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?

Velo-News

Operating System
Which operating system are you currently using?

Windows XP
Windows 2000
Windows 98
Linux
Other

Register

Basically, the same registration form that was used in a previous example.
Screen the user sees after clicking the Register button.

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 XP

Click here to view entire database.
Screen the user sees after clicking to see the entire database.

<table>
<thead>
<tr>
<th>ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>E-mail Address</th>
<th>Phone Number</th>
<th>Magazine</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Schumacher</td>
<td>Ralf</td>
<td><a href="mailto:ralf@toyota.com">ralf@toyota.com</a></td>
<td>(222)222-222</td>
<td>Cycling Weekly</td>
<td>Other</td>
</tr>
<tr>
<td>2</td>
<td>Thurau</td>
<td>Didi</td>
<td>agirl.de</td>
<td>(456)123-3345</td>
<td>Mirror du Cylisme</td>
<td>Windows 98</td>
</tr>
<tr>
<td>3</td>
<td>Jolie</td>
<td>Angelina</td>
<td><a href="mailto:aj@hollywood.com">aj@hollywood.com</a></td>
<td>(123)456-7890</td>
<td>Velo-news</td>
<td>Windows XP</td>
</tr>
<tr>
<td>4</td>
<td>Schumacher</td>
<td>Michael</td>
<td><a href="mailto:michael@ferrari.it">michael@ferrari.it</a></td>
<td>(407)823-2790</td>
<td>RadSport</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>5</td>
<td>Campbell</td>
<td>Kristi</td>
<td>im_not_sure</td>
<td>(345)987-6545</td>
<td>Cycle Sport</td>
<td>Windows 2000</td>
</tr>
<tr>
<td>6</td>
<td>Einstein</td>
<td>Albert</td>
<td>relative</td>
<td>(111)111-1111</td>
<td>Cycling Weekly</td>
<td>Other</td>
</tr>
<tr>
<td>7</td>
<td>Llewellyn</td>
<td>Mark</td>
<td><a href="mailto:markl@cs.ucf.edu">markl@cs.ucf.edu</a></td>
<td>(407)823-2790</td>
<td>Pro Cycling</td>
<td>Windows XP</td>
</tr>
</tbody>
</table>
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.
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.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!-- dynamicForm2.php -->

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

// array of name and alt values for the text input fields
$inputlist = array( "fname" => "First Name",
                    "lname" => "Last Name",
                    "email" => "Email",
                    "phone" => "Phone" );
if ( isset ( $submit ) ) {
    if ( $fname == "" ) {
        $formerrors[ "fnameerror" ] = true;
        $iserror = true;
    }
    if ( $lname == "" ) {
        $formerrors[ "lnameerror" ] = true;
        $iserror = true;
    }
    if ( $email == "" ) {
        $formerrors[ "emailerror" ] = true;
        $iserror = true;
    }
    if ( !ereg( "^[\(][0-9]{3}[\)]-[0-9]{3}-[0-9]{4}$", $phone ) ) {
        $formerrors[ "phoneerror" ] = true;
        $iserror = true;
    }
    if ( !$iserror ) {
        // build INSERT query
        $query = "INSERT INTO contacts " .
                 "(ID, LastName, FirstName, Email, Phone, Magazine, OS ) " .
                 "VALUES (null, '$lname', '$fname', '$email', " . "" . quotemeta( $phone ) . "", 'mag', 'os' )";
    }
}
// Connect to MySQL
if ( !( $database = mysql_connect( "localhost", "root", "root" ) ) )
die( "Could not connect to database" );

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

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

    <table border = '0' cellpadding = '0' cellspacing = '10'>
    <tr><td bgcolor = '#ffffaa'>Name </td>
    <td bgcolor = '#ffffbb'>Email</td>
    <td bgcolor = '#ffffcc'>Phone</td><td bgcolor = '#ffffdd'>OS</td></tr><tr>
    <td bgcolor = '#ffffaa'>&#ffffffaa'>Name </td>
    <td bgcolor = '#ffffffbb'>Email</td>
    <td bgcolor = '#ffffffcc'>Phone</td><td bgcolor = '#ffffffdd'>OS</td>
    </tr>
    </table>
<!-- print each form field's value -->
<td>$fname $lname</td>
<td>$email</td>
<td>$phone</td>
<td>$os</td></tr></table>
<br /><br /><br /><div style = 'font-size : 10pt; text-align: center'>

<div style = 'font-size : 18pt'><a href = 'formDatabase2.php'>
Click here to view entire database.</a></div></div>

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

<\/div></body></html>
<!-- create four text boxes for user input -->"

foreach ($inputlist as $inputname => $inputalt) {
    $inputtext = $inputvalues[$inputname];

    print("<img src = 'images/$inputname.gif'
        alt = '$inputalt' />
        <input type = 'text' name = '$inputname' value = "$inputtext" />
    ");
    if ($formerrors[$inputname]."error" == true) {
        print("<span style = 'color : red'>*
            </span>");
    }
    print("<br />");
}

print("<span style = 'font-size : 10pt'")
if ($formerrors["phoneerror"]) print("; color : red ");
print("">Must be in the form (555)555-5555
</span><br /><br />
<br />
alt = 'Publications' />
<br />
<br />
Which magazine would you like information about?
<br />
<br />
<!-- create drop-down list containing magazine names -->
<select name = 'mag'>
    foreach ($maglist as $currmag) {
        print("<option">
            if ($currmag == $mag) {
                print(" selected = 'true'" );
            }
            print(">$currmag</option>");
    }

The $$variable notation specifies variable variables. PHP permits the use of variable variables to allow developers to reference variables dynamically. The expression $$variable could also be written as ${$variable} for added clarity.
print( "</select><br />\n<br />
<img src = 'images/os.gif' alt = 'Operating System' /><br />
<br />
Which operating system are you currently using?<br />
<br /></span>

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

$counter = 0;

foreach ( $systemlist as $currsystem ) {
    print( "<input type = 'radio' name = 'os'
        value = '$currsystem'" );
    if ( $currsystem == $os ) print( "checked = 'checked'" );
    if ( iserror && $counter == 0 ) print( "checked = 'checked'" );

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

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

?>
<?php
extract( $_POST );
// build SELECT query
$query = "SELECT * FROM contacts";
// Connect to MySQL
if ( !( $database = mysqli_connect( "localhost",  "root", "root“, MailingList ) ) )
die( "Could not connect to database" );
// query MailingList database
if ( !( $result = mysqli_query( $database, $query ) ) ) {  
  print( "Could not execute query! <br />" );
  die( mysqli_error() );
}

<h3 style = "color: blue">
Mailing List Contacts</h3>
<?php
// fetch each record in result set
for ( $counter = 0; $row = mysqli_fetch_row( $result ); $counter++ ) {
    // build table to display results
    print( "<tr>" );
    foreach ( $row as $key => $value ) {
        print( "<td>$value</td>" );
    }
    print( "</tr>" );
}
mysqli_close( $database );
?>

</table>
</body>
</html>
Schema of the MailingList database table contacts required for the PHP database example to work. Script is available on the code page.
Connecting Apache To Tomcat

• Although it is possible for Tomcat to run standalone and server HTTP requests directly (we did this for servlets and jsps), the Apache server does a much better job of handling tasks such as static content and SSL connections.

• For this reason, Tomcat is typically used alongside an Apache server. Unlike PHP which runs as a module inside the Apache process, a JVM is external and requires a mechanism to connect it to the web server.

• Tomcat inherited the Apache JServ protocol (AJP) from the JServ project. AJP is a protocol for connecting an external process to a servlet container. It is the responsibility of an Apache module, in this case `mod_jk`, to speak this protocol to the servlet container (Tomcat).
Connecting Apache To Tomcat (cont.)

- In this last section of notes, I’ll show you how to integrate Apache and Tomcat into a single package.

- The ultimate set-up will resemble the figure shown below. Note that if you also would like Tomcat to run standalone HTTP requests, it will need to run on a different HTTP port than Apache. That’s why I set-up Apache on port 8081 and Tomcat on port 8080.
Getting The Tomcat Connector

• The first step in Apache-Tomcat integration is to obtain the Tomcat connector from Apache.

• Follow the screen shots on the next few pages to obtain the mod_jk connector.

• Windows based connector binary files will typically have the name of mod_jk.so.
From the main Tomcat webpage, select Tomcat connectors from the download section. The current documentation is also available from this page.
Apache Tomcat

Tomcat Connectors (mod_jk, mod_jk2) Downloads

We recommend you use a mirror to download our release builds, but you must verify the integrity of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from the mirrors.

You are currently using http://mirror.x10.com/mirror/apache. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are backup mirrors (at the end of the mirrors list) that should be available.

Other mirrors:  http://mirror.x10.com/mirror/apache

The KEYS link links to the code signing keys used to sign the product. The PGP link downloads the OpenPGP compatible signature from our main site.

For more information concerning Tomcat Connectors (mod_jk, mod_jk2), see the Tomcat Connectors (mod_jk, mod_jk2) site.
Tomcat Connectors (mod_jk, mod_jk2) Downloads

We recommend you use a mirror to download our release builds, but you must verify the integrity of the downloaded files using signatures downloaded from our main distribution directories. Recent releases (48 hours) may not yet be available from the mirrors.

You are currently using http://www.devlib.org/apache. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are backup mirrors (at the end of the mirrors list) that should be available.

Other mirrors: http://www.devlib.org/apache

The KEYS link links to the code signing keys used to sign the product. The PGF link downloads the OpenPGP compatible signature from our main site.

For more information concerning Tomcat Connectors (mod_jk, mod_jk2), see the Tomcat Connectors (mod_jk, mod_jk2) site.

KEYS

- JK 1.2
  - Source
    - JK 1.2.15 Source Release tar.gz
      - [pgp]
    - JK 1.2.15 Source Release zip
      - [pgp]
  - JK 1.2 Binary Releases
  - JK 2 (deprecated)
    - Source
Index of /apache/tomcat/tomcat-connectors/jk/binaries

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>24-Dec-2004 07:50</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>aix/</td>
<td>29-Sep-2003 22:05</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>freebsd/</td>
<td>20-Sep-2003 21:04</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>iseries/</td>
<td>27-Jul-2005 03:02</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>linux/</td>
<td>27-Jul-2005 03:03</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>macosx/</td>
<td>26-Jul-2005 12:55</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>netware/</td>
<td>15-Nov-2005 13:18</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>solaris/</td>
<td>06-Dec-2005 05:57</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>win32/</td>
<td>08-Nov-2005 03:54</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>win64/</td>
<td>08-Nov-2005 08:54</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The Apache Jakarta Tomcat Connector

- Current stable version is jk-1.2.15
Index of /apache/tomcat/tomcat-connectors/jk/binaries/win32

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>08-Nov-2005 08:54</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>jk-1.2.14/</td>
<td>07-Jul-2005 07:58</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>jk-1.2.15/</td>
<td>08-Nov-2005 04:01</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>jk-1.2.6/</td>
<td>20-Dec-2004 03:28</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The Apache Jakarta Tomcat Connector

- Current stable version is jk-1.2.15
Apache Tomcat JK 1.2.15 for WIN32

Here you’ll find the binaries for IIS, Apache 1.3 and 2.0 Web Server.

Built for Apache 1.3.34 and above, and 2.0.55 and above.

- mod_jk-apache-1.3.34.so is for Apache 1.3.x. Rename to mod_jk.so before putting it in your Apache/modules directory.
- mod_jk-apache-1.3.34 sapi.so is for Apache 1.3.x with SSL support. Rename to mod_jk.so before putting it in your Apache/modules directory.
- mod_jk-apache-2.0.55.so is for Apache 2.0, and works with Apache 2.0.55 and later. Rename to mod_jk.so before putting it in your Apache2/modules directory.
- isapi_redirect.dll is for IIS 5 and later Web Server.
- isapi_redirect.msi is installer package for IIS 5 and later Web Server.
- nsapi_redirect.dll is for Sun ONE Web Server 6.1 and later (formerly Netscape iPlanet).
- jk_symbols.zip contains the symbols files (.pdb) for all packages.

The Apache Tomcat Project
http://tomcat.apache.org/
Installing The Tomcat Connector

- Once you’ve downloaded the Tomcat connector, put it in the modules sub-directory of your Apache installation’s root directory.

Locate the mod_jk file in the modules subdirectory of Apache.
Enabling The Tomcat Connector

• Once you’ve put the connector file in the `modules` sub-directory of your Apache installation’s root directory, you are now ready to configure Apache to recognize and enable the module.

• To do this you’ll need to edit your Apache configuration file. This is the same file you edited to enable PHP.

• This file is located in the `conf` subdirectory and is named `httpd`. 
This was added earlier to enable PHP

Add all of this material to enable mod_jk

Add JKMount directives for any servlets and jsp's to go through Tomcat.
• The JWorkerFile directive (see previous page) refers to a separate file that configures the ajp13 protocol communications parameters. An example called workers.properties is included with mod_jk. For a basic set-up, this default file will work fine.

• There are a couple of things you should verify however: (1) workers.tomcat_home should agree with the value you’ve already set for Tomcat called CATALINA_HOME, and (2) worker.ajp13.port must be the same as the one listed in the Tomcat server.xml file as shown on the next page.

• In more robust applications, additional editing of the JWorkerFile will be required. For example, if you have more than one installation of Tomcat on your machine, you’ll need to adjust the worker.ajp.port parameter in workers.properties to make sure that mod_jk is connecting to the correct Tomcat installation as Tomcat installations will not be able to start up sharing port numbers.
The portion of the Tomcat worker.properties file showing the definition for the default AJP 1.3 worker. NOTE: the port number listed here must match the port number listed in the Tomcat server.xml file as shown on the next page.
The portion of the Tomcat server.xml file showing the definition for the AJP 1.3 connector. This connector port number listed here must match the port number in the workers.properties file as shown on the previous page.