LOYOLA UNIVERSITY CHICAGO

CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2015

Week 3

PHP and MySQL

Week 2 Exercise

- any questions?
- any issues with querying the MySQL data?
- what patterns and relationships did you test?

PHP and MySQL

Week 2 Exercise

Sample query 1

```
SELECT content.contentid, content_type.content_type_name, users.username FROM content_type_lookup, content_type,
content, users WHERE content_type_lookup.content_id=content.contentid AND users.userid=content_type_lookup.user_id
AND content_type_lookup.content_type_id=content_type.content_type_id AND content_type_lookup.user_id=1;
```

Sample query 2

```
SELECT content.contentname, content_type.content_type_name, users.username FROM content_type_lookup, content_type,
content, users WHERE content_type_lookup.content_id=content.contentid AND users.userid=content_type_lookup.user_id
AND content_type_lookup.content_type_id=content_type.content_type_id AND content_type_lookup.content_id=15;
```

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries from content_lookup etc

- modify basic connection to query multiple tables

    - SELECT content.contentid, content.contentname, users.username FROM content_lookup, content, users WHERE content_lookup.content_id=content.contentid AND users.userid=content_lookup.user_id AND content_lookup.user_id=1;

- output results in an un-ordered list

    PHP Example

PHP and MySQL

Test basic PHP MySQL connection & SELECT queries from content_lookup etc

```
<?php
//currently set to basic user query privileges
$con = mysqli_connect('localhost', '402user', 'celine59', '402framework');
if (!$con) {
    die('Could not connect: ' . mysql_error());
}

$result = mysqli_query($con,"SELECT content.contentid, content.contentname, users.username FROM content_lookup,
content, users WHERE content_lookup.content_id=content.contentid AND users.userid=content_lookup.user_id AND
content_lookup.user_id=1");

echo '<ul>';

while($row = mysqli_fetch_array($result))
  {
  echo '<li>contentid = '.$row['contentid'].' & contentname = '.$row['contentname'].' & username = '.$row['username'].'</li>';
  }

echo '</ul>';

mysqli_close($con);
?>
```

PHP and MySQL

Basic HTML output & queries - Part 1

Features include:

- output all content currently available in the framework
    - all content
    - by content type (image & text)

[PHP Example](#) | [PHP Code Example](#)

TO DO - ADD EXAMPLES FOR THE FOLLOWING USING FUNCTIONS CREATED...

- output all users currently registered in the framework

- output content per current user

PHP and MySQL

Basic HTML output & queries - Part 2

Features include:
- output all users currently registered in the framework
- output content per current user

PHP Example | PHP Code Example

TO DO - ADD EXAMPLES & REVISE FOR THE FOLLOWING USING LESS REDUNDANCY

- output all users currently registered in the framework
- output content per current user

But first...MVC

MVC

Model-View-Controller - Part 1

- avoid chaotic spaghetti programming

- MVC defines a clean separation between critical components of an app

- user interface (UI)
    - present UI elements for the user such as buttons, forms, text fields…

- handling and reacting to the UI

- application needs to store the data eg: a database

MVC

Model-View-Controller - Part 2

- defines three parts of an application called model, view, and controller

- Model
    - provides the underlying data and methods that offer information to the rest of the app
    - does not define how the app will look or how it will act

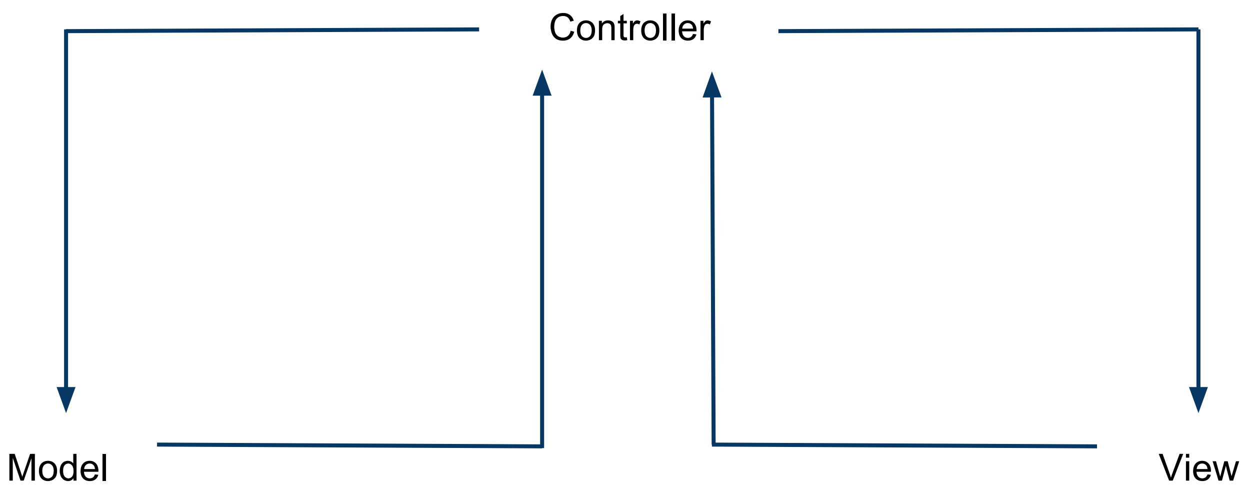
- View
    - includes different on-screen UI elements such as buttons, fields, switches...
    - multiple views make up the UI for an app

- Controller
    - manages the interaction and flow between the model and the view
    - handles actions such as user input (keyboard, mouse etc) and sends them to the model or view as required

MVC

Model-View-Controller - Part 3

402 Framework



Possible workflow patterns within the framework?

PHP and MySQL

Basic HTML output & queries - Part 3

- output all users currently registered in the framework

- then, output content per current user

PHP Example - Less Redundancy | PHP Code Example

TO DO - ABSTRACT FUNCTIONS & ADD INCLUDES TO COMBINE THE FOLLOWING

- output all content
- output all users

PHP and MySQL

Code Abstraction - PHP Include()

- allows a script to use functions, code, output etc from another PHP file

```
include(‘includes/mysql_tools.inc.php’);
```

- procedural logic still applies to PHP file

- include() places the code from another PHP file into that position within the current file

- include() allows us to include functions, scripts etc

    - anything that we need to abstract...

PHP and MySQL

ABSTRACTED FUNCTIONS & ADD INCLUDES TO COMBINE THE FOLLOWING

- output all content

- output all users

PHP Example - Abstracted | PHP Code Example (‘basicInclude’ directory in GitHub repository)

TO DO - Basic Error Checking

- check for empty results

- handle errors gracefully

PHP and MySQL

Error Handling - Basic (PHP Manual - Error Handling)

- What is error handling in programming?

    - reporting to help developers…
    - error feedback for users...

- How can we handle errors in PHP?

    - die() function or exit() function - PHP Manual
    - stops script running at point of error
    - custom error handler (see next slide)

- Graceful errors?

    - eg: check user input and logically respond to error
    - trigger_error("Age must be greater than 21!") - PHP Manual
    - exceptions (throw, try, catch) - PHP Manual

- Recording errors

    - saved to error_log on server
    - custom error_log() to send email with error number, error string

PHP and MySQL

Error Handling - Custom Error Handler

```
<?php
//error handler function
function customError($errno, $errstr)
  {
  echo "<b>Error:</b> [$errno] - $errstr";
  }

//set error handler
set_error_handler("customError");

//trigger error
echo($test);
?>

//sample output for above example
Error: [8] - Undefined variable: test
```

Example from: W3 Schools Overview

PHP and MySQL

Basic Error Checking - ~ 10 minutes

Check over http://students.ctsdh.luc.edu/teaching/demos/mysql/basicInclude/

- work your way through the site and identify potential points of error in the logic and flow

- go through the code and identify the above points where the code needs to be amended

Code = https://github.com/dighteach/source/tree/master/2015/DIGH402/week3/basicInclude

Updated with basic error checking:

http://students.ctsdh.luc.edu/teaching/demos/mysql/basicInclude2/

PHP and MySQL

TO DO - Basic Error Checking

- check for empty results
- handle errors and return feedback for the user

Handle user error reporting

- basic empty link errors
- empty result set or single empty result
- empty or invalid data returned per DB table row

…

PHP and MySQL

Basic Error Checking - BasicInclude2

- results.php - handle empty 'req' from URL - Example

- mysql_connect.inc.php - handle empty 'results' dataset returned from DB - Example

- content_viewer.php - Example

- user_viewer.php - Example

Why do we not handle errors in the following *include* files? (For this initial stage of testing...)

- content_processor.inc.php
- results_format.inc.php

PHP Example - Errors | PHP Code Example ('basicInclude2' directory in GitHub repository)

PHP and MySQL

Further abstraction in the current code - BasicInclude3

- root.inc.php
    - per required directory to allow specification of root directories
- default_includes.inc.php
    - allows us to store all cross-framework links to include files in one single file
    - eg: MySQL connection & query,
- config.inc.php
    - define ‘assets’ directory for css & javascript files
    - define ‘media’ directory for images, video, audio…
    - define MySQL DB settings, tables…

PHP and MySQL

Further abstraction in the current code

- root.inc.php

    - allows abstracted specification of root directories
    - can be project root directory or per required directory or often both!
    - mainly used to prevent unwanted repetition of directory location in ‘include’ statements
    - eg: location of default includes directory, modules…

Code Example

```
<?php
$root_base = ‘modules/base/’;
$root_content = ‘modules/content/’;
$root_user = ‘modules/users/’;
$root_images = ‘media/images/’;
$root_includes = ‘includes/’;
?>
```

PHP and MySQL

Further abstraction in the current code

- default_includes.inc.php

    - allows us to store all cross-framework links to include files in one single file
    - eg: MySQL connection & query

Code Example

```
<?php
/*DB config etc*/
include($root_includes.'config.inc.php');
include($root_includes.'mysql_connect.inc.php');
?>
```

PHP and MySQL

Further abstraction in the current code

- config.inc.php
    - allows us to store all cross-framework links to include files in one single file
    - eg: MySQL connection & query, DB tables, template settings…

Code Example

```
<?php
//database server
define('DB_SERVER', 'localhost');
//database query user login name
define('DB_USER', '402user');
?>
```

- code and files now need to be updated to reflect this latest abstraction of code and settings

PHP and MySQL

Again, more abstraction - Updated Framework layout and Model

- default framework design components
    - header, sidebar, main content, footer...
- header abstracted to modules/template in framework

PHP Example | PHP Code Example ('basicInclude3' directory in GitHub repository)

...and more to abstract and update.