CHT2520 Advanced Web Programming

Matthew Mantle m.e.mantle@hud.ac.uk

Today's Session - Basic Approaches to Writing Maintainable Code

- Using require()/include()

- Using functions

- The MVC design pattern

What's wrong with last week's example?

- Lots of duplicate code

  - In the HTML

  - When working with a database

  - Remember the DRY principle

- The code isn't structured or organised

Removing duplication using `require()`

- `require()` allows use to load one file into another.

- E.g. put database connection code in a separate file and then execute `require()` to use it in every page that needs it.

  - We can also `require()` with HTML code.

- There is also an `include()` function that works in the same way.

Using Functions

```php
//saved in fncs.php
function getConnection()
{
        $conn = new PDO('mysql:host=localhost;dbname=MyDatabase', 'MyUsername'
        return $conn;
}
```

- Put reusable code in a function.

- Use `require('fncs.php')` to load the functions file.

- Call the function to execute the code

- Better than just using `require()` - We can name the block of code

MVC - Model-View-Controller

- A design pattern (guidelines) for organising code.

- Laravel uses the MVC design pattern

  - So do lots of other web frameworks Spring, Ruby on Rails, Django etc.

MVC - Seperation of concerns

- MVC keeps code organised by splitting it into clearly defined categories: models, views and controllers

- Model

  - Business logic (code for working with the database)

- View

  - The HTML code and PHP code that generates output e.g. `echo`.

- Controller

  - Receives requests, asks the model to perform tasks, loads views

MVC - Flow

- For each action in a web application, typically:-

    1. **Controller** receives a request e.g. the user has submitted a form.

    2. The **controller** asks the **model** to perform a task e.g. insert data into the database.

    3. The **model** completes the task and returns the result to the controller.

    4. The **controller** loads a **view** e.g. a confirmation message.

    5. The user interacts with the **view** to select the next action.

Practical Work

- Re-writing the basic CRUD example from last week using `require()`, functions and the MVC design pattern.