## LAMP Stack

- LAMP stands for Linux, Apache, MySQL and PHP
- LAMP is a package of software that allows for building dynamic websites and web applications

## Installation

- First we need a Linux operating system, which is already installed on your PI's
- Second we will install Apache Web Server
- Navigate to the terminal and enter the following lines:
  - sudo apt-get update
  - sudo apt-get install apache2
- After apache has installed we can start our apache server with the following command:
  - sudo /etc/init.d/apache start
- We can now view our default server web page by going to our web browser and typing localhost in the address bar
- We can also type localhost/index.html and notice we get the same page, as the default web page is set to index.html
- Return to the terminal and navigate to the directory where the index.html file is stored by typing the following line:
  - cd /var/www/html
- using the ls command we should see a file named index.html. This is where the html files for our apache server are saved and where we will place our html and php files, so don't forget it!
- Now we will install MySQL by typing the following lines into the terminal:
  - sudo apt-get update
  - sudo apt-get install mysql-server
  - (optional) mysql_secure_installation (updates old software for security reasons)
- We can now start the MySQL database by typing the following line in the terminal:
  - sudo /etc/init.d/mysql start
- Finally we will install PHP by typing the folloing line in the terminal:
  - sudo apt-get update
  - sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql

## Creating Our Web Page

- In the terminal navigate to your apache directory by typing:
  - cd /var/www/html

- now we will begin creating our web page by typing:
    - nano form.html
- This will open a terminal based text editor in a file called form.html
- Insert the following code into the file:

```
<html>
<head>
<meta charset=”utf-8”>
</head>
<body
        <form action=”action.php” method=”POST”>
        <label>User ID:</label>
        <input type=”text” id=”uid” name=”uid” placeholder = “User ID” required>
        <label>Password:</label>
        <input type=”text” id=”passid” name=”passid” required>
        <input type=”submit” value=”Submit” />
</body>
</html>
```

- Pressing CTRL+X will allow us to save and exit the text editor
- We can now view our html file by going to our web browser and typing localhost/form.html in the web browser

## Creating Our Database

- We can access our databases by typing the following line in the terminal:
    - mysql -u root -p
- Use the password you entered when you first installed mysql when prompted
- You are now using the MariaDB monitor to interface with your database
- We will create our first database and table by typing the following SQL:
    - CREATE DATABASE usersDB;
    - USE usersDB;
    - CREATE TABLE users(userID VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL);
    - INSERT INTO users(userID, password) VALUES(“name”, “pass”);
- Now we will create a database user account and give them priviliges to access the users table with the following command;
    - CREATE USER 'user'@'localhost' identified by 'pass';
    - GRANT ALL PRIVILEGES ON usersDB.* TO 'user'@'localhost';
    - exit;

## Creating Our PHP File

- In the terminal, navigate to the apache directory:
  - cd /var/www/html
- Now we will open the nano text editor to create our PHP file:
  - nano action.php;

```
<html>
<head>
	<meta charset=”utf-8”>
</head>
<body>
	<?php
	$host = “localhost”;
	$username = “user”;
	$password = “pass”;
	$db_name = “usersDB”;
	$conn = new mysqli($host, $username, $password, $db_name);
	if($conn->connect_error){
		die(“Connection Failed: “ . $conn->connect_error);
	}
	$uid = $_POST['uid'];
	$pid = $_POST['passid'];
	$SQL = “SELECT * FROM users WHERE userID = '$uid' AND password = '$pid' “;
	$result = mysqli_query($conn, $SQL);
	if(mysqli_num_rows($result) > 0){
		echo “<h4>” . “Personal Information” . “</h4>”, “</br>”;
		$row = mysqli_fetch_assoc($result);
		echo “<p>” . “Username: “ . $row[“userID”] . “</p>”;
		echo “<p>” . “Password: “ . $row[“password”] . “</p>”;
	}else{
		echo “Invalid Username or Password”;
	}
	?>
</body>
</html>
```

- Now let us test our code by typing localhost/form.html into the web browser. It should display a form with two text fields and a submit button.
- Put in the username field and password field the userID and password you first stored in your users table. If successful, you should be sent to a second page which displays your username and password.

# SQL Injection

- Return to localhost/form.html in your web browser. Now put in the same userID, but place this string into your password:
  - ' or 'x' = 'x
- If done correctly, you should be sent to a page