

# Project Tutorial 4

## PHP and MySQL

Dr. Yuzuko Nakamura  
Dept. of Computer Science  
Y.Nakamura@ucl.ac.uk

# PHP and databases

## Basic strategy

- Connect to a database.
- Construct a SQL query as a string.
- Run the query.
  - If the query returns results (e.g., SELECT), then get the data.
- Close the connection.

# Connecting to a database

- Use the `mysqli_connect` method:  

```
$connection =  
mysqli_connect('localhost', 'exampleuser', 'mypass  
word', 'Example')
```

  - Arguments: server, username, password, database name.
- `mysqli` = MySQL improved (newer functions).

# Username and password

- Root can access all databases.
- Good practice tips:
  - Set up new usernames/passwords for each database. (Don't use root!)
  - Set a root password as well.
  - Store files such as SQL scripts on server but not in the www/htdocs directory (requestable by users).
  - Don't want hackers to get easy access to your development code.

## Connections can fail...

- Can use:

```
$connection =  
mysqli_connect('localhost','exampleuser','mypassword','Example')  
or die('Error connecting to MySQL server.' . mysql_error());
```

- die echo's the message and exits the current script.

- Can also use:

```
if (!$connection) { ... } // connection failed
```

- Allows more useful action to be taken.

## Making a SQL query

```
$query = "SELECT first_name, family_name FROM Users";  
$result = mysqli_query($connection,$query)  
    or die('Error making select users query' .  
    mysql_error());
```

- SQL query is a string (semi-colon is optional for the query string).
- Use `mysqli_query` with connection and query.
  - Returns a `mysqli_result` object.

## Getting the results

```
$row = mysqli_fetch_array($result)
```

- Fetch the next row from the object returned.
- Returns an array containing the column/value pairs for the row. Access using:

```
$row['first_name']
```

```
$row['family_name']
```



## Need to iterate to get complete result set

- Create a dynamic table from database results using PHP:

```
echo '<table border="1">';  
while ($row = mysqli_fetch_array($result))  
{  
    echo '<tr><td>' . $row['first_name'] .  
        '</td><td>' . $row['family_name'] .  
        '</td></tr>';  
}  
echo '</table>';
```

- `$row` evaluates to false when no more rows exist

# Close the connection

```
mysqli_close($connection);
```

- Important to release a connection when no longer needed (finite connection pool).

## Constructing an INSERT query from input

```
$connection = mysqli_connect('localhost','exampleuser',  
'mypassword','Example')  
    or die('Error connecting to MySQL server.' . mysql_error());  
  
$query = "INSERT INTO Users (first_name, family_name)".  
    "VALUES ('${user['firstName']}', '${user['familyName']}')";  
  
$result = mysqli_query($connection,$query)  
    or die('Error making saveToDatabase query');  
mysqli_close($connection);
```

## Tip: Beware of browser caching

- Sometimes browsers cache data to reduce the need to request data from the server.
- Frustrating experience of making changes to .php files and not seeing any changed behavior.
- Try force reload (ctrl+F5 or cmd+R) or clearing cache.

## Tip: Reuse the database connection code

- The work of opening a database connection, sending a query, and closing a database connection is the same across the site.
- Reuse these functions by placing them in a PHP file and including that file.
  - E.g. `<?php include 'database.php'; ?>`

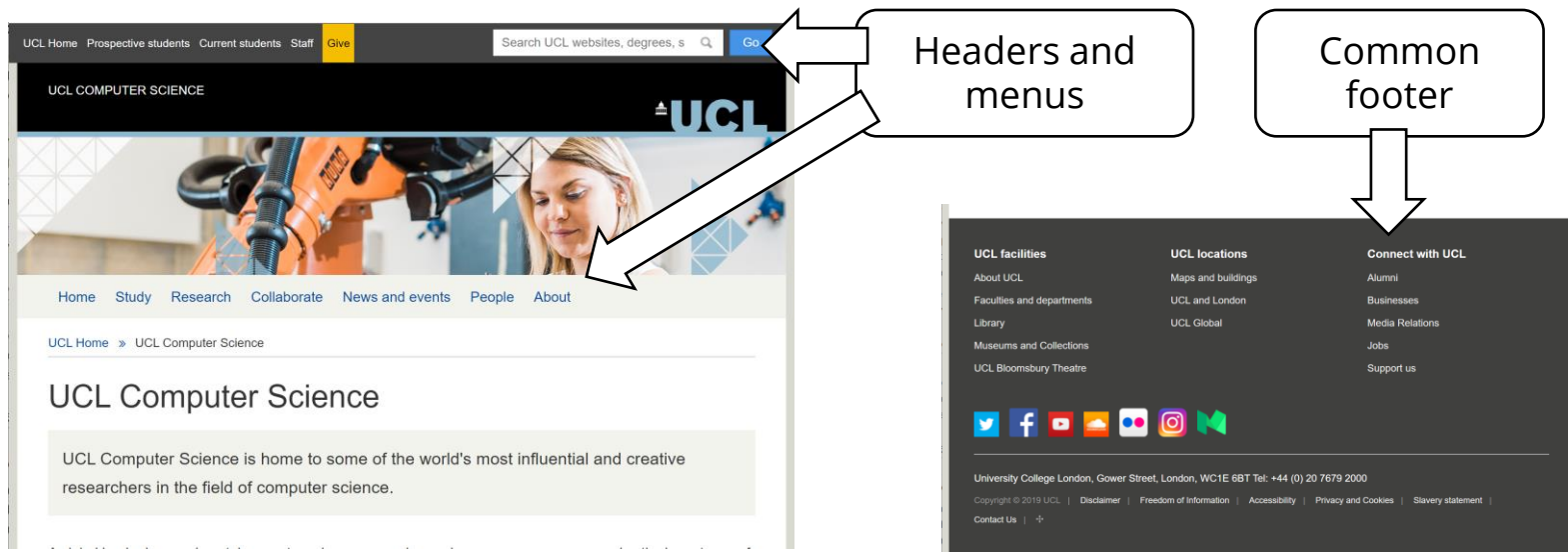
# Include vs. require

- PHP provides include and require to insert the content of one php/html file into another.
- `<?php include('header.php'); ?>`
  - If an error occurs a warning is generated but PHP script carries on.
  - Use this if you have robust error recovery.
    - Don't want to risk displaying invalid or private information.
- `<?php require('header.php'); ?>`
  - If an error occurs PHP script will exit.
  - Use this by default.

## `_once`

- PHP pages are often built with multiple `requires` or `includes`.
- There can be a problem with a file being included/required more than once.
  - This is a problem for PHP code in particular.
- `include_once`, `require_once` will not include/require a file that has already been loaded.
- Use `require_once` for your code.

# PHP includes are helpful for website layout too





# Header example using variables

- In pageheader.php:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="style.css"
type="text/css" media="screen">
    <?php echo"<title>My Application -
$pageTitle</title>" ?>
  </head>
  <body>
    <header>
      
      <p class="headertitle">Site name</p>
    </header>
```

Include same stylesheet in every page for consistent look

Can use variables to display page-specific data.

## Using the header in a page

- In a page using the header:

```
<?php
    $pageTitle = "Index";
    require_once('pageheader.php');
?>
```
- Set the page title.
- The content of the required file is directly included in the text of the document, so variables are in scope.

## TODOs this week:

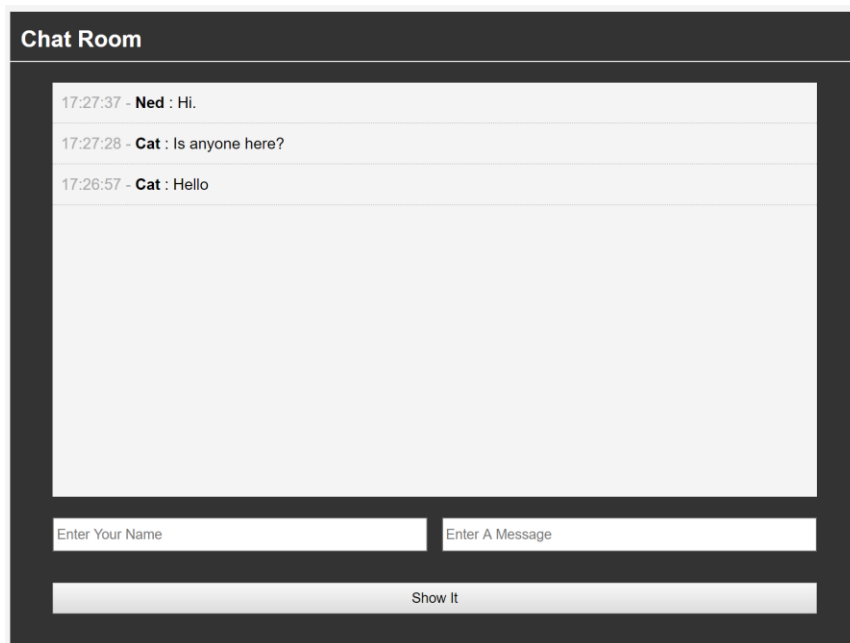
- Look at your ERD feedback.
- Add PHP scripts to your project to insert, display, or delete database rows.
- (You should now understand the full tech stack used in the project: \*AMP, PHP, SQL, git, and basic HTML/CSS.)

# PHP example

# Basic chatroom

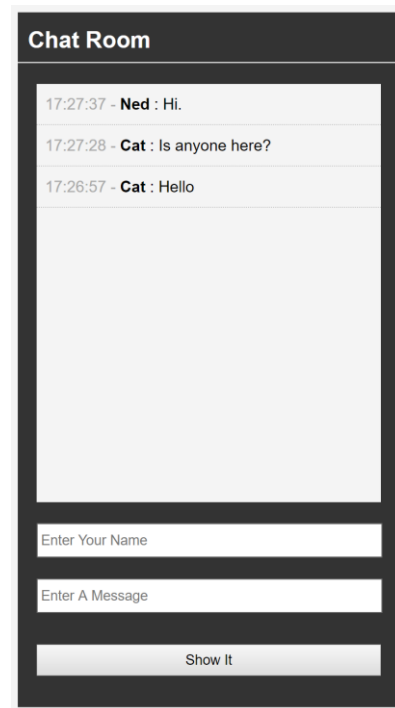
- Provides a simple chat application.
- Illustrates the combination of:
  - HTML + CSS to have a basic responsive interface.
  - PHP code running on the web server.
  - MySQL database running on the database server.

# User interface

A mockup of a chat room user interface. It has a dark grey header bar with the text 'Chat Room'. Below the header is a large white rectangular area for chat messages. Three messages are displayed: '17:27:37 - Ned : Hi.', '17:27:28 - Cat : Is anyone here?', and '17:26:57 - Cat : Hello'. At the bottom of the chat area is a dark grey bar containing two white input fields: 'Enter Your Name' and 'Enter A Message'. Below these fields is a wide, light grey button with the text 'Show It'.

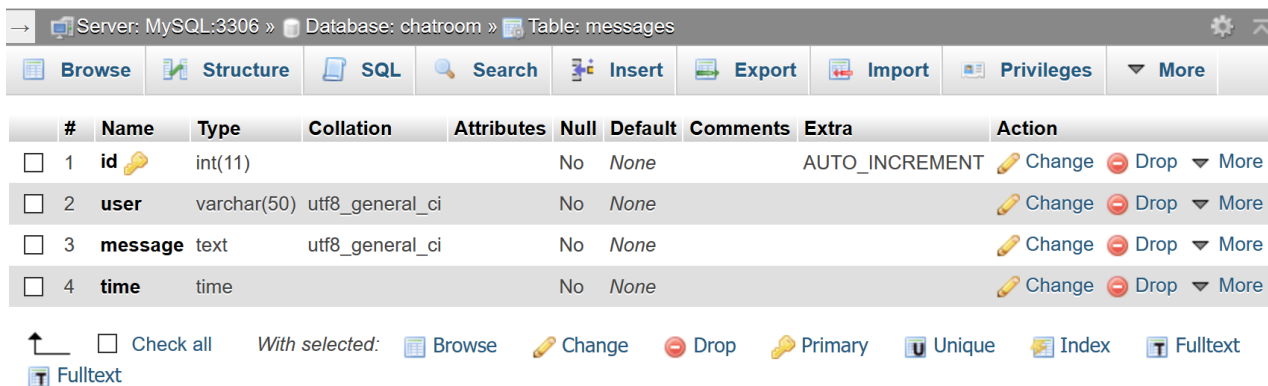
# Responsive design

- Changes layout for smaller screens.



# The database

- Created using phpMyAdmin.
- One table, 4 columns, id is the primary key.



The screenshot shows the phpMyAdmin interface for a MySQL database named 'chatroom'. The selected table is 'messages'. The table structure is displayed with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	<b>id</b>			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	<b>user</b>	varchar(50)	utf8_general_ci	No	None			Change  Drop  More
<input type="checkbox"/>	3	<b>message</b>	text	utf8_general_ci	No	None			Change  Drop  More
<input type="checkbox"/>	4	<b>time</b>	time		No	None			Change  Drop  More

At the bottom, there are checkboxes for 'Check all' and 'Fulltext', and a section for 'With selected:' containing icons for 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', and 'Fulltext'.



## index.php (1/3)

Includes / runs  
another PHP file

```
<?php include 'database.php'; ?>
<?php
    $query = "SELECT * FROM messages ORDER BY id
DESC";
    $messages = mysqli_query($connection, $query);
?>
<!doctype html>
<html>
    <head>
        <meta charset="utf-8"/>
        <title>CHAT!</title>
        <link rel="stylesheet" href="css/style.css"
type="text/css" />
    </head>
```

Retrieve all  
messages from DB

## index.php (2/3)

```
<body>
  <div id="container">
    <header>
      <h1>Chat Room</h1>
    </header>
    <div id="messages">
      <ul>
        <?php while($row = mysqli_fetch_assoc($messages)) : ?>
          <li class="message">
            <span><?php echo $row['time'] ?> - </span><strong>
              <?php echo $row['user']?></strong>
              : <?php echo $row['message'] ?>
            </li>
          <?php endwhile; ?>
        </ul>
      </div>
```

Interleaved HTML and PHP  
to display messages

## index.php (3/3)

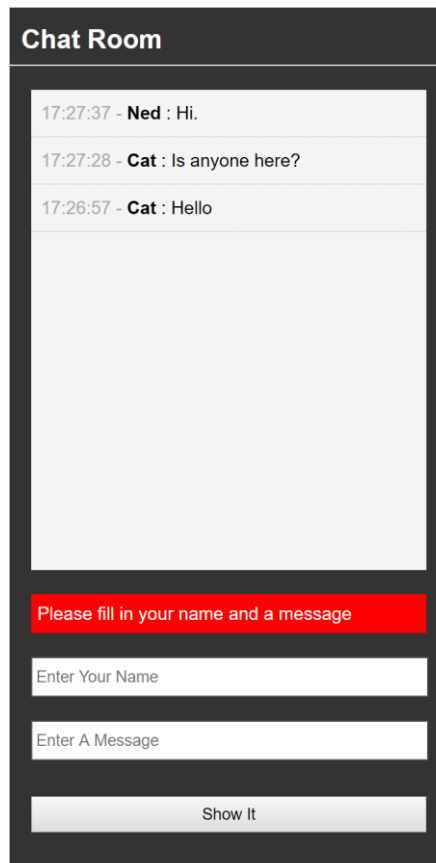
If there was an error in previous input data, display a message

```
<div id="input">
  <?php if (isset($_GET['error'])) : ?>
    <div class="error"><?php echo $_GET['error']; ?></div>
  <?php endif; ?>
  <form method="post" action="process.php">
    <input type="text" id="user" name="user"
placeholder="Enter Your Name"/>
    <input type="text" id="newmessage" name="message"
placeholder="Enter A Message"/>
    <input id="show-btn" type="submit" name="submit"
value="Show It"/>
  </form>
</div>
</body>
</html>
```

Form is submitted to process.php

# Data validation

- `index.php?error=Please+fill+in+your+name+and+a+message`

A mobile app interface for a chat room. At the top is a dark header with the text 'Chat Room'. Below this is a scrollable list of messages. The first message is '17:27:37 - Ned : Hi.' on a light grey background. The second is '17:27:28 - Cat : Is anyone here?' on a white background. The third is '17:26:57 - Cat : Hello' on a light grey background. Below the messages is a red rectangular box containing the text 'Please fill in your name and a message'. Underneath the red box are two white input fields with dark borders. The first field is labeled 'Enter Your Name' and the second is labeled 'Enter A Message'. At the bottom of the interface is a grey button with the text 'Show It'.

# database.php

```
<?php
    $connection = mysqli_connect("localhost",
    "chatadmin", "adminpassword", "chatroom");

    if (mysqli_connect_errno())
        echo 'Failed to connect to the MySQL
server: '. mysqli_connect_error();

?>
```

## process.php (1/2)

Clean up the input to avoid SQL injection attacks

```
<?php
include 'database.php';
if (isset($_POST['submit'])) {
    $user = mysqli_real_escape_string($connection,
$_POST['user']);
    $message = mysqli_real_escape_string($connection,
$_POST['message']);
    date_default_timezone_set('Europe/London');
    $time = date('H:i:s', time());
    if (!isset($user) || $user == '' || !isset($message) ||
$message == '') {
        $error = "Please fill in your name and a message";
        header("Location: index.php?error=" . urlencode($error));
        exit();
    }
}
```

## process.php (2/2)

Insert into database

```
else {  
    $query = "INSERT INTO messages (user, message, time)  
              VALUES ('$user', '$message', '$time')";  
    if (!mysqli_query($connection, $query)) {  
        die('Error: ' . mysqli_error($connection));  
    }  
    else {  
        header('Location: index.php');  
        exit();  
    }  
}  
}  
?>
```

Redirect to this URL

# The CSS

- Refer to .css file for some more examples:

```
*{  
  margin: 0;  
  padding: 0;  
}  
body {  
  font-family: arial;  
  font-size: 15px;  
  line-height: 1.5em;  
  background: #f4f4f4;  
}
```

```
#container {  
  background: #333333;  
  margin: 50px auto;  
  overflow: auto;  
  width: 60%;  
}  
header h1 {  
  color: #ffffff;  
  font-size: 22px;  
  ...
```