

H16S35 - Managing a Web Server

7 - PHP and MySQLi
Michael Ferrie

michael.ferrie@edinburghcollege.ac.uk



CONTENTS

01 PHP

What is PHP?
PHP history/versions

02 MySQLi

MySQLi examples
MySQLi in action

03 Example PHP and mysqli login system

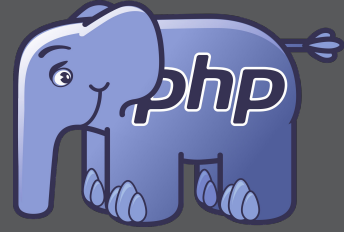
Connecting a website to a database

04 Q and A

Any questions

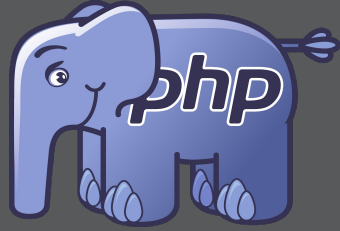
PHP

michael.ferrie@edinburghcollege.ac.uk



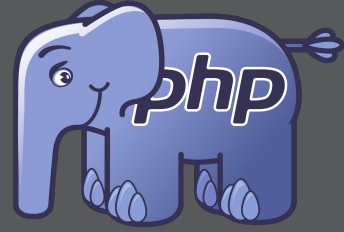
What does PHP Stand for?

- PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development
- PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Pre Hypertext Preprocessor
- It was originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group
- The elePHPant was created by Vincent Pontier



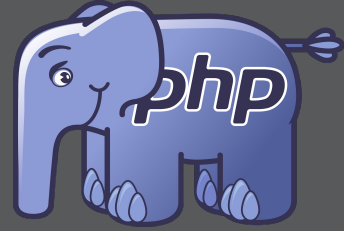
PHP operation

- PHP code may be executed with a command line interface, embedded into HTML, or used in content management systems/frameworks
- PHP is usually processed by a PHP interpreter implemented as a module in a web server
- The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data
- PHP can be used for many programming tasks outside of the web context



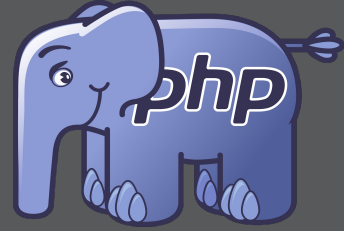
PHP Versions 1-5

- 1.0 (8 June 1995) Officially called "Personal Home Page Tools (PHP Tools)". This is the first use of the name "PHP"
- 2.0 (1 November 1997) Officially called "PHP/FI 2.0". The first release that could be characterised as PHP, being a stand-alone language, many features have endured to the present day
- 3.0 (6 June 1998) Development moves from one person to multiple developers.
- 4.0 (22 May 2000) Added more advanced two-stage parse/execute tag-parsing system called the Zend engine
- On July 14, 2004, PHP 5 was released, powered by the new Zend Engine II, official security support for PHP 5.6 ended on 31 December 2018



PHP 6 2005-2010 (Abandoned)

- PHP received mixed reviews due to lacking native Unicode support. In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP
- Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0
- A shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF-16, which is rarely used in a web context, led to delays in the project.
- As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features back-ported from PHP 6. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6



PHP 2015-present

- During 2014-2015, a new major PHP version 7 was developed
- The foundation of PHP 7 is a PHP branch that was originally dubbed PHP next generation (phpng)
- It was authored by Dmitry Stogov, Xincheng Hui and Nikita Popov, and aimed to optimize PHP performance by refactoring the Zend Engine while retaining near-complete language compatibility
- As of 14 July 2014, WordPress-based benchmarks, which served as the main benchmark suite for the phpng project, showed an almost 100% increase in performance compared to php5

MySQLi

michael.ferrie@edinburghcollege.ac.uk

MySQLi

- The MySQLi Extension (MySQL Improved) is a relational database driver used in the PHP scripting language to provide an interface with MySQL databases
- The PHP code consists of a core, with optional extensions to the core functionality
- PHP's MySQL-related extensions, such as the MySQLi extension, and the MySQL extension, are implemented using the PHP extension framework
- An extension typically exposes an API to the PHP developer, to allow its facilities to be used programmatically

MySQLi

MySQLi example

Showing how to make a connection to a database called myDB:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

MySQLi files required:

- **index.html** — Login form created with HTML5 and CSS3, we don't need to use PHP in this file so we can just save it as HTML
- **style.css** — The stylesheet (CSS) for our secure login app
- **config.php** — Connect to the database, validate form data, retrieve database results, and create new sessions
- **logout.php** — Destroy the logged in sessions and redirect the user
- **home.php** — Basic home page for logged in users.
- **profile.php** — Select the user's account from our MySQL database and display the result

MySQLi

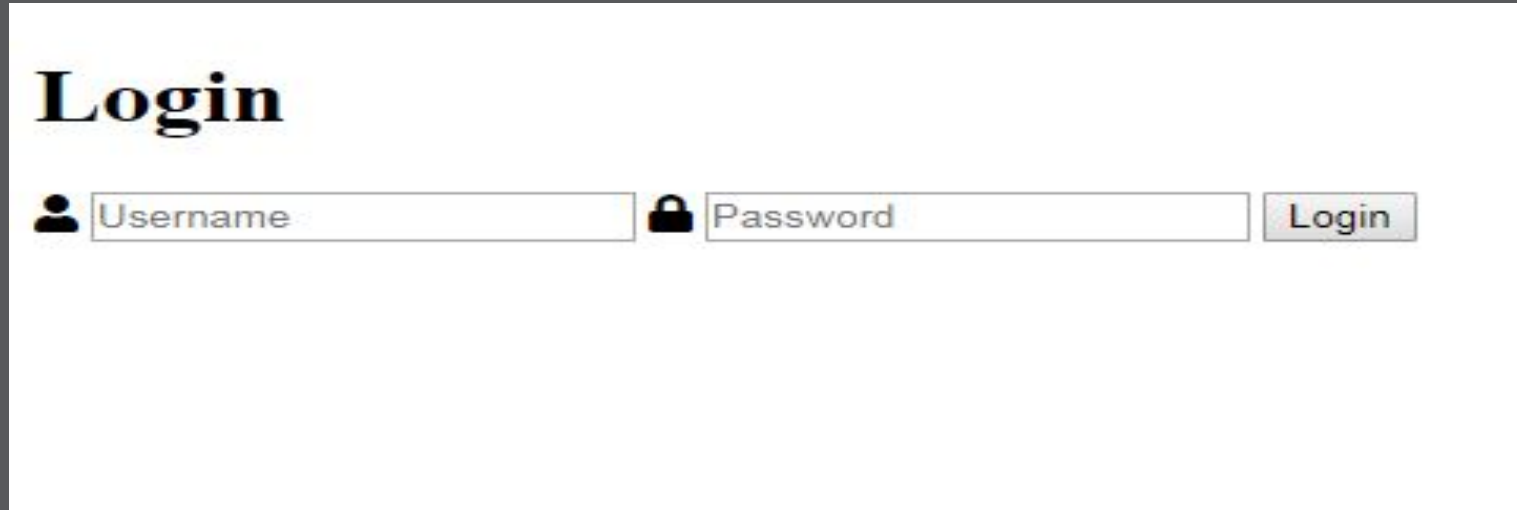
index.html

```
<head> <meta charset="utf-8">
<title>Login</title><link rel="stylesheet" href="styles.css"> </head>
<body>
<div class="login"><h1>Login</h1>
<form action="authenticate.php" method="post">
<label for="username"><i class="fas fa-user"></i>
</label>
<input type="text" name="username" placeholder="Username" id="username" required>
<label for="password"><i class="fas fa-lock"></i>
</label>
<input type="password" name="password" placeholder="Password" id="password" required>
<input type="submit" value="Login">
</form> </div> </body>
```


1

MySQLi

index.html - in browser



Login





MySQLi

styles.css

```
body {  
    background-color: #435165;  
}  
  
.login {  
    width: 400px;  
    background-color: #ffffff;  
    box-shadow: 0 0 9px 0 rgba(0, 0, 0, 0.3);  
    margin: 100px auto;  
}  
  
.login h1 {  
    text-align: center;  
    color: #5b6574;  
    font-size: 24px;  
    padding: 20px 0 20px 0;  
    border-bottom: 1px solid #dee0e4;  
}
```

1

MySQLi

html + css

The image shows a login form with a white background and a blue border. At the top, the word "Login" is written in a bold, black font. Below this, there are two input fields. The first field is labeled "Username" and has a blue icon of a person to its left. The second field is labeled "Password" and has a blue icon of a padlock to its left. At the bottom of the form, there is a blue button with the word "Login" written in white.

MySQLi

sql

```
create database phplogin;
```

```
use phplogin;
```

```
CREATE TABLE IF NOT EXISTS `accounts` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `username` varchar(50) NOT NULL,  
    `password` varchar(255) NOT NULL,  
    `email` varchar(100) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
INSERT INTO `accounts` (`id`, `username`, `password`, `email`) VALUES (1,  
'test', '$2y$10$SfhYIDtn.iOuCW7zfoFLuuZHX6lja4Ca', 'test@test.com')
```

MySQLi

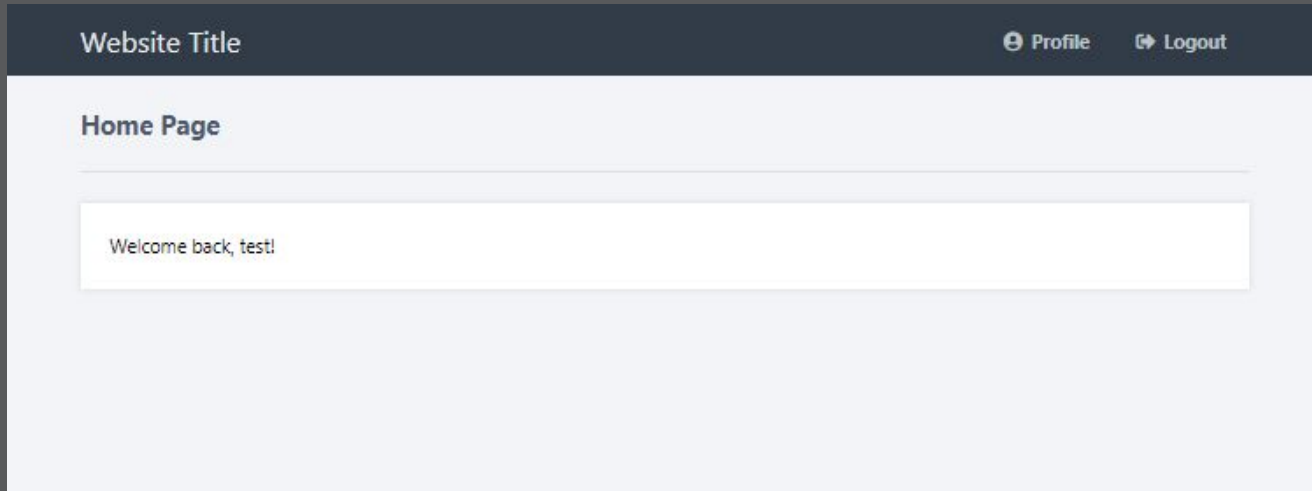
config.php

```
<?php
session_start();
// Change this to your connection info.
$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'jim';
$DATABASE_PASS = 'jims_password';
$DATABASE_NAME = 'phplogin';
// Try and connect using the info above.
$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
if ( mysqli_connect_errno() ) {
    // If there is an error with the connection, stop the script and display the error.
    die ('Failed to connect to MySQL: ' . mysqli_connect_error());
}
```

1

MySQLi

home.php



Thanks for listening. Any questions?

Additional information: <https://www.php.net/manual/en/book.mysql.php>

michael.ferrie@edinburghcollege.ac.uk