



## CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

### 402 - Introduction to Digital Humanities Design and Programming

Spring Semester 2016

Week 4

## PHP and MySQL

Test basic PHP MySQL connection & SELECT queries - against USERS table

- we can modify basic connection to query 'users' table
  - SELECT username, usercreated FROM users;
  - SELECT \* FROM users WHERE userid=3;
  - SELECT DISTINCT username FROM users;
- output results in a tabular format

[PHP Example](#)

## PHP and MySQL

### Test basic PHP MySQL connection & SELECT queries - tabular output

```
<?php
$con = MySQLi_connect('localhost', 'username', 'password', '402framework');
if (!$con) {
    die('Could not connect: ' . MySQL_error());
}

$result = MySQLi_query($con,"SELECT * FROM users");

echo '<table>';
echo '<th>username</th>';
echo '<th>created</th>';

while($row = MySQLi_fetch_array($result))
{
    echo '<tr>';
    echo '<td>'.$row['username'].'</td>';
    echo '<td>'.$row['usercreated'].'</td>';
    echo '</tr>';
}

echo '</table>';

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

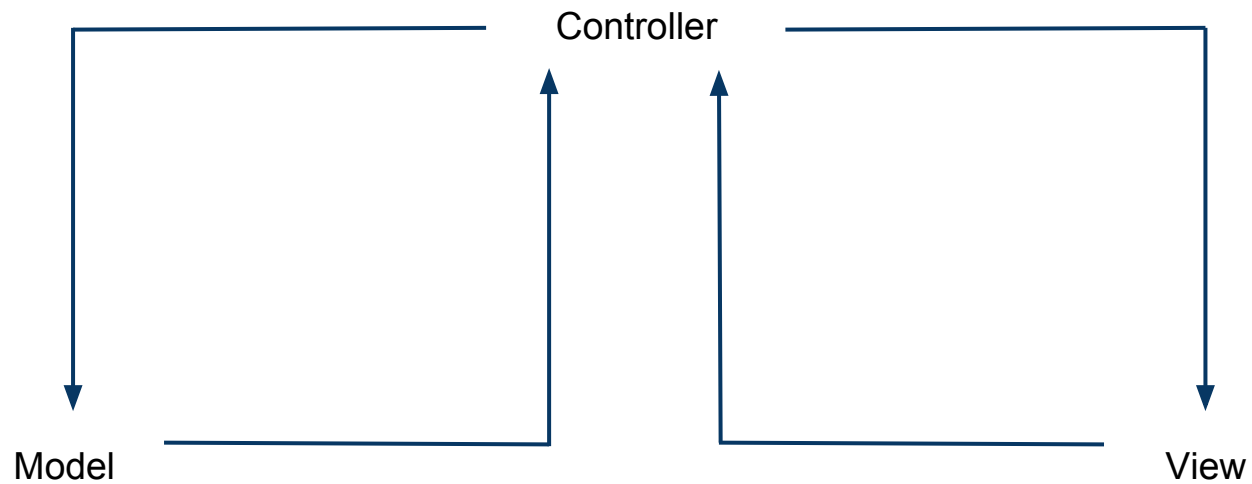
## Model-View-Controller - Part 2

- 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