# Vulnerable Web App (SQL Injection) – Installation Guide

This guide will help you run a simple PHP-based vulnerable web application on Windows using XAMPP. The app demonstrates SQL Injection (SQLi) and how to fix it using prepared statements.

## Step 1: Install XAMPP (Apache + PHP + MySQL)

1. Go to [XAMPP Official Site](https://www.apachefriends.org/download.html).
2. Click Download for Windows.
3. Run the installer and follow the prompts:
   * Keep the default installation folder (usually C:\xampp).
   * Install Apache and MySQL components.
4. Finish installation.

## Step 2: Start Apache and MySQL

1. Open XAMPP Control Panel (search for XAMPP in Start Menu).
2. Click Start next to Apache and MySQL.
   * Both should turn green.
3. If Windows Firewall pops up, allow access.

## Step 3: Set up the Database

### Step 3.1: Open phpMyAdmin

* Start Apache and MySQL in XAMPP Control Panel.
* Open your browser → go to: <http://localhost/phpmyadmin>

### Step 3.2: Create the database

1. On the left sidebar, click New.
2. Name the database vuln\_demo (exactly).
3. Leave collation as utf8mb4\_general\_ci (or default) and click Create.

### Step 3.3: Import the SQL file

1. With vuln\_demo selected, click the Import tab.
2. Under File to import, browse your computer and choose your create\_db.sql file or create a new .sql file with the following content:

-- create\_db.sql  
CREATE DATABASE IF NOT EXISTS vuln\_demo;  
USE vuln\_demo;  
  
CREATE TABLE IF NOT EXISTS users (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 username VARCHAR(50) NOT NULL UNIQUE,  
 password VARCHAR(255) NOT NULL  
);  
  
-- Sample user (plaintext password for demo only)  
INSERT INTO users (username, password) VALUES ('admin', 'admin123');

1. Character set: leave as utf8 or choose utf8mb4.
2. Leave other options as default.
3. Format: SQL (default).
4. Click Go.

*If import succeeds, you’ll see a green success message, and the users table will appear under vuln\_demo.*

## Step 4: Place PHP Files in Web Folder

1. Go to your XAMPP installation folder, e.g., C:\xampp\htdocs.
2. Create a folder named vuln\_demo.
3. Create the following files in that folder:

### **db\_config.php**

<?php  
$DB\_HOST = '127.0.0.1';  
$DB\_NAME = 'vuln\_demo';  
$DB\_USER = 'root';  
$DB\_PASS = '';  
  
function get\_db\_conn() {  
 global $DB\_HOST, $DB\_USER, $DB\_PASS, $DB\_NAME;  
 $conn = new mysqli($DB\_HOST, $DB\_USER, $DB\_PASS, $DB\_NAME);  
 if ($conn->connect\_error) {  
 die('DB connect error: ' . $conn->connect\_error);  
 }  
 return $conn;  
}  
?>

### **vulnerable\_login.php**

<?php  
require\_once 'db\_config.php';  
$conn = get\_db\_conn();  
  
$message=''; $message\_type='';  
if ($\_SERVER['REQUEST\_METHOD']==='POST'){  
 $username=$\_POST['username']??''; $password=$\_POST['password']??'';  
 $sql="SELECT \* FROM users WHERE username='$username' AND password='$password'";  
 $res=$conn->query($sql);  
 if($res && $res->num\_rows===1){ $message="Login successful! Welcome $username"; $message\_type='success'; }  
 else{ $message="Login failed"; $message\_type='error'; }  
}  
?>  
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Vulnerable Login</title>  
<style>  
body{font-family:Arial,sans-serif;background:#eef2f7;margin:0;display:flex;justify-content:center;align-items:center;height:100vh;}  
.container{background:white;padding:30px 40px;border-radius:12px;box-shadow:0 4px 12px rgba(0,0,0,0.1);width:320px;text-align:center;}  
input,button{width:100%;padding:12px;margin:10px 0;border-radius:6px;border:1px solid #ccc;box-sizing:border-box;font-size:16px;}  
button{border:none;background:#3498db;color:white;font-weight:bold;cursor:pointer;}  
button:hover{background:#2980b9;}  
.message{padding:10px;margin-bottom:15px;border-radius:6px;font-weight:bold;display:none;}  
.message.success{background:#d4edda;color:#155724;display:block;}  
.message.error{background:#f8d7da;color:#721c24;display:block;}  
</style>  
</head>  
<body>  
<div class="container">  
<h2>Vulnerable Login</h2>  
<?php if($message): ?><div class="message <?= $message\_type ?>"><?= $message ?></div><?php endif; ?>  
<form method="post">  
<input name="username" type="text" placeholder="Username" required />  
<input name="password" type="text" placeholder="Password" required />  
<button type="submit">Login</button>  
</form>  
<p style="font-size:12px;margin-top:10px;">Try SQLi payload: <code>' OR '1'='1</code></p>  
</div>  
</body>  
</html>

### **fixed\_login.php**

<?php  
require\_once 'db\_config.php';  
$conn = get\_db\_conn();  
  
$message=''; $message\_type='';  
if ($\_SERVER['REQUEST\_METHOD']==='POST'){  
 $username=$\_POST['username']??''; $password=$\_POST['password']??'';  
 $stmt=$conn->prepare("SELECT id FROM users WHERE username=? AND password=? LIMIT 1");  
 $stmt->bind\_param("ss",$username,$password); $stmt->execute(); $stmt->store\_result();  
 if($stmt->num\_rows===1){ $message="Login successful! Welcome $username"; $message\_type='success'; }  
 else{ $message="Login failed"; $message\_type='error'; }  
 $stmt->close();  
}  
?>  
<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>Fixed Login</title>  
<style>  
body{font-family:Arial,sans-serif;background:#eef2f7;margin:0;display:flex;justify-content:center;align-items:center;height:100vh;}  
.container{background:white;padding:30px 40px;border-radius:12px;box-shadow:0 4px 12px rgba(0,0,0,0.1);width:320px;text-align:center;}  
input,button{width:100%;padding:12px;margin:10px 0;border-radius:6px;border:1px solid #ccc;box-sizing:border-box;font-size:16px;}  
button{border:none;background:#3498db;color:white;font-weight:bold;cursor:pointer;}  
button:hover{background:#2980b9;}  
.message{padding:10px;margin-bottom:15px;border-radius:6px;font-weight:bold;display:none;}  
.message.success{background:#d4edda;color:#155724;display:block;}  
.message.error{background:#f8d7da;color:#721c24;display:block;}  
</style>  
</head>  
<body>  
<div class="container">  
<h2>Fixed Login</h2>  
<?php if($message): ?><div class="message <?= $message\_type ?>"><?= $message ?></div><?php endif; ?>  
<form method="post">  
<input name="username" type="text" placeholder="Username" required />  
<input name="password" type="text" placeholder="Password" required />  
<button type="submit">Login</button>  
</form>  
<p style="font-size:12px;margin-top:10px;">Prepared statements prevent SQLi.</p>  
</div>  
</body>  
</html>

## Step 5: Test the Vulnerable Page

1. Open in browser: <http://localhost/vuln_demo/vulnerable_login.php>
2. Enter a test username and password.
3. Click Login → You should see “Login successful! Welcome [username]” if correct.

## Step 6: Test the Fixed Page

1. Open in browser: <http://localhost/vuln_demo/fixed_login.php>
2. Enter the same credentials.
3. Click Login → Login will only succeed with correct credentials.