Computer Science Department  
Michigan State University  
CSE480 Database Systems, Spring 2003  
Lab Week #8: PHP Scripting Language for Generating Dynamic Web Pages

YOUR NAME:

OBJECTIVES: In this lab you will use scripting language PHP for generating web pages dynamically. The objective is to develop web based database applications using PHP. In the first part of this lab you will learn PHP syntax. In the second part you will access Oracle through PHP.

You may not have to type the programs. A text file is posted for you to copy and paste. Part I:

(40 minutes) You need a cgi-bin directory. Please apply for it from DECS as soon as possible. The web address for the information is at:  
http://www.eegr.msu.edu/decs/request/cgi.php

This will allow you to create ~/web/cgi-bin directory in your account. You cannot proceed any further in this lab without the authorization from DECS. You need to note the following before writing any php code.

1. PHP script must be stored in a file with php extension (e.g., file.php) and must be in your ~/web/cgi-bin directory.

   execute
   chmod 755 ~/web
   chmod 755 ~/web/cgi-bin

2. The first line of the file.php file must be

   #!/opt/bin/php

   This is necessary when using DEC server.

3. When using WINDOWS EDITORS LIKE NOTE PAD, file.php must not contain \M in the end of each line. To remove them, you may run the following:
   dos2unix file.php file.php

1
4. Because you may put oracle account/password in the php file, you should make the file executable only by running:

    chmod 711 file.php

Now you will create and run PHP code:

1. Getting Started: Hello World PHP

```php
#!/opt/bin/php
<html>
    <head>
        <title>Example 1</title>
    </head>
    <body>
        <?
        echo "<h1>Hello World!</h1>";
        ?>
    </body>
</html>
```

(a) Store the above into `Helloworld.php` file.
(b) Put it into `/web/cgi-bin` directory.
(c) chmod 711 Helloworld.php
(d) See it at www.egr.msu.edu/~yourusername/cgi-bin/helloworld.php

2. Variable, String and Datatypes

```php
#!/opt/bin/php
<html>
    <head>
        <title>Example 2</title>
    </head>
    <body>
        <?
        $website = "http://www.bitafterbit.com";
        echo "<br>Surf to: $website";
        echo 'Surf to: $website';
```
(a) A variable is defined by a preceding $ sign.
(b) Store the above into Variable.php file.
(c) Put it into ~/web/cgi-bin directory.
(d) chmod 711 Variable.php
(e) See it at
   www.egr.msu.edu/~yourusername/cgi-bin/Variable.php
(f) Which quotes above, single or double, use the variable $website?
(g) It is not necessary to declare the type of the variable in php.
   In PHP the following atomic datatypes are admitted. Integer;
   Floating Point number; String;

3. Now try the following example code for FOR LOOP.
   The code draws a 4*2 table.

   ```php
   #!/opt/bin/php
   <HTML>
   <HEAD>
     <TITLE>Example of For</TITLE>
   </HEAD>
   <BODY>
     <?
     echo("<TABLE ALIGN=CENTER BORDER=1 CELLSPACING=5>");
     for ($j=1; $j<=4; $j++) {
       echo("<TR>");
       for ($k=1; $k<=2; $k++)
         echo("<TD> Line $j, Cell $k </TD>");
       echo("</TR>");
     }
     echo("</TABLE>");
     ?>
   </BODY>
   </HTML>
   
   (a) Store the above into Forloop.php file.
(b) Put it into ~/web/cgi-bin directory.
(c) chmod 711 Forloop.php
(d) See it at
   www.egr.msu.edu/~yourusername/cgi-bin/Forloop.php

4. Now you write a PHP script using the syntax of WHILE LOOP to
draw a 4x2 table the same way as the example of FOR LOOP.
syntax for while loop is as follows:

   While loop:
   while (condition){
      BodyOfLoop;
   }

5. PHP also provides other control structures such as IF, and capability
to define functions and arrays.

   Tutorial References:
   1) http://www.freewebmasterhelp.com/tutorials/php

   For your project you read the whole tutorial carefully and try the
   examples outside the lab. Make sure that you are able to write some
   basic programs in PHP.

PART II ORACLE IN PHP
(45 minutes)

1. You will be using the concept of cursor in this part of the lab.

   The idea of a CURSOR:
   Result of a select statement is a table. Without naming the table,
   one can access this table, one tuple at a time, through the use of a
cursor. You may think of a cursor as a pointer which goes through the
resulting table sequentially, one tuple at a time.

2. Oracle INSERT using PHP:

   Run the following to see what is in the table Faculty. Try the following
   and see what is being inserted.

   Select * from Faculty in sqlPlus.

   Now try the following and see what is being inserted in the table
   Faculty.
#!/opt/bin/php
<?php
// set up environment
putenv("ORACLE_HOME=/opt/oracle/9.2.0");
putenv("ORACLE_SID=cse480");
putenv("TWO_TASK=cse480");

// setup connection with Oracle
// replace the cse480/cse480 with your own Oracle username and password
$connection = OCILogon ("cse480", "cse480");
if ($connection == false){
    echo OCIError($connection)."<BR>";
    exit;
}

// Defining an SQL command and store it in a variable $query$.

$query = "insert into Faculty(FacName, FacSSNo, OfficeAddress, WorksFor) " .
    "values ('Mike', '000000010', 'Engineering Building', 'CSE')";

// Define a cursor for the SQL command stored in the variable $query$.

$cursor = OCIExecute ($connection, $query);
if ($cursor == false){
    echo OCIError($cursor)."<BR>";
    exit;
}

// Run the cursor

$result = OCIExecute ($cursor, OCI_DEFAULT);
if ($result == false){
    echo OCIError($cursor)."<BR>";
    exit;
}

// commit the result
OCICcommit ($connection);
// close the connection with oracle
OCILogoff ($connection);
?>

Select * from Faculty in sqlPlus

Note: In the OCIExecute, we use the option OCI_DEFAULT, which only executes the insert command without committing. If we use the option OCI_COMMIT_ON_SUCCESS, which is the default option of OCIExecute(), the statements will be committed automatically after the execution of OCIExecute().

3. Now you write a PHP script which will update the department attribute of Faculty member, Mike, to EE. Find out the small portion of the above code that needs to be changed to achieve this.

4. Run the following example query to Display all tuples in table Faculty.

```php
#!/opt/bin/php
<?
// set up environment
putenv("ORACLE_HOME=/opt/oracle/9.2.0");
putenv("ORACLE_SID=cse480");
putenv("TWO_TASK=cse480");

// setup connection with Oracle
// replace the cse480/cse480 with your own Oracle username and password
$connection = OCILogon ("cse480", "cse480");
if ($connection == false){
    echo OCIError($connection)."<BR>";
    exit;
}

// this is the SQL command to be executed
$query = "select * from Faculty";
// parse the SQL command
$cursor = OCIParse ($connection, $query);
= OCIParse ($connection, $query);
```
if ($cursor == false){
    echo OCIError($cursor)."<BR>";
    exit;
}

// execute the command
$result = OCIExecute ($cursor);
if ($result == false){
    echo OCIError($cursor)."<BR>";
    exit;
}

// display the results
echo "<table border=1>";
echo "<tr><td><b>Name</b></td><b>SSN</b></td><b>Address</b></td><b>Department</b></tr>

// fetch the result from the cursor one by one
while (OCIFetchInto ($cursor, $values)){
    $name = $values[0];
    $ssn = $values[1];
    $address = $values[2];
    $dept = $values[3];

    echo "<tr><td>$name</td><td>$ssn</td><td>$address</td><td>$dept</td></tr>

} echo "</table>";

// close the connection with oracle
OCILogoff ($connection);
?>

5. Now you write a PHP script which displays the name, SSN and level of those students who are from Lansing.

6. References:

http://www.phpworld.com/articles/2000.03/oracle_000.html
PART III SESSION MANAGEMENT

(25 minutes)

References:


This part involves managing web sessions using php. You will need this part for your project 1. First part of this section is to copy a few files and run it. You must complete this part. Second part of this section involves answering questions. The idea is to understand the code. If you do not have time to answer all of them, ask the TA during his office hours.

Do the following to run the given php code for login:

(a) Create a table named **visitor**. The schema for visitor table is given in the file login.schema

(b) Insert the following two rows into the visitor table:
   'smith44', 'David Smith', 'happy10', NULL
   'gonzal09', 'Susan Gonzales', 'Lucky09', NULL

(c) Copy the file login.html into your **web directory**.

(d) Copy loginResponse.php, welcomePage.php and verifySession.php into your **cgi-bin directory**.

(e) Now in netscape you open the web page at address:
   http://www.egr.msu.edu/~youruserid/login.html

(f) You will see a web page displayed prompting for user id.

(g) Type the user id **smith44**

(h) Understand the **FORM** statement in login.html file.
   Answer the following:
   i. What does the FORM statement do?
   ii. What is the function of the parameter **submit**.
   iii. What gets stored into the variable **txtname**?

(i) What should be the values of the variable **$values[0]** in the loginResponse.php program.
(j) The following statement in the loginResponse.php program creates a unique session id by generating a random number. Unique session Id is stored into the variable $sessionId.

$SessionID = md5(uniqid(rand()));

Give the tuple that is stored in the table VSession. How does the tuple get into the table?

- The purpose of inserting the above tuple into the Vsession table is to identify valid users for subsequent web pages being generated in the session.
- The statement
  
  Header("Location:http://www.egr.msu.edu/~yourUserID/cgi-bin/
  welcome.php?SessionID=$SessionID");

  invokes the php program welcomePage.php and passes the value of $SessionID into this program.

(k) The first line of the welcomePage.php invokes verifySession.php. The purpose of invoking this php code is to verify a valid user. This is done by getting the user id from the Vsession table to check for valid user. Note that it is possible to invoke this page without going through the login session and access the database information.

  What does the rest of the code in welcomePage.php do?

PART IV OTHER REFERENCES


The link to ”Alphabetical function list by PHP.net” may be useful to lookup the explanation of a particular function. Other links are also useful when you use the PHP more and more for programming.