PHP

PHP stands for "PHP: **Hypertext Preprocessor**," a recursive acronym that reflects its role in processing web content. It is a general-purpose, open-source scripting language primarily used for server-side web development, enabling the creation of dynamic and interactive websites by embedding code directly within HTML. PHP code is processed on the web server before being sent to the client's browser, where it generates HTML output based on the script's execution. The language is widely supported across various operating systems and web servers, and it can interface with numerous databases, including MySQL, PostgreSQL, and SQLite3.

program

Index.php

```
<?php
require_once __DIR__ . '/functions.php';

$errors = [];
$old = ['email' => ''];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (
$errors[] = "Invalid CSRF token.";
} !check_csrf()) {

$email = trim($_POST['email'] ?? '');
$password = $_POST['password'] ?? '';
$old['email'] = $email;

if (!filter_var($email, FILTER_VALIDATE_EMAIL)) $errors[] = "Enter a valid email.";
if (empty($password)) $errors[] = "Enter your password.";

if (empty($errors)) {
$stmt = $pdo->prepare("SELECT id, username, password FROM users WHERE email = :e
LIMIT 1");
$stmt->execute([':e' => $email]);
$user = $stmt->fetch();

if ($user && password_verify($password, $user['password'])) {
session_regenerate_id(true);
$_SESSION['user_id'] = $user['id'];
$_SESSION['username'] = $user['username'];
flash('success', 'Welcome back, ' . $user['username'] . '!');
header('Location: https://recyclezone.neocities.org/');
exit;
} else {
$errors[] = "Email or password incorrect.";
}
}
}

$token = csrf_token();
$success_msg = flash('success');
$error_msg = flash('error');
?>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Login — MyApp</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="bg">
<main class="center">
<div class="card animate-up">
<h2>Welcome back</h2>

<?php if ($success_msg): ?>
<div class="msg success bounce"><?= e($success_msg) ?></div>
<?php endif; ?>

<?php if (!empty($errors)): ?>
<div class="msg error shake"><ul><?php foreach ($errors as $err) echo '<li>'.e($err).'</li>';
?></ul></div>
<?php endif; ?>

<form id="loginForm" method="post" novalidate>
<input type="hidden" name="csrf_token" value="<?= e($token) ?>">
<div class="field">
<label>Email</label>
<input name="email" type="email" required value="<?= e($old['email']) ?>"/>
</div>
```

```
<div class="field">
<label>Password</label>
<div class="pw">
<input name="password" type="password" required id="login_password"/>
<button type="button" class="show-pw" data-target="login_password">Show</button>
</div>
</div>

<button class="btn" type="submit">Log in</button>
<p class="center small">Don't have an account? <a href="register.php">Register</a></p>
</form>
</div>
</main>

<script src="assets/js/main.js"></script>
</body>
</html>
```

Register.php

```
<?php
require_once __DIR__ . '/functions.php';

$errors = [];
$old = ['username'=>'', 'email'=>''];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (!check_csrf()) {
$errors[] = "Invalid CSRF token.";
}
$username = trim($_POST['username'] ?? '');
$email = trim($_POST['email'] ?? '');
$password = $_POST['password'] ?? '';
$password_confirm = $_POST['password_confirm'] ?? '';

$old['username'] = $username;
$old['email'] = $email;

// Server-side validation
if (strlen($username) < 3) $errors[] = "Username must be at least 3 characters.";
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) $errors[] = "Invalid email.";
if (strlen($password) < 6) $errors[] = "Password must be at least 6 characters.";
if ($password !== $password_confirm) $errors[] = "Passwords do not match.";

if (empty($errors)) {
// Check duplicates
$stmt = $pdo->prepare("SELECT id, username, email FROM users WHERE username = :u OR
email = :e LIMIT 1");
$stmt->execute([':u' => $username, ':e' => $email]);
$row = $stmt->fetch();
if ($row) {
if ($row['username'] === $username) $errors[] = "Username already taken.";
if ($row['email'] === $email) $errors[] = "Email already registered.";
} else {
// Insert user
$hash = password_hash($password, PASSWORD_DEFAULT);
$ins = $pdo->prepare("INSERT INTO users (username, email, password) VALUES (:u, :e, :p)");
$ins->execute([':u' => $username, ':e' => $email, ':p' => $hash]);

flash('success', 'Registration successful. Please log in.');
header('Location: login.php');
exit;
}
}
}

$token = csrf_token();
?>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Register — MyApp</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="bg">
<main class="center">
<div class="card animate-up">
```

```
<h2>Create account</h2>

<?php if (!empty($errors)): ?>
<div class="msg error shake"><ul><?php foreach ($errors as $err) echo '<li>'.e($err).'</li>';
?></ul></div>
<?php endif; ?>

<form id="registerForm" method="post" novalidate>
<input type="hidden" name="csrf_token" value="<?= e($token) ?>">
<div class="field">
<label>Username</label>
<input name="username" required minlength="3" value="<?= e($old['username']) ?>" />
</div>

<div class="field">
<label>Email</label>
<input name="email" type="email" required value="<?= e($old['email']) ?>" />
</div>

<div class="field">
<label>Password</label>
<div class="pw">
<input name="password" type="password" required minlength="6" id="password"/>
<button type="button" class="show-pw" data-target="password">Show</button>
</div>
</div>

<div class="field">
<label>Confirm Password</label>
<div class="pw">
<input name="password_confirm" type="password" required minlength="6"
id="password_confirm"/>
<button type="button" class="show-pw" data-target="password_confirm">Show</button>
</div>
</div>

<button class="btn" type="submit">Register</button>
<p class="center small">Already have an account? <a href="index.php">Log in</a></p>
</form>
</div>
```

```
</main>

<script src="assets/js/main.js"></script>
</body>
</html>
```

config.php

```
<?php
// config.php
// Development settings: show errors (disable on production)
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

// Start session (single place)
session_start([
'cookie_httponly' => true,
'cookie_samesite' => 'Lax',
// 'cookie_secure' => true, // enable on HTTPS
]);

// Database credentials — change these
$DB_HOST = 'localhost';
$DB_NAME = 'myapp';
$DB_USER = 'root';
$DB_PASS = ''; // put your password

$dsn = "mysql:host={$DB_HOST};dbname={$DB_NAME};charset=utf8mb4";
$options = [
PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
];

try {
$pdo = new PDO($dsn, $DB_USER, $DB_PASS, $options);
} catch (PDOException $e) {
// For development only: show detailed error. In production, log instead.
die('Database connection failed: ' . $e->getMessage());
}
```

functions.php

```
<?php
// functions.php
require_once __DIR__ . '/config.php';

/** Flash messages **/
function flash($key, $msg = null) {
if ($msg === null) {
if (!empty($_SESSION['flash'][$key])) {
$m = $_SESSION['flash'][$key];
unset($_SESSION['flash'][$key]);
return $m;
}
return null;
}
$_SESSION['flash'][$key] = $msg;
}

/** CSRF Token helpers **/
function csrf_token() {
if (empty($_SESSION['csrf_token'])) {
$_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}
return $_SESSION['csrf_token'];
}

function check_csrf() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (empty($_POST['csrf_token']) || empty($_SESSION['csrf_token']) ||
!hash_equals($_SESSION['csrf_token'], $_POST['csrf_token'])) {
return false;
}
}
return true;
}

/** Escape output **/
```

```
function e($str) {
return htmlspecialchars($str ?? '', ENT_QUOTES | ENT_SUBSTITUTE, 'UTF-8');
}

/** Auth helpers **/
function is_logged_in() {
return !empty($_SESSION['user_id']);
}

function require_login() {
if (!is_logged_in()) {
flash('error', 'You must log in first.');
header('Location: login.php');
exit;
}
}
```

home.php

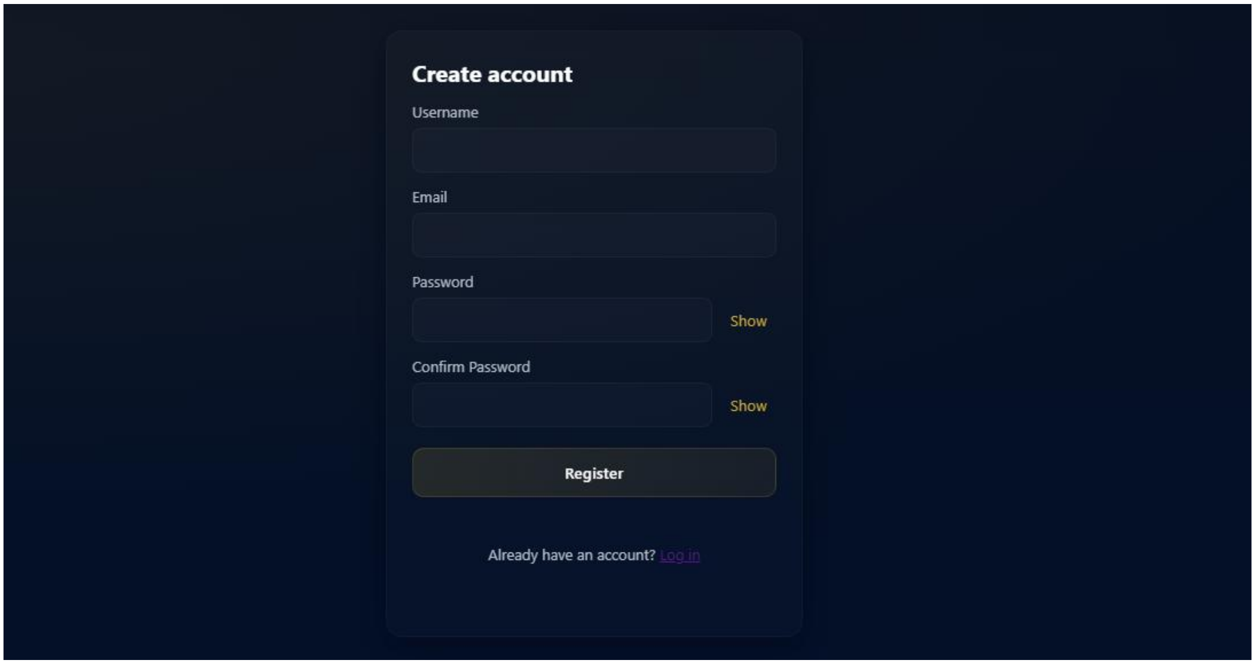
```
<?php
require_once __DIR__ . '/functions.php';
require_login();

$username = $_SESSION['username'] ?? 'User';
$success = flash('success');
?>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Home — MyApp</title>
<meta name="viewport" content="width=device-width,initial-scale=1">
<link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="bg">
<header class="top">
<div class="wrap">
<h3>MyApp</h3>
<nav>
<span>Hello, <?= e($username) ?></span>
<a class="btn small" href="logout.php">Logout</a>
```

```
</nav>
</div>
</header>

<main class="wrap content">
<?php if ($success): ?>
<div class="msg success"><?= e($success) ?></div>
<?php endif; ?>

<section class="card animate-up">
<h2>Dashboard</h2>
<p>This page is protected and visible only after login.</p>
</section>
</main>
</body>
</html>
```

logout.php

```
<?php
require_once __DIR__ . '/functions.php';
// Clear session and logout
session_unset();
session_destroy();
session_start();
flash('success', 'You have been logged out.');
header('Location: login.php');
exit;
```

Output



Create account

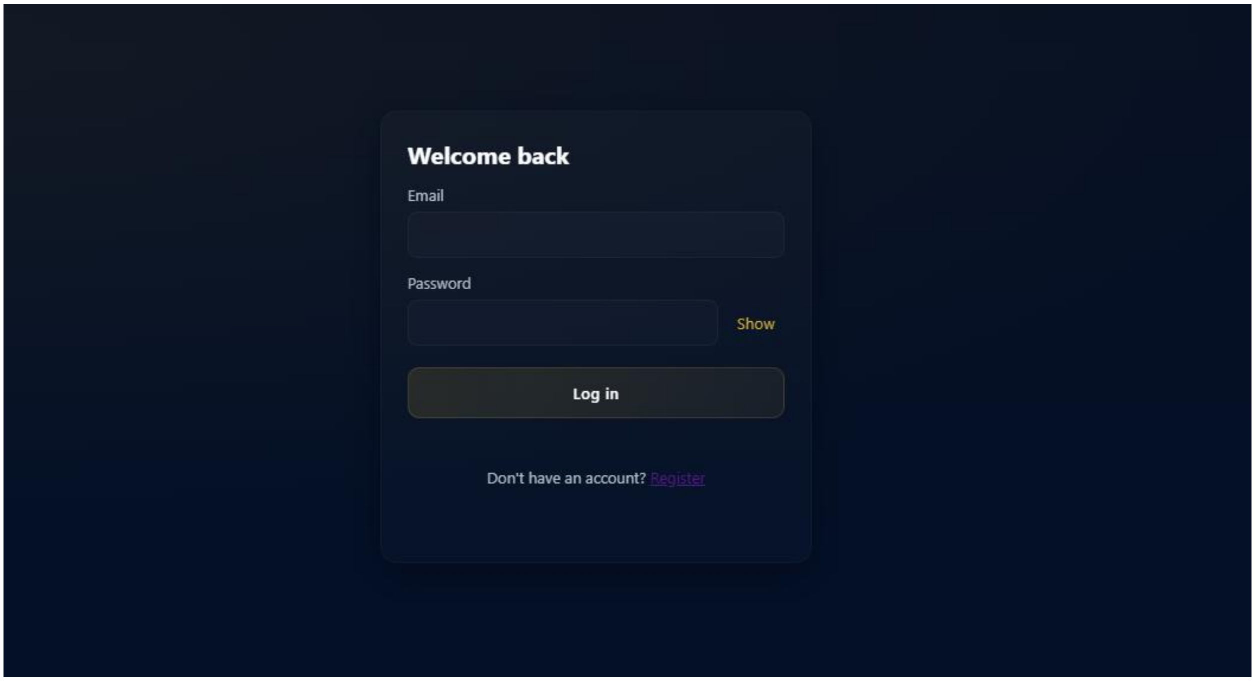
Username

Email

Password

Confirm Password

Already have an account? Log in

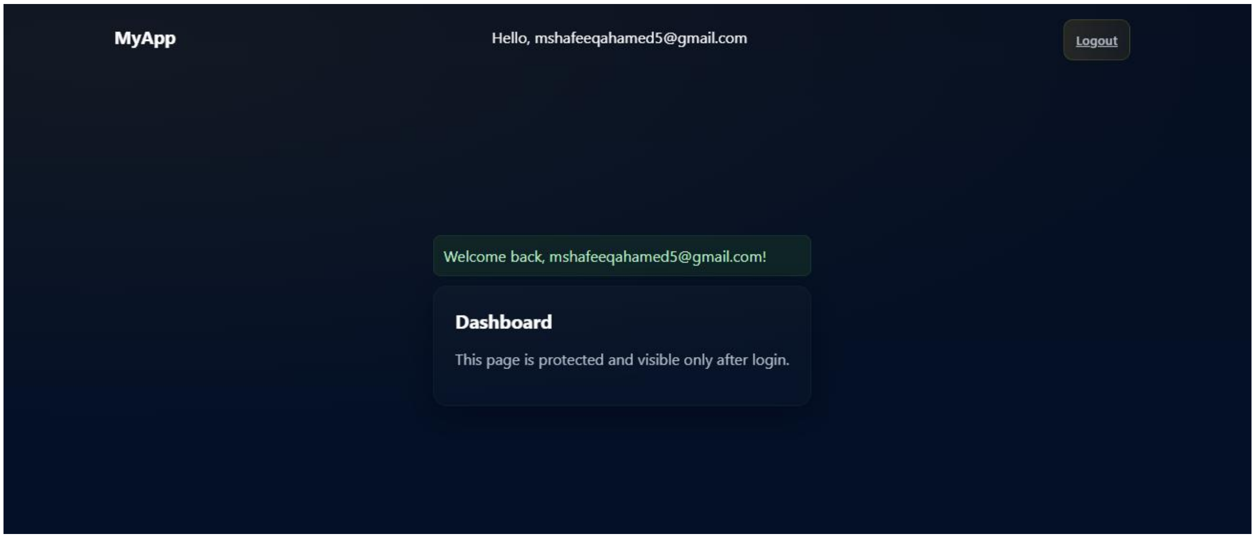


Welcome back

Email

Password

Don't have an account? Register



Hello, mshafeeqahamed5@gmail.com

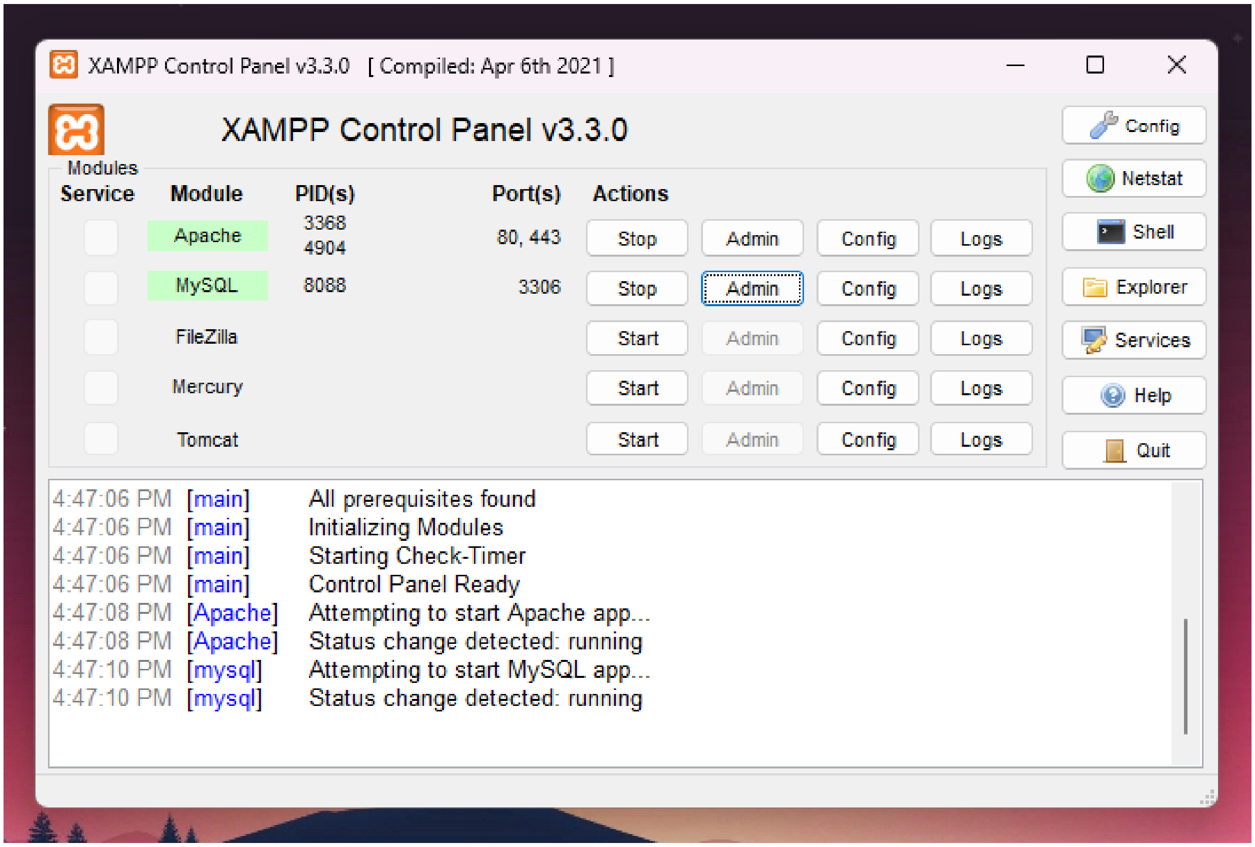
Welcome back, mshafeeqahamed5@gmail.com!

Dashboard

This page is protected and visible only after login.

XAMPP

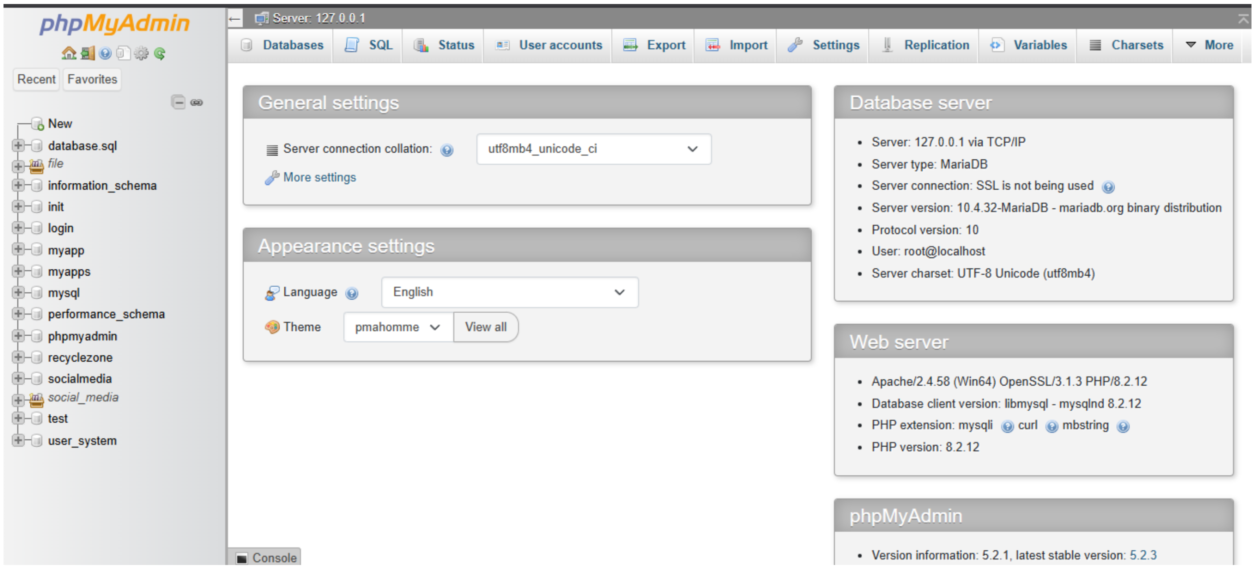
XAMPP is **a free and open-source cross-platform web server solution stack package developed by Apache Friends, designed primarily for local web development and testing**. It provides a complete environment for building and testing web applications on a personal computer without requiring an internet connection or external server access. The name XAMPP is a recursive acronym that stands for "XAMPP Apache + MariaDB + PHP + Perl," reflecting its core components: the Apache HTTP Server, MariaDB database, and interpreters for PHP and Perl scripting languages. Although it originally stood for Cross-Platform, Apache, MySQL, PHP, and Perl, the inclusion of MySQL was replaced with MariaDB starting with versions 5.5.30 and 5.6.14, released on October 19, 2015. XAMPP simplifies the setup of a WAMP (Windows, Apache, MySQL, PHP) or LAMP (Linux, Apache, MySQL, PHP) stack, allowing developers to quickly deploy a local server environment. It includes additional tools such as phpMyAdmin for database management, FileZilla FTP Server, and support for popular content management systems like WordPress and Joomla! through Bitnami. XAMPP is available for Windows, macOS, and Linux, making it a cross-platform solution. It is intended for development and testing purposes only, not for production environments, due to default security features being disabled.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	3368 4904	80, 443	Stop	Admin	Config	Logs
□	MySQL	8088	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
4:47:06 PM  [main]     All prerequisites found
4:47:06 PM  [main]     Initializing Modules
4:47:06 PM  [main]     Starting Check-Timer
4:47:06 PM  [main]     Control Panel Ready
4:47:08 PM  [Apache]   Attempting to start Apache app...
4:47:08 PM  [Apache]   Status change detected: running
4:47:10 PM  [mysql]    Attempting to start MySQL app...
4:47:10 PM  [mysql]    Status change detected: running
```

MYSQL



- New
- database.sql
- *file*
- information_schema
- init
- login
- myapp
- myapps
- mysql
- performance_schema
- phpmyadmin
- recyclezone
- socialmedia
- *social_media*
- test
- user_system

General settings

Server connection collation: utf8mb4_unicode_ci

More settings

Appearance settings

Language: English

Theme: pmahomme View all

Database server

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.32-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

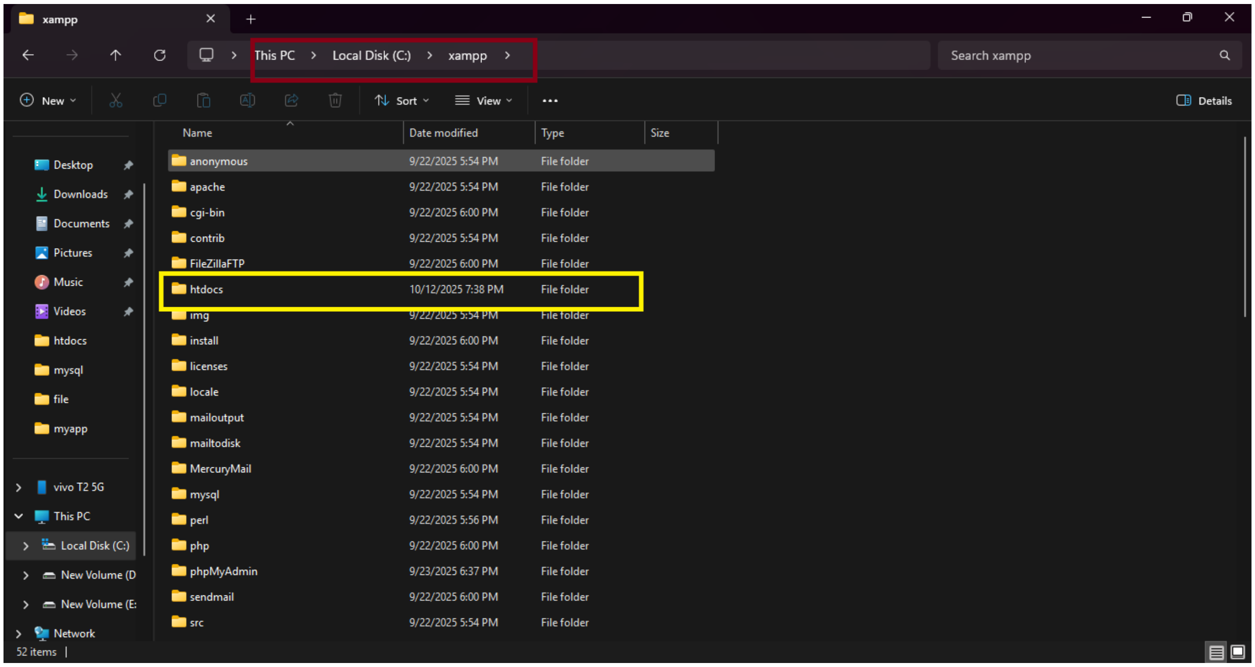
Web server

- Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
- Database client version: libmysql - mysqlnd 8.2.12
- PHP extension: mysqli curl mbstring
- PHP version: 8.2.12

phpMyAdmin

- Version information: 5.2.1, latest stable version: 5.2.3

File location



Name	Date modified	Type	Size
anonymous	9/22/2025 5:54 PM	File folder	
apache	9/22/2025 5:54 PM	File folder	
cgi-bin	9/22/2025 6:00 PM	File folder	
contrib	9/22/2025 5:54 PM	File folder	
FileZillaFTP	9/22/2025 6:00 PM	File folder	
htdocs	10/12/2025 7:38 PM	File folder	
img	9/22/2025 5:54 PM	File folder	
install	9/22/2025 6:00 PM	File folder	
licenses	9/22/2025 5:54 PM	File folder	
locale	9/22/2025 5:54 PM	File folder	
mailoutput	9/22/2025 5:54 PM	File folder	
mailtodisk	9/22/2025 5:54 PM	File folder	
MercuryMail	9/22/2025 6:00 PM	File folder	
mysql	9/22/2025 5:54 PM	File folder	
perl	9/22/2025 5:56 PM	File folder	
php	9/22/2025 6:00 PM	File folder	
phpMyAdmin	9/23/2025 6:37 PM	File folder	
sendmail	9/22/2025 6:00 PM	File folder	
src	9/22/2025 5:54 PM	File folder	