

CNT 4714: Enterprise Computing Fall 2013

PHP – Database Connectivity

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
<http://www.cs.ucf.edu/courses/cnt4714/fall2013>

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



Form Processing and Business Logic

- HTML 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 HTML 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

```
1 <!DOCTYPE html>
2 <!-- form.html -->
3 <!-- Form for use with the form.php program -->
4 <html lang="en">
5 <head>
6 <title>Sample form to take user input in XHTML</title>
7 </head>
8 <body style = "font-family: arial, sans-serif;
9 background-color: #856363" background=image1.jpg>
10 <h1>This is a sample registration form.</h1>
11 Please fill in all fields and click Register.
12 <!-- post form data to form.php -->
13 <form method = "post" action = "form.php">
14 <img src = "images/user.gif" alt = "User" /><br />
15 <span style = "color: blue">
16 Please fill out the fields below.<br />
17 </span>
18 <!-- create four text boxes for user input -->
19 <img src = "images/fname.gif" alt = "First Name" />
20 <input type = "text" name = "fname" /><br />
21 <img src = "images/lname.gif" alt = "Last Name" />
22 <input type = "text" name = "lname" /><br />
23 <img src = "images/email.gif" alt = "Email" />
24 <input type = "text" name = "email" /><br />
25 <img src = "images/phone.gif" alt = "Phone" />
```

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





form.html

```

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

```





form.html

```

42     </select>
43     <br /><br />
44     <img src = "images/os.gif" alt = "Operating System" />
45     <br /><span style = "color: blue">
46         Which operating system are you currently using?
47     <br /></span>
48     <!-- create five radio buttons -->
49     <input type = "radio" name = "os" value = "Windows Vista" checked = "checked" />
50         Windows Vista
51     <input type = "radio" name = "os" value = "Windows 7" />
52         Windows 7
53     <input type = "radio" name = "os" value = "Windows 8" />
54         Windows 8
55     <input type = "radio" name = "os" value = "Linux" />
56         Linux
57     <input type = "radio" name = "os" value = "Other" />
58         Other<br />
59     <!-- create a submit button -->
60     <br /><input type = "submit" value = "Register" />
61 </form>
62 </body>
63 </html>

```



form.php

```

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

```

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.



form.php

```
27  ?>
28  <p>Hi
29      <span style = "color: blue">
30          <strong>
31              <?php print( "$fname" ); ?>
32          </strong>
33      </span>.
34      Thank you for completing the survey.<br />
35      You have been added to the
36      <span style = "color: blue">
37          <strong>
38              <?php print( "$magazine " ); ?>
39          </strong>
40      </span>
41      mailing list.
42  </p>
43  <strong>The following information has been saved
44      in our database:</strong><br />
45
46  <table border = "0" cellpadding = "0" cellspacing = "10">
47      <tr>
48          <td bgcolor = "#ffffaa">Name </td>
49          <td bgcolor = "#ffffbb">Email</td>
50          <td bgcolor = "#ffffcc">Phone</td>
51          <td bgcolor = "#ffffdd">OS</td>
52      </tr>
53
54      <tr>
```



form.php

```

45
46 <table border = "0" cellpadding = "0" cellspacing = "10">
47 <tr>
48 <td bgcolor = "#ffffaa">Name </td>
49 <td bgcolor = "#ffffbb">Email</td>
50 <td bgcolor = "#ffffcc">Phone</td>
51 <td bgcolor = "#ffffdd">OS</td>
52 </tr>
53
54 <tr>
55 <?php
56
57 // print each form field's value
58 print( "<td>$fname $lname</td>
59 <td>$email</td>
60 <td>$phone</td>
61 <td>$os</td>" );
62 ?>
63 </tr>
64 </table>
65
66 <br /><br /><br />
67 <div style = "font-size: 10pt; text-align: center">
68 This is only a sample form.
69 You have not been added to a mailing list.
70 </div>
71 </body>
72 </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

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 8 Linux Other

Register



This is a sample registration form.

Please fill in all fields and click Register.

User Information

Please fill out the fields below.

First Name	<input type="text" value="Mark"/>
Last Name	<input type="text" value="Llewellyn"/>
Email	<input type="text" value="markl@cs.ucf.edu"/>
Phone	<input type="text" value="(407)823-2790"/>

Must be in the form (555)555-5555

Publications

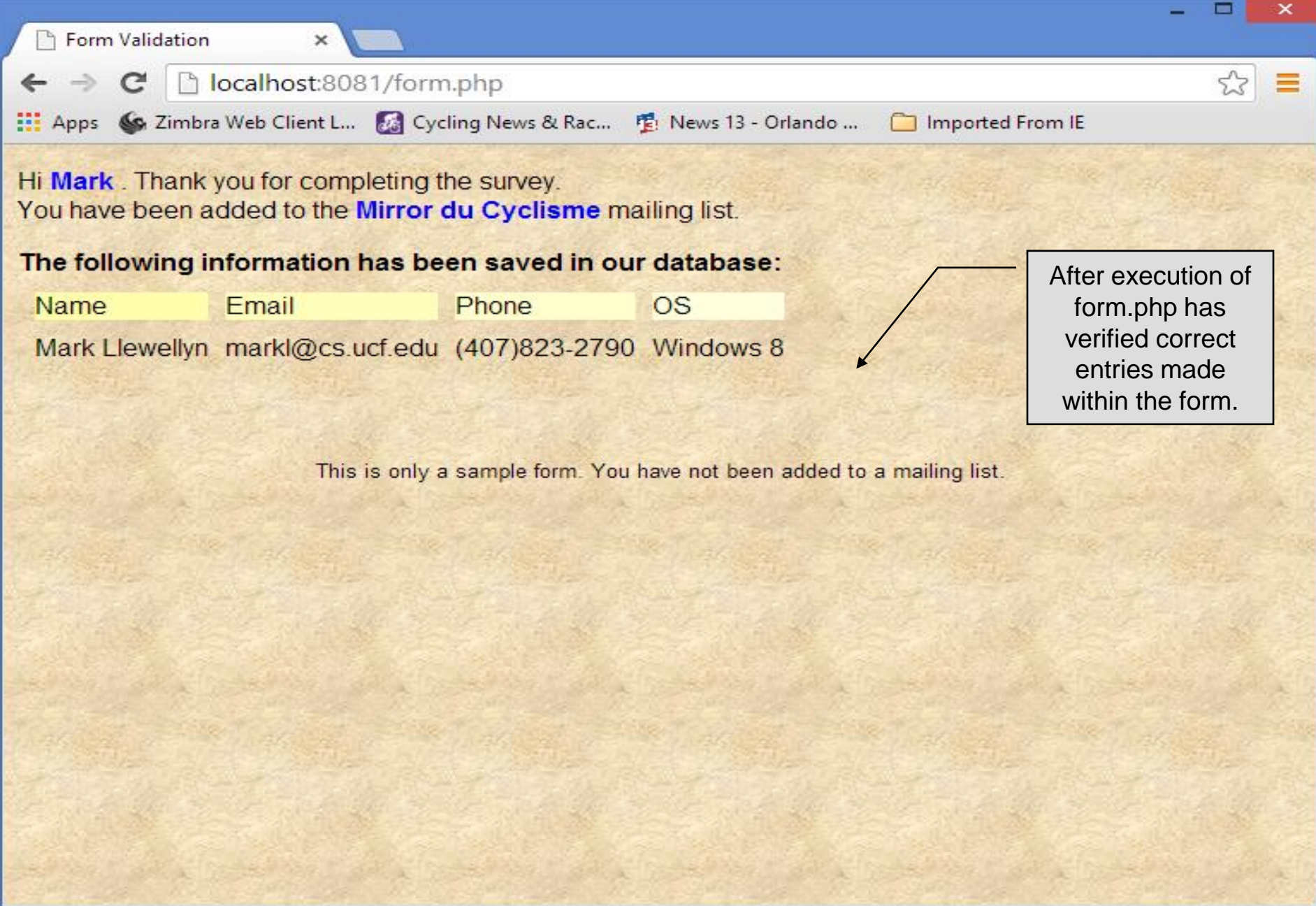
Which publication would you like information about?

Operating System

Which operating system are you currently using?

Windows Vista Windows 7 Windows 8 Linux Other





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?

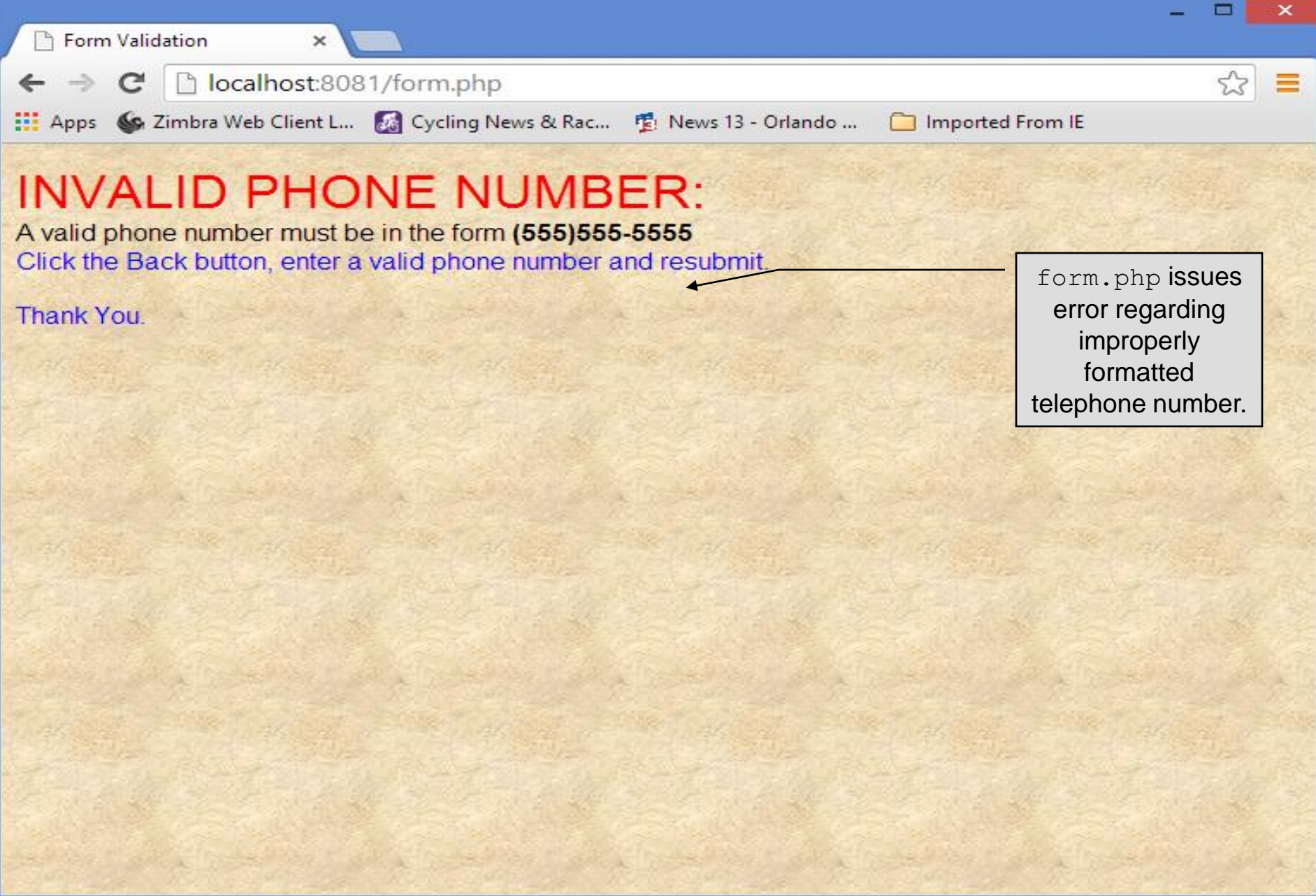
Operating System

Which operating system are you currently using?

Windows Vista Windows 7 Windows 8 Linux Other

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 HTML form and the PHP script “communicate” via the name-value pairs, it is a good idea to make the HTML object names meaningful so that the PHP script that retrieves the data is easier to understand.



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">
2 <!-- expression.php -->
3 <!-- Using regular expressions -->
4 <html lang="en">
5 <head>
6 <title>Regular expressions</title>
7 </head>
8 <body style = "font-family: arial, sans-serif;
9 background-color: #856363" background=image1.jpg>
10
11 <?php
12     $search = "Now is the time";
13     print( "Test string is: '$search'<br /><br />" );
14
15     // call function ereg to search for pattern 'Now'
16     // in variable search
17     if ( ereg( "Now", $search ) )
18         print( "String 'Now' was found.<br />" );
19
20     // search for pattern 'Now' in the beginning of
21     // the string
22     if ( ereg( "^Now", $search ) )
23         print( "String 'Now' found at beginning
24             of the line.<br />" );
25
26
```

^ matches at beginning
of a string



File Edit Search View Encoding Language Settings Macro Run Plugins Window ?



php.ini x homework.html x zipcode.py x project6.html x form.html x form.php x expression.php x

```

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

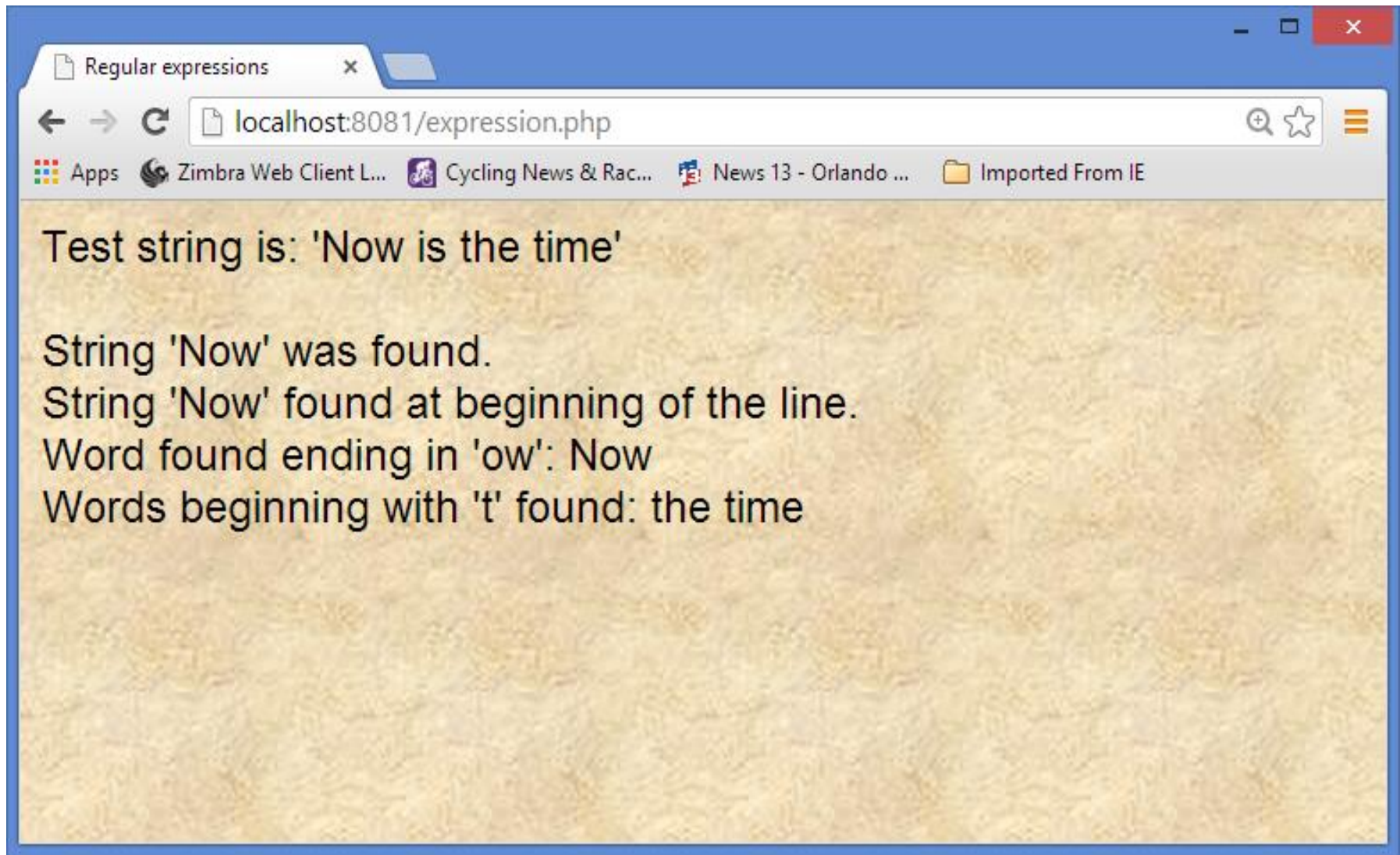
```

\$ 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 HTML form that queries a user for a username and password. The fields `USERNAME` and `PASSWORD` are posted to the PHP script `password.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>
2 <!-- password.html -->
3 <!-- HTML form sent to password.php for verification -->
4 <html lang="en">
5 <head>
6 <title>Verifying a username and a password.</title>
7 <style type = "text/css">
8 <td { background-color: #DDDDDD }
9 </style>
10 </head>
11 <body style = "font-family: arial", "font-size: 2em",
12 background-color: #856363" background=image1.jpg>
13 <p style = "font-size: 18pt">
14 <font color=red><B>
15 Welcome to the CNT 4714 High Security WebPage
16 </B></font>
17 <HR>
18 <p style = "font-size: 13pt">
19 Type in your username and password below.
20 <br />
21 <span style = "color: #0000FF; font-size: 10pt;
22 font-weight: bold">
23 Note that password will be sent as plain text - encryption not used in this ap
24 </span>
25 </p>
```



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



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




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



```
26 // write username and password to file and call function userAdded
27     fputs( $file, "$USERNAME,$PASSWORD\n" );
28     userAdded( $USERNAME );
29 }
30 else {
31
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 $field = split( ",", $line, 2 );
54 // verify username
55 if ( $USERNAME == $field[ 0 ] ) {
56     $userVerified = 1;
57
58     // call function checkPassword to verify user's password
59     if ( checkPassword( $PASSWORD, $field ) == true )
60         accessGranted( $USERNAME );
61     else
62         wrongPassword();
63 }
64 }
65 // close text file
66 fclose( $file );
67
68 // call function accessDenied if username has not been verified
69 if ( !$userVerified )
70     accessDenied();
71 }
72
73 // verify user password and return a boolean
74 function checkPassword( $userpassword, $filedata )
75 {
76     if ( $userpassword == $filedata[ 1 ] )
77         return true;
```



File Edit Search View Encoding Language Settings Macro Run Plugins Window ?



php.ini x homework.html x zipcode.py x project6.html x form.html x form.php x expression.php x password.html x password.php x

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



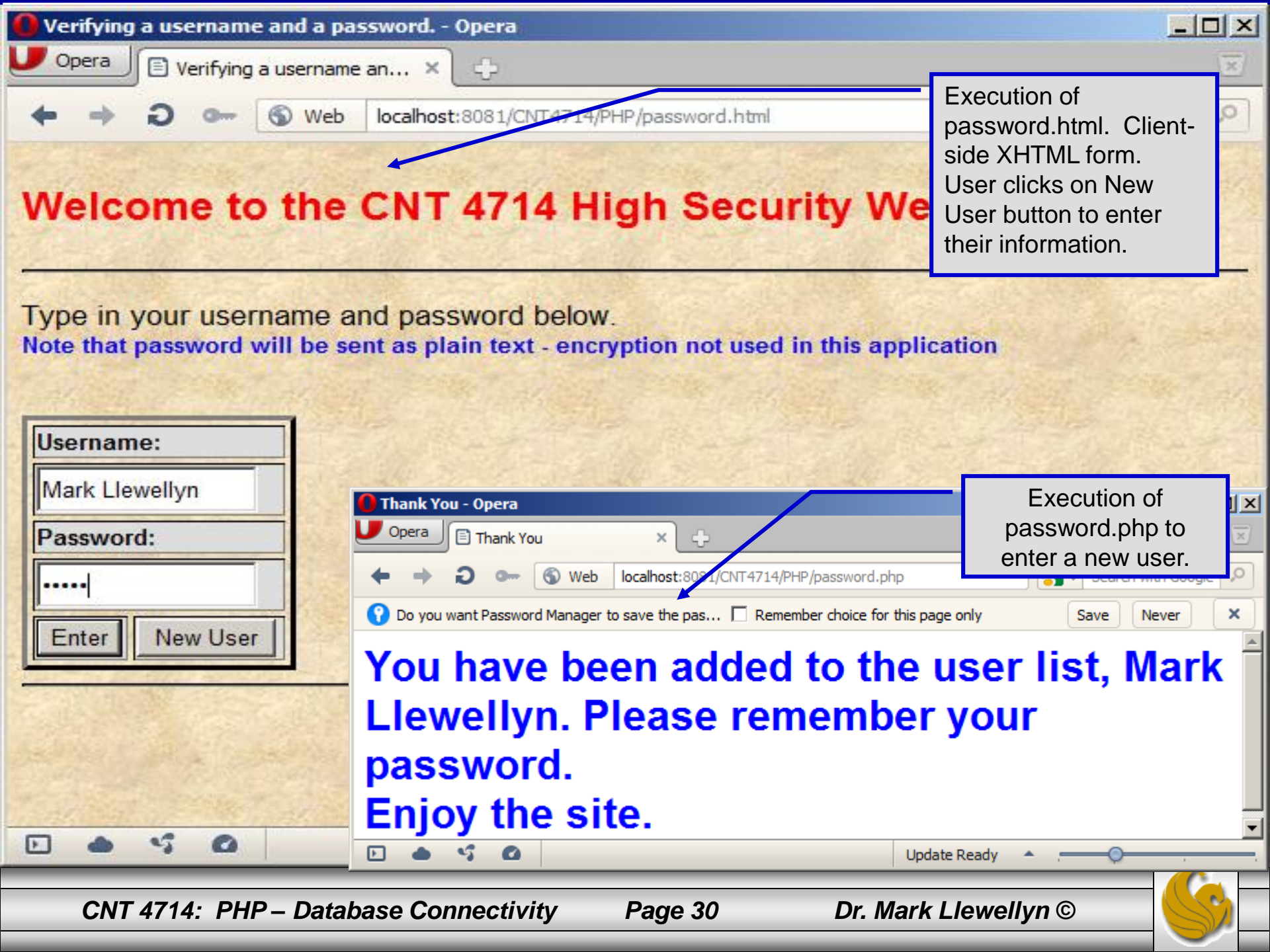
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?



php.ini x homework.html x zipcode.py x project6.html x fom.html x fom.php x expression.php x password.html x password.php x

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





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:
.....

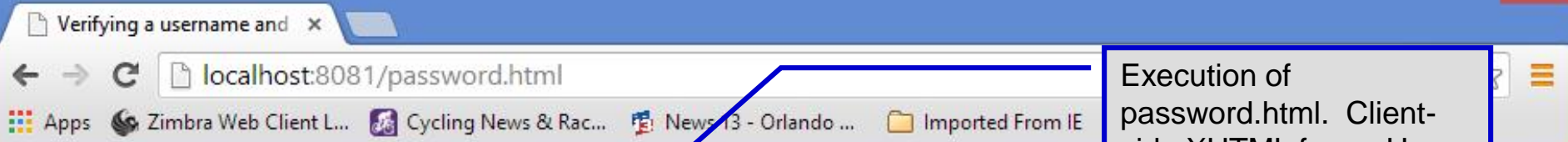
Enter New User

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

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

Execution of password.php to enter a new user.





Welcome to the CNT 4714 High Security

Type in your username and password below.

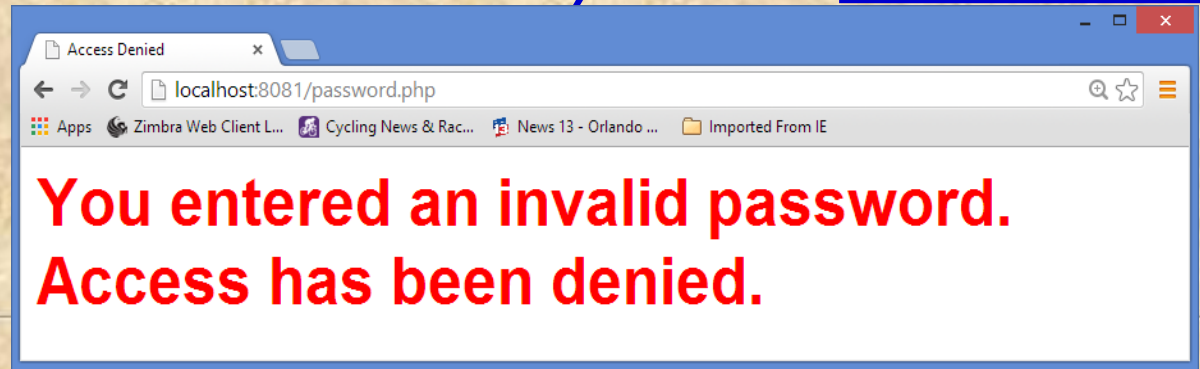
Note that password will be sent as plain text - encryption not used in this application

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

Username:

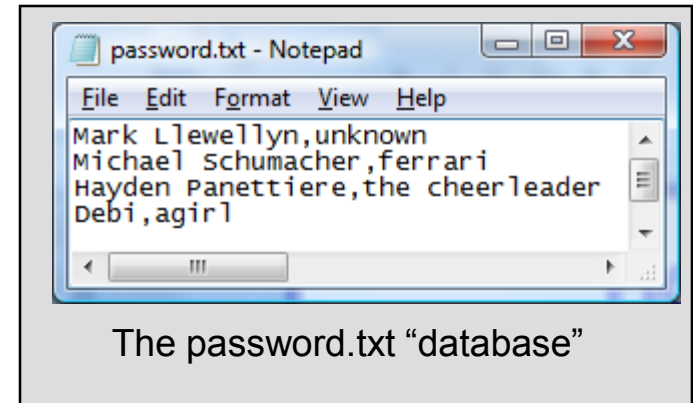
Password:

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 `$NewUser` has been set.
- When the user submits the `password.html` form to the server, they click either **Enter** or **New User** button. After calling function `extract`, either variable `$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 5.5.5 with only basic extensions loaded, you'll need to edit your `php.ini` file to enable the MySQL dynamic extensions. The following pages illustrate the changes you need to make to your `php.ini` file.
 - Versions of MySQL greater than 4.1.0 use MySQLi extensions.
 - Versions of MySQL less than 4.1.0 use MySQL extensions.



```
606 ; PHP's default setting is "&".
607 ; NOTE: Every character in this directive is considered as separator!
608 ; http://php.net/arg-separator.input
609 ; Example:
610 ;arg_separator.input = ";&"
611
612 ; This directive determines which super global arrays are registered when PHP
613 ; starts up. G,P,C,E & S are abbreviations for the following respective super
614 ; globals: GET, POST, COOKIE, ENV and SERVER. There is a performance penalty
615 ; paid for the registration of these arrays and because ENV is not as commonly
616 ; used as the others, ENV is not recommended on productions servers. You
617 ; can still get access to the environment variables through getenv() should you
618 ; need to.
619 ; Default Value: "EGPCS"
620 ; Development Value: "GPCS"
621 ; Production Value: "GPCS";
622 ; http://php.net/variables-order
623 variables_order = "GPCS"
624
625 ; This directive determines which super global data (G,P,C,E & S) should
626 ; be registered into the super global array REQUEST. If so, it also determines
627 ; the order in which that data is registered. The values for this directive are
628 ; specified in the same manner as the variables_order directive, EXCEPT one.
629 ; Leaving this value empty will cause PHP to use the value set in the
630 ; variables_order directive. It does not mean it will leave the super globals
```

If it is not already here, add this line in your php.ini file. This line determines which super_global arrays are registered when PHP starts,



```
720 ; http://php.net/doc-root
721 doc_root =c:\apache24\htdocs
722
723 ; The directory under which PHP opens the script using /~username used only
724 ; if nonempty.
725 ; http://php.net/user-dir
726 user_dir =
727
728 ; Directory in which the loadable extensions (modules) reside.
729 ; http://php.net/extension-dir
730 ; extension_dir = "."
731 ; On windows:
732 ; extension_dir = "ext"
733 extension_dir = "c:/php/ext"
734
735 ; Directory where the temporary files should be placed.
736 ; Defaults to the system default (see sys\_get\_temp\_dir)
737 ; sys\_temp\_dir = "/tmp"
738
739 ; Whether or not to enable the dl\(\) function. The dl\(\) function does NOT work
740 ; properly in multithreaded servers, such as IIS or Zeus, and is automatically
741 ; disabled on them.
742 ; http://php.net/enable-dl
743 enable_dl = Off
744
```

Set the path to the loadable extensions directory.





```
873 ; Be sure to appropriately set the extension_dir directive.
874 ;
875 ;extension=php_bz2.dll
876 ;extension=php_curl.dll
877 ;extension=php_fileinfo.dll
878 ;extension=php_gd2.dll
879 ;extension=php_gettext.dll
880 ;extension=php_gmp.dll
881 ;extension=php_intl.dll
882 ;extension=php_imap.dll
883 ;extension=php_interbase.dll
884 ;extension=php_ldap.dll
885 ;extension=php_mbstring.dll
886 ;extension=php_exif.dll ; Must be after mbstring as it
887 extension=php_mysql.dll
888 extension=php_mysqli.dll
889 ;extension=php_oci8.dll ; Use with Oracle 10gR2 Instant Client
890 ;extension=php_oci8_11g.dll ; Use with Oracle 11gR2 Instant Client
891 ;extension=php_openssl.dll
892 ;extension=php_pdo_firebird.dll
893 ;extension=php_pdo_mysql.dll
894 ;extension=php_pdo_oci.dll
895 ;extension=php_pdo_odbc.dll
896 ;extension=php_pdo_pgsql.dll
897 ;extension=php_pdo_sqlite.dll
```

Turn on both the mysql and mysqli extensions by uncommenting these two lines.



```
1122
1123 ; Maximum number of persistent links.  -1 means no limit.
1124 ; http://php.net/mysql.max-persistent
1125 mysql.max_persistent = -1
1126
1127 ; Maximum number of links (persistent + non-persistent).  -1 means no limit.
1128 ; http://php.net/mysql.max-links
1129 mysql.max_links = -1
1130
1131 ; Default port number for mysql_connect().  If unset, mysql_connect() will use
1132 ; the $MYSQL_TCP_PORT or the mysql-tcp entry in /etc/services or the
1133 ; compile-time value defined MYSQL_PORT (in that order).  Win32 will only look
1134 ; at MYSQL_PORT.
1135 ; http://php.net/mysql.default-port
1136 mysql.default_port = 3310
1137
1138 ; Default socket name for local MySQL connects.  If empty, uses the built-in
1139 ; MySQL defaults.
1140 ; http://php.net/mysql.default-socket
1141 mysql.default_socket =
1142
1143 ; Default host for mysql_connect() (doesn't apply in safe mode).
1144 ; http://php.net/mysql.default-host
1145 mysql.default_host =
1146
```

Specify the default port on which the MySQL Server is listening for the mysql extensions.





```
1182 ; Maximum number of links. -1 means no limit.
1183 ; http://php.net/mysql.max-links
1184 mysqli.max_links = -1
1185
1186 ; If mysqlnd is used: Number of cache slots for the internal result set cache
1187 ; http://php.net/mysql.cache\_size
1188 mysqli.cache_size = 2000
1189
1190 ; Default port number for mysqli_connect(). If unset, mysqli_connect() will use
1191 ; the $MYSQL_TCP_PORT or the mysql-tcp entry in /etc/services or the
1192 ; compile-time value defined MYSQL_PORT (in that order). Win32 will only look
1193 ; at MYSQL_PORT.
1194 ; http://php.net/mysql.default-port
1195 mysqli.default_port = 3310
1196
1197 ; Default socket name for local MySQL connects. If empty, uses the built-in
1198 ; MySQL defaults.
1199 ; http://php.net/mysql.default-socket
1200 mysqli.default_socket =
1201
1202 ; Default host for mysql_connect() (doesn't apply in safe mode).
1203 ; http://php.net/mysql.default-host
1204 mysqli.default_host =
1205
1206 ; Default user for mysql_connect() (doesn't apply in safe mode).
```

Specify the default port on which the MySQL Server is listening for the 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	mysqlnd 5.0.11-dev - 20120503 - \$Id: 40933630edef551dfaca71298a83fad8d03d62d4 \$

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	<i>no value</i>	<i>no value</i>
mysql.default_password	<i>no value</i>	<i>no value</i>
mysql.default_port	3310	3310
mysql.default_socket	<i>no value</i>	<i>no value</i>
mysql.default_user	<i>no value</i>	<i>no value</i>
mysql.max_links	Unlimited	Unlimited



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.

mysqli

Mysqli Support	enabled
Client API library version	mysqlnd 5.0.11-dev - 20120503 - \$Id: 40933630edef551dfaca71298a83fad8d03d62d4 \$
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	<i>no value</i>	<i>no value</i>
mysqli.default_port	3310	3310
mysqli.default_pw	<i>no value</i>	<i>no value</i>
mysqli.default_socket	<i>no value</i>	<i>no value</i>
mysqli.default_user	<i>no value</i>	<i>no value</i>
mysqli.max_links	Unlimited	Unlimited

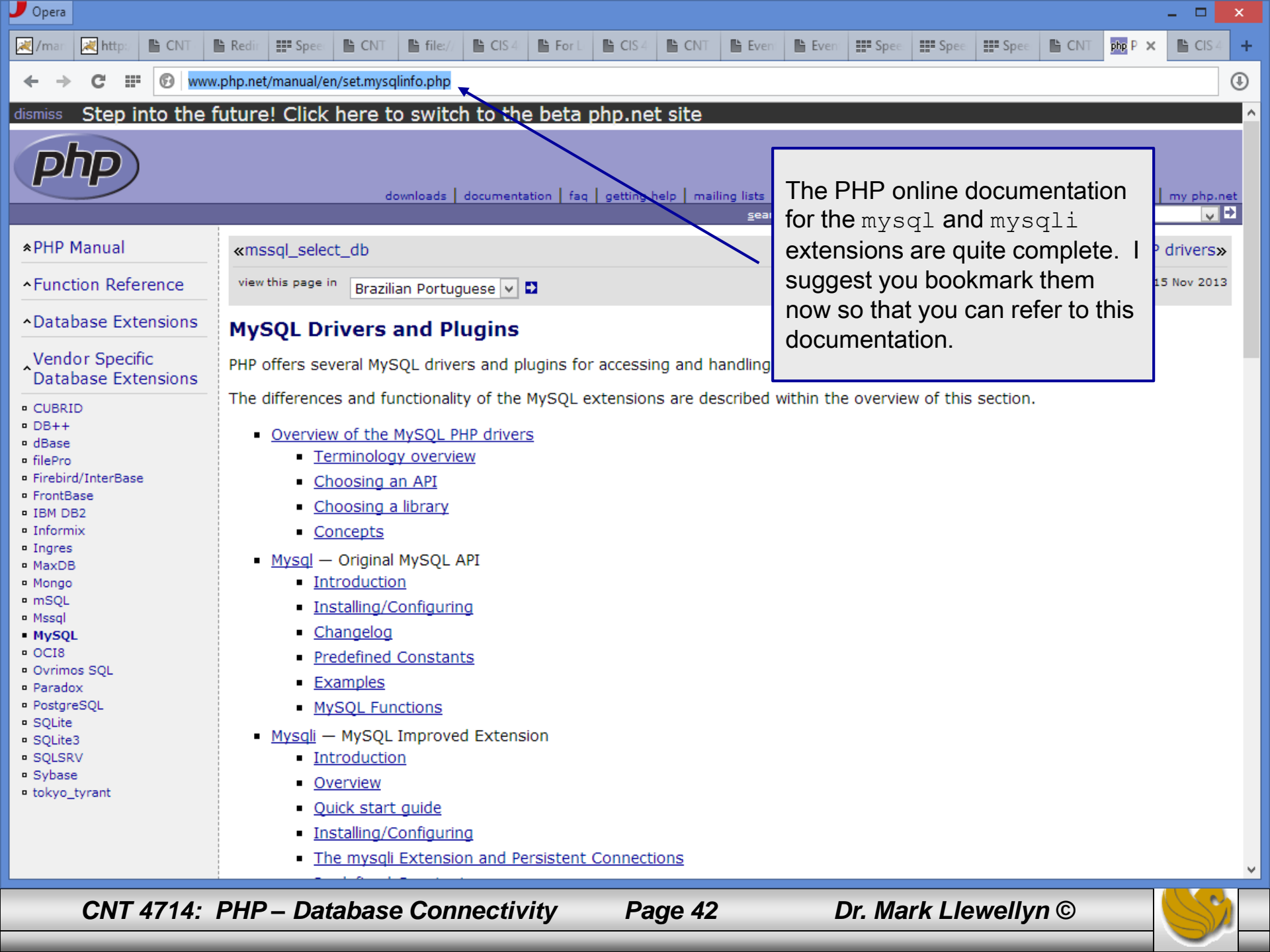


PHP and Database Connectivity (cont.)

- 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://www.php.net/manual/en/set.mysqlinfo.php>





www.php.net/manual/en/set.mysqlinfo.php

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.

MySQL Drivers and Plugins

PHP offers several MySQL drivers and plugins for accessing and handling MySQL databases. The differences and functionality of the MySQL extensions are described within the overview of this section.

- [Overview of the MySQL PHP drivers](#)
 - [Terminology overview](#)
 - [Choosing an API](#)
 - [Choosing a library](#)
 - [Concepts](#)
- [Mysql](#) — Original MySQL API
 - [Introduction](#)
 - [Installing/Configuring](#)
 - [Changelog](#)
 - [Predefined Constants](#)
 - [Examples](#)
 - [MySQL Functions](#)
- [Mysqli](#) — MySQL Improved Extension
 - [Introduction](#)
 - [Overview](#)
 - [Quick start guide](#)
 - [Installing/Configuring](#)
 - [The mysqli Extension and Persistent Connections](#)



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 HTML 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

```

1  <!DOCTYPE html>
2  <!-- data.html -->
3  <!-- Querying a MySQL Database From a PHP Script -->
4  <html lang="en">
5  <head>
6      <title>Sample Database Query From PHP</title>
7  </head>
8  <body style = "background-color: #545454" background=imagem1.jpg > <!--F0E68C -->
9  <h2 style = "font-family: arial color: blue">
10     Querying a MySQL database from a PHP Script.
11 </h2>
12 <form method = "post" action = "database.php">
13     <p>Select a field to display:
14         <!-- add a select box containing options -->
15         <!-- for SELECT query -->
16         <select name = "select">
17             <option selected = "selected">*</option>
18             <option>bikename</option>
19             <option>size</option>
20             <option>color</option>
21             <option>cost</option>
22             <option>purchased</option>
23             <option>mileage</option>
24         </select>
25     </p>
26     <input type = "submit" value = "Send Query" style = "background-color: blue;
27         color: yellow; font-weight: bold" />
28 </form>
29 </body>

```



database.php

Server side

Page 1

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.

```
1 <!DOCTYPE html>
2 <!-- database.php -->
3 <!-- Program to query a database and send results to the client.
4
5 <html lang="en">
6 <head>
7 <title>Database Search Results</title>
8 </head>
9 <body style = "font-family: arial, sans-serif"
10 style = "background-color: #4A766E" back
11 <?php
12 extract( $_POST );
13 // build SELECT query
14 $query = "SELECT " . $select . " FROM bikes";
15 // Connect to MySQL
16 if ( !( $database = mysqli_connect( "localhost:3310",
17 "root", "root", "bikedb" ) ) )
18 die( "Could not connect to database" );
19 // query bikedb database
20 if ( !( $result = mysqli_query( $database, $query ) ) ) {
21 print( "Could not execute query! <br />" );
22 die( mysqli_error() );
23 }
24 ?>
25 <h3 style = "color: blue">
26 Database Search Results</h3>
27 <table border = "1" cellpadding = "3" cellspacing =
28 style = "background-color: #00FFFF"> <!-- ADD8E
29
```



database.php

Server side

Page 2

Get metadata for the query

Display metadata in the top row of the table

```
29
30 <?php
31 // fetch meta-data
32 $metadata = mysqli_fetch_fields( $result);
33 print("<tr>");
34 for ($i=0; $i<count($metadata); $i++){
35     print("<td>");
36     printf("%s", $metadata[$i]->name);
37     print("</td>");
38 }
39 print("</tr>");
40
41 // fetch each record in result set
42 for ( $counter = 0;
43     $row = mysqli_fetch_row( $result );
44     $counter++ ){
45
46     // build table to display results
47     print( "<tr>" );
48
49     foreach ( $row as $key => $value )
50         print( "<td>$value</td>" );
51
52     print( "</tr>" );
53 }
54 mysqli_close( $database );
55 ?>
56 </table>
57 <br />Your search yielded <strong>
```



database.php

Server side

Page 3

```

40
41 // fetch each record in result set
42 for ( $counter = 0;
43     $row = mysqli_fetch_row( $result );
44     $counter++ ){
45
46     // build table to display results
47     print( "<tr>" );
48
49     foreach ( $row as $key => $value )
50         print( "<td>$value</td>" );
51
52     print( "</tr>" );
53 }
54 mysqli_close( $database );
55 ?>
56 </table>
57 <br />Your search yielded <strong>
58 <?php print( "$counter" ) ?> results.<br /><br /></strong>
59 <h5>Please email comments to
60     <a href = "mailto:markl@cs.ucf.edu">
61     markl@cs.ucf.edu
62     </a>
63 </h5>
64 </body>
65 </html>
66
67

```



Execution of data.html – Client side

Sample Database Query Fr x

localhost:8081/data.html

Apps Zimbra Web Client L... Cycling News & Rac... News 13 - Orlando ... Imported From IE

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

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

Results of query **SELECT * FROM bikes**. Display indicates that 14 rows were included in the result.

Your search yielded **14 results**.

Please email comments to markl@cs.ucf.edu



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.



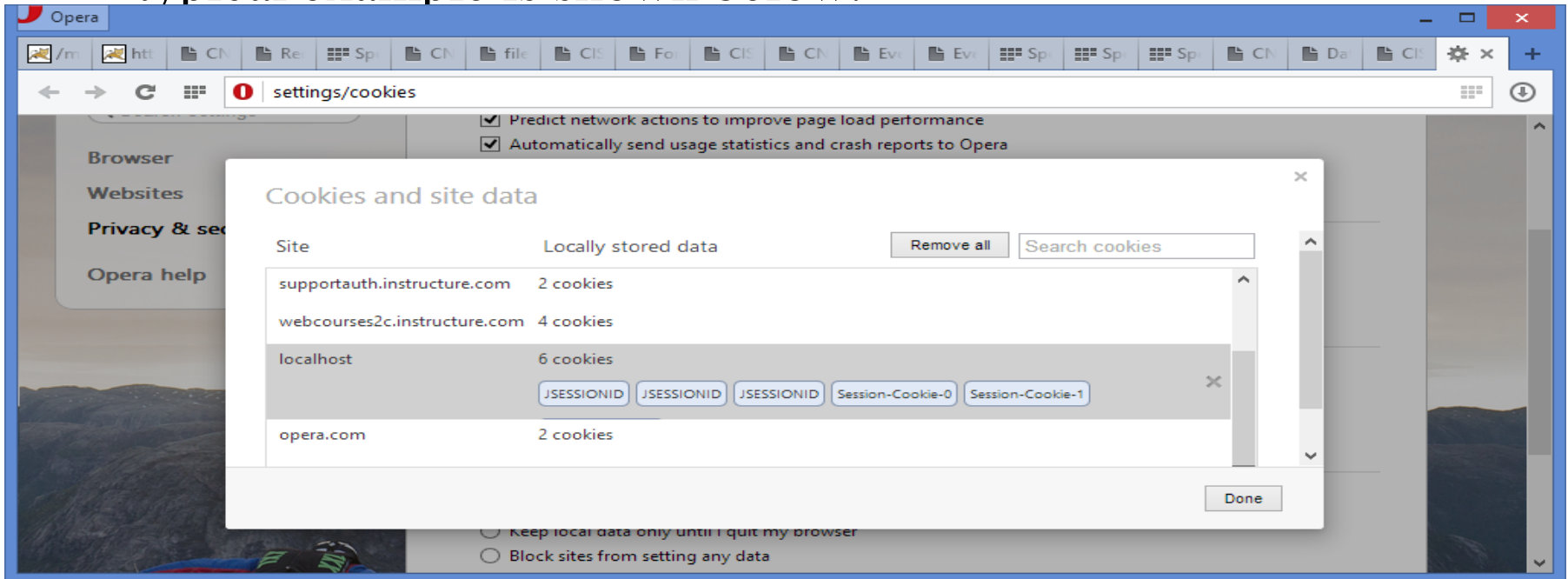
Cookies (cont.)

- 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 HTML output that reflects the preferences or information that is stored in the cookie.
- The location of the cookie file varies from browser to browser. In Opera you can get to your cookies via the settings/privacy & security settings.



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 HTML document and the PHP script are shown on the next pages. The HTML document `cookies.html` is on page 55 and the PHP script `cookies.php` appears on page 56.



```
1 <!DOCTYPE html>
2 <!-- cookies.html -->
3 <!-- Writing a Cookie -->
4
5 <html lang="en">
6 <head>
7     <title>Writing a cookie to the client computer</title>
8 </head>
9 <body style = "font-family: arial, sans-serif;
10     background-color: #856363" background=image1.jpg>
11 <h2>Click Write Cookie to save your cookie data.</h2>
12 <form method = "post" action = "cookies.php"
13     style = "font-size: 10pt" background-color: #856363">
14     <strong>Name:</strong><br />
15     <input type = "text" name = "NAME" /><br />
16     <strong>Height:</strong><br />
17     <input type = "text" name = "HEIGHT" /><br />
18     <strong>Favorite Color:</strong><br />
19     <input type = "text" name = "COLOR" /><br />
20 <p>
21     <input type = "submit" value = "Write Cookie"
22         style = "background-color: #0000FF; color: yellow; font-weight: bold" />
23 </p>
24 </form>
25 </body>
26 </html>
27
28
29
```



cookies.php – page 1

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

```

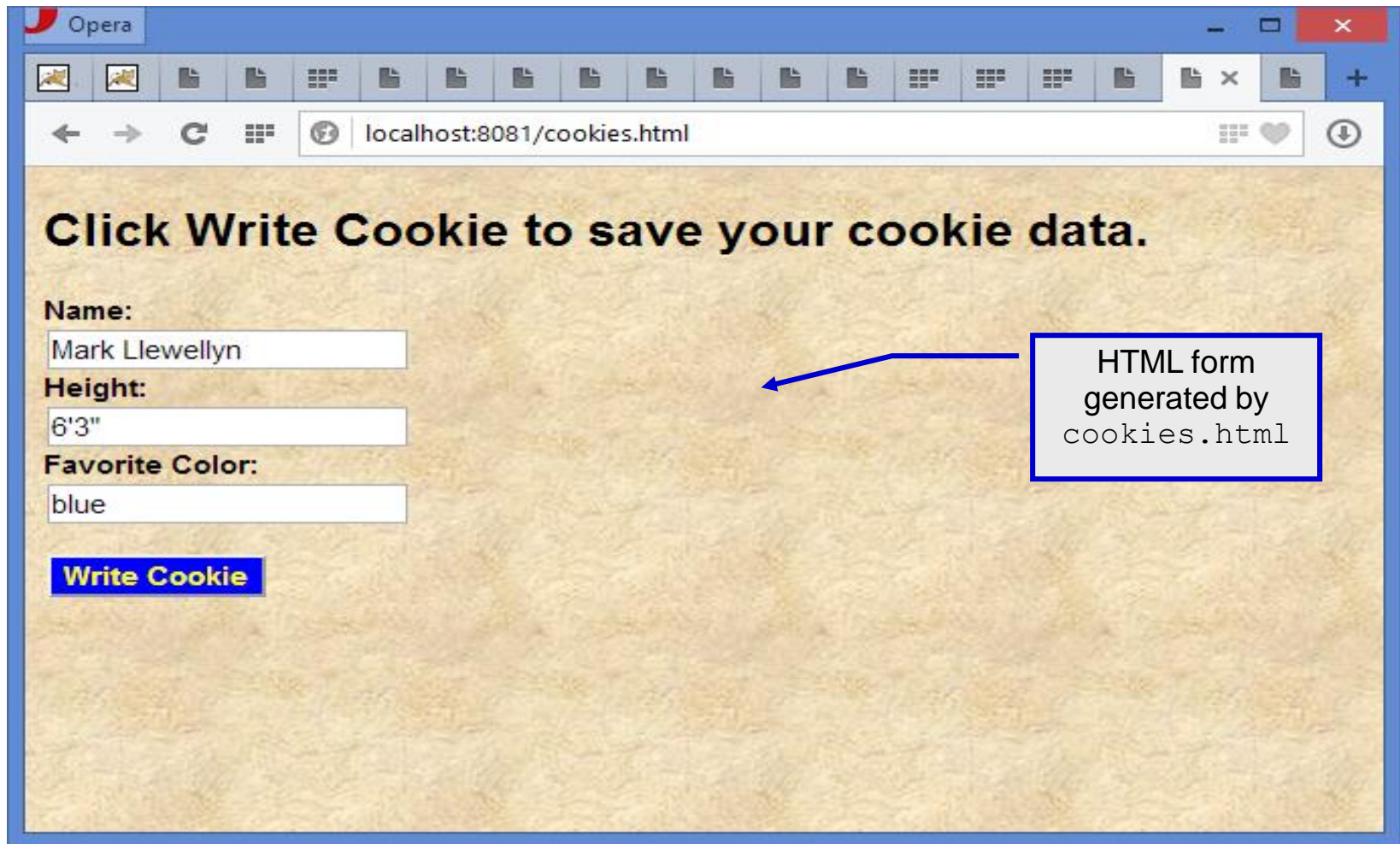
1  <?php
2  // cookies.php
3  // Program to write a cookie to a client's machine
4
5  extract( $_POST );
6  // write each form field's value to a cookie and set t
7  // cookie's expiration date
8  setcookie( "Name", $NAME, time() + 60 * 60 * 24 * 5 );
9  setcookie( "Height", $HEIGHT, time() + 60 * 60 * 24 * 5 );
10 setcookie( "Color", $COLOR, time() + 60 * 60 * 24 * 5 );
11 ?>
12
13 <!DOCTYPE html>
14 <html lang="en">
15 <head>
16     <title>Cookie Saved</title>
17 </head>
18 <body style = "font-family: arial, sans-serif", background=
19 <p><b>The cookie has been set with the following data:</b>
20 <!-- print each form field's value -->
21 <br /><span style = "color: blue">Name:</span>
22     <?php print( $NAME ) ?><br />
23 <span style = "color: blue">Height:</span>
24     <?php print( $HEIGHT ) ?><br />
25 <span style = "color: blue">Favorite Color:</span>
26 <span style = "color: <?php print( "$COLOR\">$COLOR" )
27 </span><br />
28 <p>Click <a href = "readCookies.php">here</a>
29     to read the saved cookie.</p>

```

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.)



The screenshot shows an Opera browser window with the address bar displaying `localhost:8081/cookies.html`. The page content includes the following elements:

- Header:** "Click Write Cookie to save your cookie data."
- Form Fields:**
 - Name:** Input field containing "Mark Llewellyn".
 - Height:** Input field containing "6'3\"".
 - Favorite Color:** Input field containing "blue".
- Button:** A blue button labeled "Write Cookie".

A blue arrow points from a text box to the form area. The text box contains the text: "HTML form generated by cookies.html".



Cookies (cont.)

Opera

localhost:8081/cookies.php

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.



```
1 <!DOCTYPE html>
2 <!-- readCookies.php -->
3 <!-- Program to read cookies from the client's computer -->
4 <html lang="en">
5   <head><title>Read Cookies</title></head>
6   <body style = "font-family: arial, sans-serif" background=image1.jpg>
7     <p>
8       <strong>
9         The following data is saved in a cookie on your computer.
10      </strong>
11     </p>
12     <table border = "5" cellspacing = "0" cellpadding = "10">
13       <?php
14         // iterate through array $_COOKIE and print
15         // name and value of each cookie
16         foreach ( $_COOKIE as $key => $value )
17           print( "<tr>
18             <td bgcolor=#F0F68C">$key</td>
19             <td bgcolor=#FFA500">$value</td>
20             </tr>" );
21       ?>
22     </table>
23   </body>
24 </html>
```

Superglobal array
holding cookie.



Cookies (cont.)

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.



Cookies (cont.)

The screenshot shows the Opera browser's settings page for cookies. A modal window titled "Cookies and site data" is open, displaying a list of cookies under the "Locally stored data" tab. A blue arrow points from a text box to the "Locally stored data" tab. The text box contains the text "Cookie data as stored/viewed in Opera 17.0". The cookie details shown are:

Name:	Name
Content:	Mark+Llewellyn
Domain:	localhost
Path:	/
Send for:	Any kind of connection
Accessible to script:	Yes
Created:	Tuesday, November 19, 2013 3:53:02 PM
Expires:	Sunday, November 24, 2013 3:53:02 PM

Buttons for "Remove all", "Search cookies", and "Remove" are visible. The "Done" button is at the bottom right of the modal.

Cookie data as stored/viewed in Opera 17.0



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 HTML 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 8

Linux Other

User fills in the form and clicks the Register button.



Hi **Mark**. Thank you for completing the survey.
You have been added to the **Mirror du Cyclisme** 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 8

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

[Click here to view entire database.](#)



Database Search Results x

localhost:8081/formDatabase2.php

Apps Zimbra Web Client L... Cycling News & Rac... News 13 - Orlando ... Imported From IE

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	Mirror du Cyclisme	Windows 8
0000000002	Panettiere	Hayden	savethecheerleader.com	(555)123-6789	Velo-News	Linux
0000000003	Schumacher	Michael	michael@mercedes.de	(123)222-4567	RadSport	Windows 7
0000000004	Swanepoel	Candice	vickisecret	(999)999-9999	Pro Cycling	Windows 8
0000000005	Einstein	Albert	its-relative.com	(111)111-1111	Cycling Weekly	Windows 8

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



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 (235)987-2344

Must be in the form (555)555-5555

Publications

Which magazine would you like information about?

Cycling Weekly

Operating System

Which operating system are you currently using?

Windows Vista Windows 7 Windows 8

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.



Local instance MySQL56 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 hw3dbscript-V2 SQL File 2*

SCHEMAS

Filter objects

bikedb
 colorsurvey
 guestbook
 Tables
 Views
 Stored Procedur
 Functions
 largeco
 mailinglist
 Tables
 contacts
 Views
 Stored Procedur
 Functions
 project2
 project3
 project4
 project5

Management Schemas

Information

Schema: mailinglist

Query 1

```
select * from contacts
```

Result Set Filter:

ID	LastName	FirstName	Email	Phone	Magazine	OS
0000000001	Llewellyn	Mark	markl@cs.ucf.edu	(407)823-2790	Mirror du Cyclisme	Windows 8
0000000002	Panettiere	Hayden	savethecheerleader.com	(555)123-6789	Velo-News	Linux
0000000003	Schumacher	Michael	michael@mercedes.de	(123)222-4567	RadSport	Windows 7
0000000004	Swanepoel	Candice	vickisecret	(999)999-9999	Pro Cycling	Windows 8
0000000005	Einstein	Albert	its-relative.com	(111)111-1111	Cycling Weekly	Windows 8
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

contacts 1 x Apply Cancel

Output

Object Info Session Action Output

SQLAdditions

SELECT

Topic: SELECT

Syntax:

```
SELECT
  [ALL | DISTINCT | DISTINCTROW
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG
  RESULT]
  [SQL_CACHE | SQL_NO_CACHE]
  select_expr [, select_expr ..
  [FROM table_references
  [PARTITION partition_list]
  [WHERE where_condition]
  [GROUP BY {col_name | expr |
  [ASC | DESC], ... [WITH ROL
  [HAVING where_condition]
  [ORDER BY {col_name | expr |
  [ASC | DESC], ...]
  [LIMIT {[offset,] row_count |
  offset}]
  [PROCEDURE procedure_name(arg
  [INTO OUTFILE 'file_name'
  [CHARACTER SET charset_na
  export_options
  | INTO DUMPFIL 'file_name'
  | INTO var_name [, var_name
  [FOR UPDATE | LOCK IN SHARE N
```

SELECT is used to retrieve rows selected from tables, and can include UNION statements, JOINs, and subqueries.

The most commonly used clauses of the SELECT statement are:

- FROM: Each select_expr indicates a connection to a table or view.

Context Help Snippets



```
1 <!DOCTYPE html>
2
3 <!-- dynamicForm2.php -->
4
5 <html lang="en">
6 <head>
7 <title>Sample form to take user input in HTML</title>
8 </head>
9 <body style = "font-family: arial, sans-serif; background-color: #856363" background=ima
10 <?php
11     extract ( $_POST );
12     $iserror = false;
13     // array of magazine titles
14     $maglist = array( "Velo-News",
15         "Cycling Weekly",
16         "Pro Cycling",
17         "Cycle Sport",
18         "RadSport",
19         "Mirror du Cyclisme" );
20     // array of possible operating systems
21     $systemlist = array( "Windows Vista",
22         "Windows 7",
23         "Windows 8",
24         "Linux",
25         "Other");
26     // array of name and alt values for the text input fields
27     $inputlist = array( "fname" => "First Name",
28         "lname" => "Last Name",
```



```

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

```



```

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

```




```

89     <div style = 'font-size : 10pt; text-align: center'>
90     <div style = 'font-size : 18pt'>
91     <a href = 'formDatabase2.php'>
92     Click here to view entire database.</a>
93     </div>
94 </div></body></html>" );
95     die();
96     }
97     }
98     print( "<h1>This is a sample registration form.</h1>
99     Please fill in all fields and click Register." );
100     if ( $iserror ) {
101         print( "<br /><span style = 'color : red'>
102             Fields with * need to be filled in properly.</span>
103     }
104     print( "<!-- post form data to dynamicForm2.php -->
105         <form method = 'post' action = 'dynamicForm2.php'>
106         <img src = 'images/user.gif' alt = 'User' /><br />
107         <span style = 'color: blue'>
108         Please fill out the fields below.<br />
109         </span>
110 <!-- create four text boxes for user input -->" );
111     foreach ( $inputlist as $inputname => $inputalt ) {
112         $inputtext = $inputalt[ $inputname ];
113
114         print( "<img src = 'images/$inputname.gif'
115             alt = '$inputalt' /><input type = 'text' name = '$inputname' value = '' . $$i
116         if ( $formerrors[ ( $inputname )."error" ] == true )

```

Invoke PHP script to see contents of entire database if user clicks this link. Code begins on page 75 .

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



```

106 <img src = 'images/user.gif' alt = 'User' /><br />
107 <span style = 'color: blue'>
108 Please fill out the fields below.<br />
109 </span>
110 <!-- create four text boxes for user input -->" );
111 foreach ( $inputlist as $inputname => $inputalt ) {
112     $inputtext = $inputalt[ $inputname ];
113
114     print( "<img src = 'images/$inputname.gif'
115         alt = '$inputalt' /><input type = 'text' name = '$inputname' value = '" . $$inputname . "' />" );
116     if ( $formerrors[ ( $inputname )."error" ] == true )
117         print( "<span style = 'color : red'>*</span>" );
118     print( "<br />" );
119 }
120 print( "<span style = 'font-size : 10pt" );
121 if ( $formerrors[ "phoneerror" ] ) print( "; color : red" );
122 print( "'>Must be in the form (555)555-5555
123 </span><br /><br />
124 <img src = 'images/downloads.gif'
125 alt = 'Publications' /><br />
126 <span style = 'color: blue'>
127 Which magazine would you like information about
128 </span><br />
129 <!-- create drop-down list containing magazine
130 <select name = 'mag'>" );
131 foreach ( $maglist as $currmag ) {
132     print( "<option" );
133     if ( ( $currmag == $mag ) )

```

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>
2
3 <!-- formDatabase2.php      -->
4 <!-- Program to query a database and send results to the client. -->
5
6 <html lang="en">
7   <head>     <title>Database Search Results</title>   </head>
8   <body style = "font-family: arial, sans-serif"
9     style = "background-color: #F0E68C" background=image1.jpg>
10  <?php
11     extract( $_POST );
12     // build SELECT query
13     $query = "SELECT * FROM contacts";
14     // Connect to MySQL
15     if ( !( $database = mysqli_connect( "localhost:3310", "root", "root", MailingList ) ) )
16         die( "Could not connect to database" );
17     // query MailingList database
18     if ( !( $result = mysqli_query( $database, $query ) ) ) {
19         print( "Could not execute query! <br />" );
20         die( mysqli_error() );
21     }
22  ?>
23  <h3 style = "color: blue">
24  Mailing List Contacts</h3>
25  <table border = "1" cellpadding = "3" cellspacing = "2"
26    style = "background-color: #ADD8E6">
27    <tr>
28      <td>ID</td>
29      <td>Last Name</td>

```



```
23 <h3 style = "color: blue">
24 Mailing List Contacts</h3>
25 <table border = "1" cellpadding = "3" cellspacing = "2"
26 style = "background-color: #ADD8E6">
27 <tr>
28 <td>ID</td>
29 <td>Last Name</td>
30 <td>First Name</td>
31 <td>E-mail Address</td>
32 <td>Phone Number</td>
33 <td>Magazine</td>
34 <td>Operating System</td>
35 </tr>
36 <?php
37 // fetch each record in result set
38 for ( $counter = 0;
39 $row = mysqli_fetch_row( $result );
40 $counter++ ){
41 // build table to display results
42 print( "<tr>" );
43 foreach ( $row as $key => $value )
44 print( "<td>$value</td>" );
45 print( "</tr>" );
46 }
47 mysqli_close( $database );
48 ?>
49 </table>
50 </body>
51 </html>
```



Local instance MySQL56 x

File Edit View Query Database Server Tools Scripting Help



Navigator

SCHEMAS

Filter objects

- ▶ bikedb
- ▶ colorsurvey
- ▼ guestbook
 - ▶ Tables
 - ▶ Views
 - ▶ Stored Procedur
 - ▶ Functions
- ▶ largeco
- ▼ mailinglist
 - ▼ Tables
 - ▶ contacts
 - ▶ Views
 - ▶ Stored Procedur

Management Schemas

Information

Schema: mailinglist

Query 1 hw3dbscript-V2 SQL File 2* x



1 describe contacts

Result Set Filter: Export: Wrap Cell Content: IA

	Field	Type	Null	Key	Default	Extra
▶	ID	int(10) unsigned zerofill	NO	PRI	NULL	auto_increment
	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 2 x

Output

Action Output

SQLAdditions

Jump to

No Context Help

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



```

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
cookies.html cookies.php readCookies.php dynamicForm2.php fomDatabase2.php php.ini dynamicForm2.php mailing
1 # SQL commands to create and populate the MySQL database for
2 # CNT 4714 - Fall 2013
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 course code page.

