Web Programming MySQL

INSERTing Data into a Table within a Database

Dr Stewart Blakeway

Lecturer in Computer Science
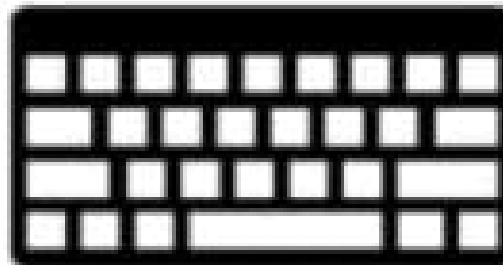
COMP23111
Database Systems

Intro

Getting User Input

- A common approach to getting user input is to implement a form.

- Let's recap (with some variation on the approach we have seen)

A basic HTML Form

```html
<form>
 <label for="username">Username</label>
 <input type="text" id="username" name="username">

 <label for="password">Password</label>
 <input type="password" id="password" name="password">

 <input type="submit" value="Register">
</form>
```

Username

Password

Register

Data sent to itself?

```
<form action='$_SERVER[PHP_SELF]' method='POST'>
```

```php
<?php
function registerForm()
{
  return '
    <form method="POST">
      <label for="username">Username</label>
      <input type="text" id="username" name="username">

      <label for="password">Password</label>
      <input type="password" id="password" name="password">

      <input type="submit" value="Register">
    </form>
  ';
}
?>
```
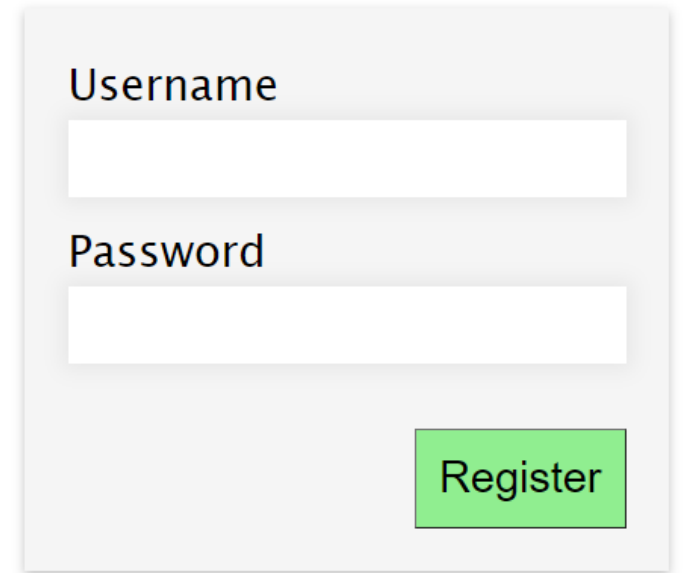
Functions

- Invoking a function is easy enough, we just call the function name

    `registerForm();`

-

But… the page will deal with both getting the data, and processing the data so we need some flow control!

- If we don't have the user input, then get the user input
- Else, process the user input.

How do we know if we have User Input?

- We can use PHP to check we have user input.

- If we don't have user input the POST variable will either not exist or will be empty.

- How do we know if the POST variable is empty?
    - We simply ask if the POST array is empty.
    - If the form has not been submitted, then a POST will be empty (or not set).

```
empty($_POST)
```

```php
<?php
  if (empty($_POST)) registerForm();
  else // do something with the data
?>
```

```php
if (empty($_POST)) echo(registerForm());
else echo("we have post");


if (empty($_POST))
{
  $theForm = registerForm();
  // do a bunch of other stuff
  // ...
  // sometime later
  echo($theForm);
}
else // do something with the data
```

So far…

- We check if a posted value is **empty** or not

- If this evaluates to true we get the form and display it

- Otherwise (else) we process the input.

- But… we have not yet created a function
  to deal with the processing of the input.

- We will do this next.

Function Definition

```
function processInput()
{
  $un = $_POST['username'];
  $pw = $_POST['password'];
}
```

```php
function registerUser()
{
  $un = $_POST['username'];
  $pw = $_POST['password'];

  $sql = "INSERT INTO user (username, password)
          VALUES (:userId, :userPassword)";
```

```php
function registerUser()
{ ...
  $sql = "INSERT INTO user (username, password)
          VALUES (:userId, :userPassword)";

  $pdo = new pdo('mysql:host=localhost;dbname=myDB', 'root', '');
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
}
```

```php
function registerUser()
{ ...
  $sql = "INSERT INTO user (username, password)
          VALUES (:userId, :userPassword)";

  $pdo = new pdo('mysql:host=localhost;dbname=myDB', 'root', '');
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
}
```

```php
function registerUser()
{
    $un = $_POST['username'];
    $pw = $_POST['password'];

    $sql = "INSERT INTO user (username, password)
            VALUES (:userId, :userPassword)";
    ...
    $stmt = $pdo->prepare($sql);
    $stmt->execute([
                    'userId'       => $un,
                    'userPassword' => $pw
                ]);
}
```

Password Hash

- I always set my passwords to: "incorrect"

- That way, when I forget my password the program will remind me what it is.

- Your password is incorrect. Try again!

Password Hash

```
function registerUser()
{
  $un = $_POST['username'];
  $pw = $_POST['password'];
  $pw = password_hash($pw, PASSWORD_DEFAULT);
  ...
```

To conclude

- One file (register.php)
- Two functions
  - registerForm and registerUser
- A bit of flow control to call the correct function
  - An if statement and the use of the empty function and POST
- Result...
  - We obtained some user input
  - Processed that input
  - Inserted the input into a table in our database

Web Programming MySQL

INSERTing Data into a Table within a Database

Dr Stewart Blakeway

Lecturer in Computer Science

COMP23111
Database Systems