Managing a Web Server

Lab 7: MariaDB Authenticated php Login Form

Aim: In this lab you use PHP to submit form data, which is authenticated by a database.

Introduction: This will require a Debian installation with Apache 2 running and the default page showing. PHP, and MariaDB installed and phpinfo() displaying correctly. This will work in any directory in /var/www/html, so you could put this into /var/www/html/site2/3/4. You need to be confident in doing this, you need to know each step exactly, make a log book on exactly what you did… you will need this to do the final assessment. Although I will look at the logbook, it is actually ***your*** notes for the course…

Part 1: Set Up the Database

1.1 Connect to the MariaDB server as root, create a new user and a new database, then grant all privileges on the new database, to the new user. Look at the parts in red and see how they relate to each other.

```
mariadb -u root -p

GRANT ALL ON *.* TO 'admin_user'@'localhost' IDENTIFIED BY
'admin_password' WITH GRANT OPTION;

CREATE DATABASE staff_website_db default character set utf8 default
collate utf8_bin;

GRANT ALL privileges on staff_website_db.* to 'admin_user'@'localhost'
identified by 'admin_password';

flush privileges;

exit;
```

1.2 Connect to the new database and make a table where we will store the staff usernames and passwords, we will use an auto increment id for the primary key. Make sure you can see the staff database, when you run show grants.

```
mariadb -u admin_user -p

show grants;

use staff_website_db;

CREATE TABLE users (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Part 2: Create PHP Files

2.1 We are going to create several files make a new directory in the www root that we will place the new files into, you need to copy and paste the data in from the lab, if the file is too big download it from the Moodle and paste it in. As this isn't a web development unit, we won’t be writing the php from scratch.

When the information is on the clipboard, in vi **NORMAL mode** you have to type :set paste, **then** press ‘i’ to get into insert mode, then either press SHIFT+INSERT or CTRL+SHIFT+V to paste.

2.2 File 1 Create **`config.php`** and paste in the php for this, however the following changes need to be made, what lovely colours vim uses for php. These values need to be changed to the values defined in part 1.1.

```
define('DB_SERVER', 'localhost');

define('DB_USERNAME', 'admin_user');

define('DB_PASSWORD', 'mypassword');

define('DB_NAME', 'staff_website_db');
```

2.3 In the same directory as the php config file, create the files: **`register.php, login.php, welcome.php, logout.php & reset-password.php`**

If you go to the files in the browser just now they will just display as text, this is due to permissions, set the permissions now. You need to **`cd`** to the parent directory of your website then run:

```
chmod -R 777 /var/www/this_parent_directory/*
```

In this example - my website is in /html so I **cd** up one level in the hirearchy, to /var/www and run this on the html directory:

```
root@debian:/var/www/html# cd ..
root@debian:/var/www# chmod 777 -R /var/www/html/*
```

Part 3: Create a User and Log in

3.1 If you have the login form up, create a new user, and to verify this has worked return to mariadb and run, if you see the user, you can see that we are hashing the password too thanks php, add another one and run the query.

```
MariaDB [(none)]> use staff_website_db

MariaDB [staff_website_db]> select * from users ;
```

Part 4: Upload to Moodle

Upload completed work to the Moodle as a PDF.