CNT 4714: Enterprise Computing Spring 2011

PHP – Database Connectivity

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cnt4714/spr2011

Department of Electrical Engineering and Computer Science
University of Central Florida

Form Processing and Business Logic

- XHTML forms enable web pages to collect data from users and send it to a web server for processing.
- Interaction of this kind between users and web servers is vital to e-commerce applications. Such capabilities allow users to purchase products, request information, send and receive web-based email, perform on-line paging and take advantage of various other online services.
- The XHTML document on the next few pages collects information from a user for the purposes of adding them to a mailing list.
- The PHP file on page 3 validates the data entered by the user through the form and "registers" them in the mailing list database.

form.html Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- form.html                    -->
<!-- Form for use with the form.php program -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Sample form to take user input in XHTML</title>
   </head>
   <body>

      <h1>This is a sample registration form.</h1>

      Please fill in all fields and click Register.
      <!-- post form data to form.php -->
      <form method = "post" action = "form.php">
         <img src = "images/user.gif" alt = "User" /><br />
         <span style = "color: blue">
            Please fill out the fields below.<br />
         </span>
         <!-- create four text boxes for user input -->
         <img src = "images/fname.gif" alt = "First Name" />
         <input type = "text" name = "fname" /><br />
```

This XHTML document generates the form that the user will submit to the server via form.php

```
<img src = "images/lname.gif" alt = "Last Name" />
<input type = "text" name = "lname" /><br />
<img src = "images/email.gif" alt = "Email" />
<input type = "text" name = "email" /><br />
<img src = "images/phone.gif" alt = "Phone" />
<input type = "text" name = "phone" /><br />
<span style = "font-size: 10pt">
   Must be in the form (555)555-5555</span>
<br /><br />
<img src = "images/downloads.gif"
   alt = "Products" /><br />

<span style = "color: blue">
   Which publication would you like information about?
</span><br />

<!-- create drop-down list containing magazine names -->
<select name = "magazine">
   <option>Velo-News</option>
   <option>Cycling Weekly</option>
   <option>Pro Cycling</option>
   <option>Cycle Sport</option>
              <option>RadSport</option>
              <option>Mirror du Cyclisme</option>
</select>
<br /><br />
```

```
        <img src = "images/os.gif" alt = "Operating System" />
        <br /><span style = "color: blue">
          Which operating system are you currently using?
        <br /></span>
        <!-- create five radio buttons -->
        <input type = "radio" name = "os" value = "Windows XP"
          checked = "checked" />
          Windows XP
        <input type = "radio" name = "os" value =
          "Windows 2000" />
          Windows 2000
        <input type = "radio" name = "os" value =
          "Windows 98" />
          Windows 98<br />
        <input type = "radio" name = "os" value = "Linux" />
          Linux

        <input type = "radio" name = "os" value = "Other" />
          Other<br />

         <!-- create a submit button -->
        <input type = "submit" value = "Register" />
      </form>

    </body>
  </html>
```

form.php Example

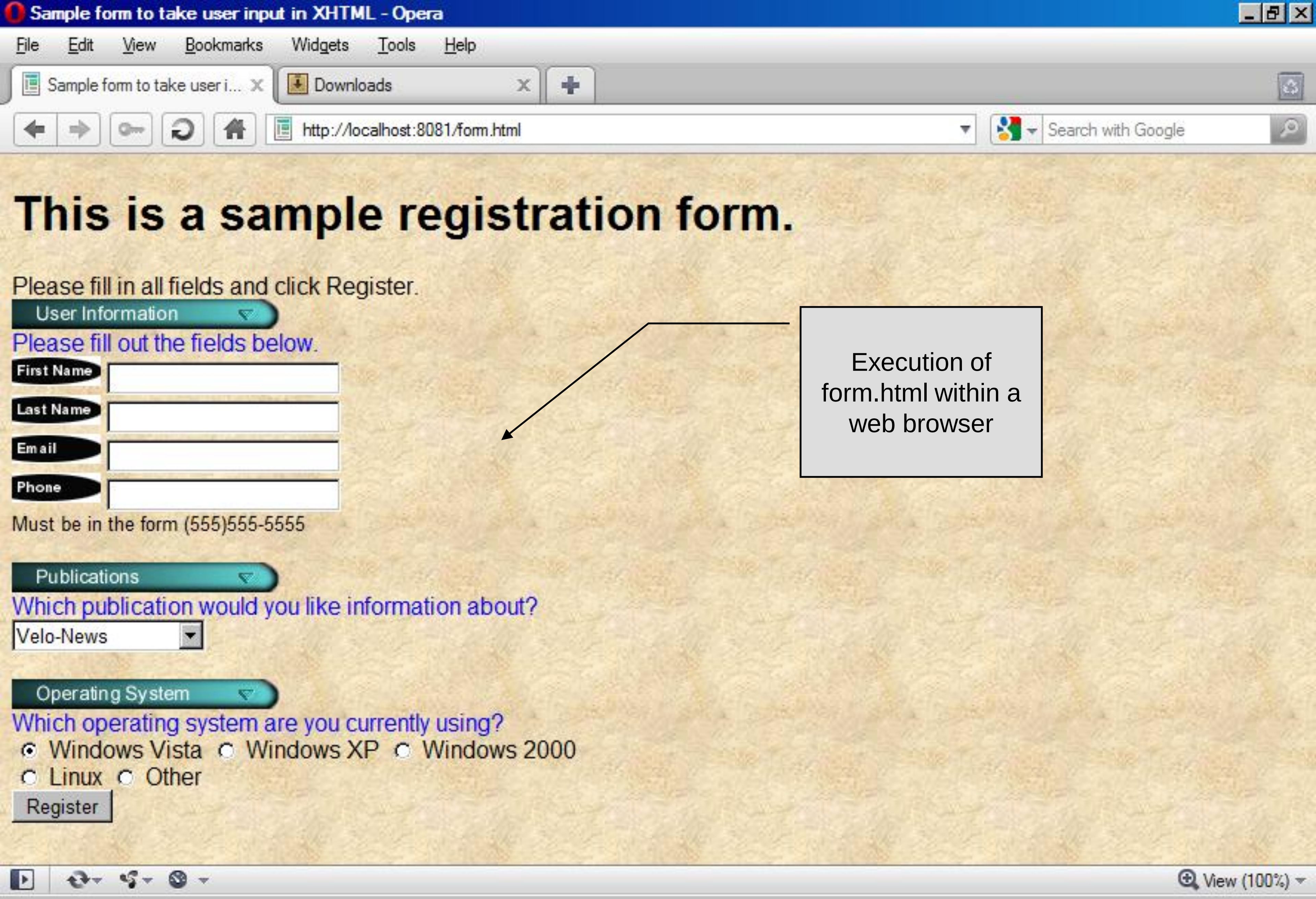
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- form.php     -->
<!-- Read information sent from form.html -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Form Validation</title>
   </head>
   <body style = "font-family: arial,sans-serif">

      <?php
         extract($_POST);
         // determine whether phone number is valid and print an error message if not
         if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$",
            $phone ) ){
            print( "<p><span style = \"color: red; font-size: 2em\">
               INVALID PHONE NUMBER:</span><br />
               A valid phone number must be in the form
               <strong>(555)555-5555</strong><br />
               <span style = \"color: blue\">
               Click the Back button, enter a valid phone number and resubmit.<br /><br />
               Thank You.</span></p></body></html>" );
            die(); // terminate script execution
         }
      ?>
```

Function extract (associativeArray) creates a variable-value pair corresponding to each key-value pair in the associative array $_POST.

See page 17 for explanation of regular expressions.

Function die() terminates script execution. An error has occurred, no need to continue.

```
    <p>Hi
      <span style = "color: blue"> <strong> <?php print( "$fname" ); ?>  </strong> </span>.
      Thank you for completing the survey.<br />
      You have been added to the  <span style = "color: blue">
        <strong>  <?php print( "$magazine " ); ?>  </strong> </span> mailing list.
    </p>
    <strong>The following information has been saved in our database:</strong><br />
     <table border = "0" cellpadding = "0" cellspacing = "10">
      <tr>
        <td bgcolor = "#ffffaa">Name </td>
        <td bgcolor = "#ffffbb">Email</td>
        <td bgcolor = "#ffffcc">Phone</td>
        <td bgcolor = "#ffffdd">OS</td>
      </tr>
      <tr>
        <?php
          // print each form field's value
          print( "<td>$fname $lname</td> <td>$email</td> <td>$phone</td> <td>$os</td>" );
        ?>
      </tr>
    </table>
   <br /><br /><br />
    <div style = "font-size: 10pt; text-align: center">
      This is only a sample form.   You have not been added to a mailing list.
    </div>
  </body>
</html>
```


Sample form to take user input in XHTML - Opera
File Edit View Bookmarks Widgets Tools Help
Sample form to take user i...
Downloads
http://localhost:8081/form.html
Search with Google
View (100%)

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name	
Last Name	
Email	
Phone	

Must be in the form (555)555-5555

Publications

Which publication would you like information about?

Velo-News

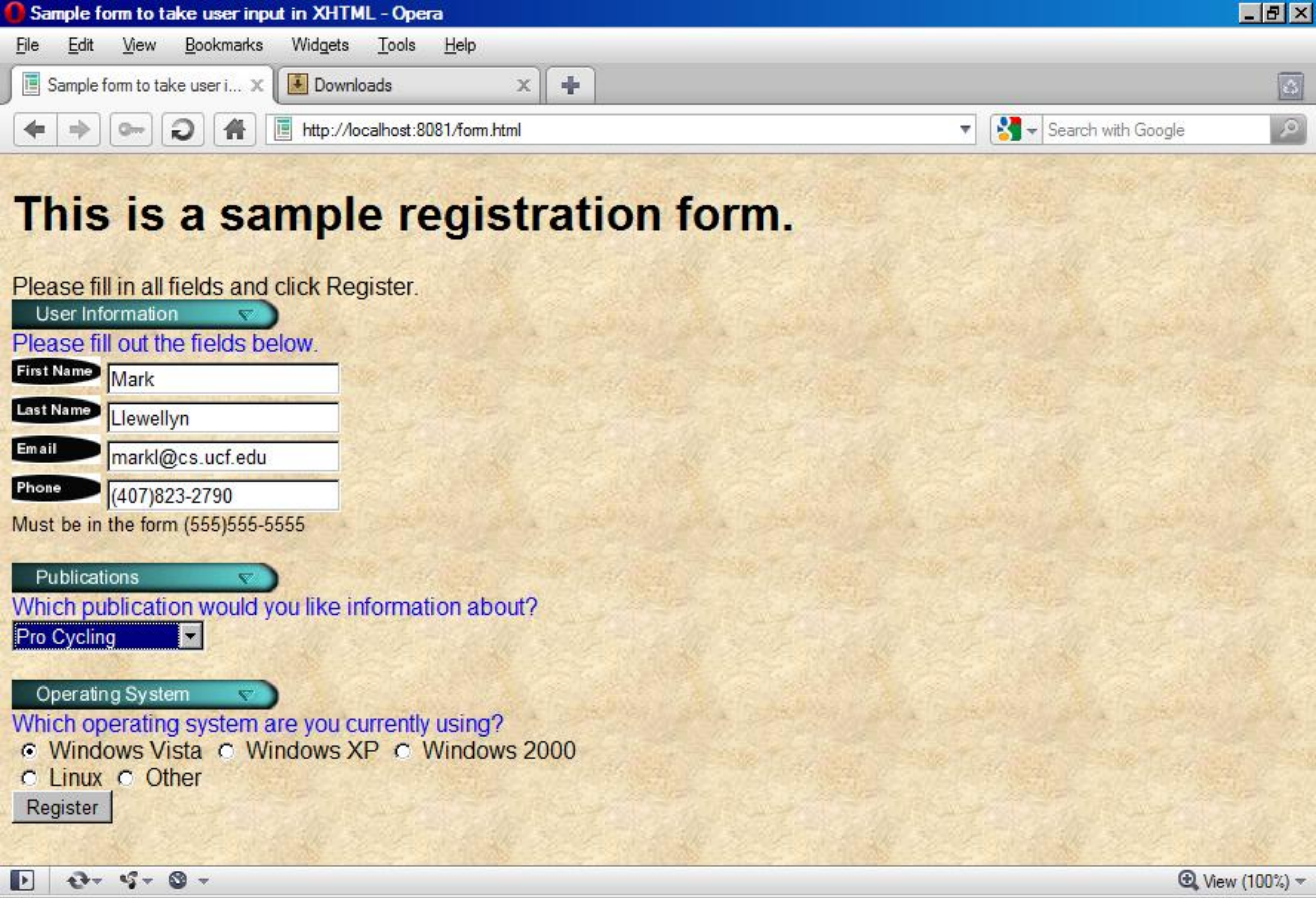
Operating System

Which operating system are you currently using?

[✓ Windows Vista] Windows XP Windows 2000 Linux Other

Register

Execution of form.html within a web browser


Sample form to take user input in XHTML - Opera
File Edit View Bookmarks Widgets Tools Help
Sample form to take user i...
Downloads
http://localhost:8081/form.html
Search with Google
View (100%)

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name	Mark
Last Name	Llewellyn
Email	markl@cs.ucf.edu
Phone	(407)823-2790

Must be in the form (555)555-5555

Publications

Which publication would you like information about?

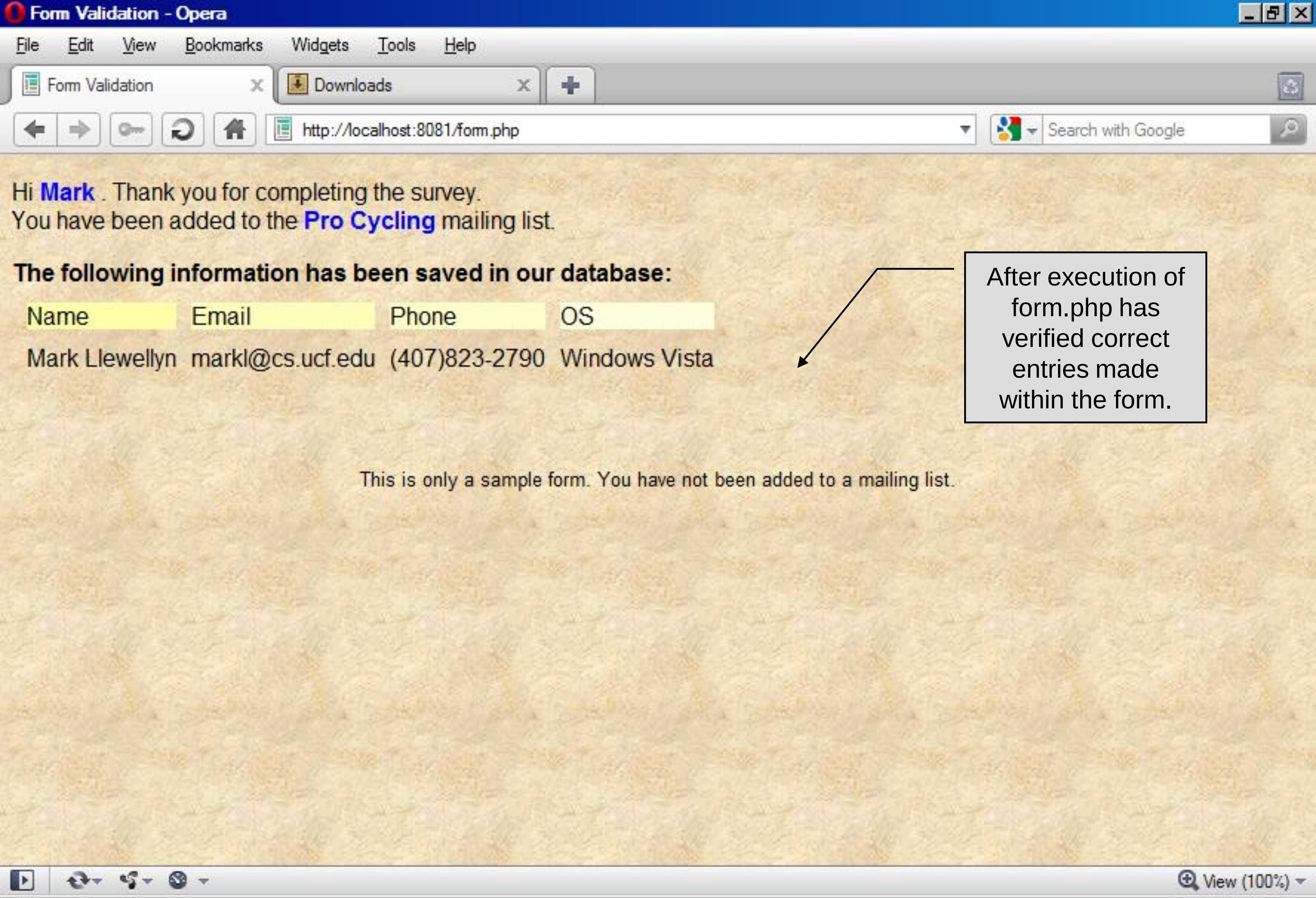
Pro Cycling

Operating System

Which operating system are you currently using?

[✓ Windows Vista] Windows XP Windows 2000
Linux Other

Register



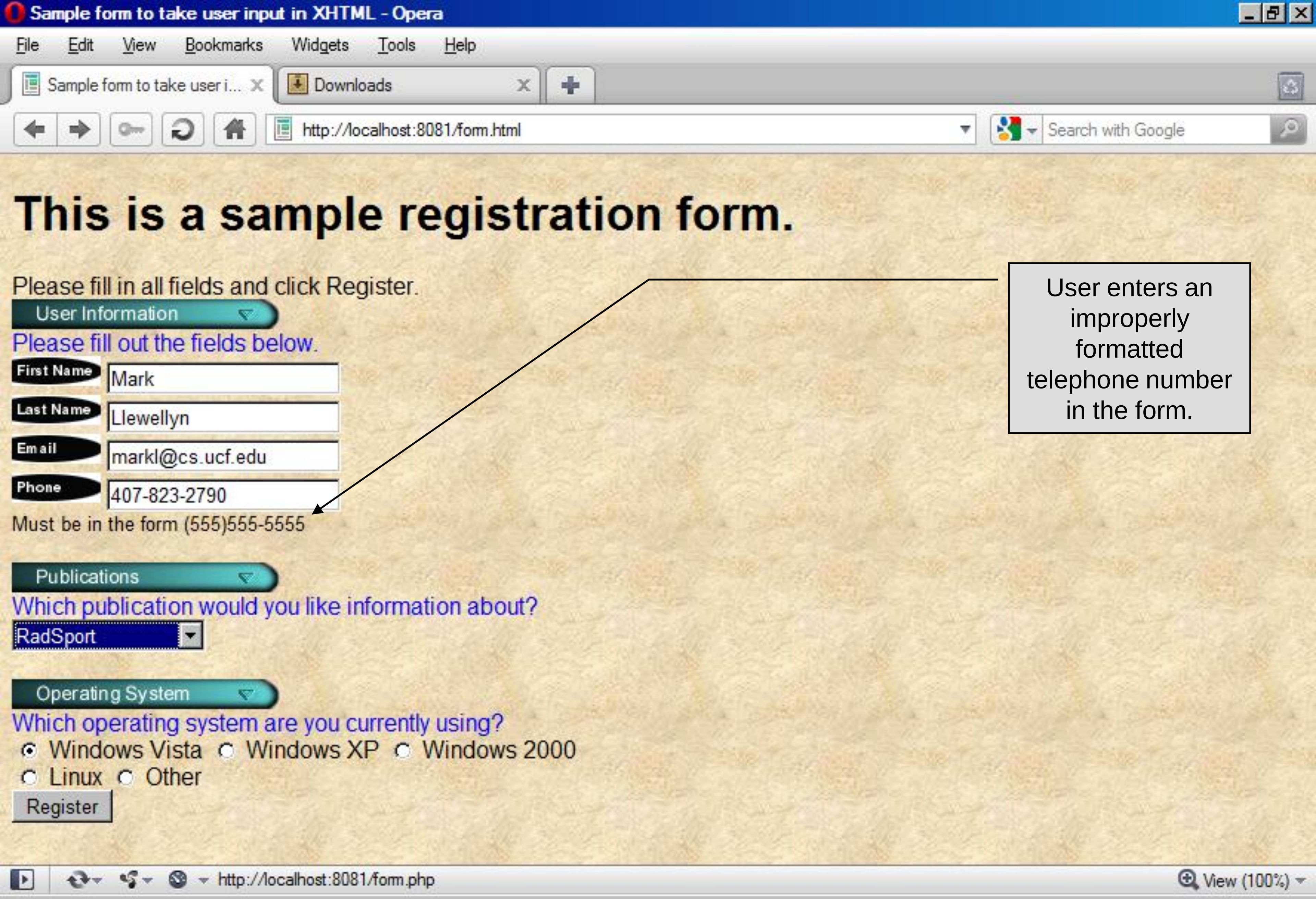
Hi **Mark** . Thank you for completing the survey.
You have been added to the **Pro Cycling** mailing list.

The following information has been saved in our database:

Name	Email	Phone	OS
Mark Llewellyn	markl@cs.ucf.edu	(407)823-2790	Windows Vista

This is only a sample form. You have not been added to a mailing list.

After execution of form.php has verified correct entries made within the form.


Sample form to take user input in XHTML - Opera

http://localhost:8081/form.html

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name: Mark

Last Name: Llewellyn

Email: markl@cs.ucf.edu

Phone: 407-823-2790

Must be in the form (555)555-5555

Publications

Which publication would you like information about?

RadSport

Operating System

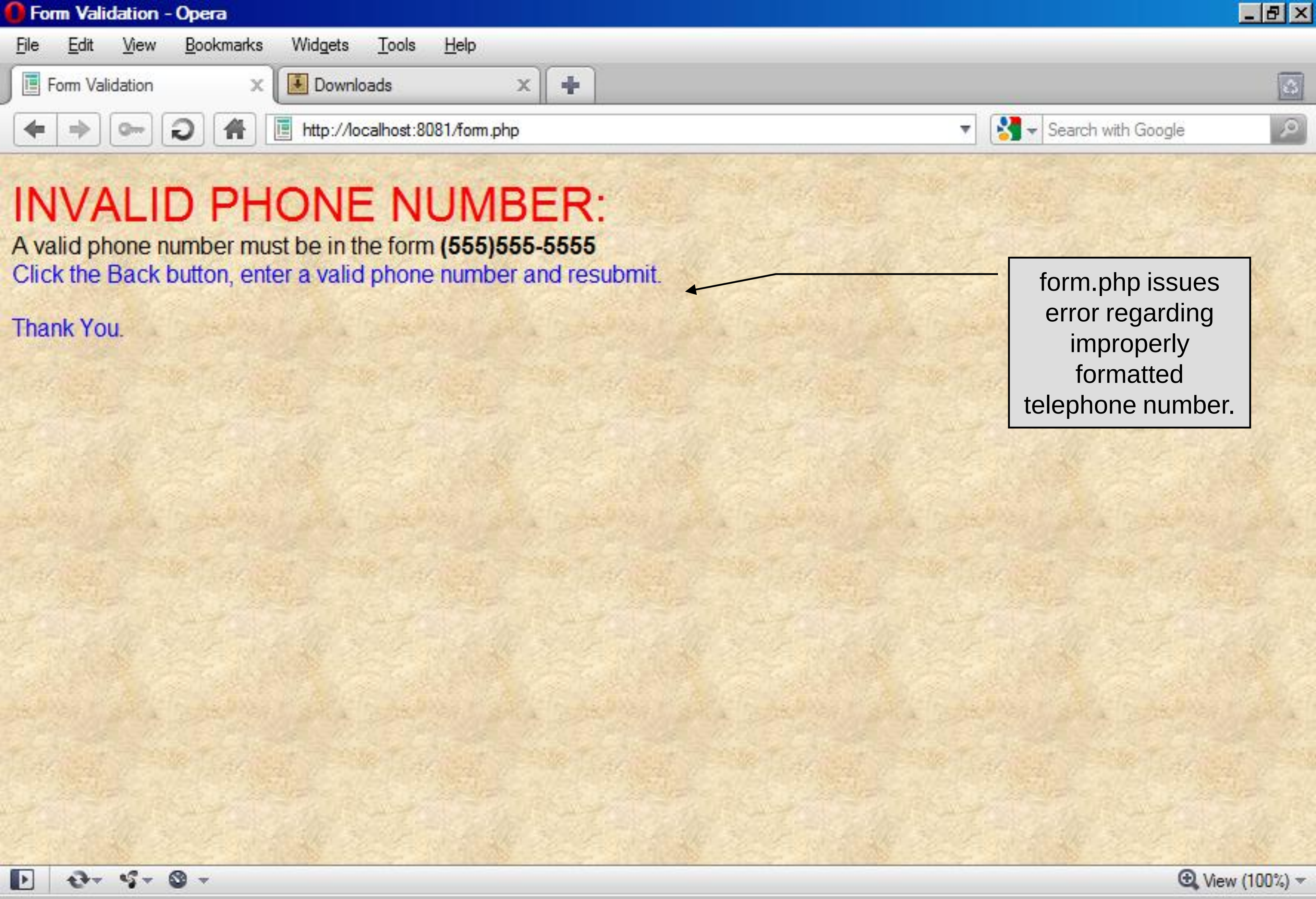
Which operating system are you currently using?

[✓ Windows Vista] Windows XP Windows 2000 Linux Other

Register

http://localhost:8081/form.php

User enters an improperly formatted telephone number in the form.



INVALID PHONE NUMBER:

A valid phone number must be in the form **(555)555-5555**

Click the Back button, enter a valid phone number and resubmit.

Thank You.

form.php issues error regarding improperly formatted telephone number.

How the Form Example Works

- The `action` attribute of the `form` element, indicates that when the user clicks the `Register` button, the form data will be posted to `form.php` for processing.
- Using `method = "post"` appends the form data to the browser request that contains the protocol (i.e., HTTP) and the requested resource's URL. Scripts located on the web server's machine (or accessible through the network) can access the form data sent as part of the request.
- Each of the form's input fields are assigned a unique name. When `Register` is clicked, each field's `name` and `value` are sent to the web server.
- Script `form.php` then accesses the value for each specific field through the global array `$_POST`.

How the Form Example Works (cont.)

- The superglobal arrays are associative arrays predefined by PHP that hold variable acquired from the user input, the environment, or the web server and are accessible in any variable scope.
  - If the information from the form had been submitted via the HTTP method `get`, then the superglobal array `$_GET` would contain the name-value pairs.
- Since the XHTML form and the PHP script “communicate” via the name-value pairs, it is a good idea to make the XHTML object names meaningful so that the PHP script that retrieves the data is easier to understand.

Register_globals

- In PHP versions 4.2 and higher, the directive `register_globals` is set to `Off` by default for security reasons.
- Turning off `register_globals` means that all variables sent from an XHTML form to a PHP document now must be accessed using the appropriate superglobal array (either `$_POST` or `$_GET`).
- When this directive was turned `On`, as was the default case in PHP versions prior to 4.2, PHP created an individual global variable corresponding to each form field.

Validation of Form Generated Data

- The form example illustrates an important concept in the validation of user input. In this case, we simply checked the validity of the format of the telephone number entered by the client user.

- In general, it is crucial to validate information that will be entered into database or used in mailing lists. For example, validation can be used to ensure that credit-card numbers contain the proper number of digits before the numbers are encrypted to a merchant.

- In this case, the form.php script is implementing the business logic or business rules for our application.

Pattern Matching in PHP

- For powerful string comparisons (pattern matching), PHP provides functions `ereg` and `preg_match`, which use regular expressions to search a string for a specified pattern.
- Function `ereg` uses Portable Operating System Interface (POSIX) extended regular expressions.
  - POSIX-extended regular expressions are a standard to which PHP regular expression conform.
- Function `preg_match` provides Perl-compatible regular expressions.
- Perl-compatible regular expressions are more widely used that POSIX regular expressions. PHP's support for Perl-compatible regular expressions eases migration from Perl to PHP. The following examples illustrates these concepts.

expression.php - Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- expression.php -->
<!-- Using regular expressions -->
<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Regular expressions</title>
   </head>
   <body>
      <?php
         $search = "Now is the time";
         print( "Test string is: '$search'<br /><br />" );
         // call function ereg to search for pattern 'Now'  in variable search
         if ( ereg( "Now", $search ) )
            print( "String 'Now' was found.<br />" );

         // search for pattern 'Now' in the beginning of  the string
         if ( ereg( "^Now", $search ) )
            print( "String 'Now' found at beginning of the line.<br />" );

         // search for pattern 'Now' at the end of the string
         if ( ereg( "Now$", $search ) )
            print( "String 'Now' was found at the end of the line.<br />" );
```

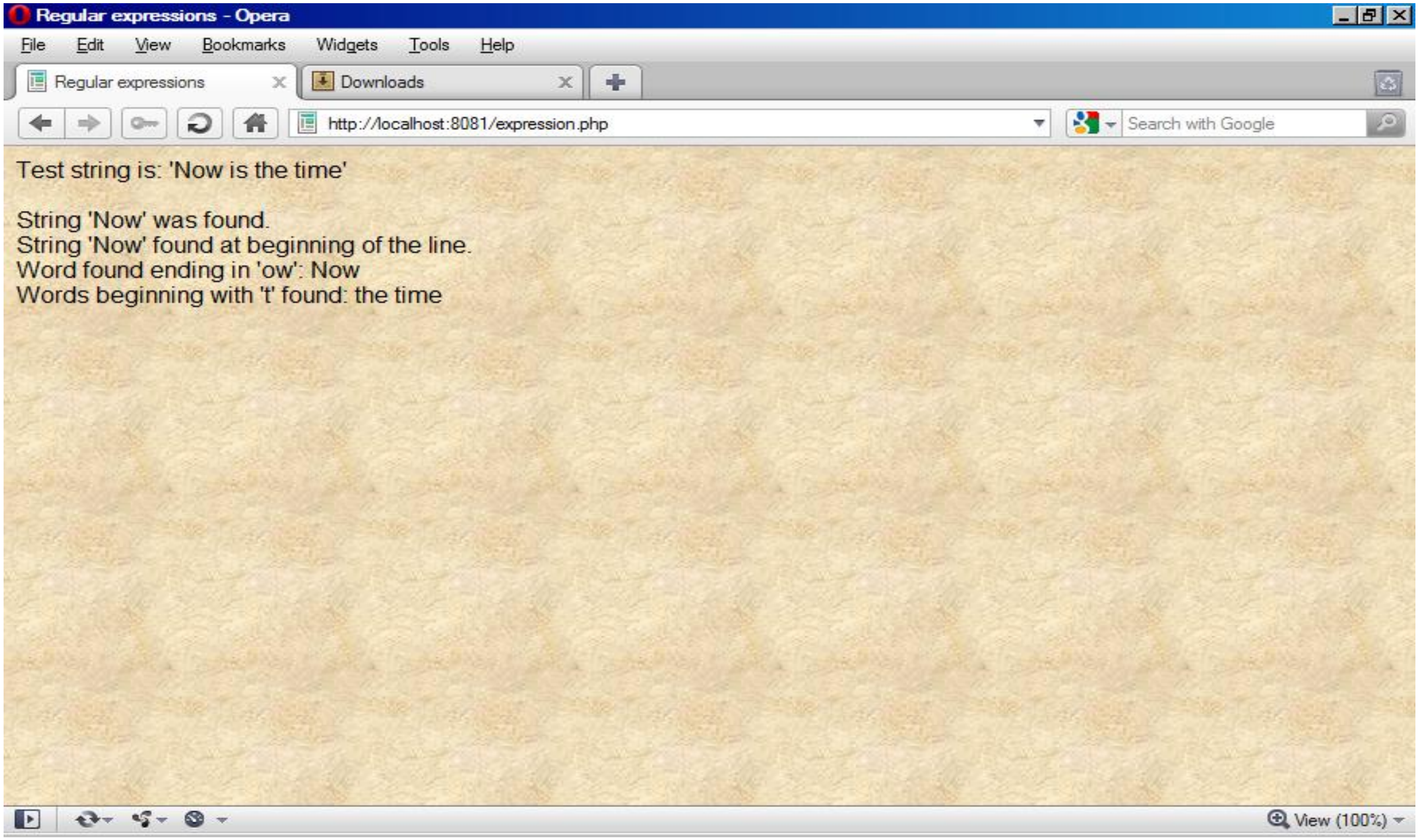
^ matches at beginning of a string

$ matches at end of a string

```
// search for any word ending in ’ow’
if ( ereg( "[[:<:]]([a-zA-Z]*ow)[[:>:]]", $search,
   $match ) )
   print( "Word found ending in 'ow': " .
      $match[ 1 ] . "<br />" );

// search for any words beginning with ’t’
print( "Words beginning with 't' found: ");

while ( eregi( "[[:<:]](t[[:alpha:]]+)[[:>:]]",
   $search, $match ) ) {
   print( $match[ 1 ] . " " );

  // remove the first occurrence of a word beginning
  // with ’t’ to find other instances in the string
  $search = ereg_replace( $match[ 1 ], "", $search );
}

print( "<br />" );
   ?>
  </body>
</html>
```

Uses a regular expression to match a word ending in “ow”.

Output From `expression.php` - Example


Regular expressions - Opera
File Edit View Bookmarks Widgets Tools Help
Regular expressions
Downloads
http://localhost:8081/expression.php
Search with Google
View (100%)

```
Test string is: 'Now is the time'

String 'Now' was found.
String 'Now' found at beginning of the line.
Word found ending in 'ow': Now
Words beginning with 't' found: the time
```

Verifying a Username and Password Using PHP

- It is often the case that a private website is created which is accessible only to certain individuals.
- Implementing privacy generally involves username and password verification.
- In the next example, we'll see an XHTML form that queries a user for a username and password. The fields USERNAME and PASSWORD are posted to the PHP script `verify.php` for verification.
  - For simplicity, data is not encrypted before sending it to the server.
  - For more information on PHP encryption functions visit: http://www.php.net/manual/en/ref.mcrypt.php.

password.html – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- password.html                          -->
<!-- XHTML form sent to password.php for verification -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Verifying a username and a password.</title>
      <style type = "text/css">
         td { background-color: #DDDDDD }
      </style>
   </head>
   <body style = "font-family: arial">
      <p style = "font-size: 18pt">
      <font color=red><B> Welcome to the CNT 4714 High Security WebPage </B></font><HR>
      <p style = "font-size: 13pt">
         Type in your username and password below.
         <br />
         <span style = "color: #0000FF; font-size: 10pt;
            font-weight: bold">
            Note that password will be sent as plain text - encryption not used in this application
         </span>
      </p>
```

password.html – page 2

```
<!-- post form data to password.php -->
<form action = "password.php" method = "post">
   <br />
      <table border = "3" cellspacing = "3"  style = "height: 90px; width: 150px;
      font-size: 10pt" cellpadding = "1">
      <tr>
         <td colspan = "3">  <strong>Username:</strong>  </td>
      </tr>
      <tr>
         <td colspan = "3"> <input size = "40" name = "USERNAME"
               style = "height: 22px; width: 115px" />              </td>
      </tr>
      <tr>
         <td colspan = "3">  <strong>Password:</strong>    </td>
      </tr>
      <tr>
         <td colspan = "3">  <input size = "40" name = "PASSWORD"
               style = "height: 22px; width: 115px"  type = "password" /> <br/></td>
      </tr>
  <tr>
         <td colspan = "1">
            <input type = "submit" name = "Enter"  value = "Enter" style = "height: 23px;
               width: 47px" />        </td>
         <td colspan = "2">  <input type = "submit" name = "NewUser"   value = "New User"
               style = "height: 23px" />
         </td>
      </tr>
   </table>     </form>     <HR>   </body>    </html>
```

password.php – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- password.php      -->
<!-- Searching a database for usernames and passwords. -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <?php
         extract( $_POST );
         // check if user has left USERNAME or PASSWORD field blank
         if ( !$USERNAME || !$PASSWORD ) {
            fieldsBlank();
            die();
         }
         // check if the New User button was clicked
         if ( isset( $NewUser ) ) {
            // open password.txt for writing using append mode
            if ( !( $file = fopen( "password.txt", "a" ) ) ) {

               // print error message and terminate script
               // execution if file cannot be opened
               print( "<title>Error</title></head><body>
                 Could not open password file
                 </body></html>" );
               die();
            }
```

password.php – page 2

```
      // write username and password to file and call function userAdded
      fputs( $file, "$USERNAME,$PASSWORD\n" );
      userAdded( $USERNAME );
   }
   else {

      // if a new user is not being added, open file
      // for reading
      if ( !( $file = fopen( "password.txt",  "r" ) ) ) {
         print( "<title>Error</title></head>
            <body>Could not open password file
            </body></html>" );
         die();
      }

      $userVerified = 0;

      // read each line in file and check username and password
      while ( !feof( $file ) && !$userVerified ) {

         // read line from file
         $line = fgets( $file, 255 );

         // remove newline character from end of line
         $line = chop( $line );

         // split username and password using comma delimited string
         $field = split( ",", $line, 2 );
```

password.php – page 3

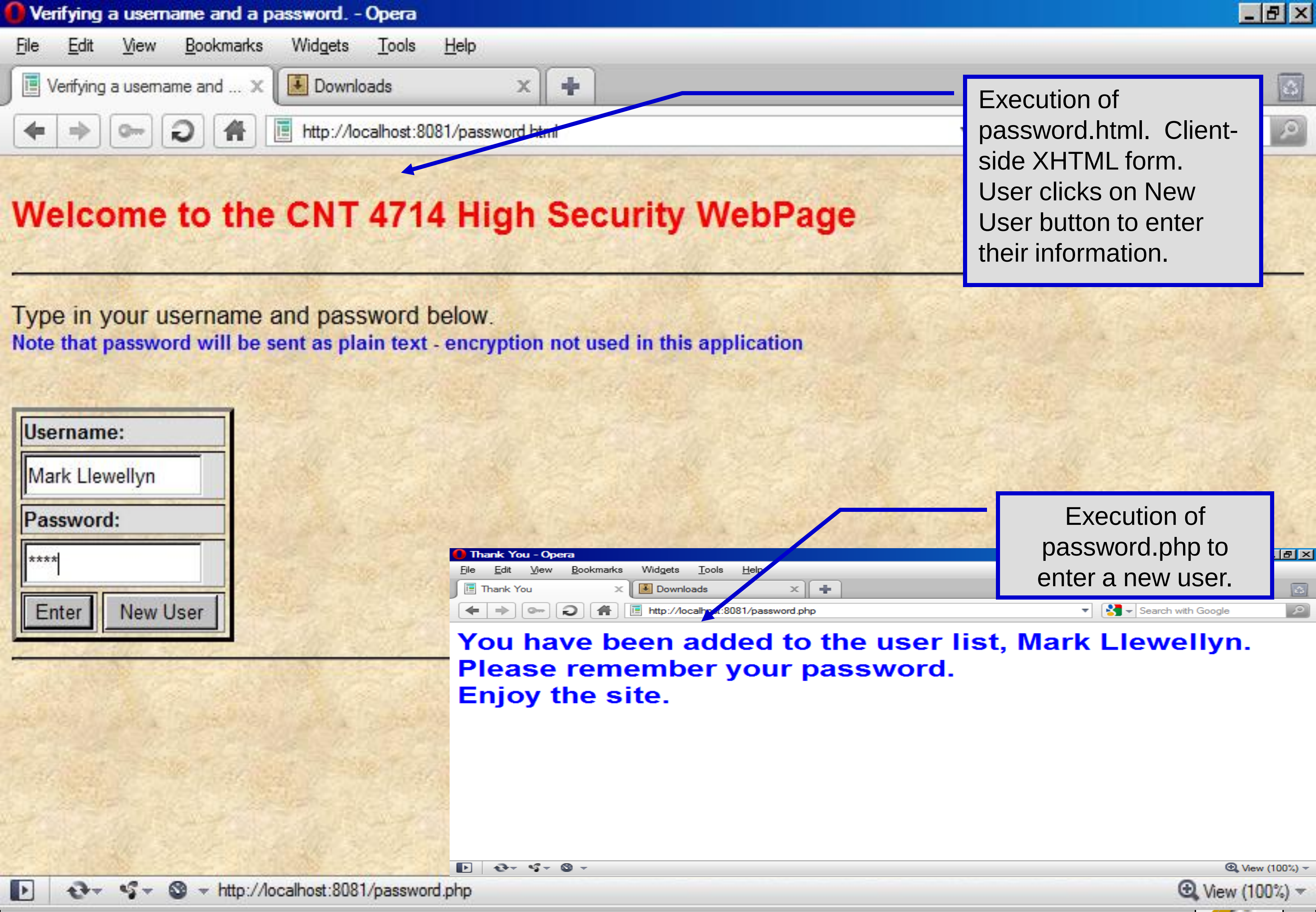
```
      // verify username
      if ( $USERNAME == $field[ 0 ] ) {
         $userVerified = 1;

         // call function checkPassword to verify user's password
         if ( checkPassword( $PASSWORD, $field ) == true )
            accessGranted( $USERNAME );
         else
            wrongPassword();
      }
   }

   // close text file
   fclose( $file );

   // call function accessDenied if username has not been verified
   if ( !$userVerified )
      accessDenied();
}

 // verify user password and return a boolean
function checkPassword( $userpassword, $filedata )
{
   if ( $userpassword == $filedata[ 1 ] )
      return true;
   else
      return false;
}
```

password.php – page 4

```
 // print a message indicating the user has been added
 function userAdded( $name )  {
    print( "<title>Thank You</title></head>
       <body style = \"font-family: arial;
       font-size: 1em; color: blue\">
       <strong>You have been added
       to the user list, $name.  Please remember your password.
       <br />Enjoy the site.</strong>" );
 }

  // print a message indicating permission has been granted
 function accessGranted( $name )  {
    print( "<title>Thank You</title></head>
       <body style = \"font-family: arial;
       font-size: 1em; color: blue\">
       <strong>Permission has been
       granted, $name. <br />
       Enjoy the site.</strong>" );
 }
// print a message indicating password is invalid
 function wrongPassword()   {
    print( "<title>Access Denied</title></head>
       <body style = \"font-family: arial;
       font-size: 1em; color: red\">
       <strong>You entered an invalid
       password.<br />Access has
       been denied.</strong>" );
 }
```

password.php – page 5

```
// print a message indicating access has been denied
      function accessDenied()   {
        print( "<title>Access Denied</title></head>
          <body style = \"font-family: arial;
          font-size: 1em; color: red\">
          <strong>
          You were denied access to this server.
          <br /></strong>" );
      }

       // print a message indicating that fields
       // have been left blank
      function fieldsBlank()   {
        print( "<title>Access Denied</title></head>
          <body style = \"font-family: arial;
          font-size: 1em; color: red\">
          <strong>
          Please fill in all form fields.
          <br /></strong>" );
      }
    ?>
  </body>
</html>
```


Verifying a username and a password. - Opera
File Edit View Bookmarks Widgets Tools Help
Verifying a username and ...
Downloads
http://localhost:8081/password.html
Enter
New User
http://localhost:8081/password.php
View (100%)
Thank You - Opera
Thank You
Downloads
http://localhost:8081/password.php
Search with Google

Welcome to the CNT 4714 High Security WebPage

Type in your username and password below.

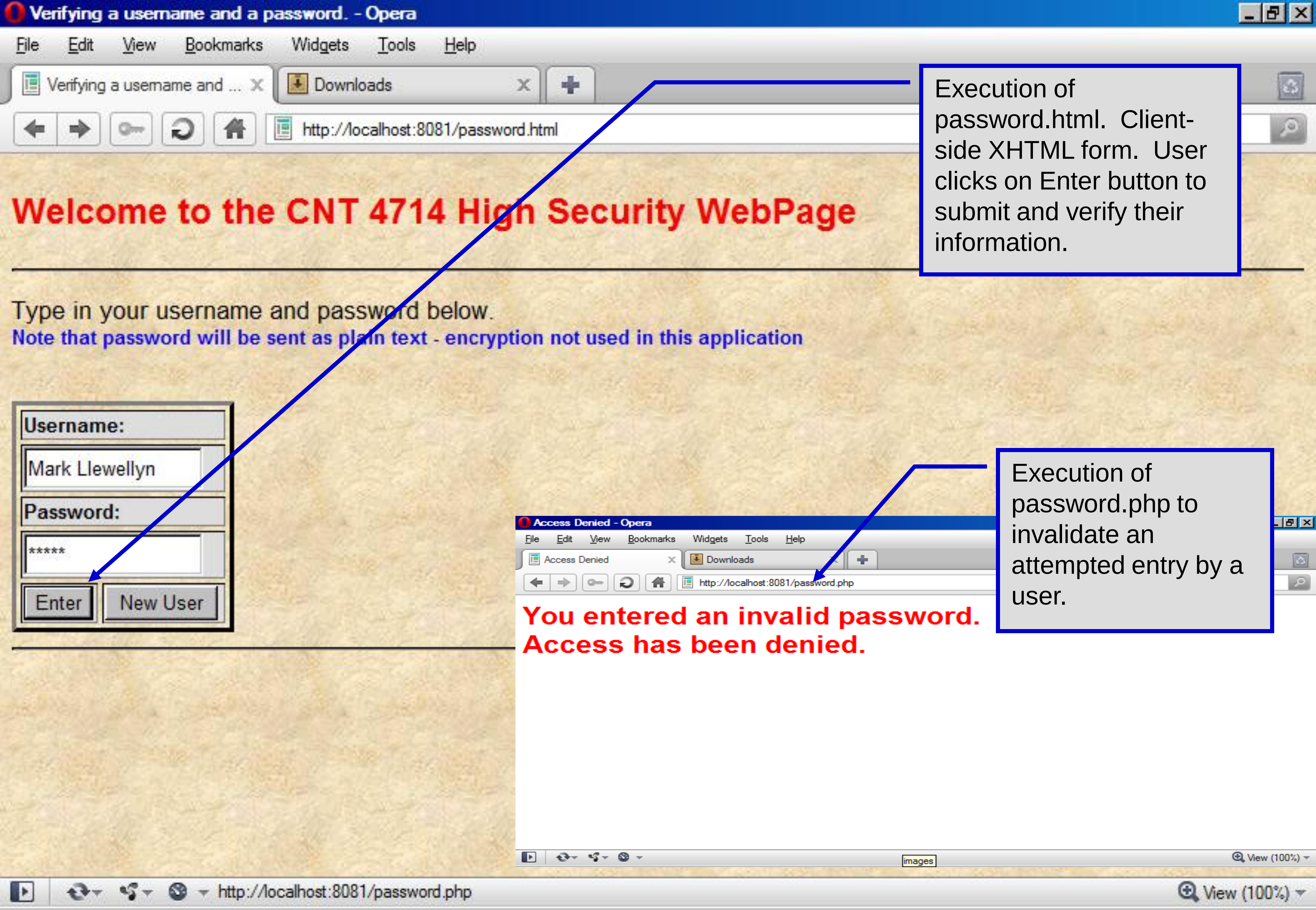
Note that password will be sent as plain text - encryption not used in this application

Username:
Mark Llewellyn
Password:
****

You have been added to the user list, Mark Llewellyn.
Please remember your password.
Enjoy the site.

Execution of password.html. Client-side XHTML form. User clicks on New User button to enter their information.

Execution of password.php to enter a new user.


Verifying a username and a password. - Opera
File Edit View Bookmarks Widgets Tools Help
Verifying a username and ...
Downloads
http://localhost:8081/password.html

Welcome to the CNT 4714 High Security WebPage

Type in your username and password below.
Note that password will be sent as plain text - encryption not used in this application

Username: Mark Llewellyn
Password: *****
Enter | New User

Access Denied - Opera
File Edit View Bookmarks Widgets Tools Help
Access Denied
Downloads
http://localhost:8081/password.php

You entered an invalid password.
Access has been denied.

images
View (100%)
http://localhost:8081/password.php
View (100%)

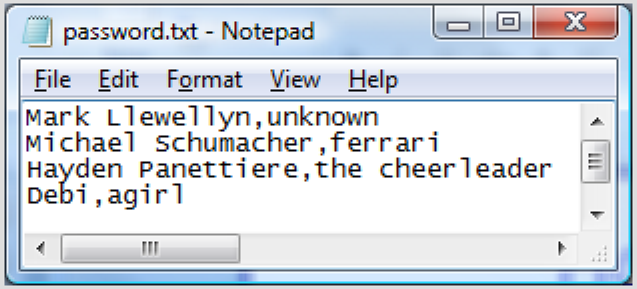
Execution of password.html. Client-side XHTML form. User clicks on Enter button to submit and verify their information.

Execution of password.php to invalidate an attempted entry by a user.

How password.php Works

- The PHP script `password.php` verifies the client's username and password by querying a database. For this example, the "database" of usernames and passwords is just a text file (for simplicity). Existing users are validated against this file, and new users are appended to it.

- Whether we are dealing with a new user is determined by calling function `isset` to test if variable `$NewUser` has been set.



```
Mark Llewellyn,unknown
Michael Schumacher,ferrari
Hayden Panettiere,the cheerleader
Debi,agirl
```

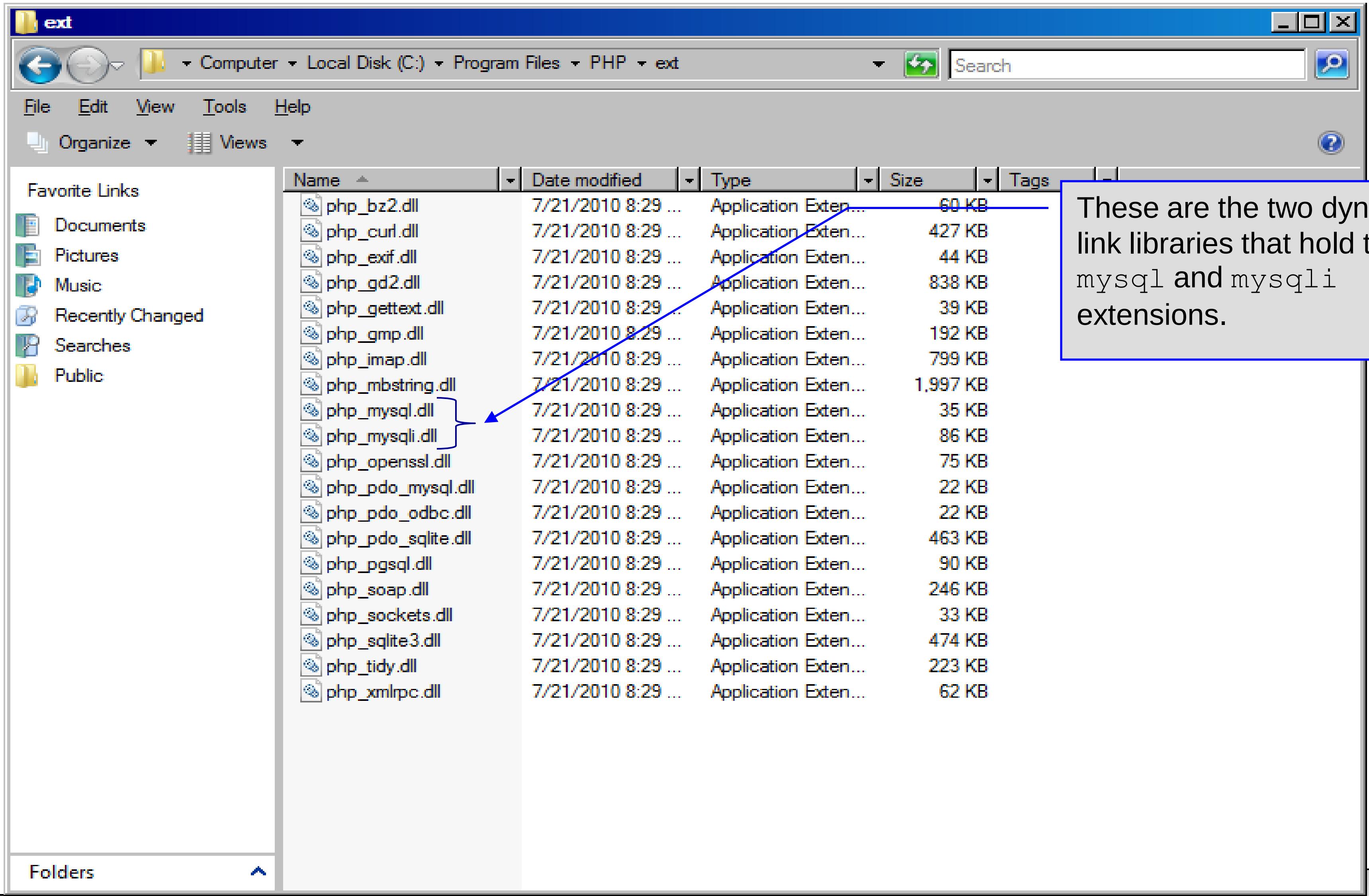
The password.txt "database"

- When the user submits the password.html form to the server, they click either Enter or New User button. After calling function extract, either variable `$NewUser` or `$Enter` is created depending on which button was selected. If `$NewUser` has not been set, we assume the user clicked Enter.

PHP and Database Connectivity

- PHP offers built-in support for a wide variety of database systems from Unix DBM through relational systems such as MySQL to full size commercial systems like Oracle.
- We'll continue to use MySQL as the underlying database system so that you can easily compare the work we've done with MySQL using Java servlets and JSPs.
- PHP 5.3.3 should already be setup to accept MySQL. You can verify this as shown on page 34.
  - Versions of MySQL greater than 4.1.0 use MySQLi extensions.
  - Versions of MySQL less than 4.1.0 use MySQL extensions.

PHP and Database Connectivity (cont.)

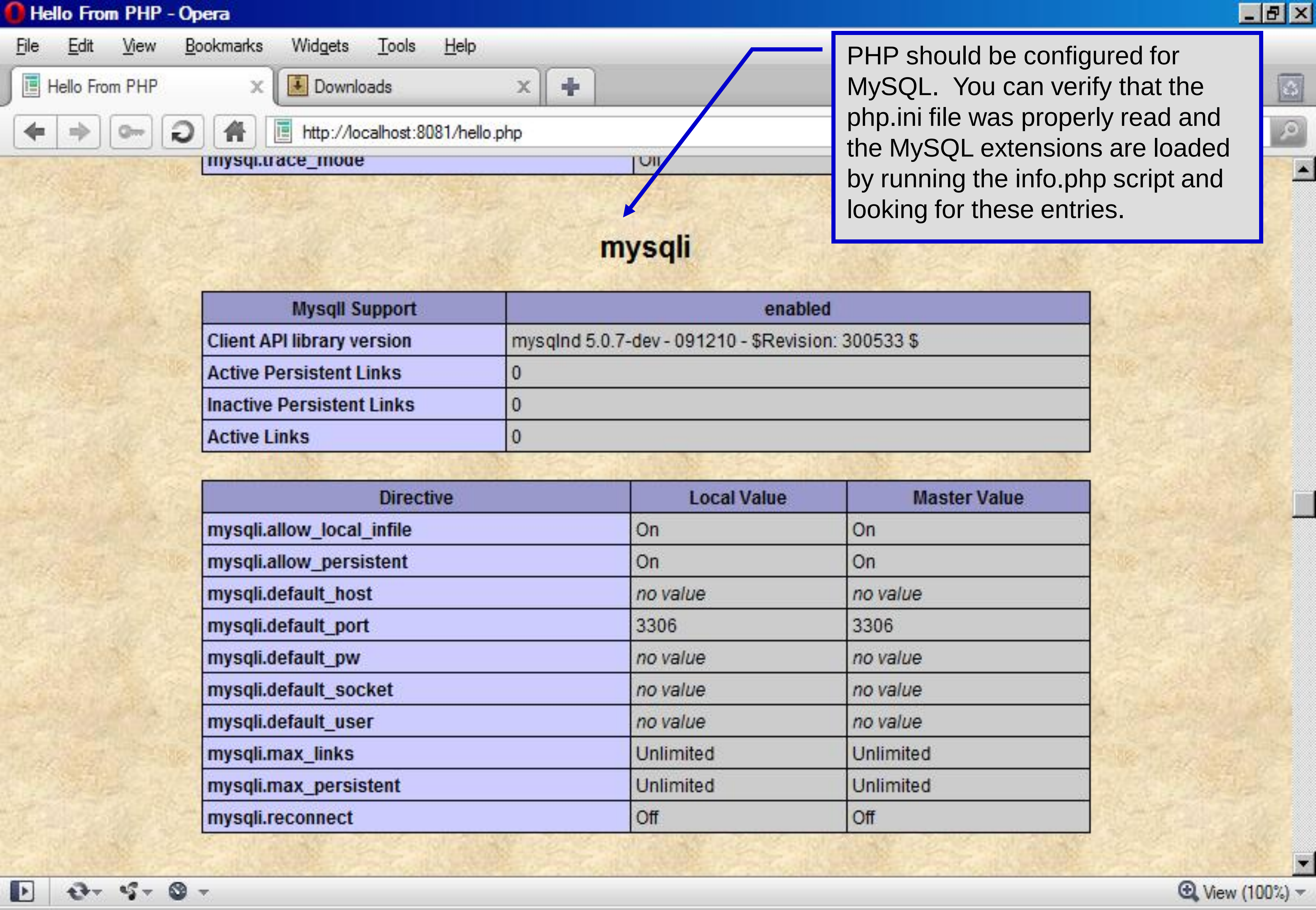


ext

Computer ▸ Local Disk (C:) ▸ Program Files ▸ PHP ▸ ext

Name	Date modified	Type	Size	Tags
php_bz2.dll	7/21/2010 8:29 ...	Application Exten...	60 KB	
php_curl.dll	7/21/2010 8:29 ...	Application Exten...	427 KB	
php_exif.dll	7/21/2010 8:29 ...	Application Exten...	44 KB	
php_gd2.dll	7/21/2010 8:29 ...	Application Exten...	838 KB	
php_gettext.dll	7/21/2010 8:29 ...	Application Exten...	39 KB	
php_gmp.dll	7/21/2010 8:29 ...	Application Exten...	192 KB	
php_imap.dll	7/21/2010 8:29 ...	Application Exten...	799 KB	
php_mbstring.dll	7/21/2010 8:29 ...	Application Exten...	1,997 KB	
php_mysql.dll	7/21/2010 8:29 ...	Application Exten...	35 KB	
php_mysqli.dll	7/21/2010 8:29 ...	Application Exten...	86 KB	
php_openssl.dll	7/21/2010 8:29 ...	Application Exten...	75 KB	
php_pdo_mysql.dll	7/21/2010 8:29 ...	Application Exten...	22 KB	
php_pdo_odbc.dll	7/21/2010 8:29 ...	Application Exten...	22 KB	
php_pdo_sqlite.dll	7/21/2010 8:29 ...	Application Exten...	463 KB	
php_pgsql.dll	7/21/2010 8:29 ...	Application Exten...	90 KB	
php_soap.dll	7/21/2010 8:29 ...	Application Exten...	246 KB	
php_sockets.dll	7/21/2010 8:29 ...	Application Exten...	33 KB	
php_sqlite3.dll	7/21/2010 8:29 ...	Application Exten...	474 KB	
php_tidy.dll	7/21/2010 8:29 ...	Application Exten...	223 KB	
php_xmlrpc.dll	7/21/2010 8:29 ...	Application Exten...	62 KB	

These are the two dynamic link libraries that hold the `mysql` and `mysqli` extensions.


Hello From PHP - Opera
File Edit View Bookmarks Widgets Tools Help
Hello From PHP
Downloads
http://localhost:8081/hello.php
View (100%)

PHP should be configured for MySQL. You can verify that the php.ini file was properly read and the MySQL extensions are loaded by running the info.php script and looking for these entries.

mysqli.trace_mode | Off

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.7-dev - 091210 - $Revision: 300533 $
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host	*no value*	*no value*
mysqli.default_port	3306	3306
mysqli.default_pw	*no value*	*no value*
mysqli.default_socket	*no value*	*no value*
mysqli.default_user	*no value*	*no value*
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off

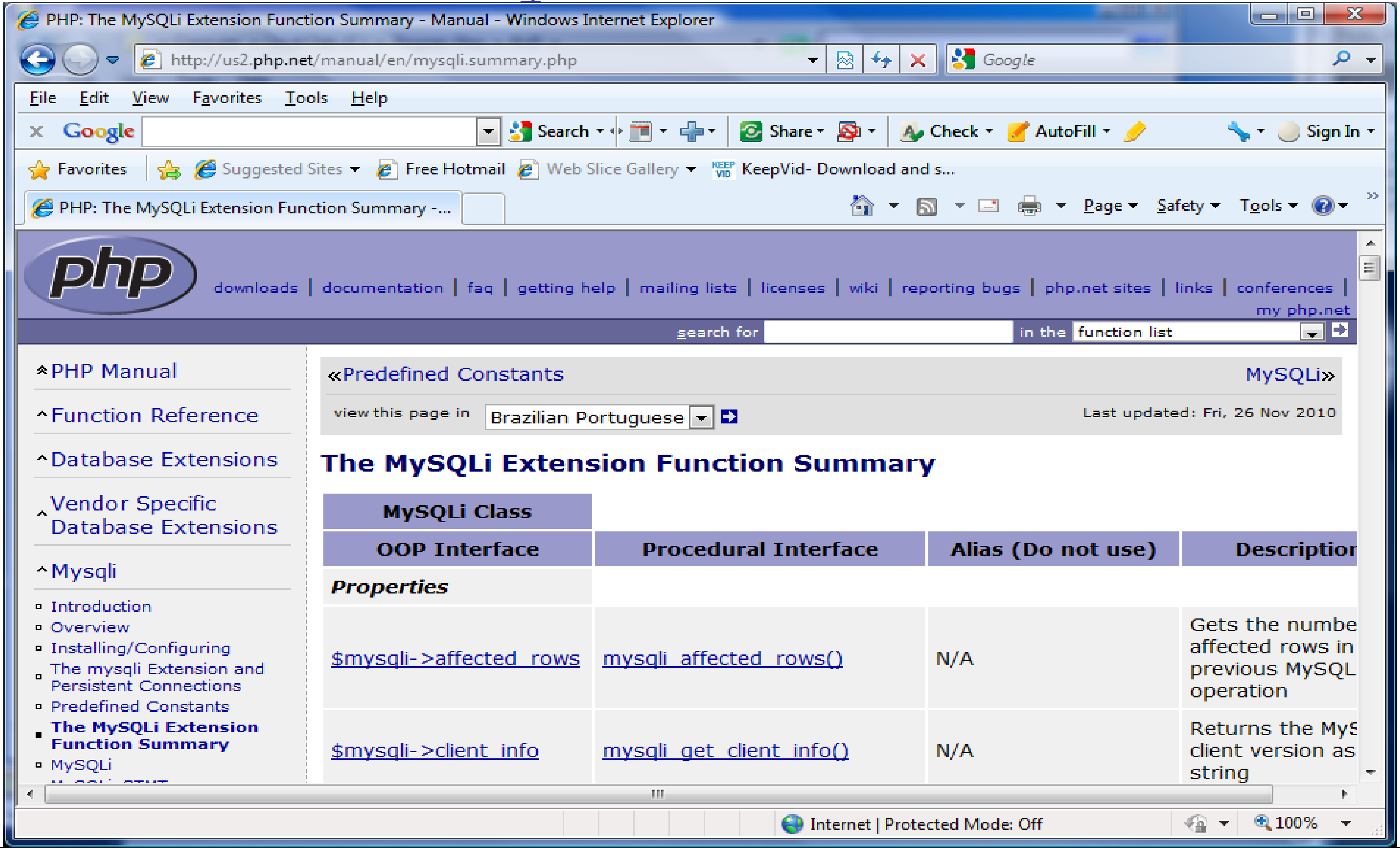
PHP and Database Connectivity (cont.)

- PHP contains a fairly extensive set of commands that can be used to access and manipulate MySQL databases.
- A very brief listing of some of these commands appears on the next page.
- For a complete listing see:

  http://us2.php.net/manual/en/print/ref.mysql.php.

  http://us2.php.net/manual/en/print/ref.mysqli.php.

PHP MySQLi Extensions


PHP: The MySQLi Extension Function Summary - Manual - Windows Internet Explorer
http://us2.php.net/manual/en/mysqli.summary.php
File Edit View Favorites Tools Help
Search Share Check AutoFill Sign In
Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid- Download and s...
Page Safety Tools
downloads | documentation | faq | getting help | mailing lists | licenses | wiki | reporting bugs | php.net sites | links | conferences | my php.net
search for in the function list
PHP Manual
Function Reference
Database Extensions
Vendor Specific Database Extensions
Mysqli
Introduction
Overview
Installing/Configuring
The mysqli Extension and Persistent Connections
Predefined Constants
The MySQLi Extension Function Summary
MySQLi
«Predefined Constants
MySQLi»
view this page in Brazilian Portuguese
Last updated: Fri, 26 Nov 2010
Internet | Protected Mode: Off
100%

The MySQLi Extension Function Summary

MySQLi Class			
OOP Interface	Procedural Interface	Alias (Do not use)	Descriptior
Properties			
$mysqli->affected_rows	mysqli_affected_rows()	N/A	Gets the numbe affected rows in previous MySQL operation
$mysqli->client_info	mysqli_get_client_info()	N/A	Returns the MyS client version as string

PHP and Database Connectivity (cont.)

- Now that you have PHP set to accept MySQL extensions, let’s connect to the bike database that we used for examples with Java servlets and JSPs.
- The following example is a simple database connection process in PHP where the client interacts with the database from an XHTML form that simply asks them to select which attributes from the bikes table that they would like to display. This is done through the `data.html` file.
- When the client clicks the submit query button, the `database.php` script executes by connecting to the database, posting the query, retrieving the results, and displaying them to the client.

data.html

Client side

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- data.html     -->
<!-- Querying a MySQL Database From a PHP Script -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>      <title>Sample Database Query From PHP</title>   </head>
   <body style = "background-color: #545454" background=image1.jpg >
      <h2 style = "font-family: arial color: blue"> Querying a MySQL database from a PHP Script. </h2>
      <form method = "post" action = "database.php">
         <p>Select a field to display:
            <!-- add a select box containing options for SELECT query   -->
            <select name = "select">
               <option selected = "selected">*</option>
               <option>bikename</option>
               <option>size</option>
               <option>color</option>
               <option>cost</option>
               <option>purchased</option>
               <option>mileage</option>
            </select>
         </p>
         <input type = "submit" value = "Send Query"   style = "background-color: blue;
            color: yellow; font-weight: bold" />
      </form>
   </body>   </html>
```

database.php

Server side

Page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- database.php       -->
<!-- Program to query a database and send results to the client.     -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>      <title>Database Search Results</title>   </head>

   <body style = "font-family: arial, sans-serif"
      style = "background-color: #4A766E" background=image1.jpg link=blue vlink=blue>
      <?php

         extract( $_POST );

         // build SELECT query
         $query = "SELECT " . $select . " FROM bikes";

         // Connect to MySQL
         if ( !( $database = mysqli_connect( "localhost",
            "root", "root", bikedb ) ) )
            die( "Could not connect to database" );
```

Default query is to select the attributes chosen by the client for use in a SELECT query.

Connect to MySQL database. URL, username, password, and database all specified.

```
   // query bikedb database
   if ( !( $result = mysql_query( $database, $query ) ) ) {
      print( "Could not execute query! <br />" );
      die( mysql_error() );
   }
?>

<h3 style = "color: blue">
Database Search Results</h3>
<table border = "1" cellpadding = "3" cellspacing = "3"
   style = "background-color: #00FFFF">  <!-- ADD8E6  -->

   <?php
            // fetch meta-data
            $metadata = mysqli_fetch_fields( $result);
            print("<tr>");
            for ($i=0; $i<count($metadata); $i++){
                              print("<td>");
                              printf("%s",$metadata[$i]->name);
                              print("</td>");
            }
            print("</tr>");
```

database.php

Server side

Page 2

Get metadata for the query

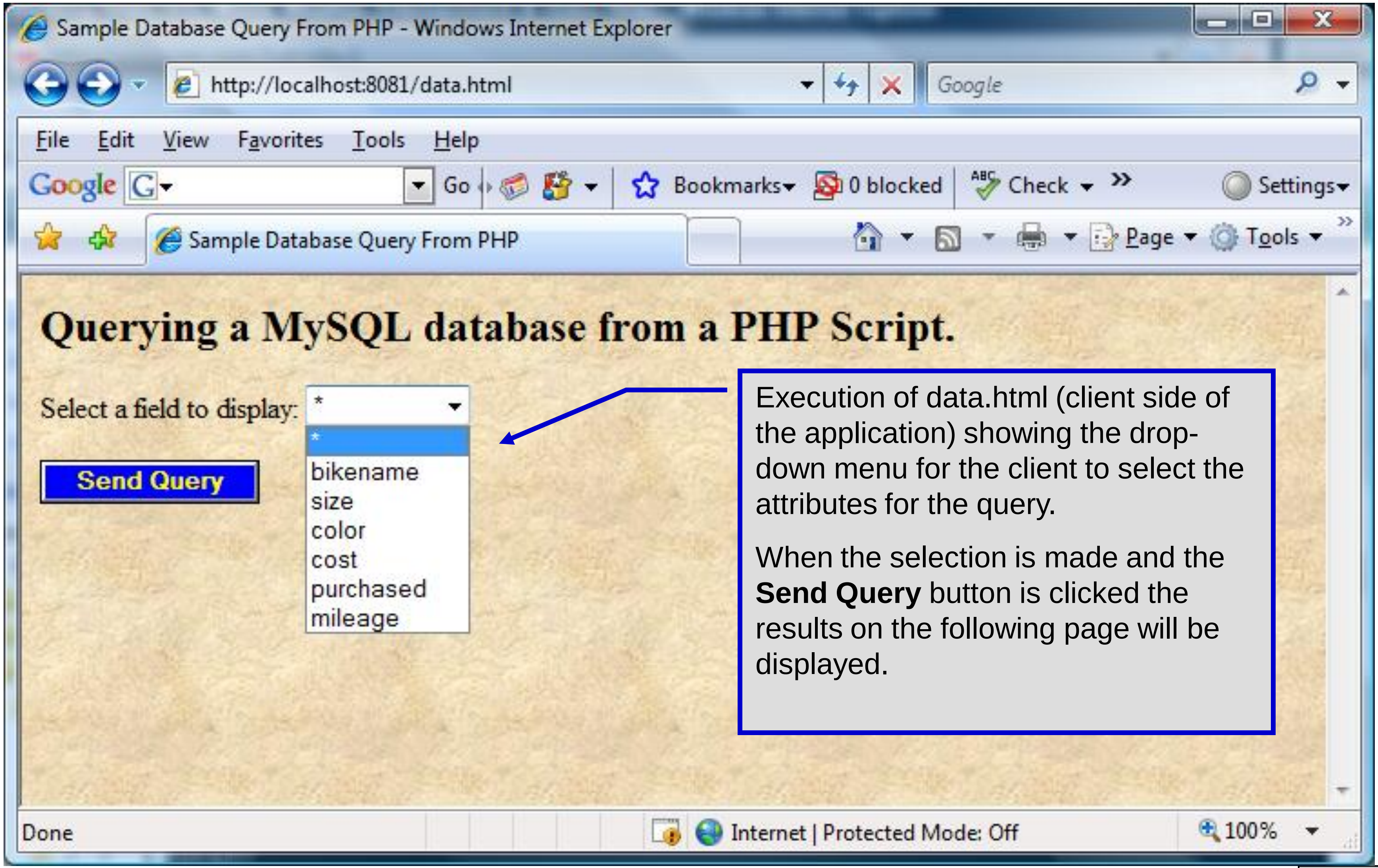
Display metadata in the top row of the table database.php

Server side

Page 3

```
        // fetch each record in result set
           for ( $counter = 0;
              $row = mysql_fetch_row( $result );
              $counter++ ){
              // build table to display results
              print( "<tr>" );
              foreach ( $row as $key => $value )
                 print( "<td>$value</td>" );
              print( "</tr>" );
           }
           mysql_close( $database );
        ?>
     </table>
     <br />Your search yielded <strong>
           <?php print( "$counter" ) ?> results.<br /><br /></strong>
           <h5>Please email comments to
           <a href = "mailto:markl@cs.ucf.edu">
                                markl@cs.ucf.edu
            </a>
          </h5>
  </body></html>
```

Execution of data.html – Client side


Sample Database Query From PHP - Windows Internet Explorer
http://localhost:8081/data.html
File Edit View Favorites Tools Help
Sample Database Query From PHP
Done
Internet | Protected Mode: Off
100%

Querying a MySQL database from a PHP Script.

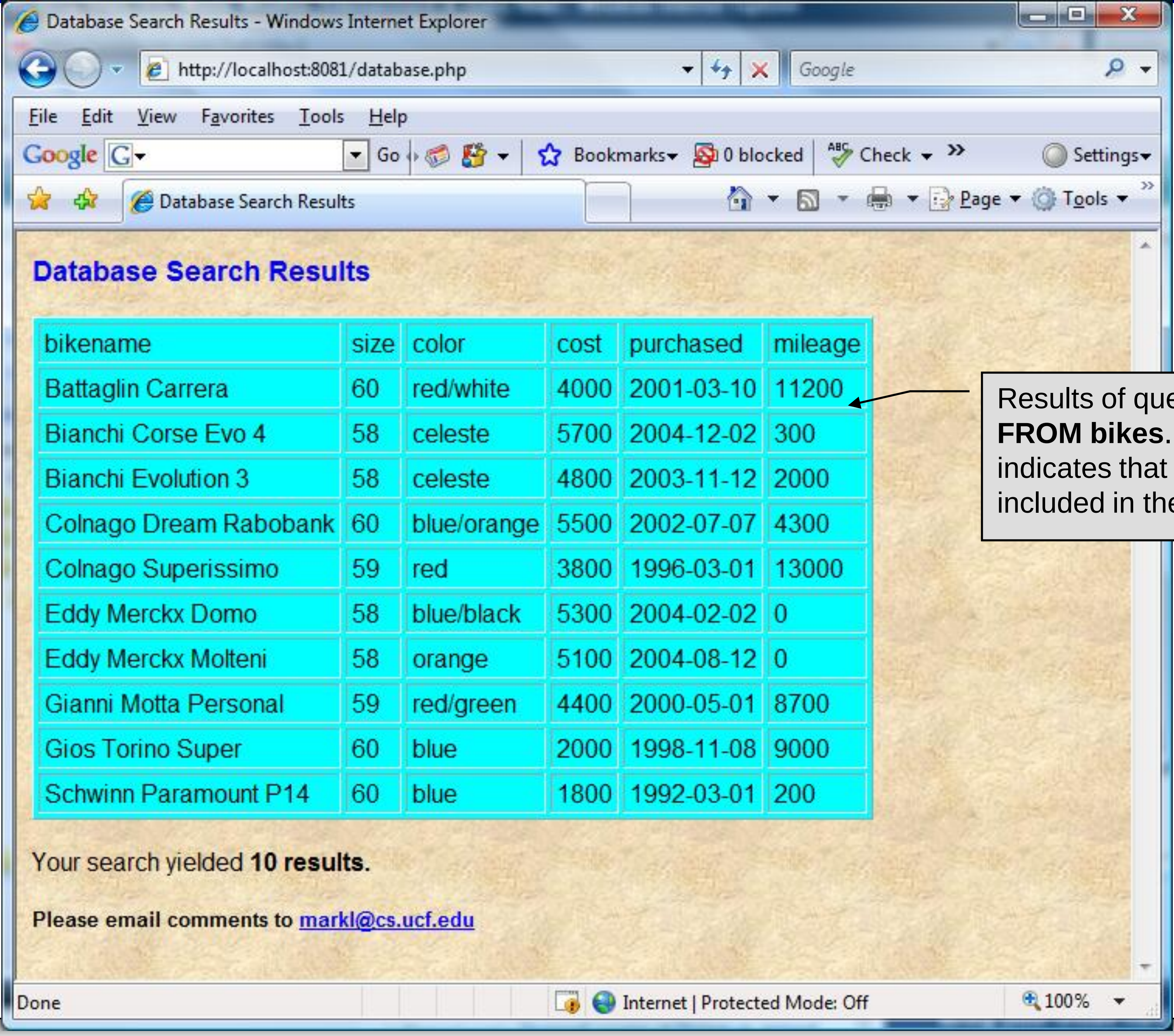
Select a field to display: *

- *
- bikename
- size
- color
- cost
- purchased
- mileage

Send Query

Execution of data.html (client side of the application) showing the drop-down menu for the client to select the attributes for the query.

When the selection is made and the **Send Query** button is clicked the results on the following page will be displayed.



Database Search Results - Windows Internet Explorer
http://localhost:8081/database.php
File Edit View Favorites Tools Help
Database Search Results
Done
Internet | Protected Mode: Off
100%

Database Search Results

bikename	size	color	cost	purchased	mileage
Battaglin Carrera	60	red/white	4000	2001-03-10	11200
Bianchi Corse Evo 4	58	celeste	5700	2004-12-02	300
Bianchi Evolution 3	58	celeste	4800	2003-11-12	2000
Colnago Dream Rabobank	60	blue/orange	5500	2002-07-07	4300
Colnago Superissimo	59	red	3800	1996-03-01	13000
Eddy Merckx Domo	58	blue/black	5300	2004-02-02	0
Eddy Merckx Molteni	58	orange	5100	2004-08-12	0
Gianni Motta Personal	59	red/green	4400	2000-05-01	8700
Gios Torino Super	60	blue	2000	1998-11-08	9000
Schwinn Paramount P14	60	blue	1800	1992-03-01	200

Your search yielded **10 results.**

Please email comments to markl@cs.ucf.edu

Results of query **SELECT * FROM bikes**. Display indicates that 10 rows were included in the result.

Cookies

- A cookie is a text file that a Web site stores on a client's computer to maintain information about the client during and between browsing sessions.

- A Web site can store a cookie on a client's computer to record user preferences and other information that the Web site can retrieve during the client's subsequent visits. For example, many Web sites use cookies to store client's zipcodes. The Web site can retrieve the zipcode from the cookie and provide weather reports and news updates tailored to the user's region.

- Web sites also use cookies to track information about client activity. Analysis of information collected via cookies can reveal the popularity of Web sites or products.

Cookies (cont.)

- Marketers use cookies to determine the effectiveness of advertising campaigns.
- Web sites store cookies on users' hard drives, which raises issues regarding security and privacy. Web sites should not store critical information, such as credit-card numbers or passwords, in cookies, because cookies are just text files that anyone can read.
- Several cookie features address security and privacy concerns. A server can access only the cookies that it has placed on the client.
- A cookies has an expiration date, after which the Web browser deletes it.

Cookies (cont.)

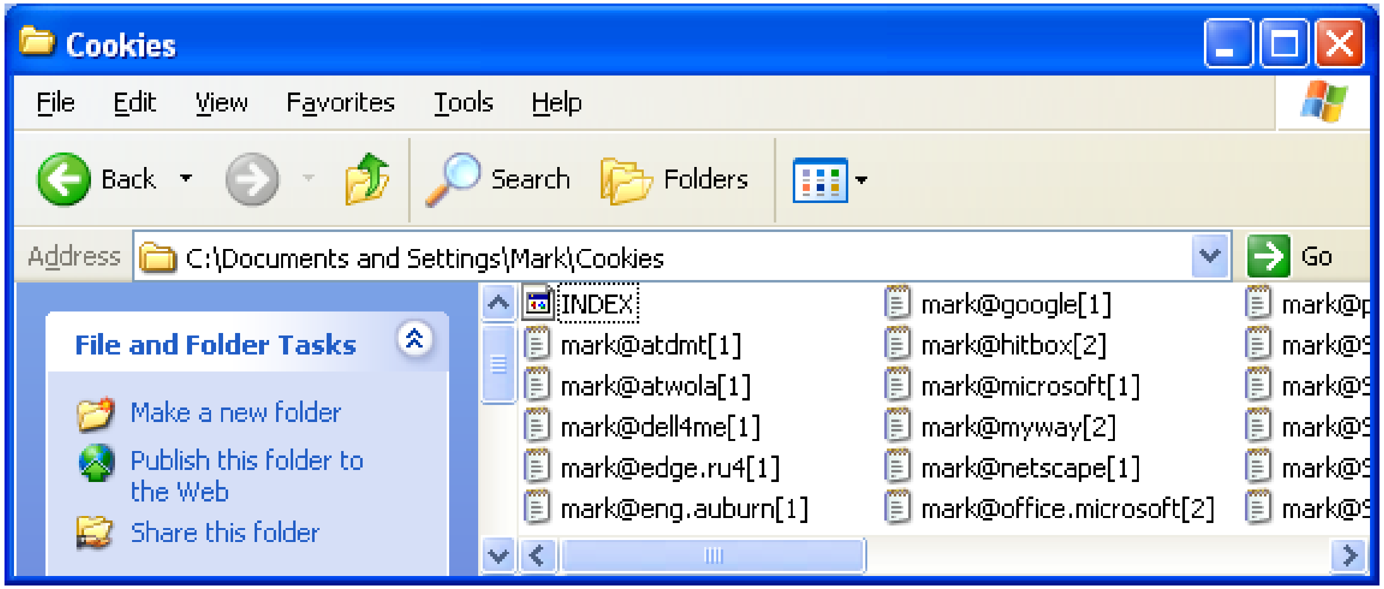
- Users who are concerned about the privacy and security implications of cookies can disable them in their Web browsers. However, the disabling of cookies can make it impossible for the user to interact with Web sites that rely on cookies to function properly.

- Information stored in the cookie is sent to the Web server from which it originated whenever the user requests a Web page from that particular server. The Web server can send the client XHTML output that reflects the preferences or information that is stored in the cookie.

- The location of the cookie file varies from browser to browser. Internet Explorer places cookies in the Cookies directory located at `C:\Documents and Settings\...\Cookies`

Cookies (cont.)

- After a cookie is created, a text file is added to this directory. While the name of the file will vary from user to user a typical example is shown below.



Cookies
File Edit View Favorites Tools Help
Back
Search
Folders
Address
C:\Documents and Settings\Mark\Cookies
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder

INDEX
mark@atdmt[1]
mark@atwola[1]
mark@dell4me[1]
mark@edge.ru4[1]
mark@eng.auburn[1]
mark@google[1]
mark@hitbox[2]
mark@microsoft[1]
mark@myway[2]
mark@netscape[1]
mark@office.microsoft[2]
mark@p
mark@s
mark@s
mark@s
mark@s
mark@s

- The contents of a cookie are shown on page 74.

Cookies (cont.)

- Now let’s create the code necessary to create our own cookie.
- In this example, a PHP script is invoked from a client-side HTML document. The HTML document creates a form for the user to enter the information that will be stored in the cookie. (Often the information that is stored in a cookie will be extracted from several different areas and may involved tracking the client’s actions at the Web site.)
- Once the user has entered their information, when they click the Write Cookie button, the `cookies.php` script executes.
- The XHTML document and the PHP script are shown on the next pages. The XHTML document `cookies.html` is on page 36 and the PHP script `cookies.php` appears on page 37.

cookies.html – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- cookies.html -->
<!-- Writing a Cookie         -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head> <title>Writing a cookie to the client computer</title>   </head>

   <body style = "font-family: arial, sans-serif;
      background-color: #856363"  background=image1.jpg>
      <h2>Click Write Cookie to save your cookie data.</h2>

      <form method = "post" action = "cookies.php" style = "font-size: 10pt"
                  background-color: #856363">
         <strong>Name:</strong><br />
         <input type = "text" name = "NAME" /><br />
         <strong>Height:</strong><br />
         <input type = "text" name = "HEIGHT" /><br />
         <strong>Favorite Color:</strong><br />
         <input type = "text" name = "COLOR" /><br />
         <p>
            <input type = "submit" value = "Write Cookie"  style = "background-color: #0000FF;
                 color: yellow;  font-weight: bold" /></p>
      </form>
   </body>  </html>
```

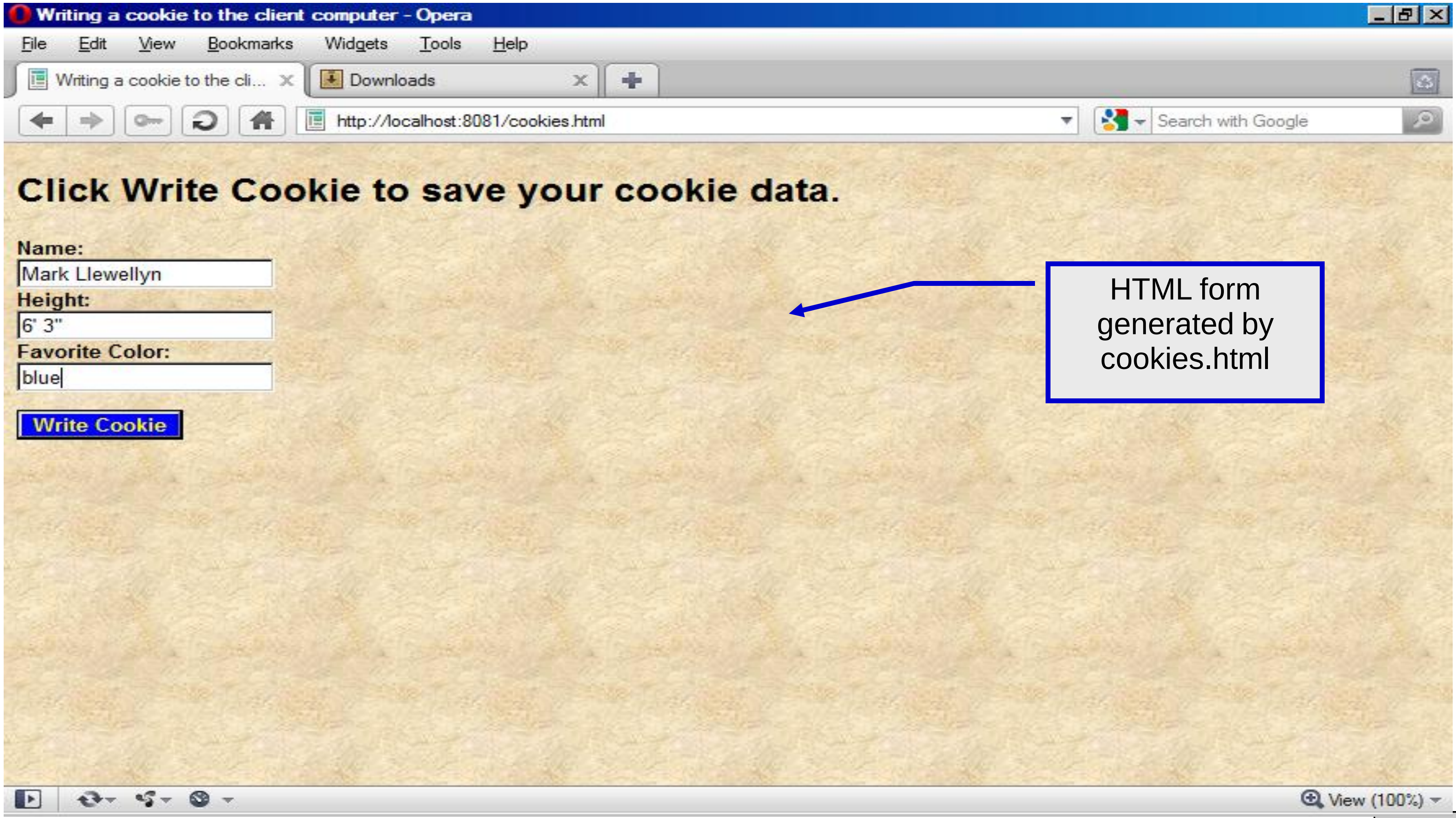
cookies.php – page 1

```
<?php
   // cookies.php
   // Program to write a cookie to a client's machine
    extract( $_POST );
   // write each form field’s value to a cookie and set the
   // cookie’s expiration date
   setcookie( "Name", $NAME, time() + 60 * 60 * 24 * 5 );
   setcookie( "Height", $HEIGHT, time() + 60 * 60 * 24 * 5 );
   setcookie( "Color", $COLOR, time() + 60 * 60 * 24 * 5 );
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>      <title>Cookie Saved</title>   </head>
   <body style = "font-family: arial, sans-serif", background=image1.jpg>
      <p><b>The cookie has been set with the following data:</b></p>
      <!-- print each form field’s value -->
      <br /><span style = "color: blue">Name:</span>
         <?php print( $NAME ) ?><br />
      <span style = "color: blue">Height:</span>
         <?php print( $HEIGHT ) ?><br />
      <span style = "color: blue">Favorite Color:</span>
      <span style = "color: <?php print( "$COLOR\">$COLOR" ) ?>
      </span><br />
      <p>Click <a href = "readCookies.php">here</a>  to read the saved cookie.</p>
   </body>  </html>
```

Function setcookie sets the cookies to the values passed from the cookies.html form. Function setcookie prints XHTML header information and therefore it needs to be called before any other XHTML (including comments) is printed.

The third argument to setcookie is optional and indicates the expiration date of the cookie. In this case it is set to expire 5 days from the current time. Function time returns the current time and then we add to this the number of seconds after which the cookie is to expire.

Cookies (cont.)


Writing a cookie to the client computer - Opera
File Edit View Bookmarks Widgets Tools Help
Writing a cookie to the cli...
Downloads
http://localhost:8081/cookies.html
Search with Google
View (100%)

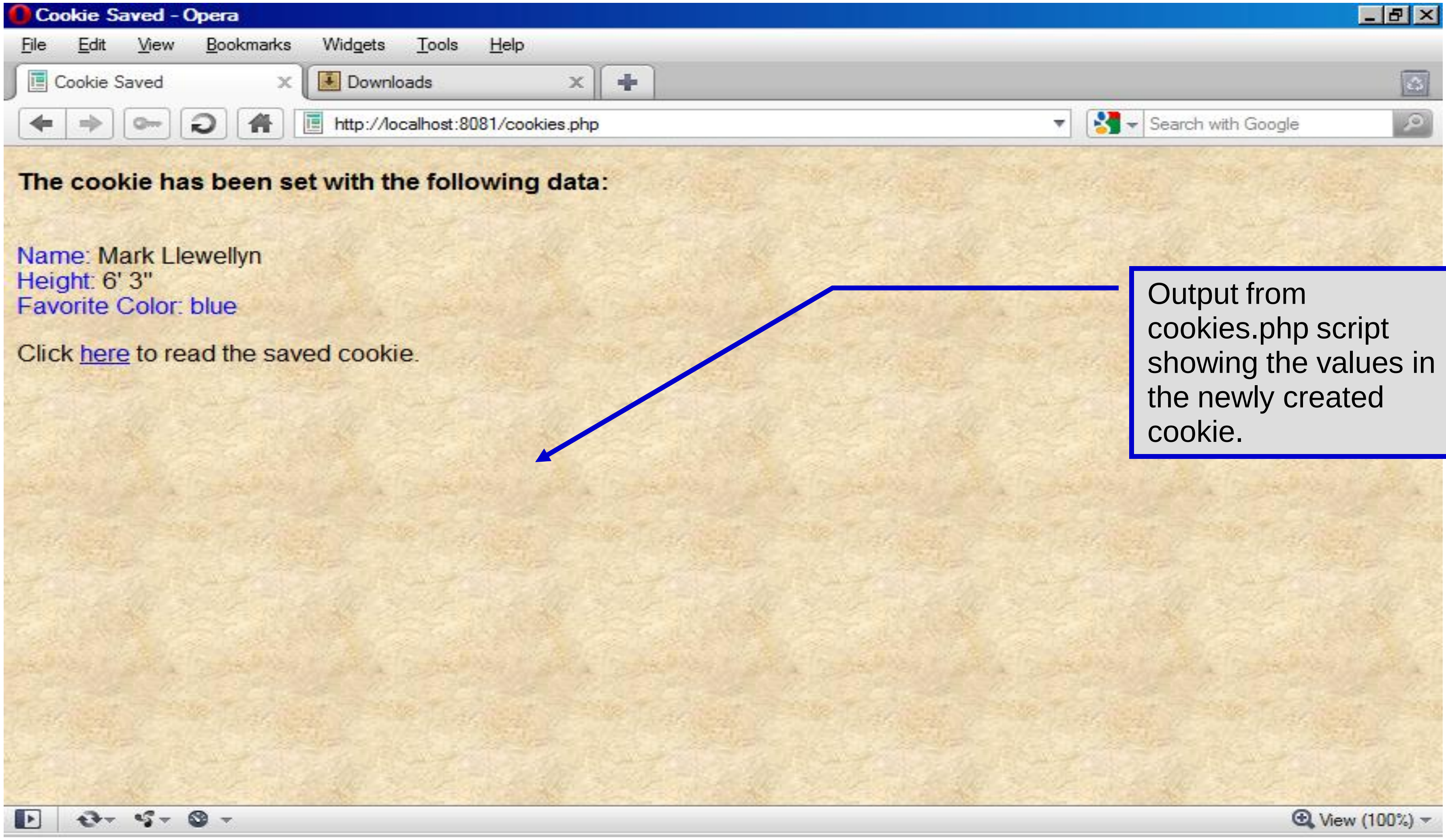
Click Write Cookie to save your cookie data.

Name:
Mark Llewellyn
Height:
6' 3"
Favorite Color:
blue

Write Cookie

HTML form generated by cookies.html

Cookies (cont.)


Cookie Saved - Opera
File Edit View Bookmarks Widgets Tools Help
Cookie Saved
Downloads
http://localhost:8081/cookies.php
Search with Google
View (100%)

The cookie has been set with the following data:

Name: Mark Llewellyn
Height: 6' 3"
Favorite Color: blue

Click here to read the saved cookie.

Output from cookies.php script showing the values in the newly created cookie.

Cookies (cont.)

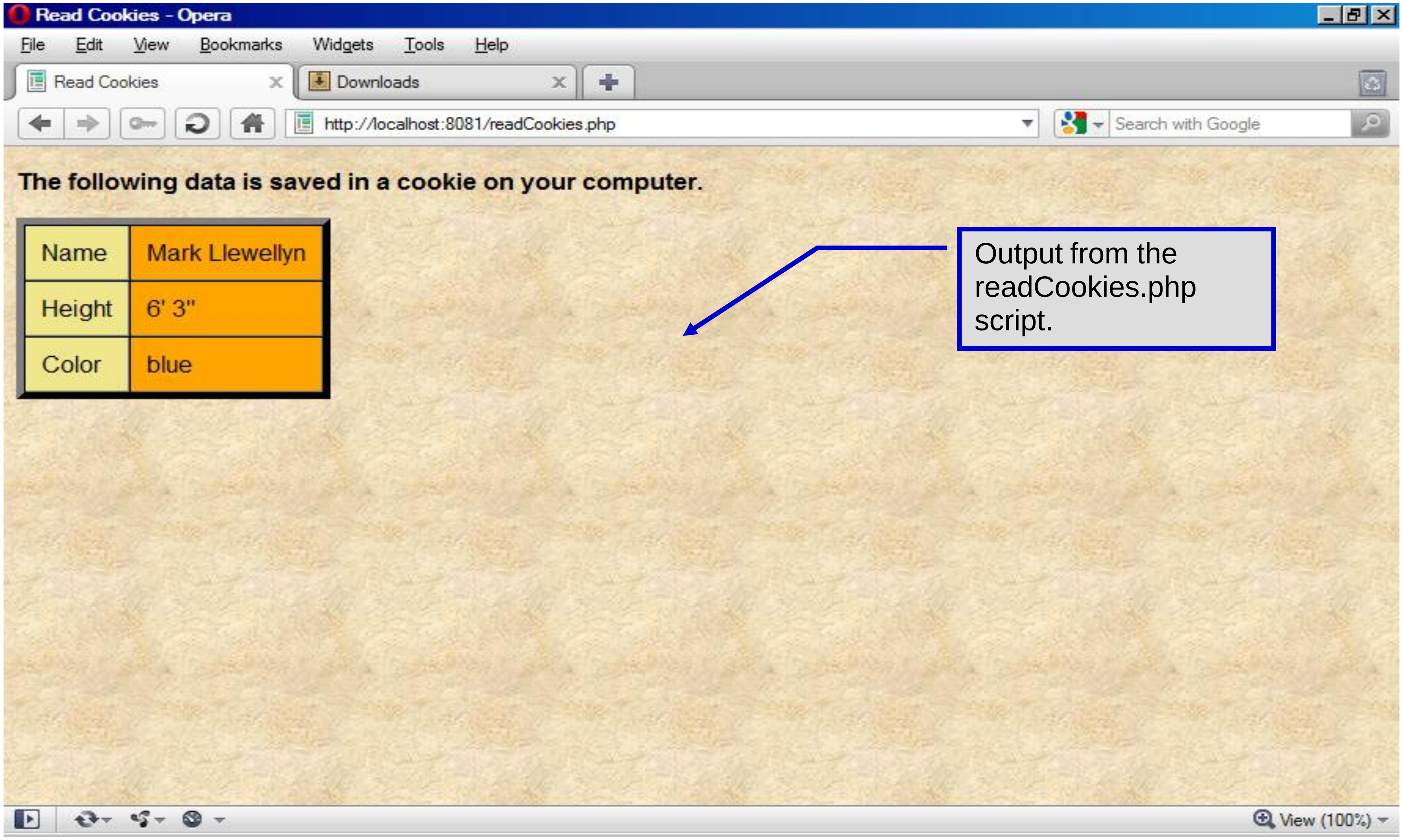
- Once the cookie has been created, the cookies.php script gives the user the chance to view the newly created cookie by invoking the readCookies.php script from within the cookies.php script by clicking on the link.
- The readCookies.php script code is illustrated on the next page followed by the output from the execution of this PHP script.

readCookies.php – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- readCookies.php                         -->
<!-- Program to read cookies from the client's computer -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head><title>Read Cookies</title></head>

   <body style = "font-family: arial, sans-serif" background=image1.jpg>
      <p>
         <strong>   The following data is saved in a cookie on your computer.
        </strong>
      </p>
      <table border = "5" cellspacing = "0" cellpadding = "10">

         <?php
            // iterate through array $_COOKIE and print
            // name and value of each cookie
            foreach ( $_COOKIE as $key => $value )
               print( "<tr>
                  <td bgcolor=\"#F0E68C\">$key</td>
                  <td bgcolor=\"#FFA500\">$value</td>
                  </tr>" );
         ?>
      </table>
   </body>  </html>
```

Superglobal array holding cookie.

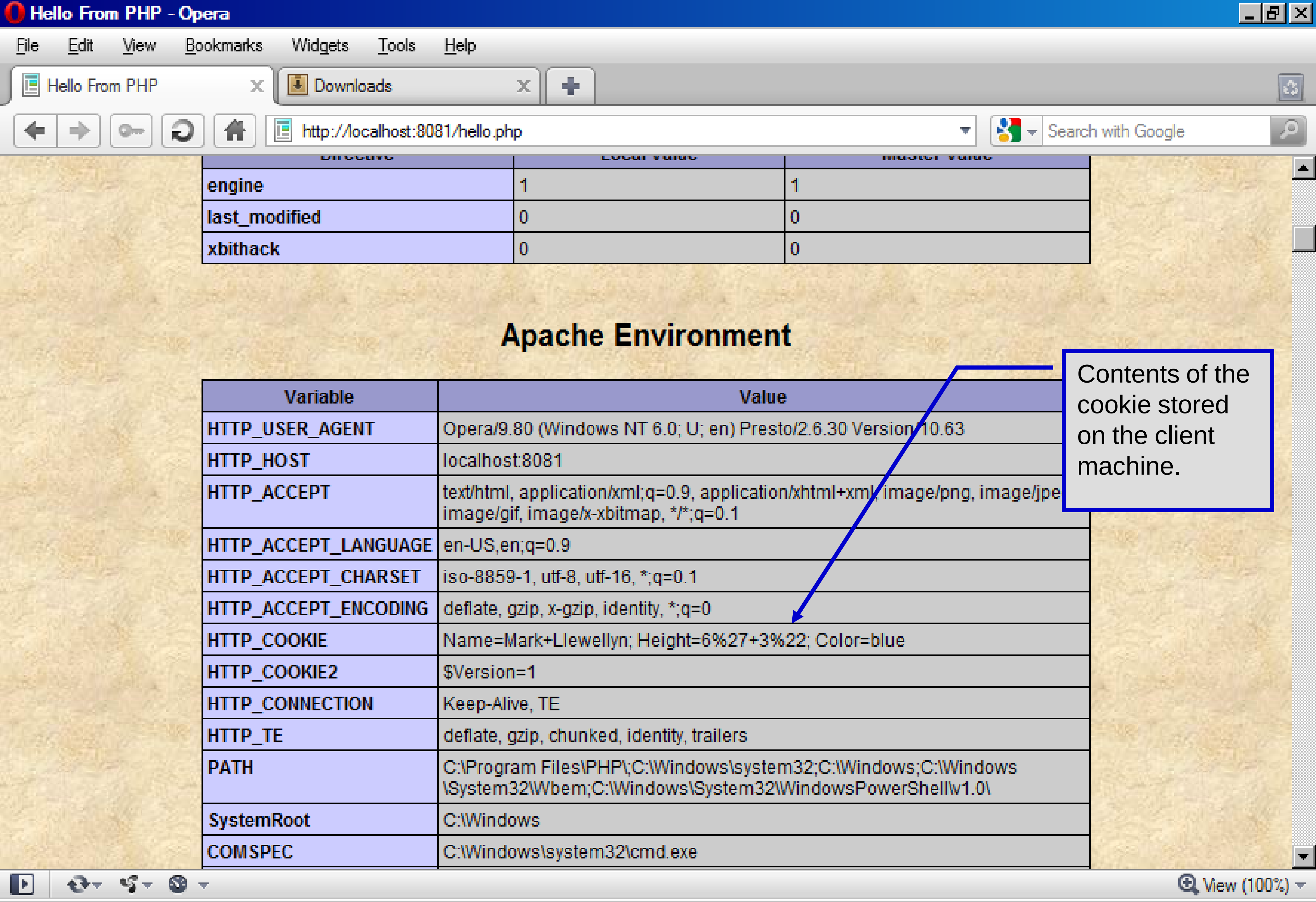
Cookies (cont.)


Read Cookies - Opera
File Edit View Bookmarks Widgets Tools Help
Read Cookies
Downloads
http://localhost:8081/readCookies.php
Search with Google
View (100%)

The following data is saved in a cookie on your computer.

Name	Mark Llewellyn
Height	6' 3"
Color	blue

Output from the readCookies.php script.


Hello From PHP - Opera
File Edit View Bookmarks Widgets Tools Help
Hello From PHP
Downloads
http://localhost:8081/hello.php
Search with Google
View (100%)

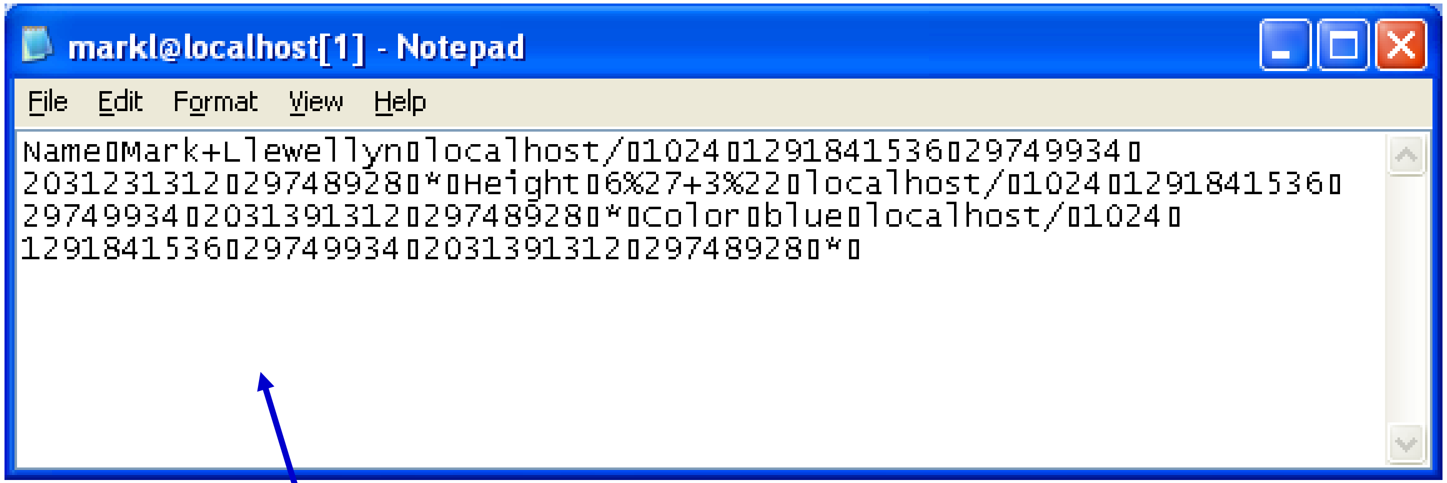
engine	1	1
last_modified	0	0
xbithack	0	0

Apache Environment

Variable	Value
HTTP_USER_AGENT	Opera/9.80 (Windows NT 6.0; U; en) Presto/2.6.30 Version/10.63
HTTP_HOST	localhost:8081
HTTP_ACCEPT	text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpe image/gif, image/x-xbitmap, */*;q=0.1
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.9
HTTP_ACCEPT_CHARSET	iso-8859-1, utf-8, utf-16, *;q=0.1
HTTP_ACCEPT_ENCODING	deflate, gzip, x-gzip, identity, *;q=0
HTTP_COOKIE	Name=Mark+Llewellyn; Height=6%27+3%22; Color=blue
HTTP_COOKIE2	$Version=1
HTTP_CONNECTION	Keep-Alive, TE
HTTP_TE	deflate, gzip, chunked, identity, trailers
PATH	C:\Program Files\PHP\;C:\Windows\system32;C:\Windows;C:\Windows \System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
SystemRoot	C:\Windows
COMSPEC	C:\Windows\system32\cmd.exe

Contents of the cookie stored on the client machine.

Cookies (cont.)



```
Name▯Mark+Llewellyn▯localhost/▯1024▯1291841536▯29749934▯
2031231312▯29748928▯*▯Height▯6%27+3%22▯localhost/▯1024▯1291841536▯
29749934▯2031391312▯29748928▯*▯Color▯blue▯localhost/▯1024▯
1291841536▯29749934▯2031391312▯29748928▯*▯
```

Actual text file holding cookie data for the cookie that was created in this example.

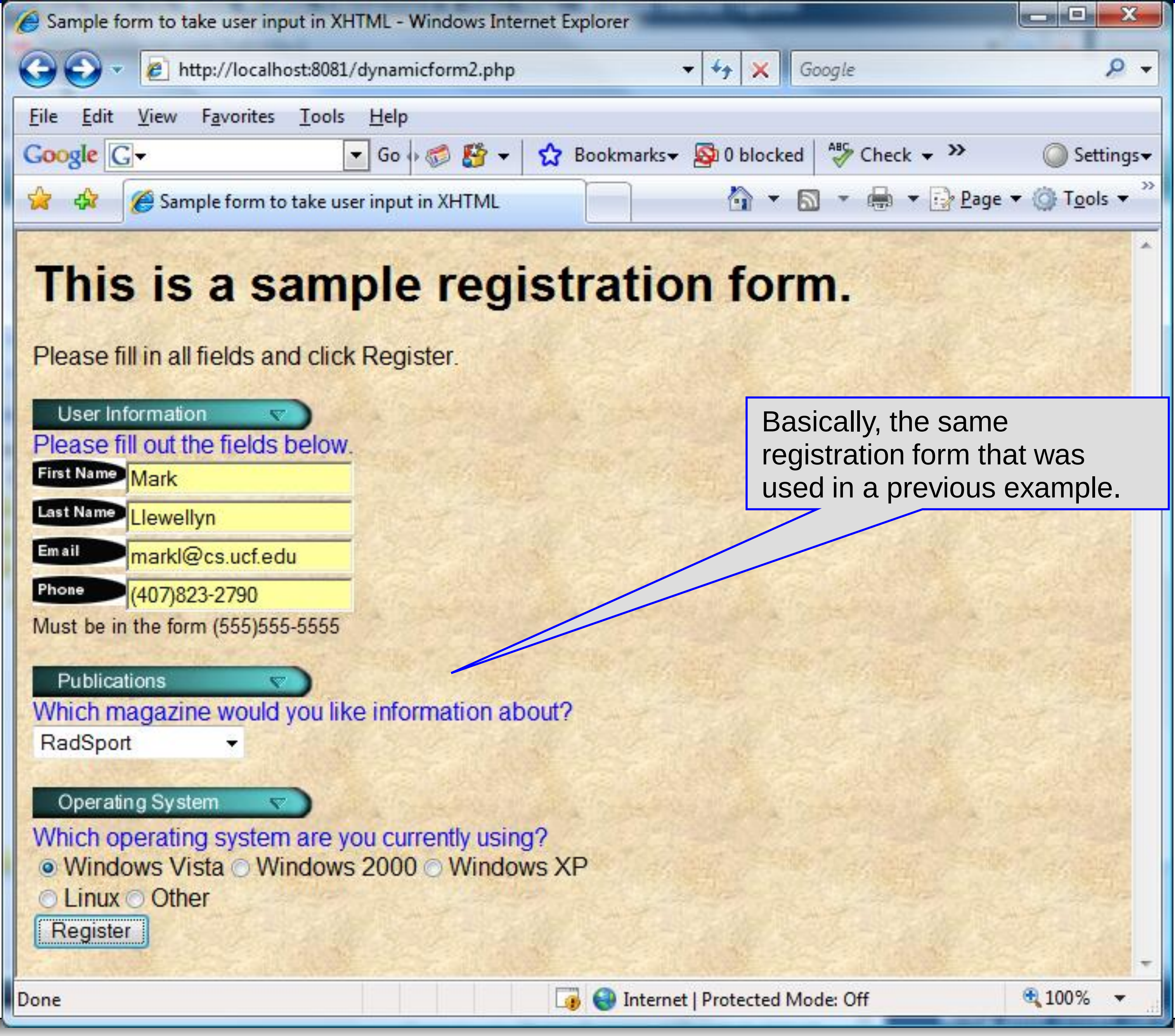
Dynamic Content in PHP

- Of all the strengths PHP exhibits as a server-side scripting language, perhaps its greatest strength lies in its ability to dynamically change XHTML output based on user input.

- In this final section of notes, we'll build on the examples we've constructed in the previous two sets of notes by combining `form.html` and `form.php` into one dynamic PHP document named `dynamicForm2.php`.

- We'll add error checking to the user input fields and inform the user of invalid entries on the form itself, rather than on an error page. If an error exists, the script maintains the previously submitted values in each form element.

- Finally, after the form has been successfully completed, we'll store the input from the user in a MySQL database.


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost:8081/dynamicform2.php
File Edit View Favorites Tools Help
Sample form to take user input in XHTML
Done
Internet | Protected Mode: Off
100%

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name	Mark
Last Name	Llewellyn
Em ail	markl@cs.ucf.edu
Phone	(407)823-2790

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

RadSport

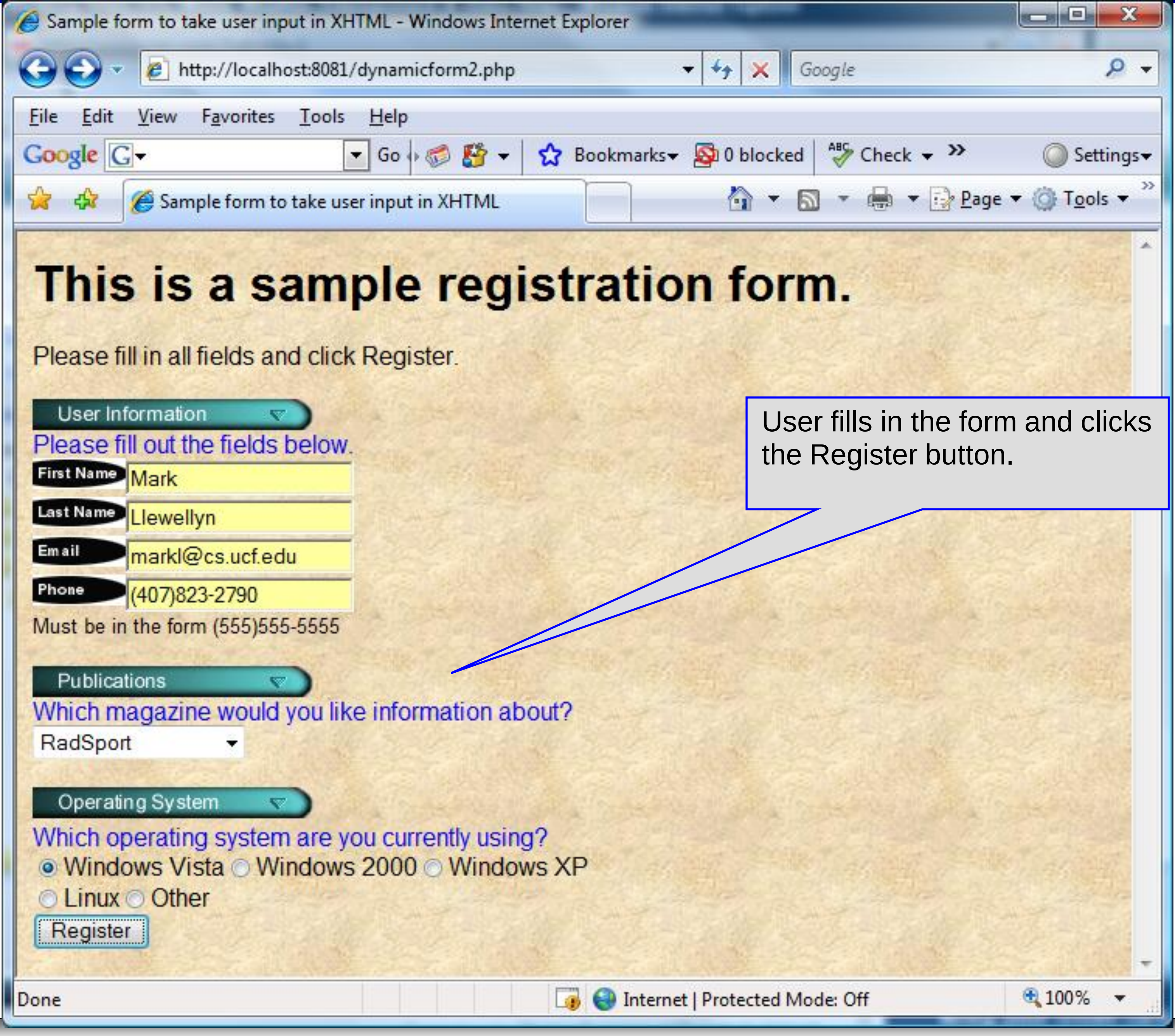
Operating System

Which operating system are you currently using?

[✓ Windows Vista] Windows 2000 Windows XP Linux Other

Register

Basically, the same registration form that was used in a previous example.


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost:8081/dynamicform2.php
File Edit View Favorites Tools Help
Sample form to take user input in XHTML
Done
Internet | Protected Mode: Off
100%

This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name	Mark
Last Name	Llewellyn
Email	markl@cs.ucf.edu
Phone	(407)823-2790

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

RadSport

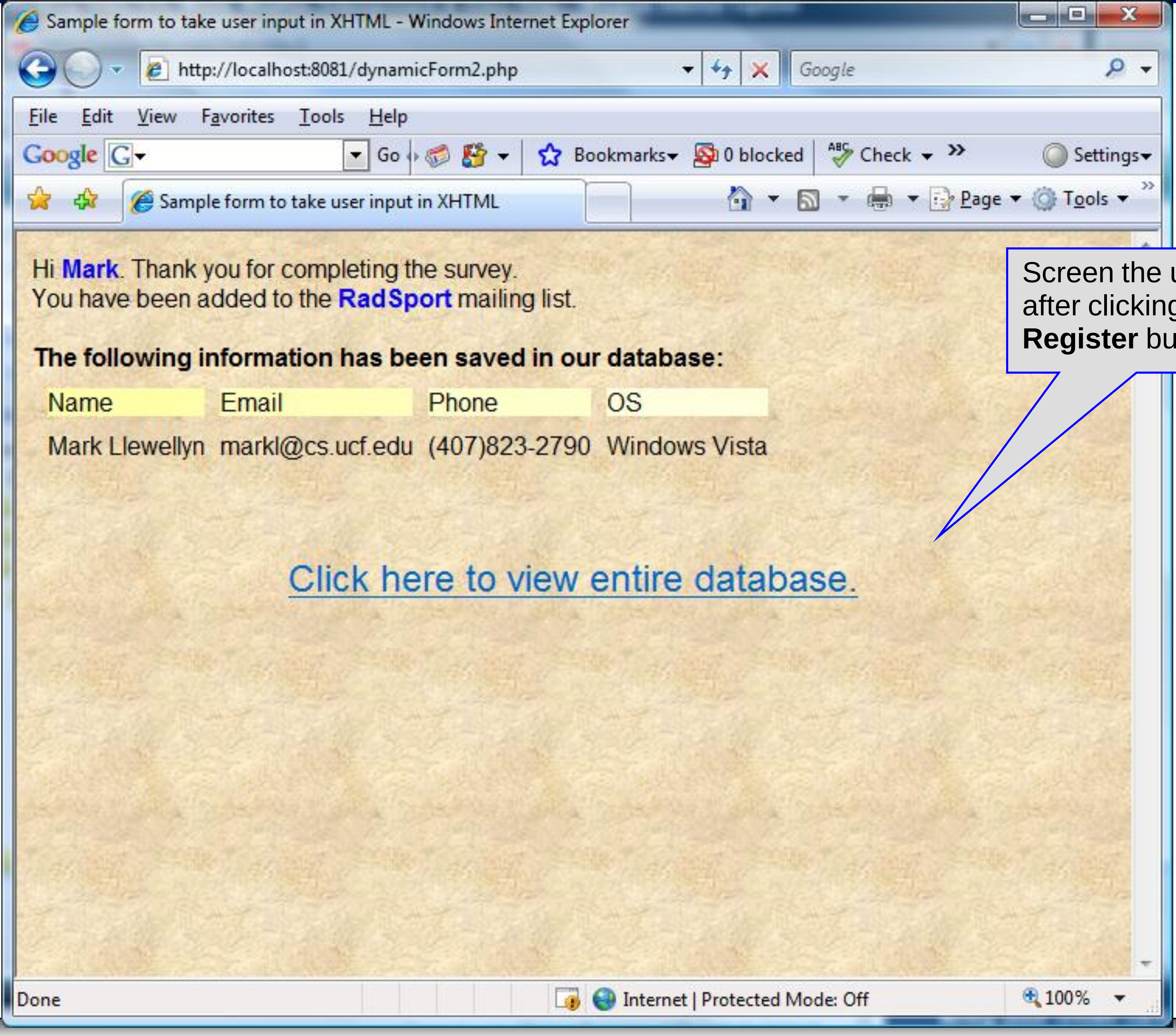
Operating System

Which operating system are you currently using?

[✓ Windows Vista] Windows 2000 Windows XP
Linux Other

Register

User fills in the form and clicks the Register button.


Sample form to take user input in XHTML - Windows Internet Explorer
http://localhost:8081/dynamicForm2.php
File Edit View Favorites Tools Help
Done
Internet | Protected Mode: Off
100%

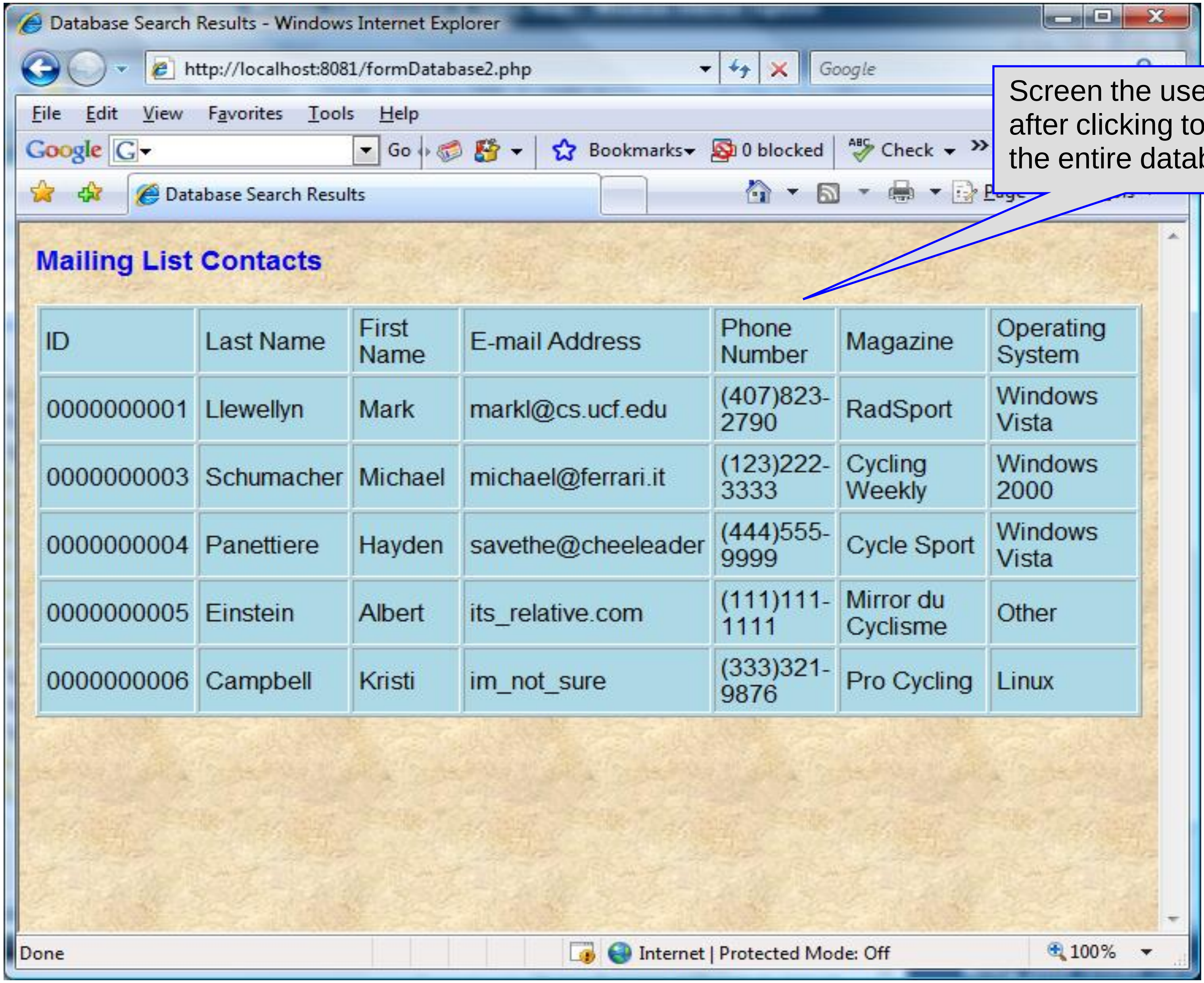
Hi **Mark**. Thank you for completing the survey.
You have been added to the **RadSport** mailing list.

The following information has been saved in our database:

Name	Email	Phone	OS
Mark Llewellyn	markl@cs.ucf.edu	(407)823-2790	Windows Vista

Click here to view entire database.

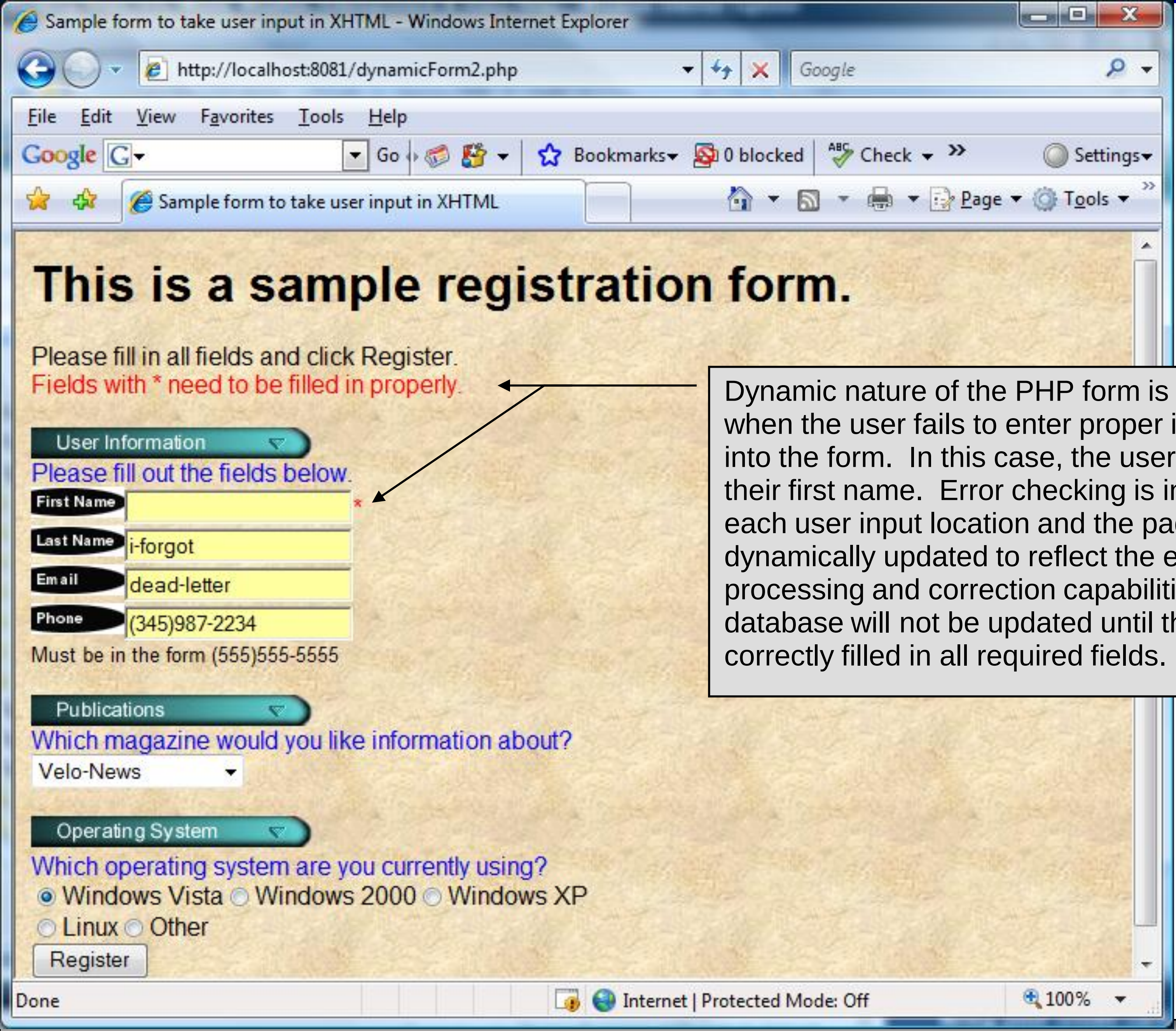
Screen the user sees after clicking the **Register** button.


Database Search Results - Windows Internet Explorer
http://localhost:8081/formDatabase2.php
File Edit View Favorites Tools Help
Database Search Results
Done
Internet | Protected Mode: Off
100%

Mailing List Contacts

ID	Last Name	First Name	E-mail Address	Phone Number	Magazine	Operating System
0000000001	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	RadSport	Windows Vista
0000000003	Schumacher	Michael	michael@ferrari.it	(123)222-3333	Cycling Weekly	Windows 2000
0000000004	Panettiere	Hayden	savethe@cheeleader	(444)555-9999	Cycle Sport	Windows Vista
0000000005	Einstein	Albert	its_relative.com	(111)111-1111	Mirror du Cyclisme	Other
0000000006	Campbell	Kristi	im_not_sure	(333)321-9876	Pro Cycling	Linux

Screen the user sees after clicking to see the entire database.



Sample form to take user input in XHTML - Windows Internet Explorer

http://localhost:8081/dynamicForm2.php

This is a sample registration form.

Please fill in all fields and click Register.
Fields with * need to be filled in properly.

User Information
Please fill out the fields below.

First Name	*
Last Name	i-forgot
Email	dead-letter
Phone	(345)987-2234

Must be in the form (555)555-5555

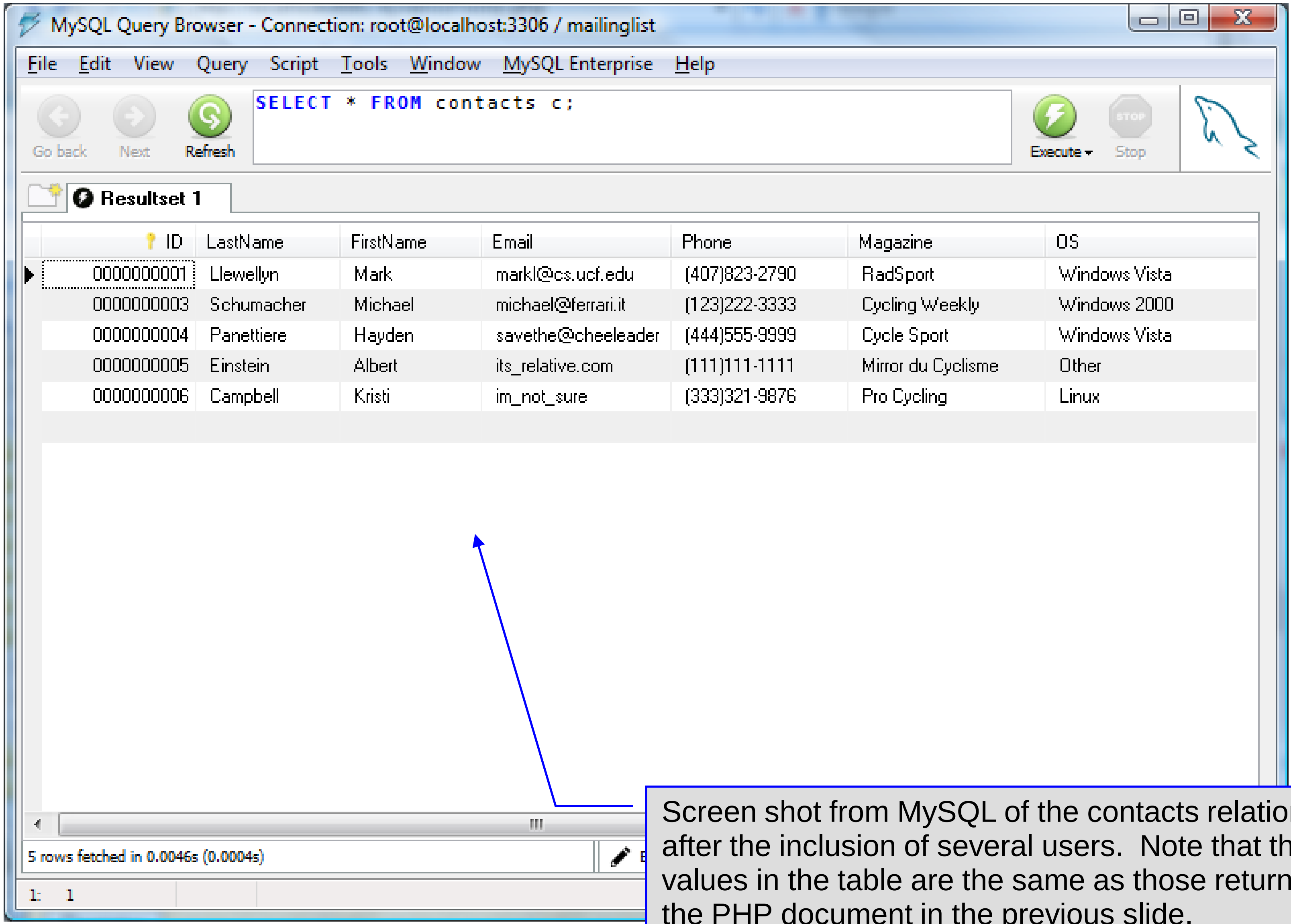
Publications
Which magazine would you like information about?
Velo-News

Operating System
Which operating system are you currently using?
[✓ Windows Vista] Windows 2000 Windows XP
Linux Other

Register

Done

Dynamic nature of the PHP form is illustrated when the user fails to enter proper information into the form. In this case, the user forgot to enter their first name. Error checking is in place on each user input location and the page is dynamically updated to reflect the error processing and correction capabilities. The database will not be updated until the user has correctly filled in all required fields.


MySQL Query Browser - Connection: root@localhost:3306 / mailinglist
File Edit View Query Script Tools Window MySQL Enterprise Help
Go back
Next
Refresh
Execute
Stop
Resultset 1

```
SELECT * FROM contacts c;
```

ID	LastName	FirstName	Email	Phone	Magazine	OS
0000000001	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	RadSport	Windows Vista
0000000003	Schumacher	Michael	michael@ferrari.it	(123)222-3333	Cycling Weekly	Windows 2000
0000000004	Panettiere	Hayden	savethe@cheeleader	(444)555-9999	Cycle Sport	Windows Vista
0000000005	Einstein	Albert	its_relative.com	(111)111-1111	Mirror du Cyclisme	Other
0000000006	Campbell	Kristi	im_not_sure	(333)321-9876	Pro Cycling	Linux

5 rows fetched in 0.0046s (0.0004s)

1: 1

Screen shot from MySQL of the contacts relation after the inclusion of several users. Note that the values in the table are the same as those returned to the PHP document in the previous slide.

dynamicForm2.php – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- dynamicForm2.php            -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>
      <title>Sample form to take user input in XHTML</title>
</head>
<body style = "font-family: arial, sans-serif;  background-color: #856363"
background=background.jpg>
   <?php
      extract ( $_POST );
      $iserror = false;
      // array of magazine titles
      $maglist = array( "Velo-News",
         "Cycling Weekly",
         "Pro Cycling",
         "Cycle Sport",
                  "RadSport",
                  "Mirror du Cyclisme" );
      // array of possible operating systems
      $systemlist = array( "Windows XP",
         "Windows 2000",
         "Windows 98",
         "Linux",
         "Other");
```

dynamicForm2.php – page 2

```
// array of name and alt values for the text input fields
   $inputlist = array( "fname" => "First Name",
      "lname" => "Last Name",
      "email" => "Email",
      "phone" => "Phone" );
   if ( isset ( $submit ) ) {
      if ( $fname == "" ) {
         $formerrors[ "fnameerror" ] = true;
         $iserror = true;
      }
      if ( $lname == "" ) {
         $formerrors[ "lnameerror" ] = true;
         $iserror = true;
      }
      if ( $email == "" ) {
         $formerrors[ "emailerror" ] = true;
         $iserror = true;
      }
      if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", $phone ) ) {
         $formerrors[ "phoneerror" ] = true;
         $iserror = true;
      }
      if ( !$iserror ) {
         // build INSERT query
         $query = "INSERT INTO contacts " .
            "(ID, LastName, FirstName, Email, Phone, Magazine, OS ) " .
            "VALUES (null, '$lname', '$fname', '$email', " .  "'" . quotemeta( $phone ) . "', '$mag', '$os' )";
```

dynamicForm2.php – page 3

```
// Connect to MySQL
 if ( !( $database = mysql_connect( "localhost",
    "root", "root" ) ) )
    die( "Could not connect to database" );

 // open MailingList database
 if ( !mysql_select_db( "MailingList", $database ) )
    die( "Could not open MailingList database" );

 // execute query in MailingList database
 if ( !( $result = mysql_query( $query, $database ) ) ) {
    print( "Could not execute query! <br />" );
    die( mysql_error() );
 }
 print( "<p>Hi
    <span style = 'color: blue'> <strong>$fname</strong></span>.
    Thank you for completing the survey.<br />
    You have been added to the <span style = 'color: blue'>
    <strong>$mag</strong></span> mailing list.             </p>
    <strong>The following information has been saved in our database:</strong><br />

    <table border = '0' cellpadding = '0' cellspacing = '10'>
    <tr>
    <td bgcolor = '#ffffaa'>Name </td>
    <td bgcolor = '#ffffbb'>Email</td>
    <td bgcolor = '#ffffcc'>Phone</td>
    <td bgcolor = '#ffffdd'>OS</td>
    </tr>
    <tr>
```

```
            <!-- print each form field’s value -->
            <td>$fname $lname</td>
            <td>$email</td>
            <td>$phone</td>
            <td>$os</td>
            </tr></table>
            <br /><br /><br />
            <div style = 'font-size : 10pt; text-align: center'>
                                          <div style = 'font-size : 18pt'>
                                          <a href = 'formDatabase2.php'>
                                          Click here to view entire database.</a>
                                          </div>
              </div></body></html>" );
          die();
        }
      }
      print( "<h1>This is a sample registration form.</h1>
        Please fill in all fields and click Register." );
      if ( $iserror ) {
        print( "<br /><span style = 'color : red'>
          Fields with * need to be filled in properly.</span>" );
      }
      print( "<!-- post form data to dynamicForm2.php -->
        <form method = 'post' action = 'dynamicForm2.php'>
        <img src = 'images/user.gif' alt = 'User' /><br />
        <span style = 'color: blue'>
        Please fill out the fields below.<br />
        </span>
```

Invoke PHP script to see contents of entire database if user clicks this link. Code begins on page 14.

The form created is self-submitting (i.e., it posts to itself). This is done by setting the action to dynamicForm2.php

dynamicForm2.php – page 5

```
<!-- create four text boxes for user input -->" );
 foreach ( $inputlist as $inputname => $inputalt ) {
    $inputtext = $inputvalues[ $inputname ];

    print( "<img src = 'images/$inputname.gif'
       alt = '$inputalt' /><input type = 'text'  name = '$inputname' value = '" . $$inputname . "' />" );
    if ( $formerrors[ ( $inputname )."error" ] == true )
       print( "<span style = 'color : red'>*</span>" );
    print( "<br />" );
 }
 print( "<span style = 'font-size : 10pt" );
 if ( $formerrors[ "phoneerror" ] )  print( "; color : red" );
 print( "'>Must be in the form (555)555-5555
    </span><br /><br />
    <img src = 'images/downloads.gif'
    alt = 'Publications' /><br />
    <span style = 'color: blue'>
    Which magazine would you like information about?
    </span><br />
    <!-- create drop-down list containing magazine names -->
    <select name = 'mag'>" );
 foreach ( $maglist as $currmag ) {
    print( "<option" );
    if ( ( $currmag == $mag ) )
       print( " selected = 'true'" );
    print( ">$currmag</option>" );
 }
```

The $$variable notation specifies variable variables. PHP permits the use of variable variables to allow developers to reference variables dynamically. The expression $$variable could also be written as ${$variable} for added clarity.

dynamicForm2.php – page 6

```
      print( "</select><br /><br />
      <img src = 'images/os.gif' alt = 'Operating System' />
      <br /><span style = 'color: blue'>
      Which operating system are you currently using?
      <br /></span>

      <!-- create five radio buttons -->" );

   $counter = 0;

   foreach ( $systemlist as $currsystem ) {
      print( "<input type = 'radio' name = 'os'
         value = '$currsystem'" );

      if ( $currsystem == $os ) print( "checked = 'checked'" );
      if ( iserror && $counter == 0 ) print( "checked = 'checked'" );

      print( " />$currsystem" );

      if ( $counter == 2 ) print( "<br />" );
      $counter++;
   }

   print( "<!-- create a submit button -->
      <br />
      <input type = 'submit' name = 'submit' value = 'Register' />
      </form></body></html>" );
?>
```

formDatabase2.php – page 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!– formDatabase2.php        -->
<!-- Program to query a database and send results to the client.     -->

<html xmlns = "http://www.w3.org/1999/xhtml">
   <head>      <title>Database Search Results</title>   </head>
   <body style = "font-family: arial, sans-serif"
      style = "background-color: #F0E68C" background=image1.jpg>
      <?php
         extract( $_POST );
         // build SELECT query
         $query = "SELECT * FROM contacts";
                 // Connect to MySQL
         if ( !( $database = mysqli_connect( "localhost",  "root", "root“, MailingList ) ) )
            die( "Could not connect to database" );
          // query MailingList database
         if ( !( $result = mysqli_query( $database, $query ) ) ) {
            print( "Could not execute query! <br />" );
            die( mysqli_error() );
         }
      ?>
      <h3 style = "color: blue">
      Mailing List Contacts</h3>
```

formDatabase2.php – page 2

```
    <table border = "1" cellpadding = "3" cellspacing = "2"
     style = "background-color: #ADD8E6">
     <tr>
        <td>ID</td>
        <td>Last Name</td>
        <td>First Name</td>
        <td>E-mail Address</td>
        <td>Phone Number</td>
        <td>Magazine</td>
        <td>Operating System</td>
     </tr>
     <?php
        // fetch each record in result set
        for ( $counter = 0;
           $row = mysqli_fetch_row( $result );
           $counter++ ){
           // build table to display results
           print( "<tr>" );
           foreach ( $row as $key => $value )
              print( "<td>$value</td>" );
           print( "</tr>" );
        }
        mysqli_close( $database );
     ?>

   </table>
  </body>
</html>
```

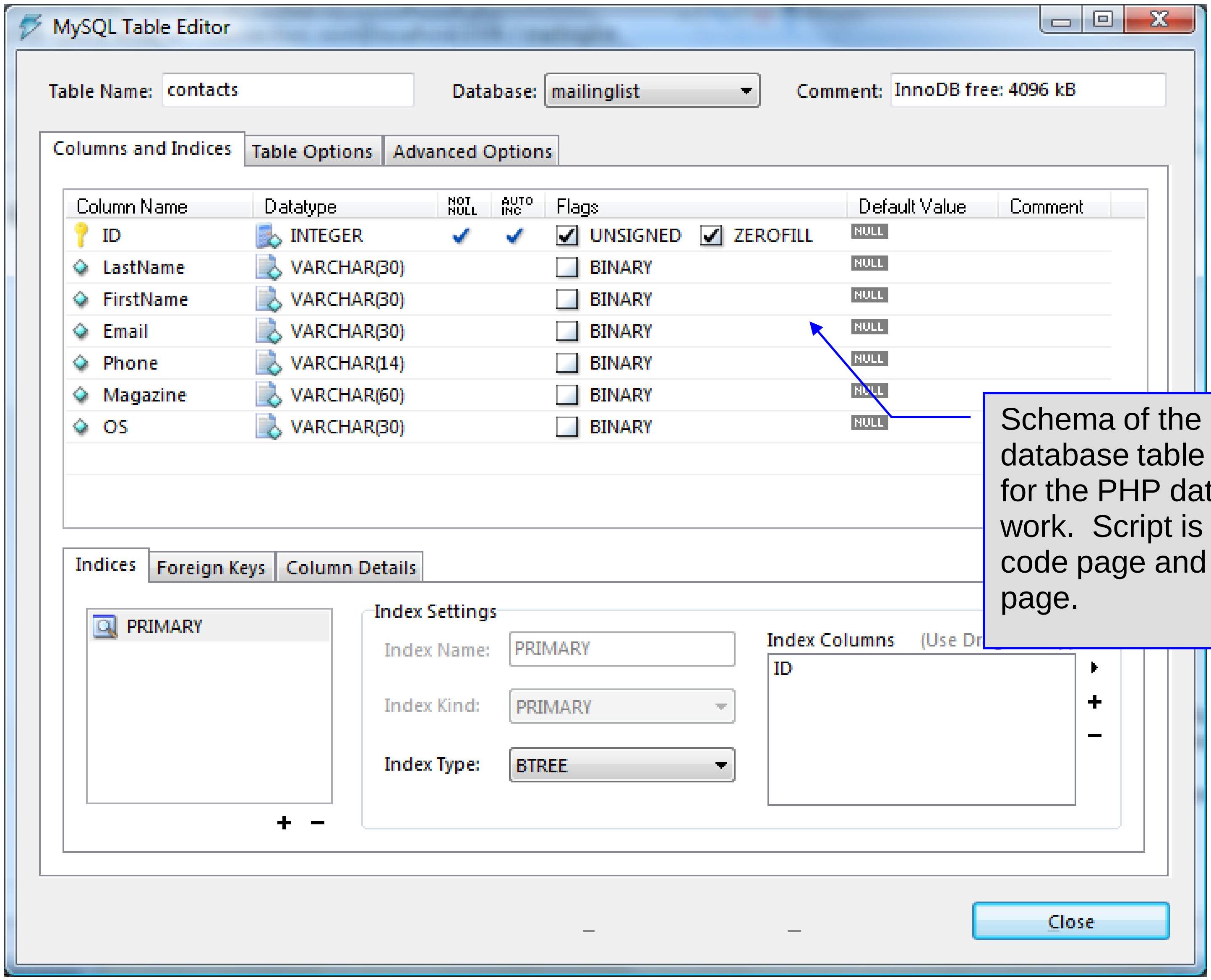


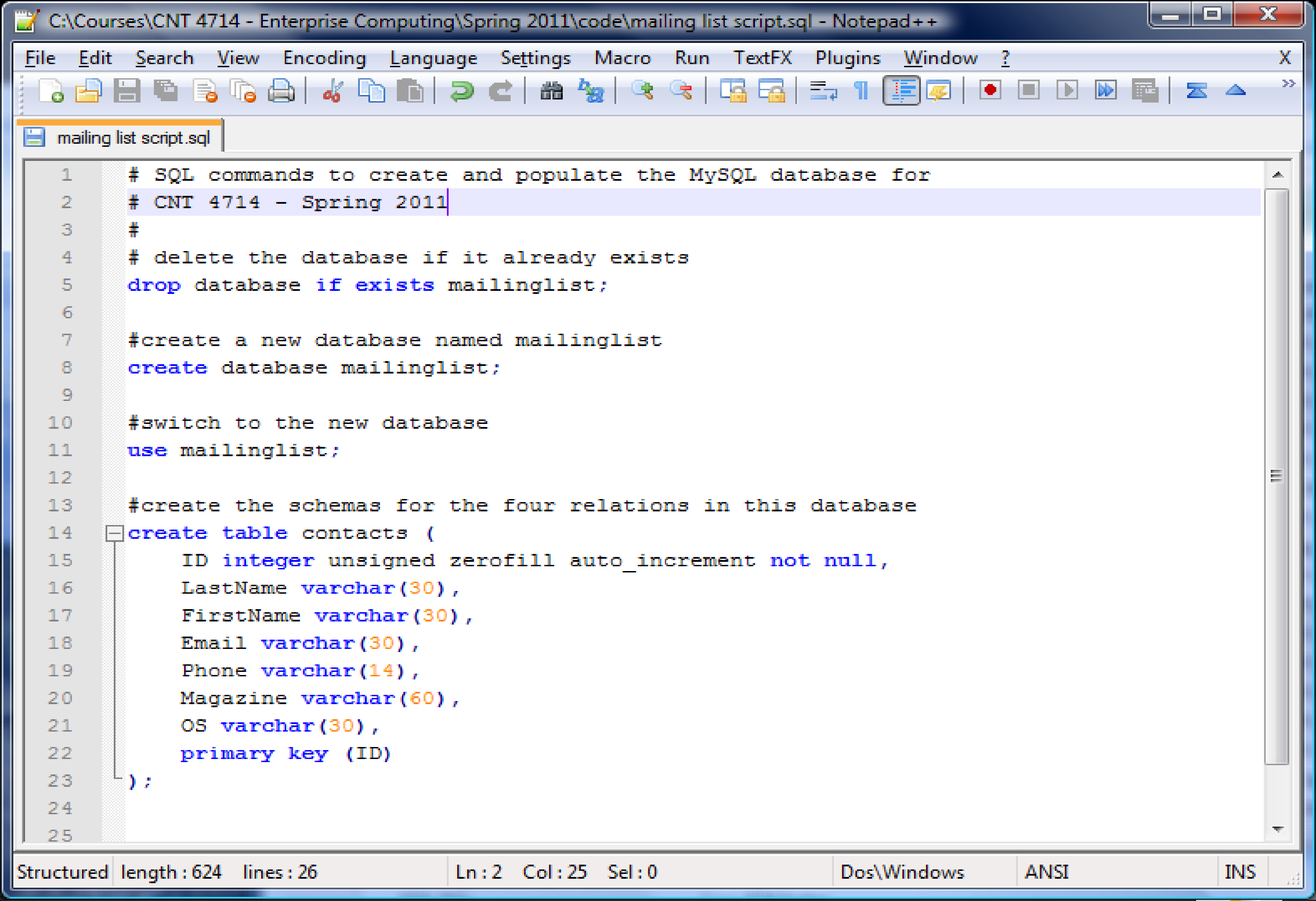
Table Name: contacts
Database: mailinglist
Comment: InnoDB free: 4096 kB

Column Name	Datatype	NOT NULL	AUTO INC	Flags		Default Value	Comment
ID	INTEGER	✓	✓	☑ UNSIGNED	☑ ZEROFILL	NULL	
LastName	VARCHAR(30)			☐ BINARY		NULL	
FirstName	VARCHAR(30)			☐ BINARY		NULL	
Email	VARCHAR(30)			☐ BINARY		NULL	
Phone	VARCHAR(14)			☐ BINARY		NULL	
Magazine	VARCHAR(60)			☐ BINARY		NULL	
OS	VARCHAR(30)			☐ BINARY		NULL	

PRIMARY

Index Settings
Index Name: PRIMARY
Index Kind: PRIMARY
Index Type: BTREE
Index Columns (Use Dr
ID

Schema of the MailingList database table contacts required for the PHP database example to work. Script is available on the code page and shown on the next page.



C:\Courses\CNT 4714 - Enterprise Computing\Spring 2011\code\mailing list script.sql - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
mailing list script.sql

```sql
# SQL commands to create and populate the MySQL database for
# CNT 4714 - Spring 2011
#
# delete the database if it already exists
drop database if exists mailinglist;

#create a new database named mailinglist
create database mailinglist;

#switch to the new database
use mailinglist;

#create the schemas for the four relations in this database
create table contacts (
    ID integer unsigned zerofill auto_increment not null,
    LastName varchar(30),
    FirstName varchar(30),
    Email varchar(30),
    Phone varchar(14),
    Magazine varchar(60),
    OS varchar(30),
    primary key (ID)
);

```

Structured length : 624 lines : 26 Ln : 2 Col : 25 Sel : 0 Dos\Windows ANSI INS