

# CNT 4714: Enterprise Computing Fall 2012

## PHP – Database Connectivity

Instructor :      Dr. Mark Llewellyn  
                         markl@cs.ucf.edu  
                         HEC 236, 407-823-2790  
                         <http://www.cs.ucf.edu/courses/cnt4714/fall2012>

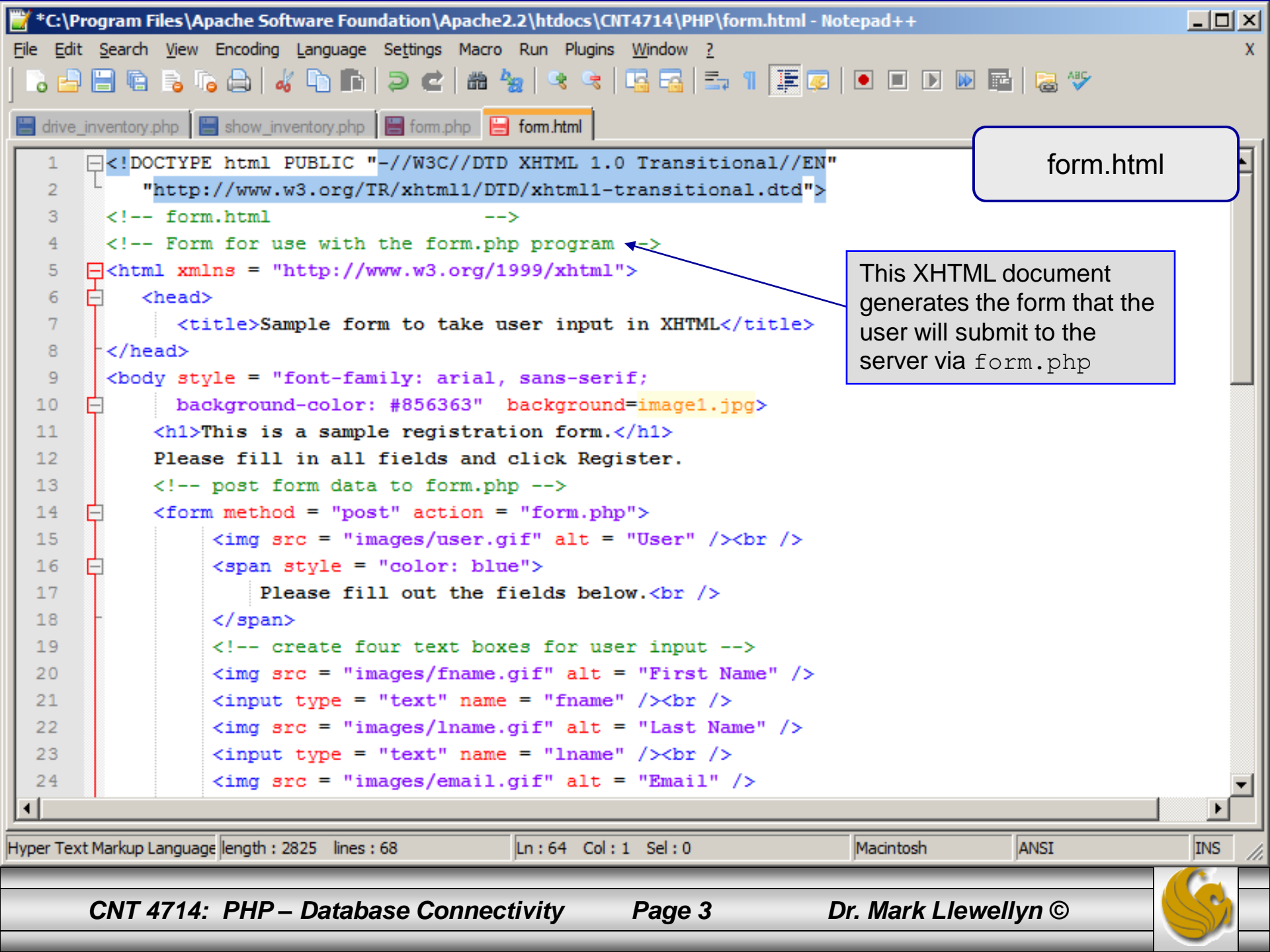
Department of Electrical Engineering and Computer Science  
Computer Science Division  
University of Central Florida



# 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

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

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- form.html -->
4 <!-- Form for use with the form.php program -->
5 <html xmlns = "http://www.w3.org/1999/xhtml">
6 <head>
7     <title>Sample form to take user input in XHTML</title>
8 </head>
9 <body style = "font-family: arial, sans-serif;
10     background-color: #856363" background=imagem1.jpg">
11 <h1>This is a sample registration form.</h1>
12 Please fill in all fields and click Register.
13 <!-- post form data to form.php -->
14 <form method = "post" action = "form.php">
15     <img src = "images/user.gif" alt = "User" /><br />
16     <span style = "color: blue">
17         Please fill out the fields below.<br />
18     </span>
19     <!-- create four text boxes for user input -->
20     <img src = "images/fname.gif" alt = "First Name" />
21     <input type = "text" name = "fname" /><br />
22     <img src = "images/lname.gif" alt = "Last Name" />
23     <input type = "text" name = "lname" /><br />
24     <img src = "images/email.gif" alt = "Email" />
```





form.html

```

25 <input type = "text" name = "email" /><br />
26 <img src = "images/phone.gif" alt = "Phone" />
27 <input type = "text" name = "phone" /><br />
28 <span style = "font-size: 10pt">
29     Must be in the form (555)555-5555</span>
30 <br /><br />
31 <img src = "images/downloads.gif" alt = "Products" /><br />
32 <span style = "color: blue">
33     Which publication would you like information about?
34 </span><br />
35 <!-- create drop-down list containing magazine names -->
36 <select name = "magazine">
37     <option>Velo-News</option>
38     <option>Cycling Weekly</option>
39     <option>Pro Cycling</option>
40     <option>Cycle Sport</option>
41     <option>RadSport</option>
42     <option>Mirror du Cyclisme</option>
43 </select>
44 <br /><br />
45 <img src = "images/os.gif" alt = "Operating System" />
46 <br /><span style = "color: blue">
47     Which operating system are you currently using?
48 <br /></span>

```



form.html

```
48 <br /></span>
49 <!-- create five radio buttons -->
50 <input type = "radio" name = "os" value = "Windows Vista" checked = "checked" />
51     Windows Vista
52 <input type = "radio" name = "os" value = "Windows 2000" />
53     Windows 7
54 <input type = "radio" name = "os" value = "Windows 98" />
55     Windows XP
56 <input type = "radio" name = "os" value = "Linux" />
57     Linux
58 <input type = "radio" name = "os" value = "Other" />
59     Other<br />
60 <!-- create a submit button -->
61 <br /><input type = "submit" value = "Register" />
62 </form>
63 </body>
64 </html>
65
66
67
68
```



form.php

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.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- form.php -->
4 <!-- Read information sent from form.html -->
5 <html xmlns = "http://www.w3.org/1999/xhtml">
6 <head>
7 <title>Form Validation</title>
8 </head>
9 <body style = "font-family: arial, sans-serif;
10 background-color: #856363" background=image1.jpg>
11 <?php
12 extract($_POST);
13 // determine whether phone number is valid and print
14 // an error message if not
15 if ( !ereg( "[0-9]{3}\-[0-9]{3}-[0-9]{4}$",
16 $phone ) ){
17 print( "<p><span style = \"color: red;
18 font-size: 2em\">
19 INVALID PHONE NUMBER:</span><br />
20 A valid phone number must be in the form
21 <strong>(555)555-5555</strong><br />
22 <span style = \"color: blue\">
23 Click the Back button, enter a valid phone
24 number and resubmit.<br /><br />
25 Thank You.</span></p></body></html>" );
26 die(); // terminate script execution
27 }

```





form.php

```

28  ?>
29  <p>Hi
30      <span style = "color: blue">
31          <strong>
32              <?php print( "$fname" ); ?>
33          </strong>
34      </span>.
35      Thank you for completing the survey.<br />
36      You have been added to the
37      <span style = "color: blue">
38          <strong>
39              <?php print( "$magazine " ); ?>
40          </strong>
41      </span>
42      mailing list.
43  </p>
44  <strong>The following information has been saved
45      in our database:</strong><br />
46
47  <table border = "0" cellpadding = "0" cellspacing = "10">
48      <tr>
49          <td bgcolor = "#ffffaa">Name </td>
50          <td bgcolor = "#ffffbb">Email</td>
51          <td bgcolor = "#ffffcc">Phone</td>
52          <td bgcolor = "#ffffdd">OS</td>
53      </tr>
54

```



```
55 <tr>
56 <?php
57
58 // print each form field's value
59 print( "<td>$fname $lname</td>
60 <td>$email</td>
61 <td>$phone</td>
62 <td>$os</td>" );
63 ?>
64 </tr>
65 </table>
66
67 <br /><br /><br />
68 <div style = "font-size: 10pt; text-align: center">
69 This is only a sample form.
70 You have not been added to a mailing list.
71 </div>
72 </body>
73 </html>
74
75
```



Sample form to take user input in XHTML - Opera

Opera Sample form to take user... x

Web localhost:8081/CNT4714/PHP/form.html Search with Google

# 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 7  Windows XP  Linux  Other

Register

Execution of form.html within a web browser

View (100%) Update Ready





Sample form to take user input in XHTML - Opera

Opera Sample form to take user... x +

Web localhost:8081/CNT4714/PHP/form.html Search with Google

# 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?

**Operating System**

Which operating system are you currently using?

Windows Vista  Windows 7  Windows XP  Linux  Other

Update Ready



Form Validation - Opera

Opera Form Validation

Web localhost:8081/CNT4714/PHP/form.php Search with Google

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

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

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

Update Ready





# 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

User enters an improperly formatted telephone number in the form.

Must be in the form (555)555-5555

Publications

Which publication would you like information about?

RadSport

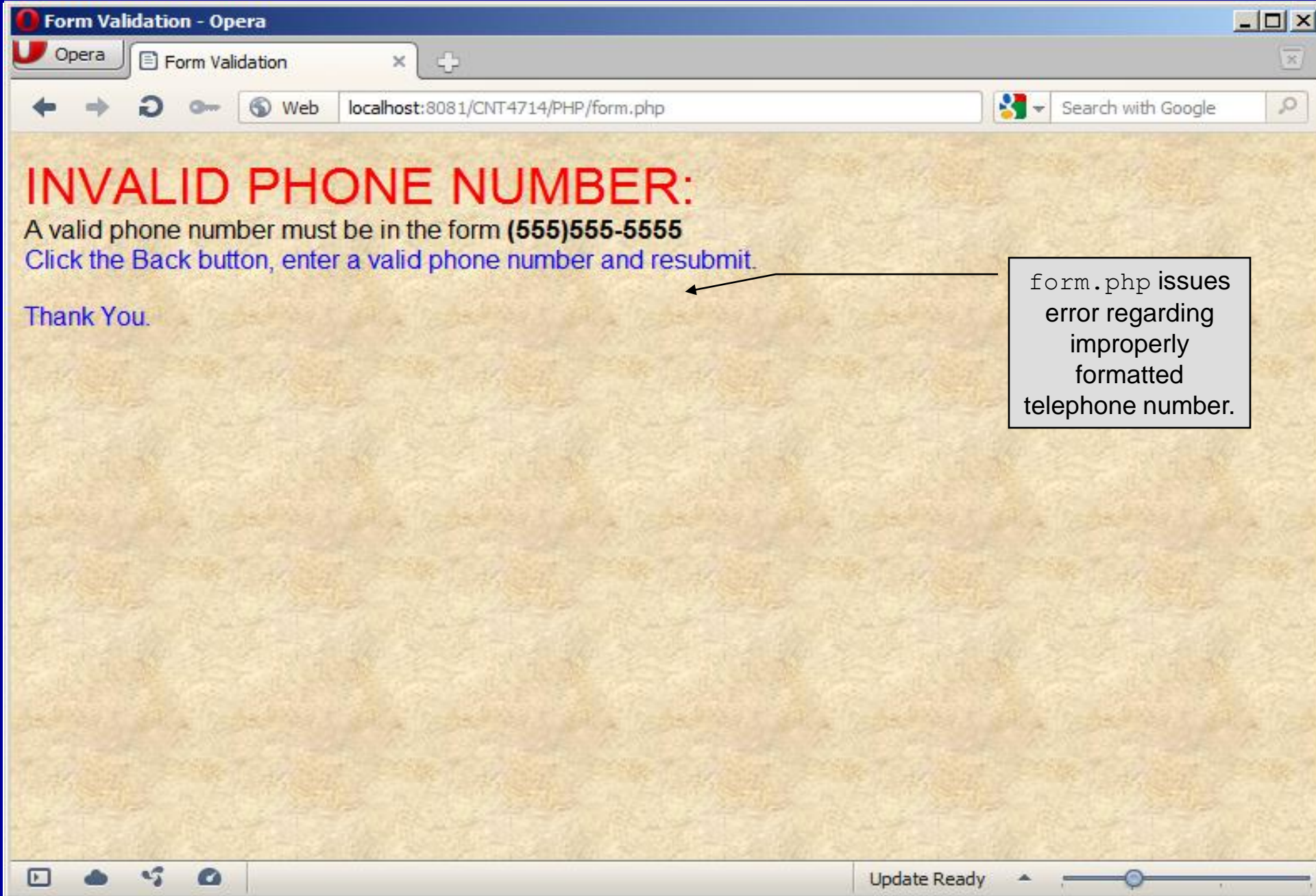
Operating System

Which operating system are you currently using?

Windows Vista  Windows 7  Windows XP  Linux  Other

Register





# 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 than POSIX regular expressions. PHP's support for Perl-compatible regular expressions eases migration from Perl to PHP. The following examples illustrate these concepts.





expression.php

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3  <!-- expression.php -->
4  <!-- Using regular expressions -->
5  <html xmlns = "http://www.w3.org/1999/xhtml">
6  <head>
7      <title>Regular expressions</title>
8  </head>
9  <body style = "font-family: arial, sans-serif;
10     background-color: #856363" background=images/background.jpg>
11
12     <?php
13         $search = "Now is the time";
14         print( "Test string is: '$search'<br /><br />" );
15
16         // call function ereg to search for pattern 'Now'
17         // in variable search
18         if ( ereg( "Now", $search ) )
19             print( "String 'Now' was found.<br />" );
20
21         // search for pattern 'Now' in the beginning of
22         // the string
23         if ( ereg( "^Now", $search ) )
24             print( "String 'Now' found at beginning
25                 of the line.<br />" );
26

```

^ matches at beginning of a string



expression.php



expression.php

```

27
28 // search for pattern 'Now' at the end of the string
29 if ( ereg( "Now$", $search ) )
30     print( "String 'Now' was found at the end
31           of the line.<br />" );
32
33 // search for any word ending in 'ow'
34 if ( ereg( "[[:<:]]([a-zA-Z]*ow)[[:>:]]", $search,
35           $match ) )
36     print( "Word found ending in 'ow': " .
37           $match[ 1 ] . "<br />" );
38
39 // search for any words beginning with 't'
40 print( "Words beginning with 't' found: " );
41
42 while ( eregi( "[[:<:]](t[[:alpha:]]+)[[:>:]]",
43             $search, $match ) ) {
44     print( $match[ 1 ] . " " );
45
46     // remove the first occurrence of a word beginning
47     // with 't' to find other instances in the string
48     $search = ereg_replace( $match[ 1 ], "", $search );
49 }
50
51 print( "<br />" );
52
53

```

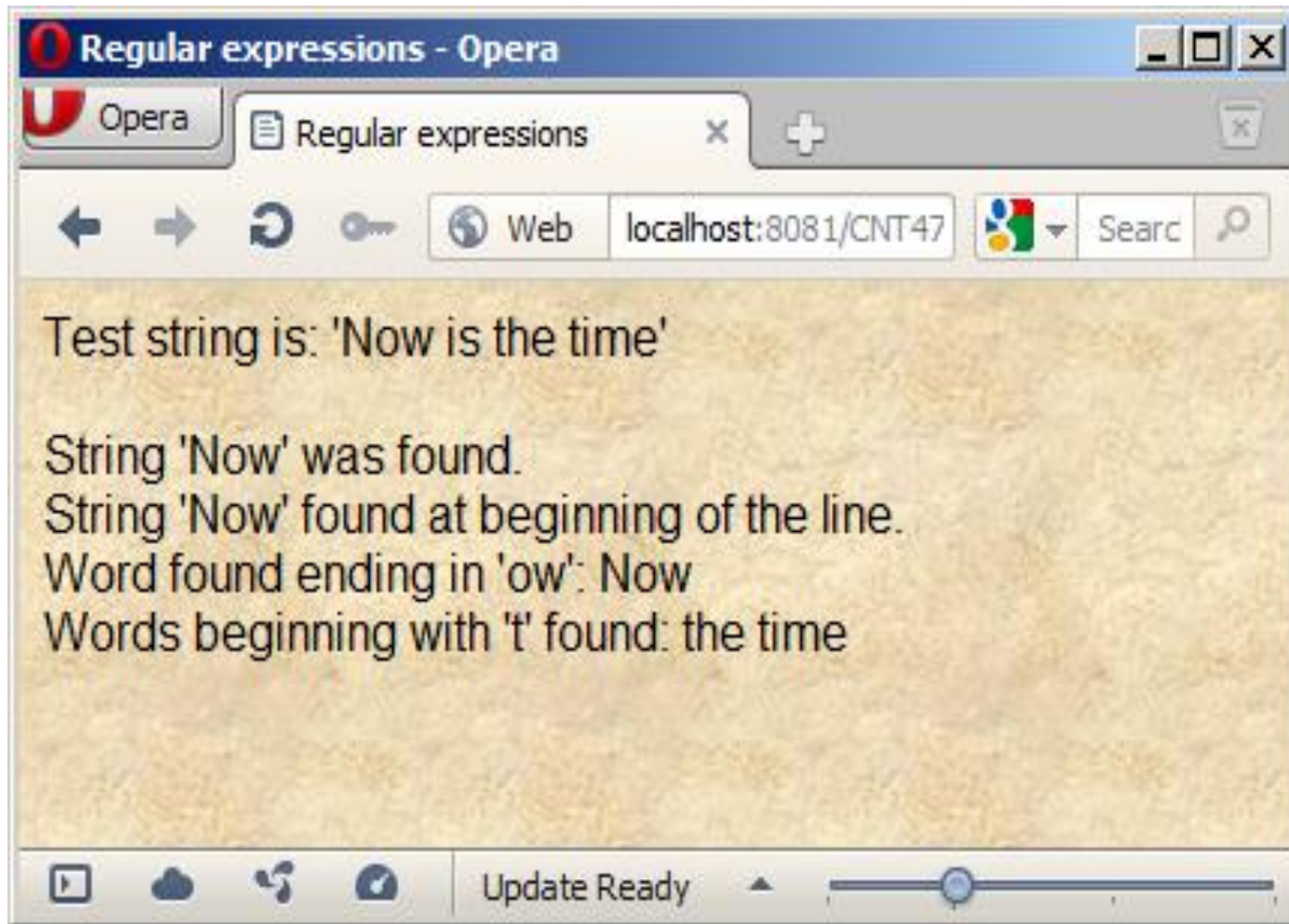
\$ matches at end of a string

Uses a regular expression to match a word ending in "ow".





# Output From `expression.php` - Example



# 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>.

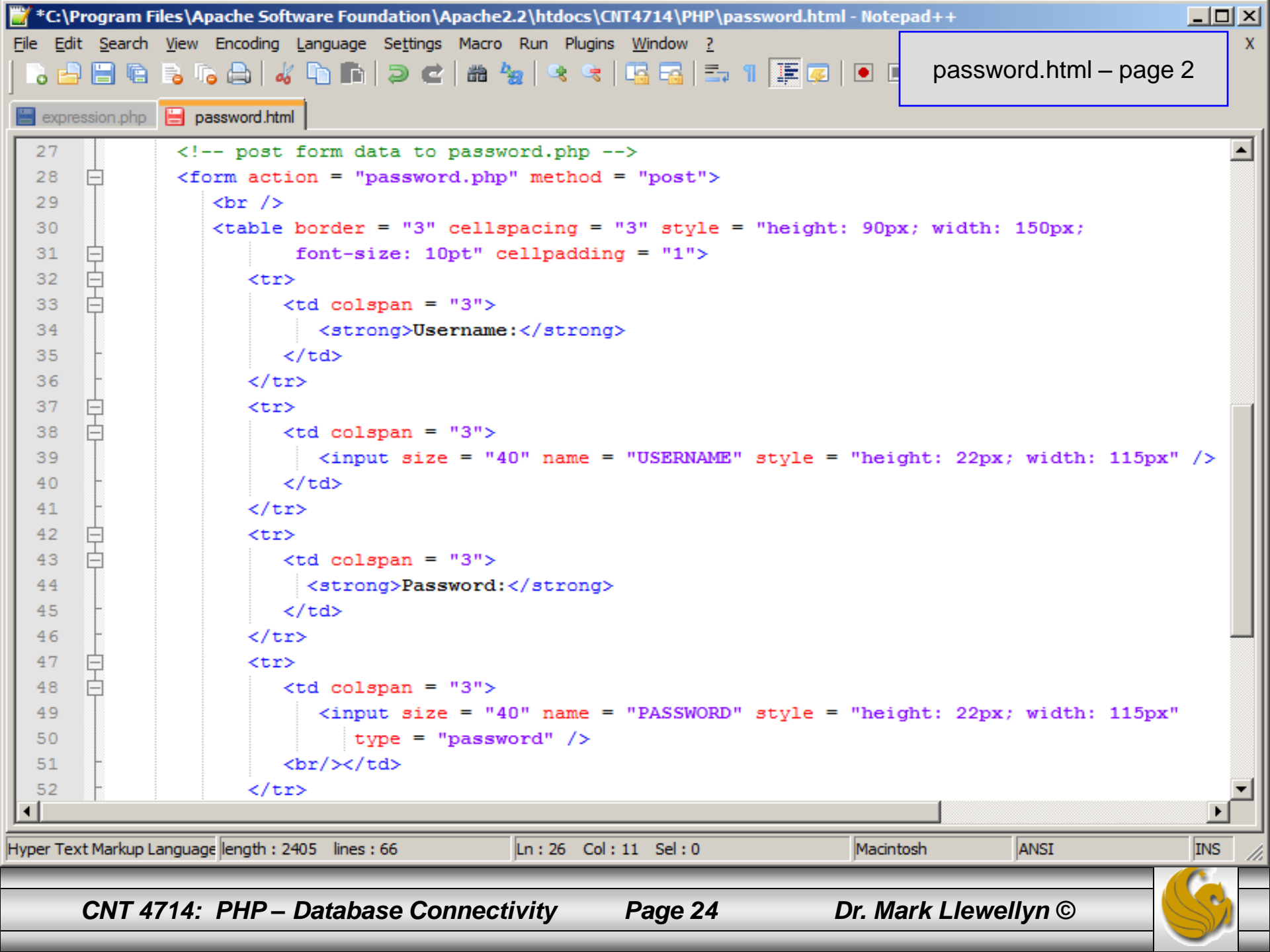


```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- password.html -->
4 <!-- XHTML form sent to password.php for verification -->
5 <html xmlns = "http://www.w3.org/1999/xhtml">
6   <head>
7     <title>Verifying a username and a password.</title>
8     <style type = "text/css">
9       td { background-color: #DDDDDD }
10    </style>
11  </head>
12  <body style = "font-family: arial", "font-size: 2em",
13    background-color: #856363" background=images/background.jpg>
14    <p style = "font-size: 18pt">
15      <font color=red><B>
16      Welcome to the CNT 4714 High Security WebPage
17      </B></font>
18      <HR>
19      <p style = "font-size: 13pt">
20        Type in your username and password below.
21        <br />
22        <span style = "color: #0000FF; font-size: 10pt;
23          font-weight: bold">
24          Note that password will be sent as plain text - encryption not used in this applica
25        </span>
26    </p>

```





expression.php password.html

```

27 <!-- post form data to password.php -->
28 <form action = "password.php" method = "post">
29   <br />
30   <table border = "3" cellspacing = "3" style = "height: 90px; width: 150px;
31     font-size: 10pt" cellpadding = "1">
32     <tr>
33       <td colspan = "3">
34         <strong>Username:</strong>
35       </td>
36     </tr>
37     <tr>
38       <td colspan = "3">
39         <input size = "40" name = "USERNAME" style = "height: 22px; width: 115px" />
40       </td>
41     </tr>
42     <tr>
43       <td colspan = "3">
44         <strong>Password:</strong>
45       </td>
46     </tr>
47     <tr>
48       <td colspan = "3">
49         <input size = "40" name = "PASSWORD" style = "height: 22px; width: 115px"
50           type = "password" />
51         <br/></td>
52     </tr>

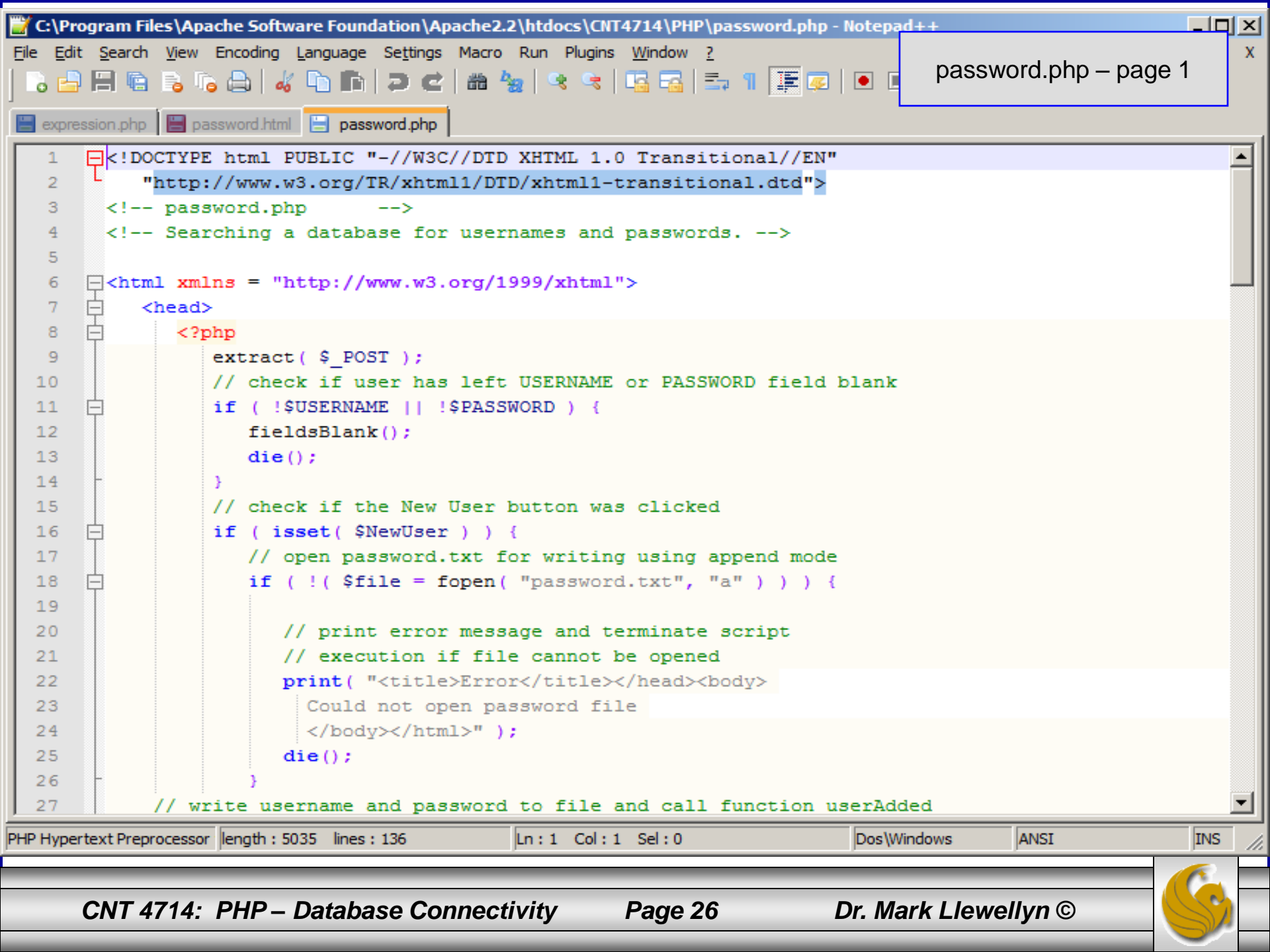
```



```
42 <tr>
43     <td colspan = "3">
44         <strong>Password:</strong>
45     </td>
46 </tr>
47 <tr>
48     <td colspan = "3">
49         <input size = "40" name = "PASSWORD" style = "height: 22px; width: 115px"
50             type = "password" />
51         <br/></td>
52 </tr>
53 <tr>
54     <td colspan = "1">
55         <input type = "submit" name = "Enter" value = "Enter" style = "height: 23px;
56             width: 47px" />
57     </td>
58     <td colspan = "2">
59         <input type = "submit" name = "NewUser" value = "New User" style = "height: 2
60     </td>
61 </tr>
62 </table>
63 </form>
64 <HR>
65 </body>
66 </html>
```







```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- password.php -->
4 <!-- Searching a database for usernames and passwords. -->
5
6 <html xmlns = "http://www.w3.org/1999/xhtml">
7 <head>
8 <?php
9     extract( $_POST );
10    // check if user has left USERNAME or PASSWORD field blank
11    if ( !$USERNAME || !$PASSWORD ) {
12        fieldsBlank();
13        die();
14    }
15    // check if the New User button was clicked
16    if ( isset( $NewUser ) ) {
17        // open password.txt for writing using append mode
18        if ( !( $file = fopen( "password.txt", "a" ) ) ) {
19
20            // print error message and terminate script
21            // execution if file cannot be opened
22            print( "<title>Error</title></head><body>
23                Could not open password file
24                </body></html>" );
25            die();
26        }
27    // write username and password to file and call function userAdded
```





```
27 // write username and password to file and call function userAdded
28     fputs( $file, "$USERNAME,$PASSWORD\n" );
29     userAdded( $USERNAME );
30 }
31 else {
32     // if a new user is not being added, open file
33     // for reading
34     if ( !( $file = fopen( "password.txt", "r" ) ) ) {
35         print( "<title>Error</title></head>
36             <body >Could not open password file
37             </body></html>" );
38         die();
39     }
40
41     $userVerified = 0;
42
43     // read each line in file and check username and password
44     while ( !feof( $file ) && !$userVerified ) {
45
46         // read line from file
47         $line = fgets( $file, 255 );
48
49         // remove newline character from end of line
50         $line = chop( $line );
51
52         // split username and password using comma delimited string
53
```

```
53         // split username and password using comma delimited string
54         $field = split( ",", $line, 2 );
55     // verify username
56     if ( $USERNAME == $field[ 0 ] ) {
57         $userVerified = 1;
58
59         // call function checkPassword to verify user's password
60         if ( checkPassword( $PASSWORD, $field ) == true )
61             accessGranted( $USERNAME );
62         else
63             wrongPassword();
64     }
65 }
66 // close text file
67 fclose( $file );
68
69 // call function accessDenied if username has not been verified
70 if ( !$userVerified )
71     accessDenied();
72 }
73
74 // verify user password and return a boolean
75 function checkPassword( $userpassword, $filedata )
76 {
77     if ( $userpassword == $filedata[ 1 ] )
78         return true;
```



```

78         return true;
79     else
80         return false;
81 }
82 // print a message indicating the user has been added
83 function userAdded( $name ) {
84     print( "<title>Thank You</title></head>
85         <body style = \"font-family: arial; font-size: 2em; color: blue;\">
86         <strong>You have been added
87         to the user list, $name. Please remember your password.
88         <br />Enjoy the site.</strong>" );
89 }
90
91 // print a message indicating permission has been granted
92 function accessGranted( $name ) {
93     print( "<title>Thank You</title></head>
94         <body style = \"font-family: arial; font-size: 2em; color: blue\">
95         <strong>Permission has been
96         granted, $name. <br />
97         Enjoy the site.</strong>" );
98 }
99 // print a message indicating password is invalid
100 function wrongPassword() {
101     print( "<title>Access Denied</title></head>
102         <body style = \"font-family: arial; font-size: 2em; color: red\">
103         <strong>You entered an invalid

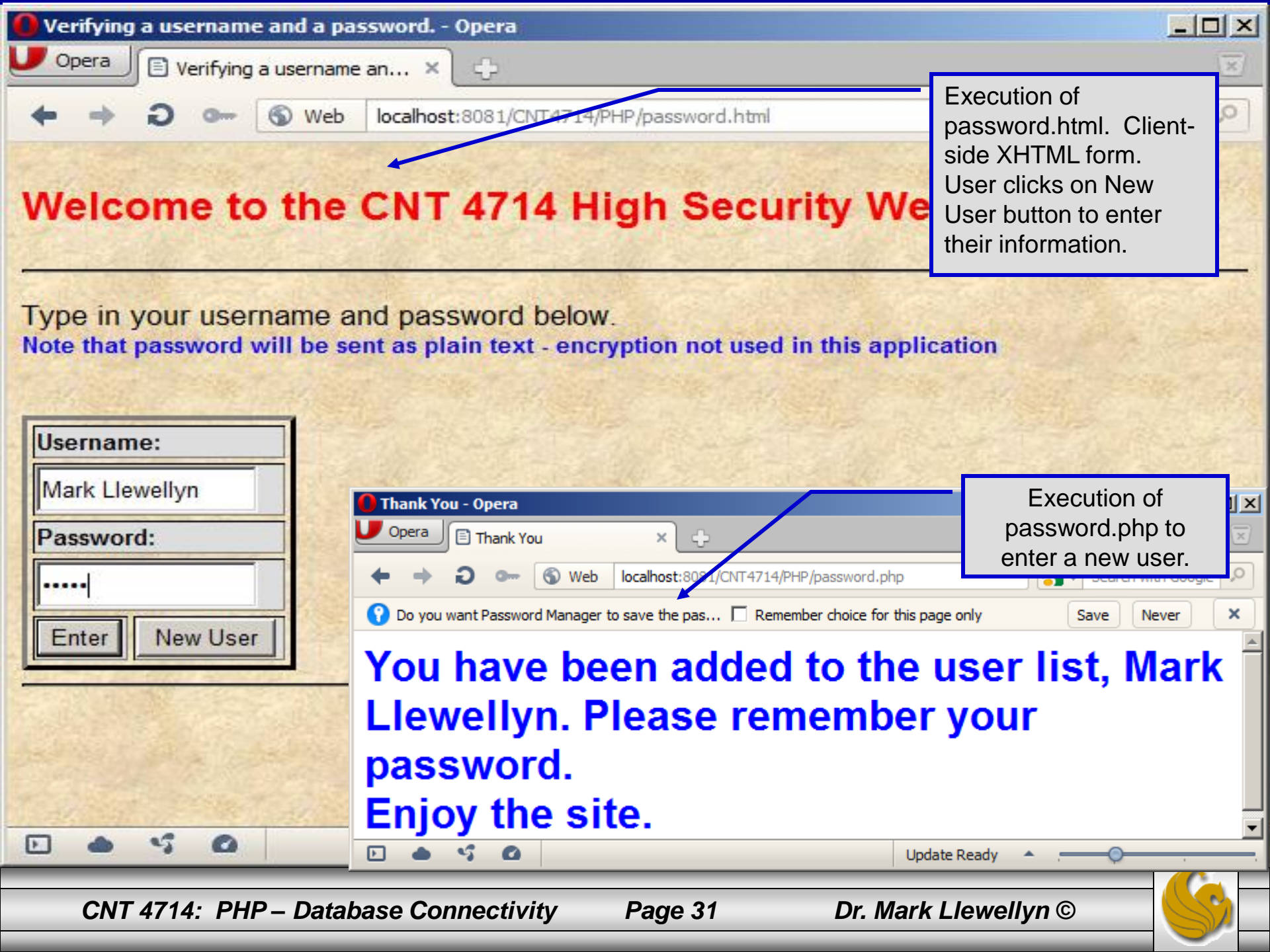
```



```
99 // print a message indicating password is invalid
100 function wrongPassword() {
101     print( "<title>Access Denied</title></head>
102           <body style = \"font-family: arial; font-size: 2em; color: red\">
103           <strong>You entered an invalid password.<br />Access has been denied.</strong>"
104         )
105 // print a message indicating access has been denied
106 function accessDenied() {
107     print( "<title>Access Denied</title></head>
108           <body style = \"font-family: arial; font-size: 2em; color: red\">
109           <strong>You were denied access to this server.
110           <br /></strong>" );
111 }
112
113 // print a message indicating that fields
114 // have been left blank
115 function fieldsBlank() {
116     print( "<title>Access Denied</title></head>
117           <body style = \"font-family: arial; font-size: 2em; color: red\">
118           <strong>Please fill in all form fields.<br /></strong>" );
119 }
120 ?>
121 </body>
122 </html>
123
124
```







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.

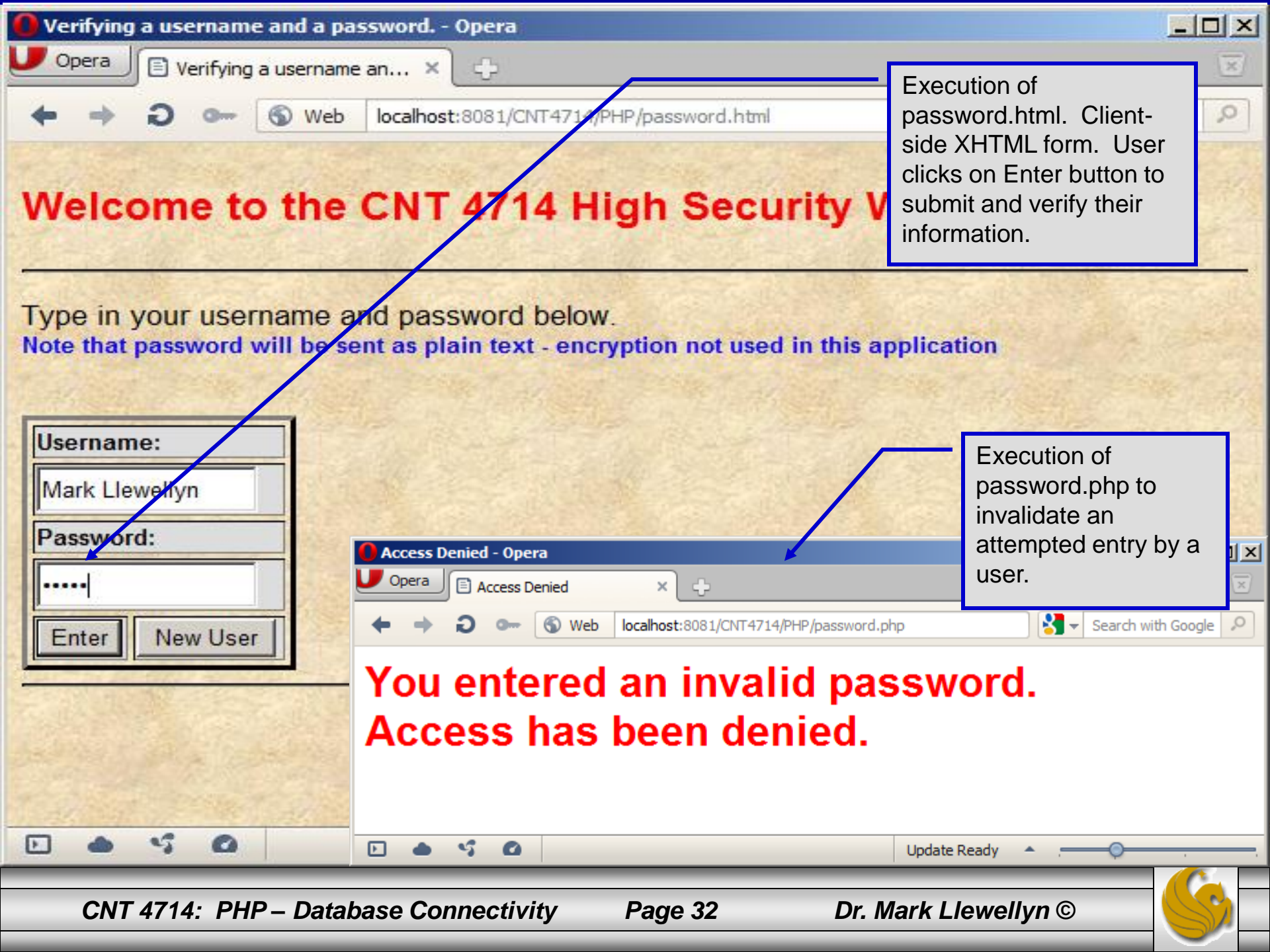
# Welcome to the CNT 4714 High Security We

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

Thank You - Opera  
Thank You  
localhost:8081/CNT4714/PHP/password.php  
Do you want Password Manager to save the pas... Remember choice for this page only Save Never  
You have been added to the user list, Mark Llewellyn. Please remember your password. Enjoy the site.  
Update Ready





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

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

Username:  
  
Password:

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

Access Denied - Opera  
Access Denied  
localhost:8081/CNT4714/PHP/password.php  
Search with Google

**You entered an invalid password.  
Access has been denied.**

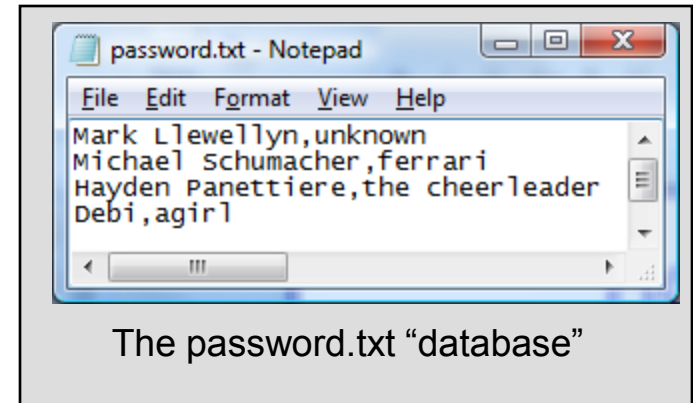
Update Ready





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

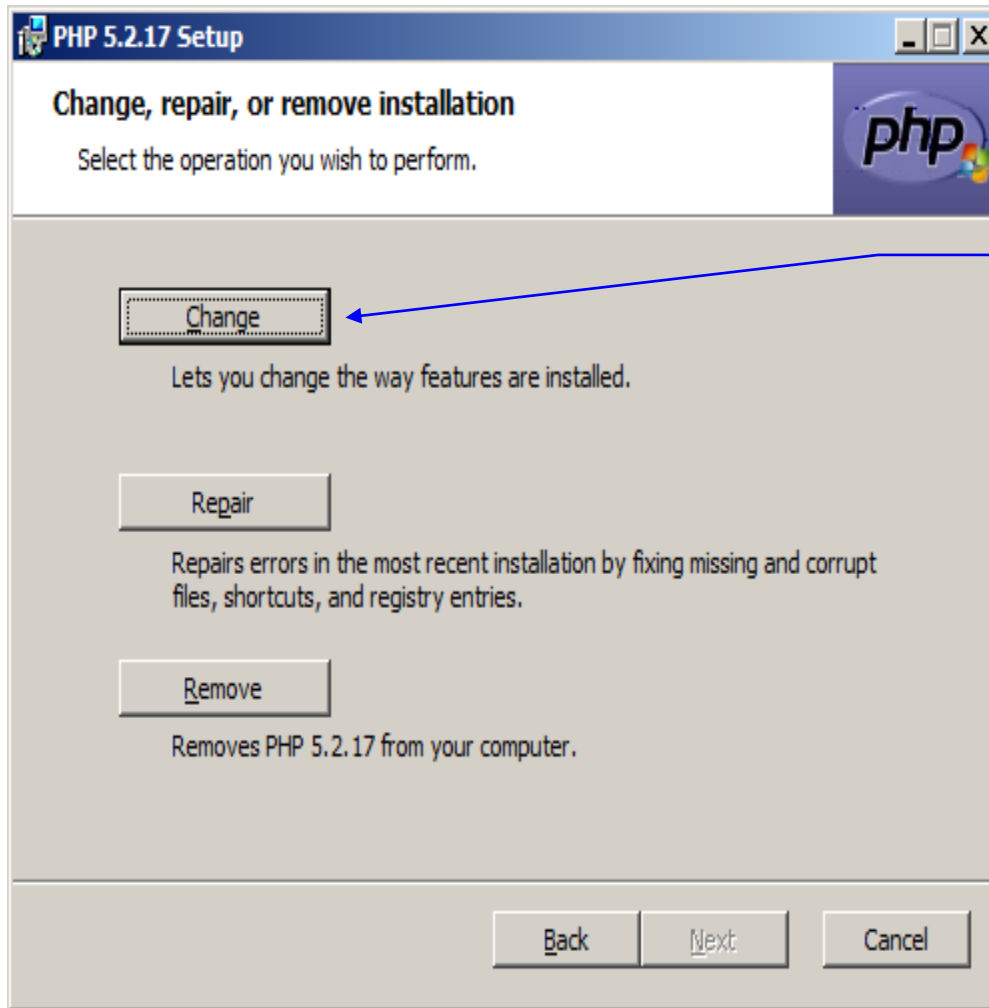


# PHP and Database Connectivity

- 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.
- Since we installed PHP 5.2.17 without any extensions loaded, you'll need to go back and re-run the PHP installer and this time, select Change (See next page).
  - Versions of MySQL greater than 4.1.0 use MySQLi extensions.
  - Versions of MySQL less than 4.1.0 use MySQL extensions.



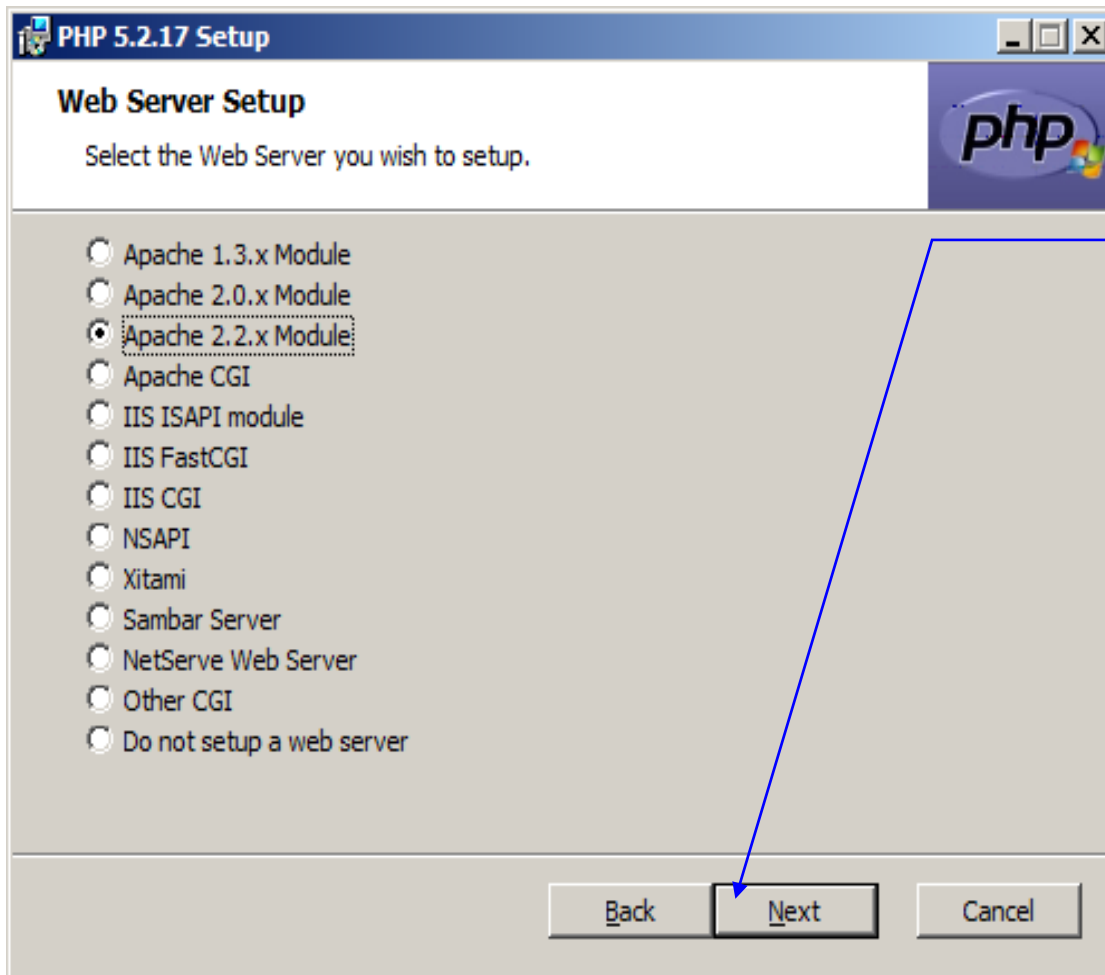
# PHP and Database Connectivity (cont.)



We need to add the `mysql` and `mysqli` extensions to PHP



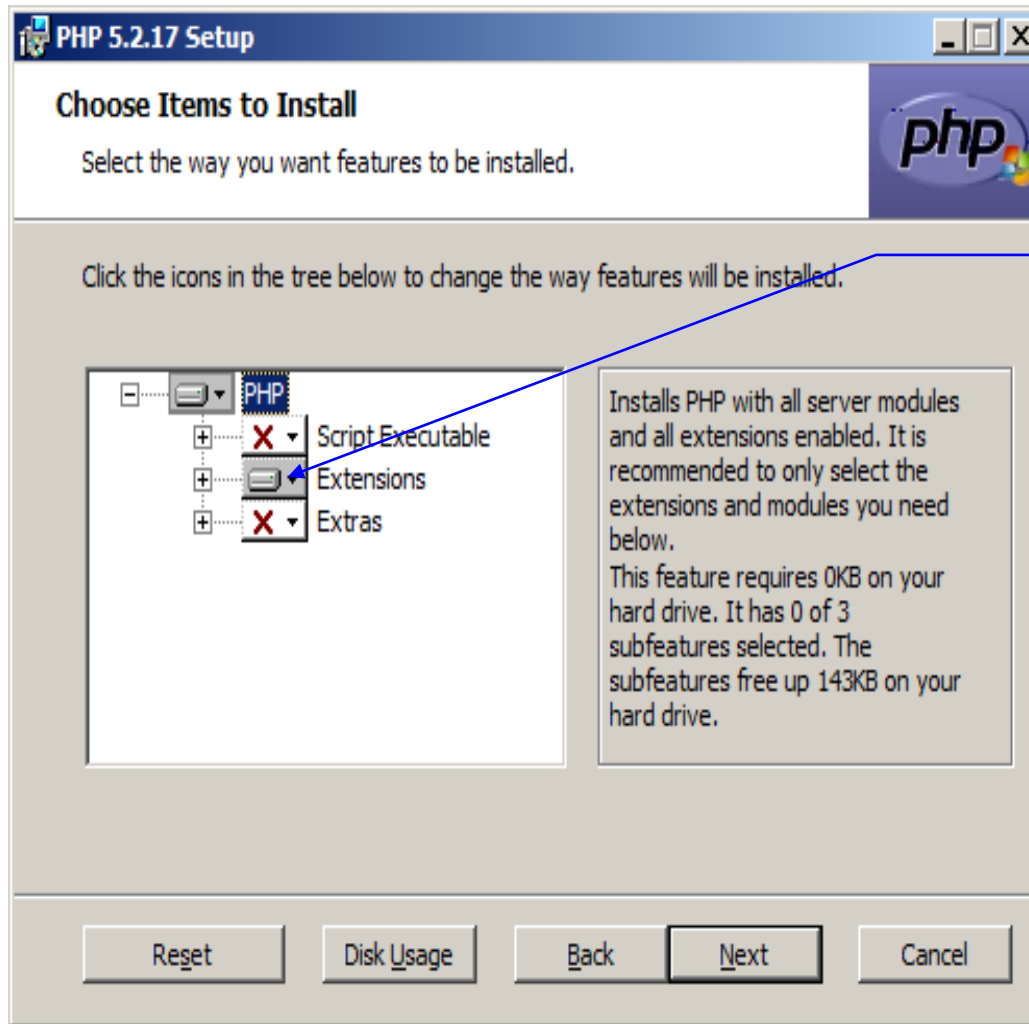
# PHP and Database Connectivity (cont.)



Leave this as it is...click Next.



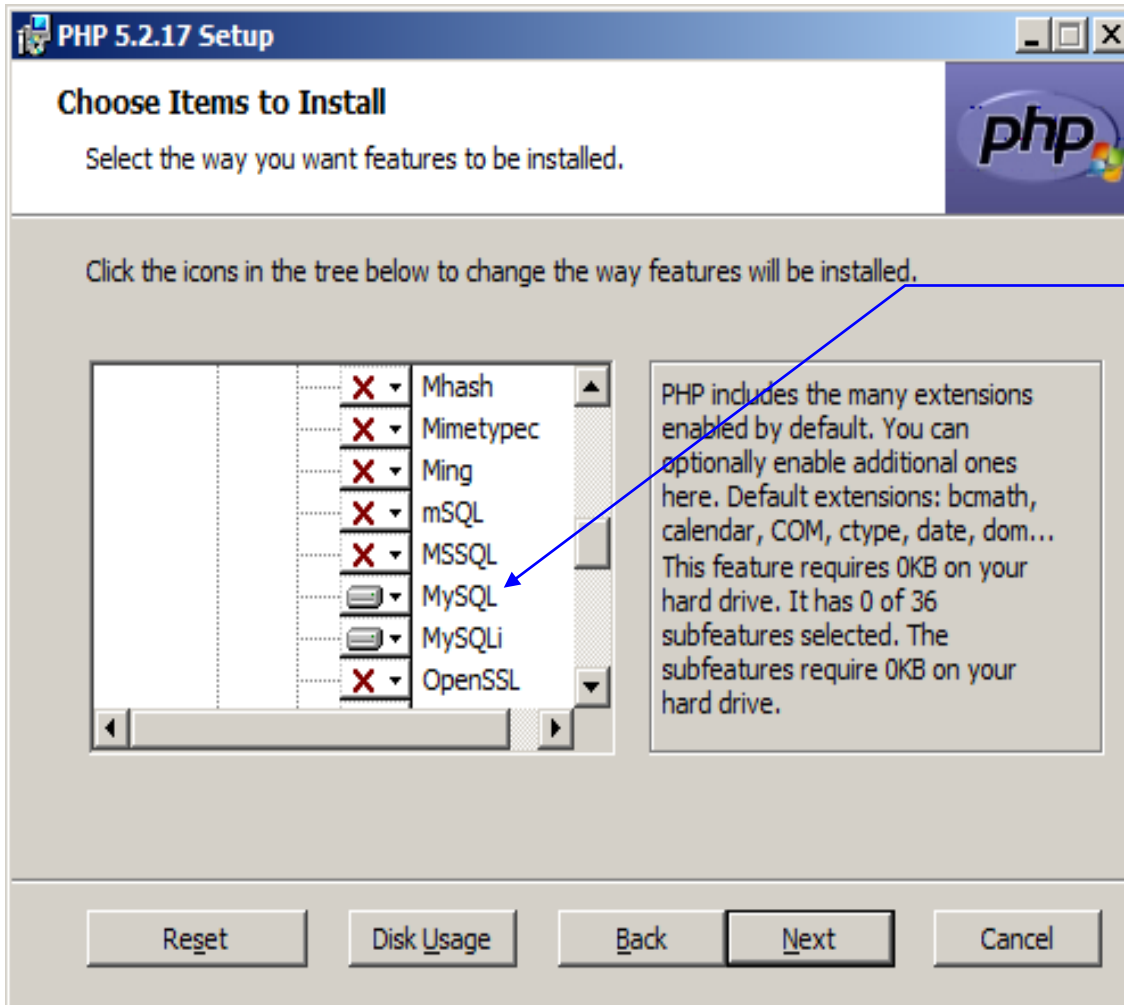
# PHP and Database Connectivity (cont.)



Select the Extensions box...it will also have a big red X in it right now. Click on the + sign to expand it.



# PHP and Database Connectivity (cont.)

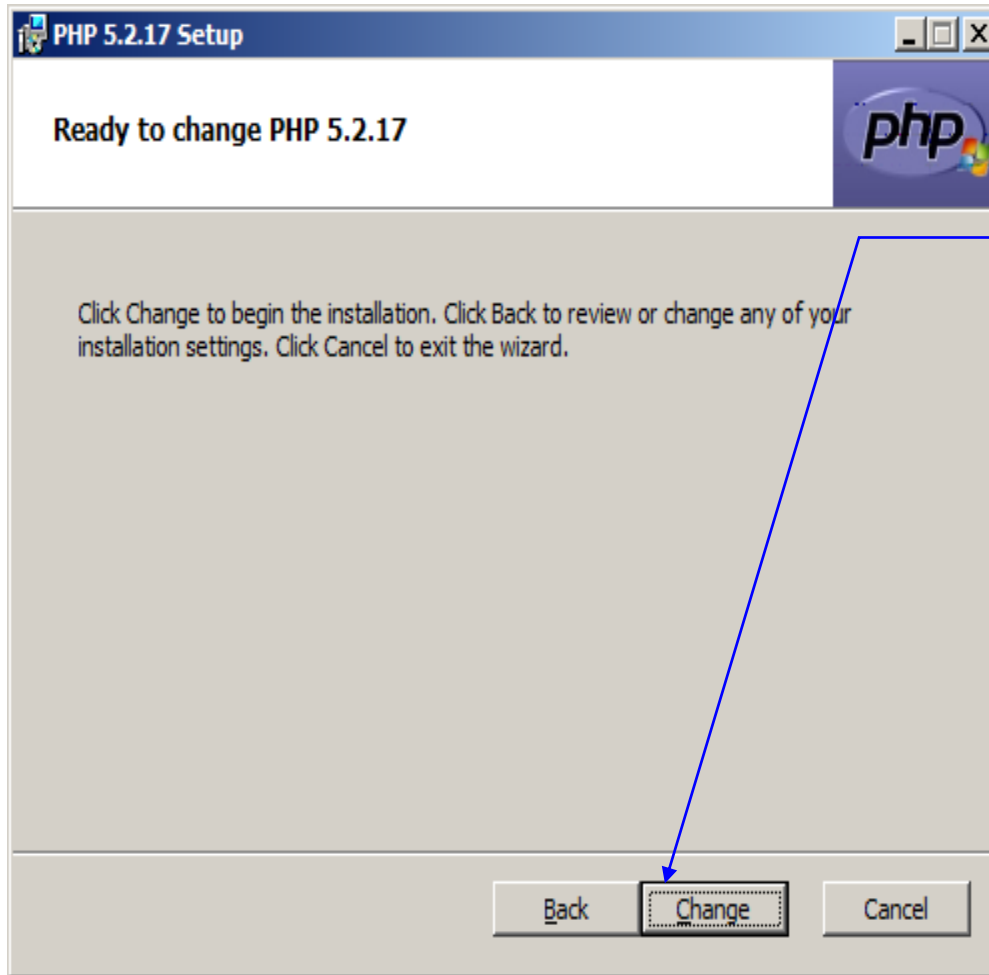


Scroll down through the list of extension until you find Mysql and MySQLi. Select both of them (the big red X will disappear). Then click Next.





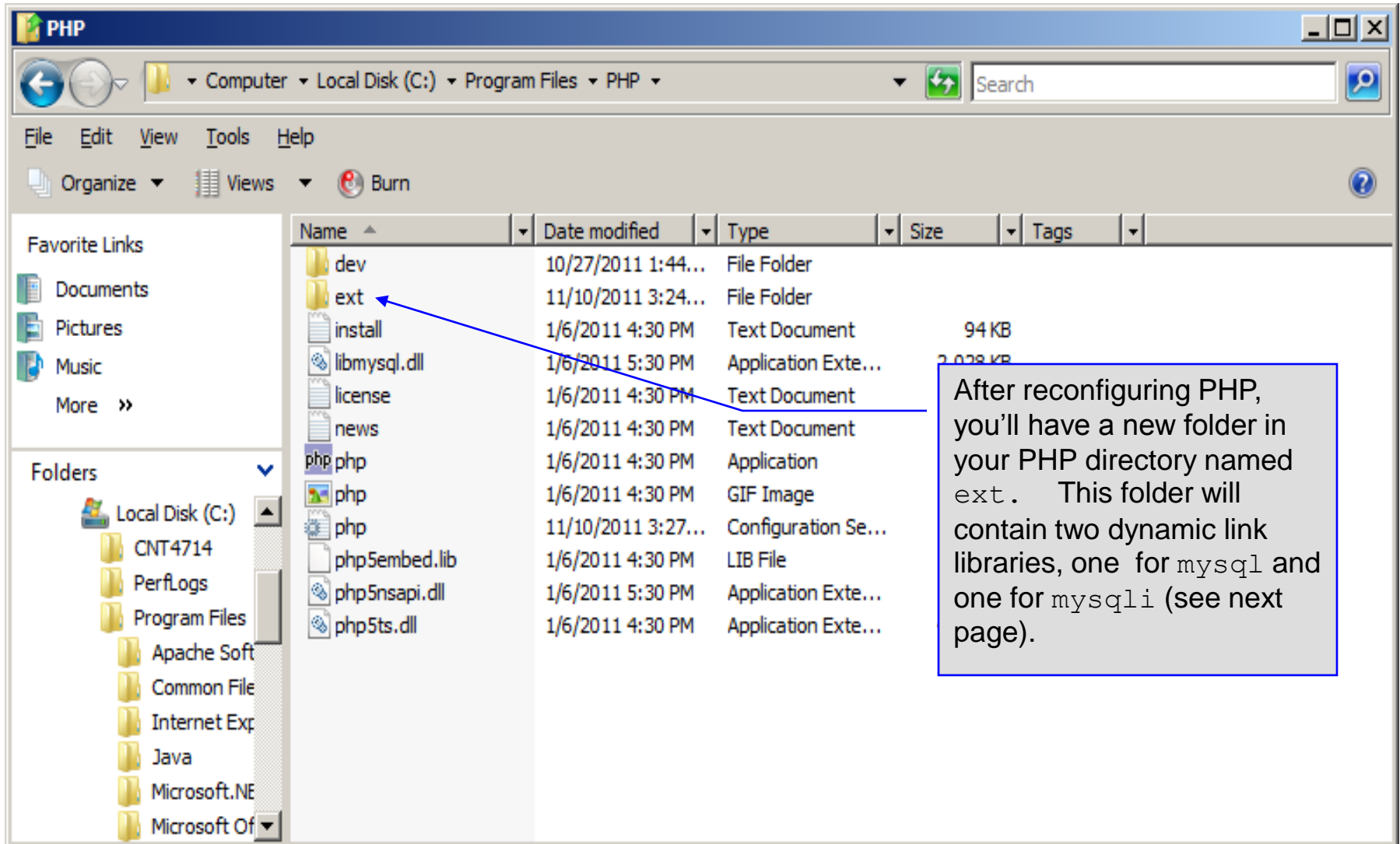
# PHP and Database Connectivity (cont.)



Click Change and you'll be done. This might require a reboot depending on your system. Just follow the PHP prompts.



# PHP and Database Connectivity (cont.)



The screenshot shows a Windows Explorer window titled 'PHP' with the address bar set to 'Computer > Local Disk (C:) > Program Files > PHP'. The main pane displays a list of files and folders:

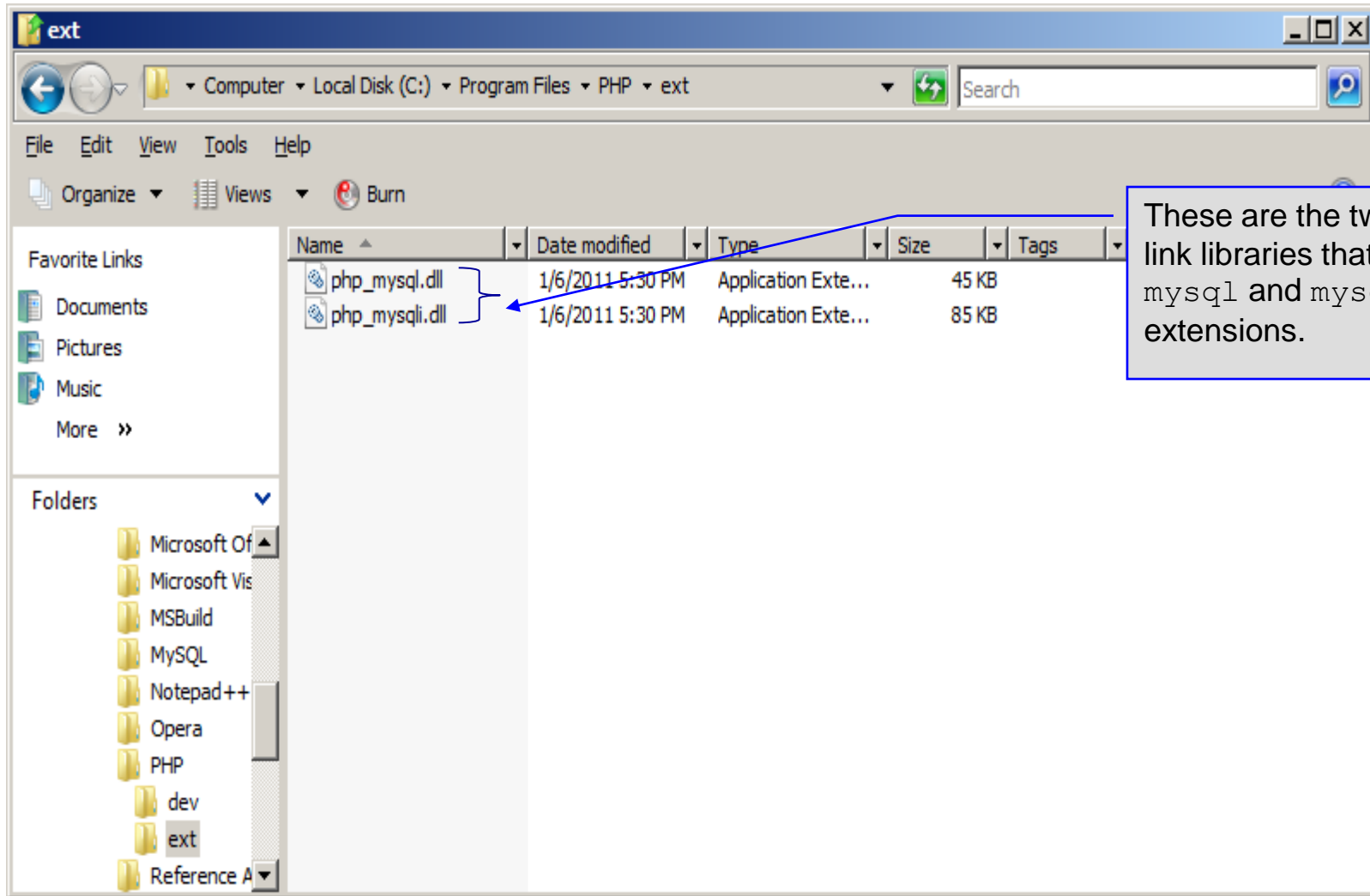
Name	Date modified	Type	Size	Tags
dev	10/27/2011 1:44...	File Folder		
ext	11/10/2011 3:24...	File Folder		
install	1/6/2011 4:30 PM	Text Document	94 KB	
libmysql.dll	1/6/2011 5:30 PM	Application Ext...	2,078 KB	
license	1/6/2011 4:30 PM	Text Document		
news	1/6/2011 4:30 PM	Text Document		
php.php	1/6/2011 4:30 PM	Application		
php	1/6/2011 4:30 PM	GIF Image		
php	11/10/2011 3:27...	Configuration Se...		
php5embed.lib	1/6/2011 4:30 PM	LIB File		
php5sapi.dll	1/6/2011 5:30 PM	Application Ext...		
php5ts.dll	1/6/2011 4:30 PM	Application Ext...		

A blue arrow points from the 'ext' folder in the list to a text box on the right. The text box contains the following text:

After reconfiguring PHP, you'll have a new folder in your PHP directory named `ext`. This folder will contain two dynamic link libraries, one for `mysql` and one for `mysql_i` (see next page).

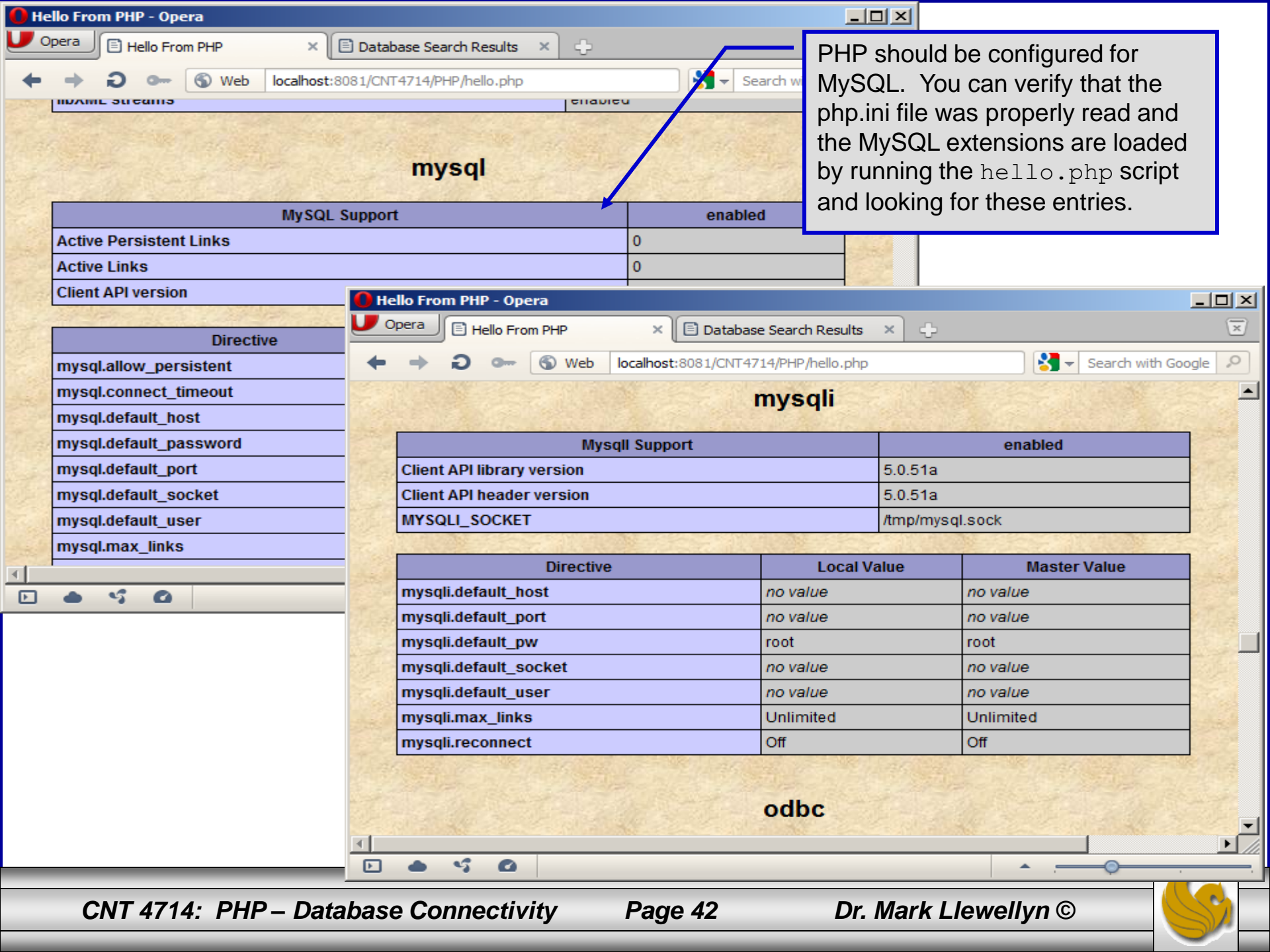


# PHP and Database Connectivity (cont.)



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





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 `hello.php` script and looking for these entries.

## mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	

Directive
mysql.allow_persistent
mysql.connect_timeout
mysql.default_host
mysql.default_password
mysql.default_port
mysql.default_socket
mysql.default_user
mysql.max_links

Second browser window showing 'mysqli' and 'odbc' sections.

## mysqli

Mysqli Support	enabled
Client API library version	5.0.51a
Client API header version	5.0.51a
MYSQLI_SOCKET	/tmp/mysql.sock

Directive	Local Value	Master Value
mysqli.default_host	no value	no value
mysqli.default_port	no value	no value
mysqli.default_pw	root	root
mysqli.default_socket	no value	no value
mysqli.default_user	no value	no value
mysqli.max_links	Unlimited	Unlimited
mysqli.reconnect	Off	Off

## odbc





# 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.mysql.php>.



PHP: Mysql - Manual - Windows Internet Explorer

http://www.php.net/manual/en/book.mysql.php

php 5.2.17 and mysql

File Edit View Favorites Tools Help

Google

Sign In

PHP: Mysql - Manual

php

downloads | documentation | faq | getting help | mailing lists | licenses

search for \_\_\_\_\_ in the function list

PHP Manual

Function Reference

Database Extensions

Vendor Specific Database Extensions

- CUBRID
- DB++
- dBase
- filePro
- Firebird/InterBase
- FrontBase
- IBM DB2
- Informix
- Ingres
- MaxDB
- Mongo
- MySQL

«mysql\_unbuffered\_query Introduction»

view this page in  [edit] Last updated: Fri, 04 Nov 2011

## MySQL Improved Extension

- [Introduction](#)
- [Overview](#)
- [Installing/Configuring](#)
  - [Requirements](#)
  - [Installation](#)
  - [Runtime Configuration](#)
  - [Resource Types](#)
- [The mysql Extension and Persistent Connections](#)
- [Predefined Constants](#)
- [Notes](#)

Internet | Protected Mode: Off

100%

The PHP online documentation for the `mysql` and `mysqli` extensions are quite complete. I suggest you bookmark them now so that you can refer to this documentation.



# 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.



```
C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\CNT4714\PHP\data.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
data.html database.php
6 <head>
7 <title>Sample Database Query From PHP</title>
8 </head>
9 <body style = "background-color: #545454" background=image1.jpg > <!--F0E68C -->
10 <h2 style = "font-family: arial color: blue">
11 Querying a MySQL database from a PHP Script.
12 </h2>
13 <form method = "post" action = "database.php">
14 <p>Select a field to display:
15 <!-- add a select box containing options -->
16 <!-- for SELECT query -->
17 <select name = "select">
18 <option selected = "selected">*</option>
19 <option>bikename</option>
20 <option>size</option>
21 <option>color</option>
22 <option>cost</option>
23 <option>purchased</option>
24 <option>mileage</option>
25 </select>
26 </p>
27 <input type = "submit" value = "Send Query" style = "background-color: blue;
28 color: yellow; font-weight: bold" />
29 </form>
Hyper Text Markup length : 1267 lines : 35 Ln : 9 Col : 64 Sel : 0 Macintosh ANSI INS
```

data.html  
Client side





database.php  
Server side  
Page 1

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- database.php -->
4 <!-- Program to query a database and send results to the client. -->
5
6 <html xmlns = "http://www.w3.org/1999/xhtml">
7 <head>
8 <title>Database Search Results</title>
9 </head>
10 <body style = "font-family: arial, sans-serif"
11 style = "background-color: #4A766E" background=image1.jpg link=blue vlink=blue>
12 <?php
13     extract( $_POST );
14     // build SELECT query
15     $query = "SELECT " . $select . " FROM bikes";
16     // Connect to MySQL
17     if ( !( $database = mysqli_connect( "localhost",
18         "root", "root", "bikedb" ) ) )
19         die( "Could not connect to database" );
20     // query bikedb database
21     if ( !( $result = mysqli_query( $database, $query ) ) ) {
22         print( "Could not execute query! <br />" );
23         die( mysqli_error() );
24     }
25     ?>
26 <h3 style = "color: blue">

```

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.





database.php  
Server side  
Page 2

```

26 <h3 style = "color: blue">
27 Database Search Results</h3>
28 <table border = "1" cellpadding = "3" cellspacing = "3"
29 style = "background-color: #00FFFF" <!-- ADD8E6 -->
30
31 <?php
32 // fetch meta-data
33 $metadata = mysqli_fetch_fields( $result);
34 print("<tr>");
35 for ($i=0; $i<count($metadata); $i++){
36 print("<td>");
37 printf("%s", $metadata[$i]->name);
38 print("</td>");
39 }
40 print("</tr>");
41
42 // fetch each record in result set
43 for ( $counter = 0;
44 $row = mysqli_fetch_row( $result );
45 $counter++ ){
46
47 // build table to display results
48 print( "<tr>" );
49
50 foreach ( $row as $key => $value )
51 print( "<td>$value</td>" );

```

Get metadata for the query

Display metadata in the top row of the table



database.php

Server side

Page 3

```

43         for ( $counter = 0;
44             $row = mysqli_fetch_row( $result );
45             $counter++ ){
46
47             // build table to display results
48             print( "<tr>" );
49
50             foreach ( $row as $key => $value )
51                 print( "<td>$value</td>" );
52
53             print( "</tr>" );
54         }
55         mysqli_close( $database );
56     ?>
57 </table>
58 <br />Your search yielded <strong>
59 <?php print( "$counter" ) ?> results.<br /><br /></strong>
60 <h5>Please email comments to
61     <a href = "mailto:markl@cs.ucf.edu">
62     markl@cs.ucf.edu
63     </a>
64 </h5>
65 </body>
66 </html>
67
68

```



# Execution of data.html – Client side

Sample Database Query From PHP - Opera

Opera Hello From PHP Sample Database Query ...

Web localhost:8081/CNT4714/PHP/data.html Search with Google

## Querying a MySQL database from a PHP Script.

Select a field to display: \*

**Send Query**

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

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 - Opera

Opera Hello From PHP Database Search Results

Web localhost:8081/CNT4714/PHP/database.php Search with Google

### 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
Bianchi Infinito	58	celeste	8900	2011-07-14	0
BMC SLC01 - Swiss	58	red/black/white	8000	2010-06-23	0
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
Ridley Damocles	58	blue/black	7500	2008-06-27	0
Ridley X-Fire	58	red/white	7500	2011-09-01	0
Schwinn Paramount P14	60	blue	1800	1992-03-01	200

Your search yielded 14 results.

Please email comments to [markl@cs.ucf.edu](mailto:markl@cs.ucf.edu)

Results of query **SELECT \* FROM bikes**. Display indicates that 14 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.



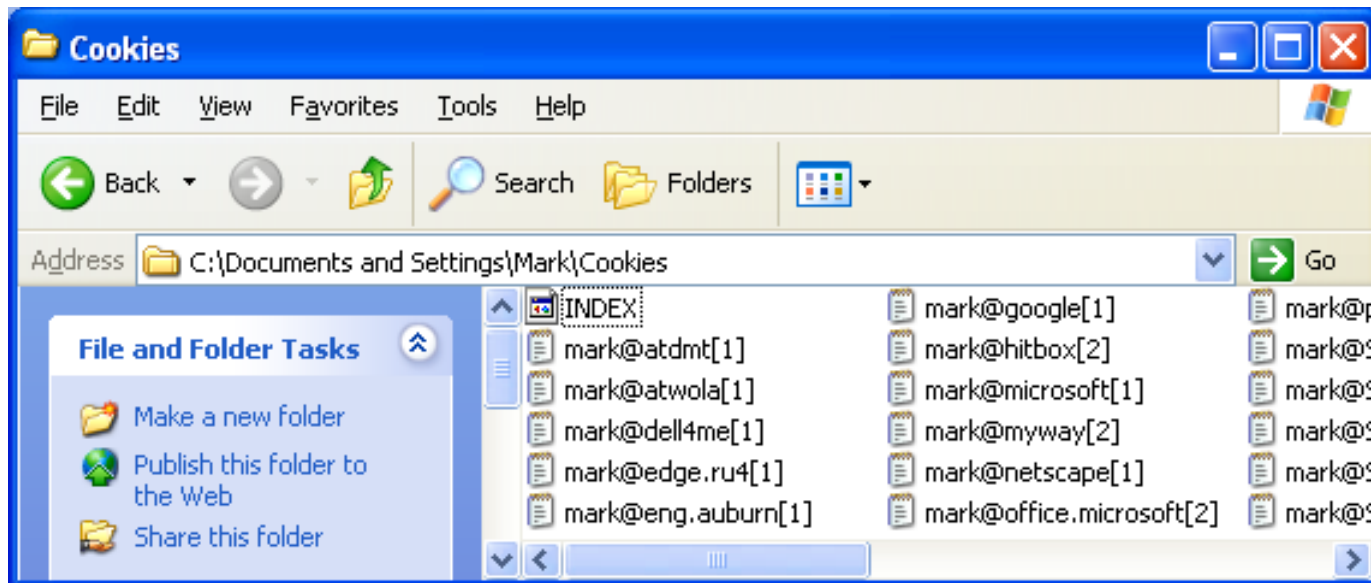
# Cookies (cont.)

- 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.



- 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.



```
*C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\CNT4714\PHP\cookies.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
cookies.html
data.html database.php cookies.html
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- cookies.html -->
4 <!-- Writing a Cookie -->
5
6 <html xmlns = "http://www.w3.org/1999/xhtml">
7 <head>
8 <title>Writing a cookie to the client computer</title>
9 </head>
10 <body style = "font-family: arial, sans-serif;
11 background-color: #856363" background=images/background.jpg>
12 <h2>Click Write Cookie to save your cookie data.</h2>
13 <form method = "post" action = "cookies.php"
14 style = "font-size: 10pt" background-color: #856363">
15 <strong>Name:</strong><br />
16 <input type = "text" name = "NAME" /><br />
17 <strong>Height:</strong><br />
18 <input type = "text" name = "HEIGHT" /><br />
19 <strong>Favorite Color:</strong><br />
20 <input type = "text" name = "COLOR" /><br />
21 <p>
22 <input type = "submit" value = "Write Cookie"
23 style = "background-color: #0000FF; color: yellow; font-weight: bold" />
24 </p>
25 </form>
26 </body>
```

Hyper Text Markup length : 1077 lines : 31 Ln : 27 Col : 1 Sel : 0 Macintosh ANSI INS





## cookies.php – page 1

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.

```
1 <?php
2 // cookies.php
3 // Program to write a cookie to a client's machine
4
5 extract( $_POST );
6 // write each form field's value to a cookie and set
7 // cookie's expiration date
8 setcookie( "Name", $NAME, time() + 60 * 60 * 24 * 5 );
9 setcookie( "Height", $HEIGHT, time() + 60 * 60 * 24 * 5 );
10 setcookie( "Color", $COLOR, time() + 60 * 60 * 24 * 5 );
11 ?>
12
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
14 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
15 <html xmlns = "http://www.w3.org/1999/xhtml">
16 <head>
17     <title>Cookie Saved</title>
18 </head>
19 <body style = "font-family: arial, sans-serif", background=
20 <p><b>The cookie has been set with the following data:</b>
21 <!-- print each form field's value -->
22 <br /><span style = "color: blue">Name:</span>
23     <?php print( $NAME ) ?><br />
24 <span style = "color: blue">Height:</span>
25     <?php print( $HEIGHT ) ?><br />
26 <span style = "color: blue">Favorite Color:</span>
27 <span style = "color: <?php print( "$COLOR\">$COLOR" ) ?>
```

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

Opera Hello From PHP Writing a cookie to the di...

localhost:8081/CNT4714/PHP/cookies.html Search with G

## Click Write Cookie to save your cookie data.

Name:

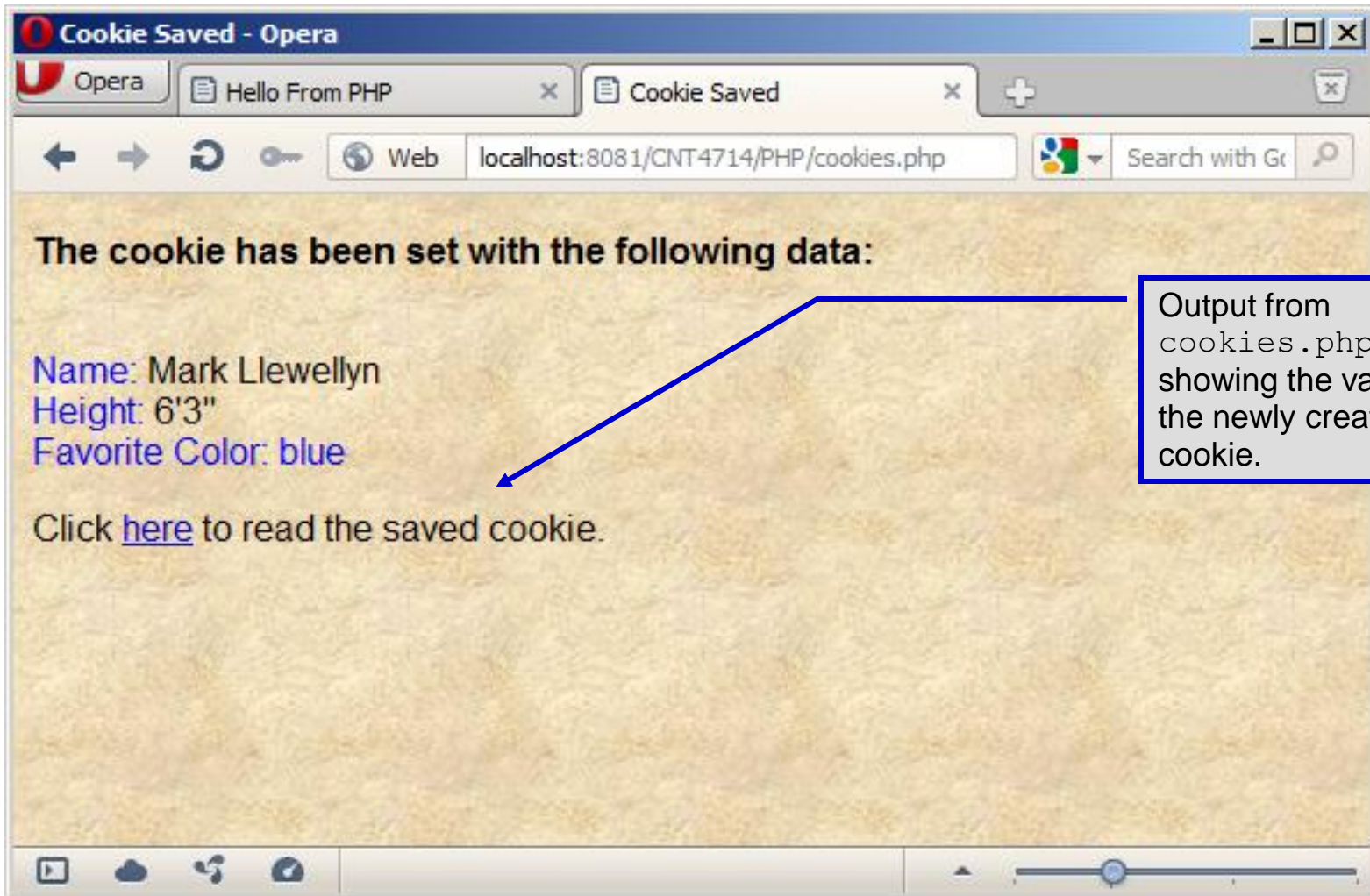
Height:

Favorite Color:

HTML form generated by cookies.html



# Cookies (cont.)

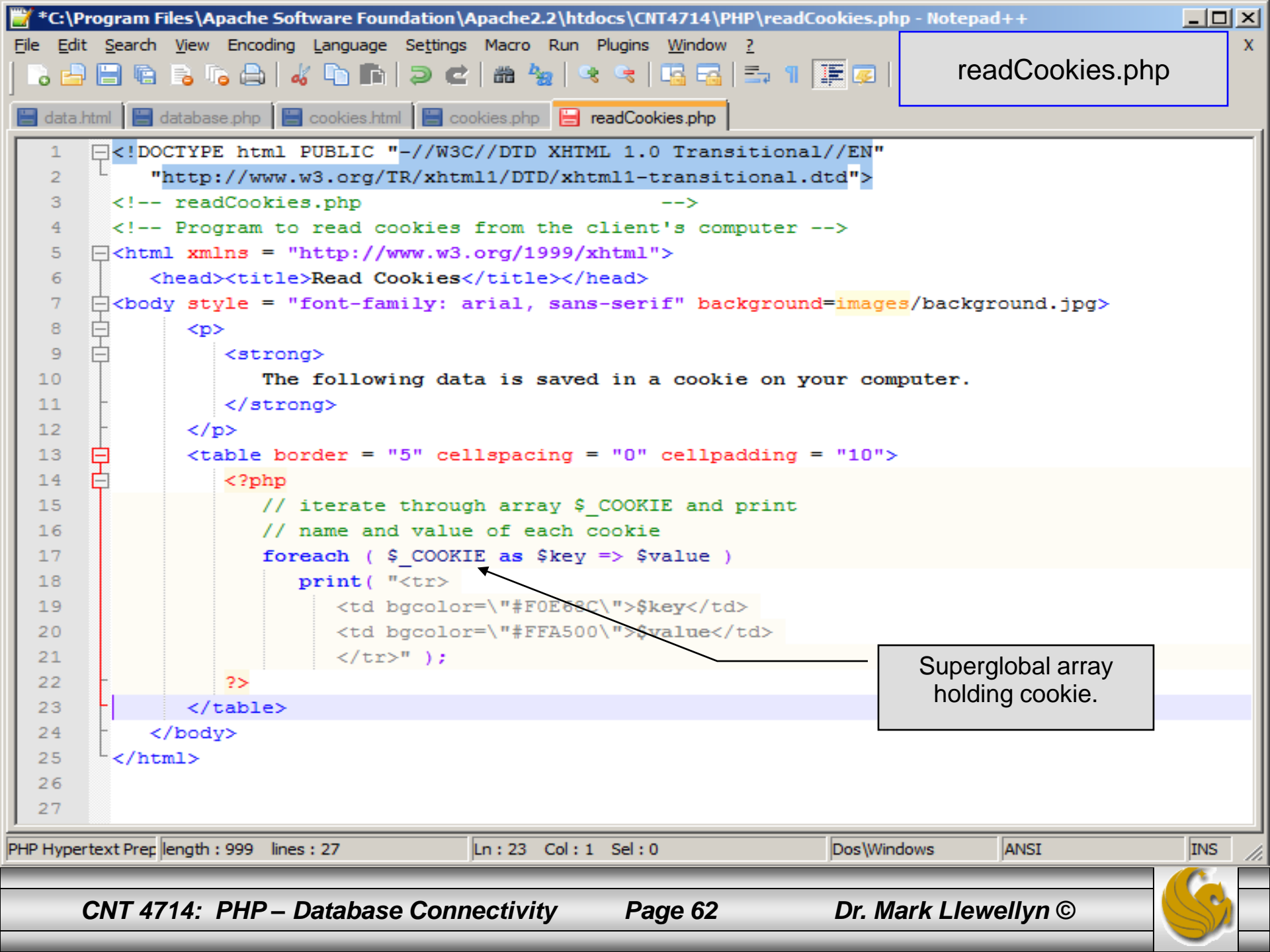


# Cookies (cont.)

- 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

data.html database.php cookies.html cookies.php readCookies.php

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <!-- readCookies.php -->
4 <!-- Program to read cookies from the client's computer -->
5 <html xmlns = "http://www.w3.org/1999/xhtml">
6   <head><title>Read Cookies</title></head>
7   <body style = "font-family: arial, sans-serif" background=images/background.jpg>
8     <p>
9       <strong>
10        The following data is saved in a cookie on your computer.
11      </strong>
12    </p>
13    <table border = "5" cellspacing = "0" cellpadding = "10">
14      <?php
15        // iterate through array $_COOKIE and print
16        // name and value of each cookie
17        foreach ( $_COOKIE as $key => $value )
18          print( "<tr>
19                <td bgcolor=#F0E68C>$key</td>
20                <td bgcolor=#FFA500>$value</td>
21                </tr>" );
22      ?>
23    </table>
24  </body>
25 </html>

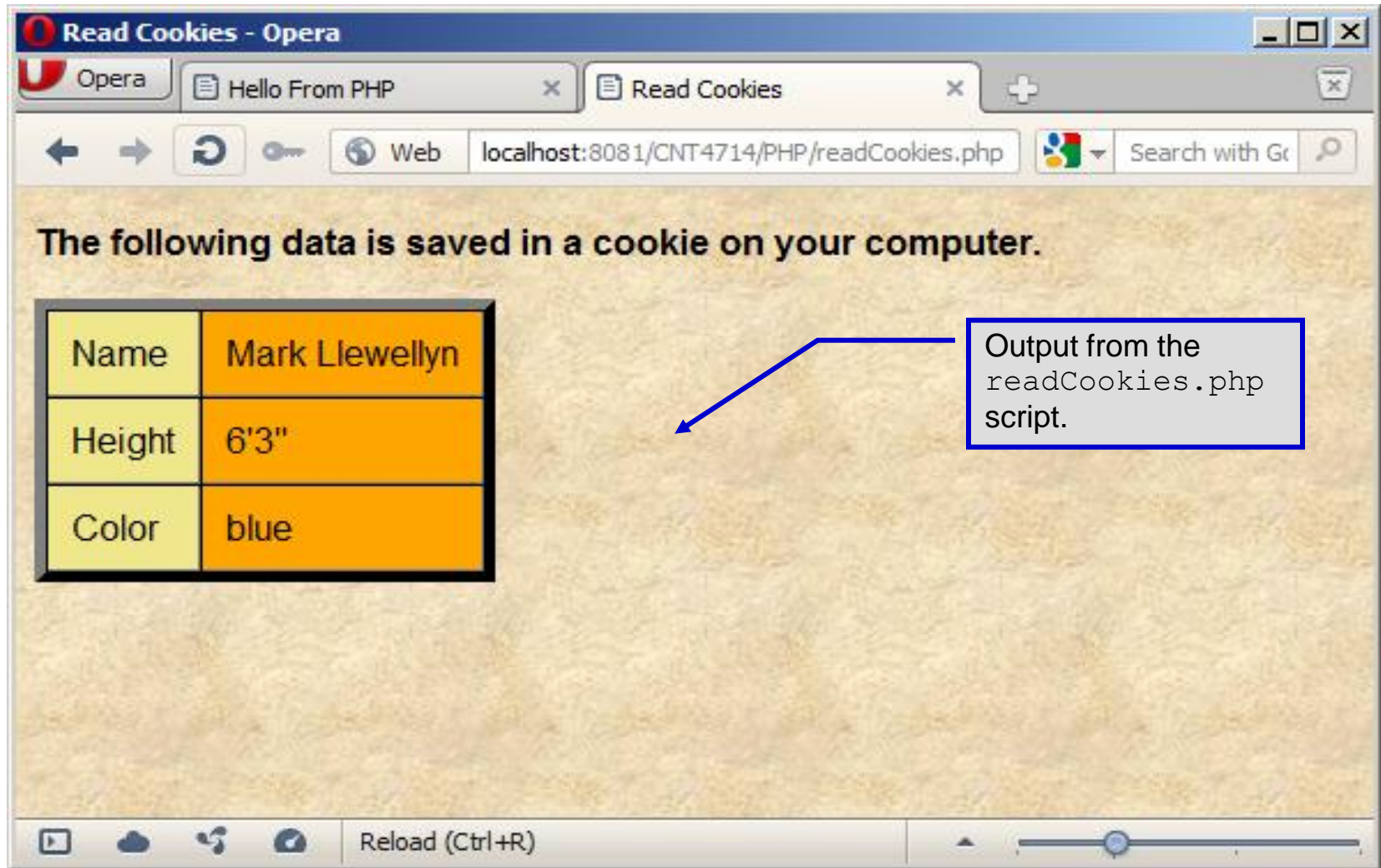
```

Superglobal array holding cookie.





# Cookies (cont.)



The screenshot shows an Opera browser window titled "Read Cookies - Opera". The address bar displays "localhost:8081/CNT4714/PHP/readCookies.php". The main content area features the text "The following data is saved in a cookie on your computer." followed by a table. A blue arrow points from a text box to the table.

Name	Mark Llewellyn
Height	6'3"
Color	blue

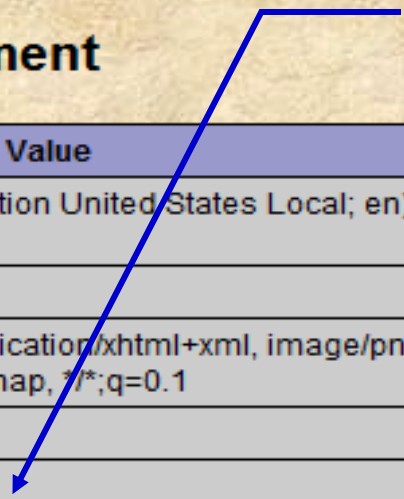
Output from the readCookies.php script.



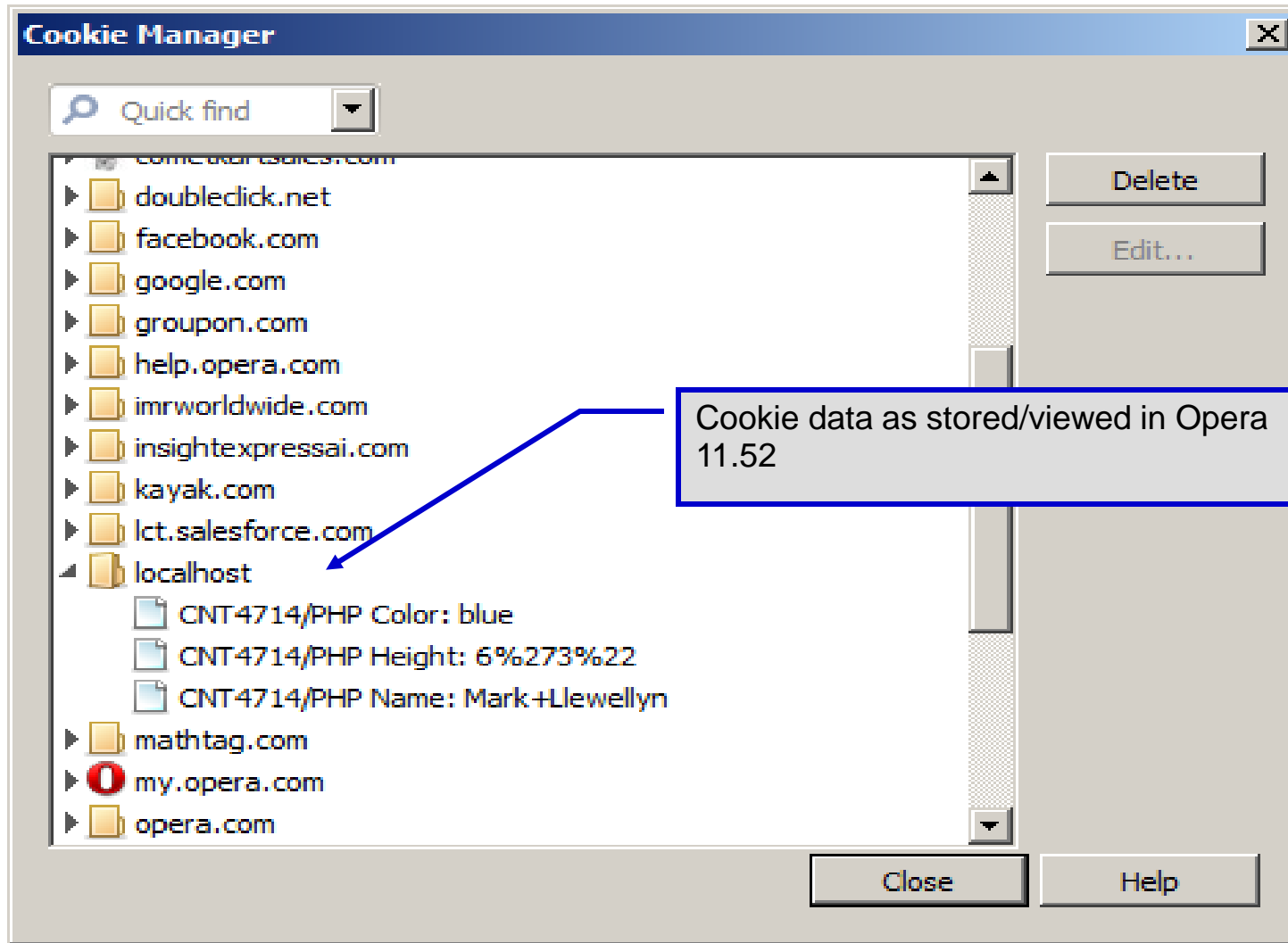
## Apache Environment

Variable	Value
HTTP_USER_AGENT	Opera/9.80 (Windows NT 6.0; U; Edition United States Local; en) Presto/2.9.168 Version/11.52
HTTP_HOST	localhost:8081
HTTP_ACCEPT	text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/webp, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.9
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_COOKIE	Name=Mark+Llewellyn; Height=6%273%22; Color=blue
HTTP_CACHE_CONTROL	no-cache
HTTP_CONNECTION	Keep-Alive
PATH	C:\Program Files\PHP\C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\MySQL\MySQL Server 5.5\bin
SystemRoot	C:\Windows
COMSPEC	C:\Windows\system32\cmd.exe
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
WINDIR	C:\Windows

Contents of the cookie stored on the client machine.



# Cookies (cont.)



# 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.





# 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 magazine would you like information about?

Operating System

Which operating system are you currently using?

Windows Vista  Windows 7  Windows XP  
 Linux  Other

User fills in the form and clicks the Register button.



Sample form to take user input in XHTML - Opera

Opera Hello From PHP Sample form to take user... x

Web localhost:8081/CNT4714/PHP/dynamicForm2.php Search with Google

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 - Opera

Opera Hello From PHP Database Search Results

Web localhost:8081/CNT4714/PHP/formDatabase2.php

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

### Mailing List Contacts

ID	Last Name	First Name	E-mail Address	Phone Number	Magazine	Operating System
0000000003	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	RadSport	Windows Vista
0000000004	Schumacher	Michael	michael@mercedes.de	(123)222-4567	RadSport	Windows 7
0000000005	Panettiere	Hayden	savethecheerleader.com	(555)123-6789	Velo-News	Other
0000000006	Swanepoel	Candice	vickisecret.com	(555)555-0987	Mirror du Cyclisme	Linux
0000000007	Einstein	Albert	its-relative.com	(111)111-1111	Pro Cycling	Other
0000000008	Cancellara	Fabian	imfast.com	(333)333-3333	Cycling Weekly	Windows Vista





Sample form to take user input in XHTML - Opera

Opera Hello From PHP Sample form to take user... x +

Web localhost:8081/CNT4714/PHP/dynamicForm2.php Search with Google

# 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

Email

Phone

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

Operating System

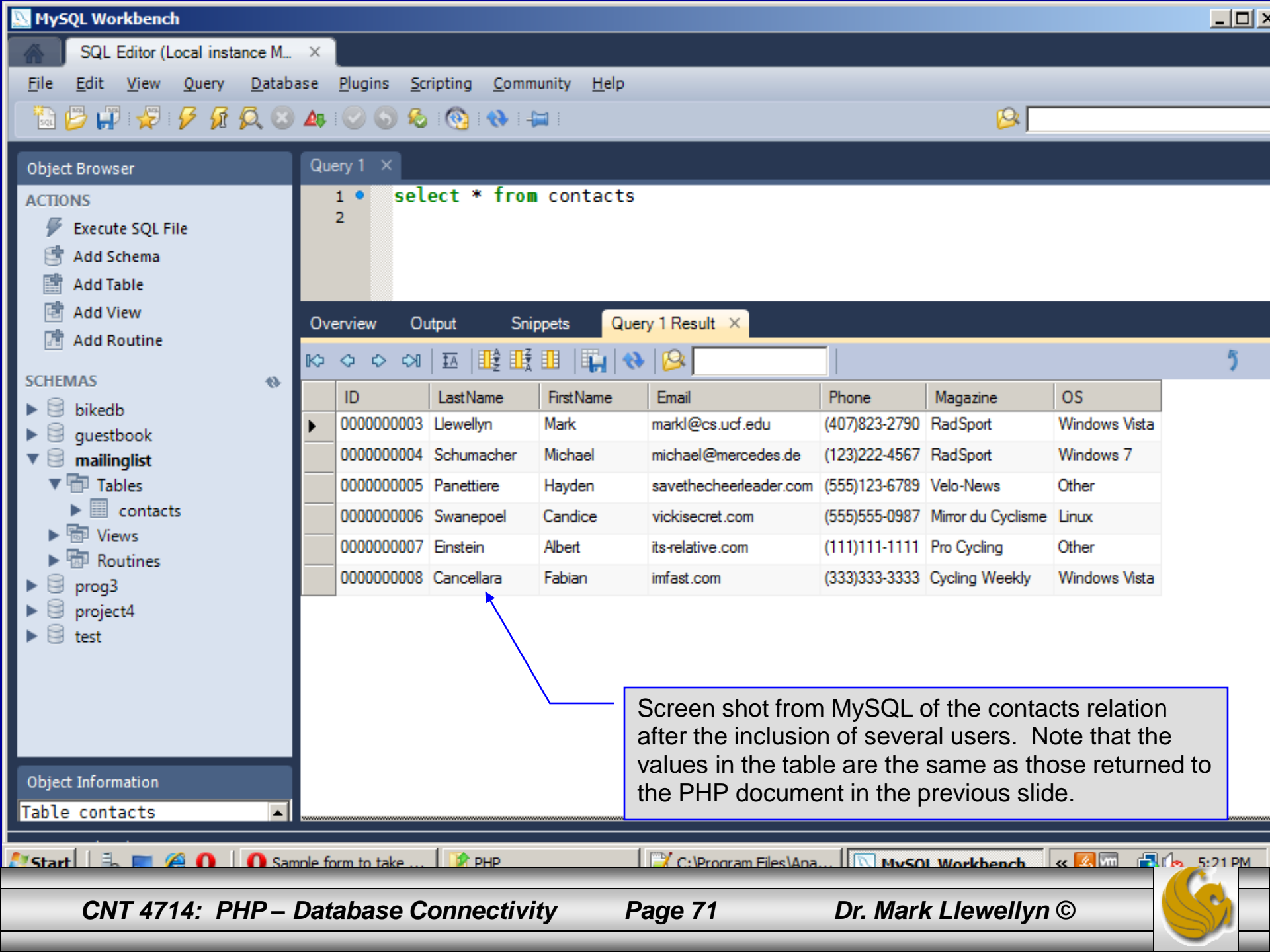
Which operating system are you currently using?

Windows Vista  Windows 7  Windows XP  
 Linux  Other

Register

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.





```
1 • select * from contacts
2
```

ID	LastName	FirstName	Email	Phone	Magazine	OS
0000000003	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	RadSport	Windows Vista
0000000004	Schumacher	Michael	michael@mercedes.de	(123)222-4567	RadSport	Windows 7
0000000005	Panettiere	Hayden	savethecheerleader.com	(555)123-6789	Velo-News	Other
0000000006	Swanepoel	Candice	vickisecret.com	(555)555-0987	Mirror du Cyclisme	Linux
0000000007	Einstein	Albert	its-relative.com	(111)111-1111	Pro Cycling	Other
0000000008	Cancellara	Fabian	imfast.com	(333)333-3333	Cycling Weekly	Windows Vista

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.



```
3
4 <!-- dynamicForm2.php -->
5
6 <html xmlns = "http://www.w3.org/1999/xhtml">
7 <head>
8 <title>Sample form to take user input in XHTML</title>
9 </head>
10 <body style = "font-family: arial, sans-serif; background-color: #856363" background=ima
11 <?php
12     extract ( $_POST );
13     $iserror = false;
14     // array of magazine titles
15     $maglist = array( "Velo-News",
16                     "Cycling Weekly",
17                     "Pro Cycling",
18                     "Cycle Sport",
19                     "RadSport",
20                     "Mirror du Cyclisme" );
21     // array of possible operating systems
22     $systemlist = array( "Windows Vista",
23                         "Windows 7",
24                         "Windows XP",
25                         "Linux",
26                         "Other");
27     // array of name and alt values for the text input fields
28     $inputlist = array( "fname" => "First Name"
```



```
28 $inputlist = array( "fname" => "First Name",
29     "lname" => "Last Name",
30     "email" => "Email",
31     "phone" => "Phone" );
32
33 if ( isset ( $submit ) ) {
34     if ( $fname == "" ) {
35         $formerrors[ "fnameerror" ] = true;
36         $iserror = true;
37     }
38     if ( $lname == "" ) {
39         $formerrors[ "lnameerror" ] = true;
40         $iserror = true;
41     }
42     if ( $email == "" ) {
43         $formerrors[ "emailerror" ] = true;
44         $iserror = true;
45     }
46     if ( !ereg( "^\([0-9]{3}\)[0-9]{3}-[0-9]{4}$", $phone ) ) {
47         $formerrors[ "phoneerror" ] = true;
48         $iserror = true;
49     }
50     if ( !$iserror ) {
51         // build INSERT query
52         $query = "INSERT INTO contacts " .
53             "(ID, LastName, FirstName, Email, Phone, Magazine, OS ) " .
54             "VALUES (null, '$lname', '$fname', '$email', " . " " . quotemeta( $phone ) .
```



```

54 // Connect to MySQL
55     if ( !( $database = mysql_connect( "localhost",
56         "root", "root" ) ) )
57         die( "Could not connect to database" );
58
59 // open MailingList database
60 if ( !mysql_select_db( "mailinglist", $database ) )
61     die( "Could not open MailingList database" );
62
63 // execute query in MailingList database
64 if ( !( $result = mysql_query( $query, $database ) ) ) {
65     print( "Could not execute query! <br />" );
66     die( mysql_error() );
67 }
68 print( "<p>Hi
69     <span style = 'color: blue'> <strong>$fname</strong></span>.
70     Thank you for completing the survey.<br />
71     You have been added to the <span style = 'color: blue'>
72     <strong>$mag</strong></span> mailing list.           </p>
73     <strong>The following information has been saved in our database:</strong><br>
74
75     <table border = '0' cellpadding = '0' cellspacing = '10'>
76     <tr>
77     <td bgcolor = '#ffffaa'>Name </td>
78     <td bgcolor = '#ffffbb'>Email</td>
79     <td bgcolor = '#ffffcc'>Phone</td>

```



```
83 <!-- print each form field's value -->
84 <td>$fname $lname</td>
85 <td>$email</td>
86 <td>$phone</td>
87 <td>$os</td>
88 </tr></table>
89 <br /><br /><br />
90 <div style = 'font-size : 10pt; text-align: center'>
91 <div style = 'font-size : 18pt'>
92 <a href = 'formDatabase2.php'>
93 Click here to view entire database.</a>
94 </div>
95 </div></body></html>" );
96 die();
97 }
98 }
99 print( "<h1>This is a sample registration form</h1>
100 Please fill in all fields and click Register." );
101 if ( $iserror ) {
102 print( "<br /><span style = 'color : red'>
103 Fields with * need to be filled in properly.</span>" );
104 }
105 print( "<!-- post form data to dynamicForm2.php -->
106 <form method = 'post' action = 'dynamicForm2.php'>
107 <img src = 'images/user.gif' alt = 'User' /><br />
108 <span style = 'color: blue'>
```

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

```

102     it( "<br /><span style = 'color : red'>
103     fields with * need to be filled in properly.</span>" );
104
105     "<!-- post form data to dynamicForm2.php -->
106     m method = 'post' action = 'dynamicForm2.php'>
107     g src = 'images/user.gif' alt = 'User' /><br />
108     n style = 'color: blue'>
109     use fill out the fields below.<br />
110     an>
111     ate four text boxes for user input -->" );
112     a ( $inputlist as $inputname => $inputalt ) {
113     ertext = $inputalt[ $inputname ];
114
115     it( "<img src = 'images/$inputname.gif'
116     alt = '$inputalt' /><input type = 'text' name = '$inputname' value = '" . $$inputname . "'
117     [ $formerrors[ ( $inputname )."error" ] == true )
118     rint( "<span style = 'color : red'>*</span>" );
119     it( "<br />" );
120
121     "<span style = 'font-size : 10pt" );
122     formerrors[ "phoneerror" ] ) print( "; color
123     "'>Must be in the form (555)555-5555
124     an><br /><br />
125     g src = 'images/downloads.gif'
126     = 'Publications' /><br />
127     n style = 'color: blue'>

```

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.



```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <!-- formDatabase2.php -->
5 <!-- Program to query a database and send results to the client. -->
6
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8   <head>     <title>Database Search Results</title>   </head>
9   <body style = "font-family: arial, sans-serif"
10     style= "background-color: #F0E68C" background=image1.jpg>
11     <?php
12         extract( $_POST );
13         // build SELECT query
14         $query = "SELECT * FROM contacts";
15         // Connect to MySQL
16         if ( !( $database = mysqli_connect( "localhost", "root", "root", MailingList ) ) )
17             die( "Could not connect to database" );
18         // query MailingList database
19         if ( !( $result = mysqli_query( $database, $query ) ) ) {
20             print( "Could not execute query! <br />" );
21             die( mysqli_error() );
22         }
23     ?>
24     <h3 style = "color: blue">
25     Mailing List Contacts</h3>
26     <table border = "1" cellpadding = "3" cellspacing = "2"

```





```

26 <table border = "1" cellpadding = "3" cellspacing = "2"
27     style = "background-color: #ADD8E6">
28     <tr>
29         <td>ID</td>
30         <td>Last Name</td>
31         <td>First Name</td>
32         <td>E-mail Address</td>
33         <td>Phone Number</td>
34         <td>Magazine</td>
35         <td>Operating System</td>
36     </tr>
37     <?php
38         // fetch each record in result set
39         for ( $counter = 0;
40             $row = mysqli_fetch_row( $result );
41             $counter++ ){
42             // build table to display results
43             print( "<tr>" );
44             foreach ( $row as $key => $value )
45                 print( "<td>$value</td>" );
46             print( "</tr>" );
47         }
48         mysqli_close( $database );
49     <?>
50 </table>
51 </body>

```



MySQL Workbench

SQL Editor (Mark-Server1VM... x)

File Edit View Query Database Plugins Scripting Community Help

ORACLE

Object Browser

SCHEMAS

- bikedb
- coloursurvey
- guestbook
- mailinglist
  - Tables
    - contacts
      - Columns
        - ID
        - LastNa
        - FirstNa
        - Email
        - Phone
        - Magaz
        - OS
      - Indexes
      - Foreign K
      - Triggers
      - Views
      - Routines
      - prog3

SQL File 1\* x

describe contacts

Filter: Export: Autosize:

Field	Type	Null	Key	Default	Extra
ID	int(10) unsigned zerofill	NO	PRI	NULL	auto_inc
LastName	varchar(30)	YES		NULL	
FirstName	varchar(30)	YES		NULL	
Email	varchar(30)	YES		NULL	
Phone	varchar(14)	YES		NULL	
Magazine	varchar(60)	YES		NULL	
OS	varchar(30)	YES		NULL	

Result 1

Output

Action Output

	Time	Action	Message
✓	1 15:40:44	describe contacts	7 row(s)

Information

Object Info

Query Completed

SQL Additions

My Snippets

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.





```

1  # SQL commands to create and populate the MySQL database for
2  # CNT 4714 - Fall 2012
3  #
4  # delete the database if it already exists
5  drop database if exists mailinglist;
6
7  #create a new database named mailinglist
8  create database mailinglist;
9
10 #switch to the new database
11 use mailinglist;
12
13 #create the schemas for the four relations in this database
14 create table contacts (
15     ID integer unsigned zerofill auto_increment not null,
16     LastName varchar(30),
17     FirstName varchar(30),
18     Email varchar(30),
19     Phone varchar(14),
20     Magazine varchar(60),
21     OS varchar(30),
22     primary key (ID)
23 );
24

```

The script to generate/populate the MailingList database table contacts required for the PHP database example to work. Script is available on the code.

