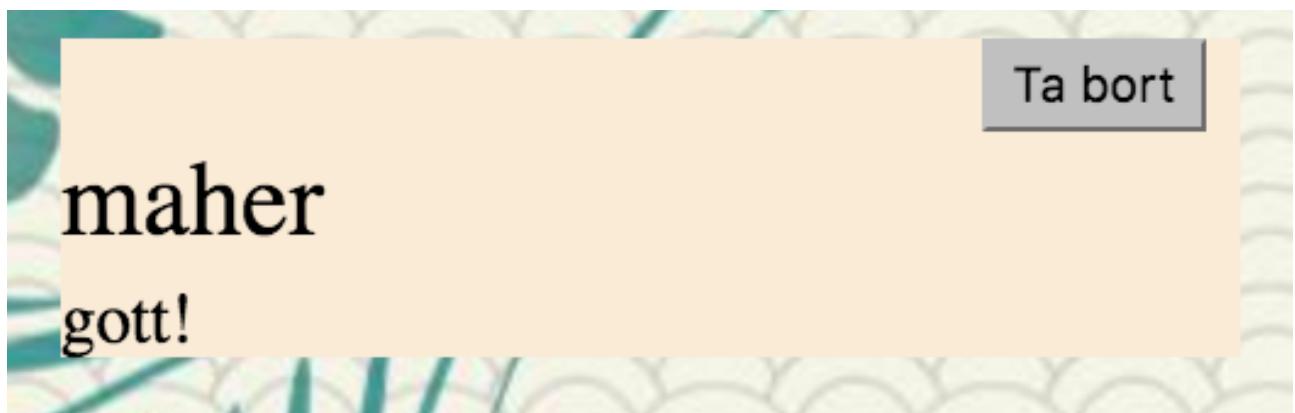
Om if-satsen är true alltså då man lyckas logga in så dyker kommentarsfältet upp ifall man inte är inloggad så syns inte kommentarsfältet och så går det inte att skriva en kommentar på receptsidorna. Jag har även inkluderat PHP filerna för att kunna skriva kommentar, posta den i receptsidan man väljer att kommentera och att användaren som är skrivit en kommentar ska kunna ta bort sin kommentar.

```

if($result->num_rows > 0){
    while($row = $result->fetch_assoc()) {
        echo '<div class="kommentar">';
        if ($row['username'] == $_SESSION['username']) {
            ?>
            <form action="" method="post">
                <input type="hidden" value=<?php echo $row['id']; ?>" name="commentid">
                <div>
                    <input class="tabort" type="submit" value="Ta bort">
                </div>
            </form>
            <?php
        }
        echo '<h6>' . $row['username'] . '</h6>';
        echo '<p>' . $row['comment'] . '</p>';
        echo '</div>';
    }
}

```

Ifall num_rows är större än 0 som i detta fall då man har skrivit en kommentar. Num_rows i detta fall är kommentaren man skrivit och så läggs kommentaren man skapat i receptsidan man kommenterat och man kan även se att jag har skapat en ta bort knapp för att man ska kunna radera sin kommentar man skrivit.



Figur 12. Kommentar med en ta bort knapp



Figur 13. Kommentaren är borttagen med hjälp av ta bort knappen

```
1  |?php
2
3 ▼ if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true) {
4 ▼   if (isset($_POST['commentid']) && $_POST['commentid'] != '') {
5     require_once("database.php");
6
7     $_POST['commentid'] = mysqli_real_escape_string($conn, $_POST['commentid']);
8
9     $sql = "DELETE FROM comments WHERE username='". $_SESSION['username']."' AND id='". $_POST['commentid']."'";
```

Figur 14.

PHP filen för att man ska kunna radera sin skapade kommentar

Denna PHP fil inkluderas i receptsidorna som gör så att användaren kan radera sin kommentar. \$sql = "DELETE FROM comments WHERE username =..." är just det som gör att användaren kan radera sin kommentar bara ifall användaren är inloggad. mysqli_real_escape_string () funktionen undgår specialtecken i en sträng för användning i en SQL-sats.

5 Diskussion

De obligatoriska uppgifterna för detta uppdrag var att göra det möjligt för en användare att logga in på webbsidan och även ut, göra det möjligt för en användare att kommentera i receptsidorna och göra det möjligt för en användare att ta bort en kommentar som den inloggade användaren har skrivit som visas i resultatavsnittet uppgifter.

Det fanns också två extra uppgifter ett var att göra det möjligt för en ny användare att registrera och den andra var att rädda recept i XML-filer. Tyvärr så gjorde jag inte någon av de extra uppgifterna efter som tiden fanns inte för det då jag blev fullt upptagen med de obligatoriska uppgifterna.

Att göra denna uppgift har jag dock lärt mig om hur man skriver PHP, jag trodde det var lite svårt i början och jag hade några problem på hur \$ SESSION och \$ POST arrayer fungerade, men när jag listat ut det gick ganska smidigt. Jag var även tvungen att kolla på massa tutorials om PHP för att veta hur jag skulle ta mig tillväga. Jag tog även hjälp av w3schools och handledningen som var stor hjälp. Till nästa seminarium kommer jag ha bättre grepp om PHP eftersom nu har fått lära mig att använda mig av lite PHP och det kändes bra.

6 Kommentarer om kursen

Jag spenderade ganska många timmar på att skriva PHP koden för att man skulle kunna logga in, kommentera och ta bort kommentarerna. Hade även svårt att komma igång men det blev bättre dag för dag. Men att skapa login from och kommentarsboxen gick ganska fort då man har börjat lära sig html och CSS. Kursen är bra upplagd som vanligt.