introduction to SCRIPTING, DATABASES, SYSTEM ARCHITECTURE

Web Services I: mailing list, restaurant rating



Claus Brabrand

((brabrand@itu.dk)))

Associate Professor, Ph.D. (((Software and Systems))) IT University of Copenhagen

Agenda

Course Evaluation

- Recap: PHP + SQL
- Sending Emails (revisited)
- Web Service: Mailing List (cont'd)
- Web Service: Restaurant Rating

ITU Online Course Evaluation



41 out of 85

Thanks for participating! Here are a few of the hi-lights (see it on mit.itu.dk)

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

[3] Nov 16, 2012

Course Evaluation (course)



Course Evaluation (course)

My time consumption for this course is too high, compared to the norm decided by the Board of Studies

I am satisfied with my effort on this course



Course Evaluation (teacher)



Course Evaluation (TAs)



Course Evaluation (qualitative)

Qualitative: Generally very positive, but also some problems:

Formulation of assignments!

- Assignments are hard
- Very difficult course
- Big workload
- TA presence (i.e., want more TA presence)
- This isn't useful to me later

Assignments:	Completely new:	No more new stuff:	TA presence:
I wish I had time	takes time + effort	Rest is "just" about	1 TA = 70 hrs
to re-formulate all	Train for the exam	combining what you	≈ 5 hrs / week
assignments !!!	(rest of course!)	have already learnt	(incl. correction)
<u> </u>			

Agenda

Course Evaluation

Recap: PHP + SQL

Sending Emails (revisited)

- Web Service: Mailing List (cont'd)
- Web Service: Restaurant Rating

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

```
<html>
                                                                           Complete
    <body>
                                                                           PHP+SQL
                                                      mysql.itu.dk:
        <?php
                                                                            Example
mysql connect("mysql.itu.dk", "claus", "pa$$word");
                                                       students
mysql select db("students");
                                                                    STUDENTS:
mysql query("DROP TABLE students;");
// CREATE students TABLE:
                                                                    STUDENTS:
mysql query("CREATE TABLE students (
                name VARCHAR(20) NOT NULL,
                                                                     name | address
                address VARCHAR(80) NOT NULL
             );");
// INSERT student records:
mysql query("INSERT INTO students (name, address) VALUES ('Anna', 'Somewhere 3');");
mysql query("INSERT INTO students (name, address) VALUES ('Brian', 'Homestreet 4');");
mysql query("INSERT INTO students (name, address) VALUES ('Claire', 'Nowhere 9b');");
                                                                    STUDENTS:
// SELECT everything and print it out nicely:
                                                                     name | address
$rows = mysql query("SELECT * FROM students;");
                                                                     Anna |
                                                                            Somewhere 3
while ($row = mysql fetch array($rows)) {
                                                                     Brian Homestreet 4
    $name = $row['name'] ;
    $addr = $row['address'] ;
                                                                     Claire | Nowhere 9b
    echo "<b>$name</b> lives: <em>$addr</em>" ;
}
mysql close();
                                                                • Anna lives: Somewhere 3
        ?>
                                                                • Brian lives: Homestreet 4
    </body>
                                                                 • Claire lives: Nowhere 9b
</html>
```

Agenda

Course Evaluation

Recap: PHP + SQL

Sending Emails (revisited)

- Web Service: Mailing List (cont'd)
- Web Service: Restaurant Rating

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

Sending Email

Sending Email:

<pre>mail('obama@whitehouse.com',</pre>	<pre>// recipient</pre>
'Congrats!',	// subject
'Dear Barack, Congrats on the elections	.'); // message

...and with sender:

mail('obama@whitehouse.com',	<pre>// recipient</pre>
'Congrats!',	// subject
'Dear Barack, Congrats on the elections',	// message
<pre>'From: mitt_romney@hotmail.com');</pre>	// sender

Note: function 'mail()' returns boolean status:

\$status = mail(...);
if (\$status) {
 echo "Your email was sent" ; // NB: no reception guarantee!
}

Send Mail Web Service

```
<html><body>
                                                <?php
                                      HTML
                                                                                       PHP
   <form action="send mail.php">
                                                include("fn headerfooter.php");
                                                include("fn input validation.php");
      To:
     <input type="text" name="to"/>
                                                show header("Sending Email");
     \langle br \rangle >
                                                $to
                                                      =  REQUEST['to']; 
      Subject:
                                                $subj = $ REQUEST['subject'];
     <input type="text" name="subject"/>
                                                $msg = $ REQUEST['message'];
     \langle br / \rangle
                                                chk email($to);
                                                $status = mail($to, $subj, $msg);
     Message:
     <textarea name="message">
     </textarea>
                                                if ($status) {
                                                  echo "Your mail was sent!" ;
                                                } else {
     echo "It wasn't possible to send!" ;
     <input type="submit" value="Send"/>
                                                show footer();
   </form>
 </body></html>
                                                ?>
                           send mail.html
                                                                           send mail.php
                                                                             [13]
Claus Brabrand, ITU, Denmark
                             SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE
                                                                                    Nov 16, 2012
```

Agenda

Course Evaluation

Recap: PHP + SQL

Sending Emails (revisited)

Web Service: Mailing List (cont'd)

Web Service: Restaurant Rating

Making Web Services

4-Step design process...:



Making Web Services

4-Step design process...:



maillest mailkst_ maillist.php add.php new.html maillist.php New: 25 SQL Name: INSERT Emil: <html><body>List: <?php include("fn mydb connect.php"); mydb connect(); \$rows = mysql query("SELECT * FROM maillist;"); while (\$row = mysql fetch array(\$rows)) { \$name = \$row['name']; \$email = \$row['email']; echo "\$name"; mysql close(); ?> NEW </body></html>





...and with **DELETE**

4-Step design process...:



maillist.php



```
<html><body>List:
<?php include("fn mydb connect.php");</pre>
   mydb connect();
    $rows = mysql query("SELECT * FROM maillist;");
    while ( $row = mysql fetch array($rows) ) {
        $name = $row['name'];
        $email = $row['email'];
        echo "
             [<a href='maillist del.php?email=$email'>X</a>]
             <a href='mailto:$email'>$name</a>
             >" ;
   mysql_close();
?>
<a href='maillist add.html'>NEW</a>
</body></html>
```

maillist_del.php



```
<?php
include("fn_mydb_connect.php");
mydb_connect();
$email = $_REQUEST['email'];
mysql_query("DELETE FROM maillist WHERE email='$email';");
header("Location: maillist.php"); // jump!
mysql_close();
?>
```

...and with **SORT**

4-Step design process...:



...and with **SENDING EMAIL**

4-Step design process...:



maillist.php

<html><body>List:



```
<?php
include("fn_mydb_connect.php");
mydb_connect();
$rows = mysql_query("SELECT * FROM maillist;");
while ( $row = mysql_fetch_array($rows) ) {
    $name = $row['name'];
```

```
$email = $row['email'];
echo "<a href='mailto:$email'>$name</a>" ;
```

```
}
mysql_close();
?>
```

```
<a href='maillist new.html'>NEW</a>
```

```
<a href='maillist_spam.html'>SPAM</a>
```

</body></html>





Agenda

Course Evaluation

- Recap: PHP + SQL
- Sending Emails (revisited)
- Web Service: Mailing List (cont'd)

• Web Service: Restaurant Rating

Restaurant Rating Web Service



and

forms:

[29] Nov 16, 2012

Restaurant Rating Web Service

4-Step design process...:



1) Create Tables

Restaurants:





Ratings:

CREATE TABLE ratings (
id INT NOT NULL,
rating INT NOT NULL,
FOREIGN KEY (id)
REFERENCES restaurants(id)
);

r.php



```
<html><body><h3>Restaurants</h3>
<?php
include("fn mydb connect.php");
mydb connect();
$rows = mysql query("SELECT r.id, name, AVG(rating) as average, COUNT(rating) as count
                    FROM restaurants as r LEFT JOIN ratings as s
                     ON r.id = s.id
                                                          Restaurants
                     GROUP BY r.id
                                                           • Maharaja: 4.67 (3 rating(s)):
                    ORDER BY average DESC;");
                                                           • <u>Viet-nam-nam</u>: 4 (1 rating(s)):
while ( $row = mysql fetch array($rows) ) {
  $id = $row['id'] ;
                                                           • <u>EatIT</u>: 1 (4 rating(s)):
  $name = $row['name'] ;
  $average = round($row['average'], 2);
  $count = $row['count'] ;
                                                          New Restaurant
  $width = $average * 100 ;
  echo "<a href='r rate.php?id=$id'>$name</a>: <b>$average</b> ($count rating(s)):
       $average
        ";
}
mysql close();
?>
/><a href="r new.html">New Restaurant</a></body></html></a>
```

r_new.html



<html></html>	
<body></body>	
<h3>New Restaurant</h3>	New Restaurant
<form action="r_new.php"></form>	
Name: 	Name:
<input name="name" type="text"/>	
<input type="submit" value="Create!"/>	Create

r_new.php



<?php

```
include("fn_mydb_connect.php");
mydb_connect();
$name = $_REQUEST['name'];
mysql_query( "INSERT INTO restaurants (name) VALUES ('$name');" ); // NB: AUTO_INCREMENT!
mysql_close();
header("Location: r.php"); // JUMP!
?>
```

r_rate.php



```
<html><body><?php
include("fn mydb connect.php");
mydb connect();
$id = $ REQUEST['id'];
$rows = mysql query( "SELECT * FROM restaurants WHERE id = $id;" );
$row = mysql fetch array($rows); // we know there is one!
                                                                    Rate Restaurant: Maharaja
$name = $row['name'];
                                                                       • 0 5
                                                                       • 0 4
echo "<h3>Rate Restaurant: $name</h3>" ;
                                                                       • 🔘 3
echo "<form action='r rate update.php'>" ;
                                                                       • 0 2
echo "" ;
                                                                       • 🔍 1
echo "<input type='hidden' name='id' value='$id' />" ;
                                                                     Rate
for ($i=5; $i>0; $i--) {
  echo "<input type='radio' name='rating' value='$i'> $i";
}
echo "" ;
echo "<input type='submit' value='Rate!'/>" ;
echo "</form>" ;
mysql close();
?></body></html>
```

r_rate_update.j



<?php

```
include("fn_mydb_connect.php");
mydb_connect();

$id = $_REQUEST['id'];
$rating = $_REQUEST['rating'];
mysql_query( "INSERT INTO ratings (id, rating) VALUES ('$id', '$rating');" );
mysql_close();
header("Location: r.php"); // JUMP !
?>
```

A9: Project Broker

Project Broker

• Project: "Man on the Moon" (by Barack Obama) [update]

Project: "Climate 3.0" (by <u>Al Gore</u>) [update]

· The project is about getting someone on the moon. We seek a PHP programmer and an astronaut.

4-Step design process:

- 1) Design data model
- 2) Develop data transactions
- 3) Develop site map and forms
- 4) Program it all in SQL+PHP

	Project Broker [Create New Project] LIST
⁻ airly Open Assignment	Title: Man on the Moon Name: Barack Obama Email: obama@whitehouse. Password: •••• This project is about putting someone on the Moon. We need a PHP programmer and an astronaut. Welcome back Barack Obama, Enter your password and updated project description for your project "Man on the Moon". This project is about putting someone on the Moon. We need a PHP programmer and an astronaut. This project is about putting someone on the Moon. We need a PHP programmer and an astronaut. Password: •••• Create Project NEW
Brabrand, ITU, Denmark	SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE [37] Nov 16, 2012

Claus Brabrand, IIU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

A9.1: Design Data Model

Destination file: A9/project_table.sql

Construct a suitable data model based on the requirements presented earlier. More concretely, define an SQL table using the CREATE TABLE statement which contains all the necessary fields, guided by the following description:

The table should contain information about the title of the project, a short project description, and a project editing password. It should furthermore contain info about the project administrator (name, email). Furthermore, each project should have a unique numerical id (use PRIMARY KEY and AUTO_INCREMENT). Finally, each project title should be forced to be unique by declaring that field as UNIQUE.

Add two projects to your table using suitable INSERT INTO commands. Save your CREATE TABLE and INSERT INTO statements in the file A9/project_table.sql

(Note: It would probably be better to split the table up into two tables, one containing data about projects, the other containing data about project administrators. You are welcome to do such a design if you want, but it is **not** mandatory for this exercise.)

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

[38] Nov 16, 2012

A9.2: Database Transactions

Destination file: **A9/project_transactions.sql**

Construct SQL queries for:

- 1) showing a list of all projects in the database
- 2) inserting a new project
- 3) updating the project description for a given project.

Save your queries in the file **A9/project_transactions.sql**

A9.3: Site Map

Destination file: A9/site_map.gif/jpg/png/pdf

Now the time has come to design the overall sitemap of the broker service. What web pages and PHP scripts are needed and what variables need to be sent between them? On the next slide is a an almost complete site map diagram showing one possible structure of a Project Broker web service.

Your task is now to:

- 1) to give name to all necessary PHP and HTML files (which are unnamed on the next page) in your project broker service solution; and
- 2) to also define the form variables that each file is expecting to send and receive (if any). Please add that info next to the arrows.

The only constraint is that the main file (the one a user typically will access first) should be given the name **A9/projects.php** so that the TAs can quickly locate and start your service easily.

Claus Brabrand, ITU, Denmark

A9.3: Site Map (cont'd)



Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

[41] Nov 16, 2012

A9.4: Program Web Service

Destination files: A9/projects.php, A9/[yourfilenames.php, yourfilenames.html], A9/includes/[yourfilenames.php] (optional)

In this part of the exercise, you will implement (construct) the files needed to generate the web pages and performing the database transactions pictured in your design (A9.3).

Note 1: When implementing the PHP file for updating a project description, remember to include code for checking the project description password. Only if you know the password for the project, should a user be allowed to update a project description.

Note 2: You do not have to do any input validation in this exercise.

Note 3: The PHP scripts whose main task is to perform database transactions will probably benefit from using the PHP function **header()** to make the user's browser "jump" to a suitable web page after the transaction has been performed; e.g.

header("Location: projects.php");

Put all your PHP and HTML files for this exercise in **A9**/, and any files you include in **A9**/includes/.

Making Web Services

4-Step design process...:

Any Questions?

(Have a nice weekend)

Claus Brabrand, ITU, Denmark

SCRIPTING, DATABASES, & SYSTEM ARCHITECTURE

Nov 16, 2012