

AUTHENTICATION WITH ROLE ACCESS for ECOMMERCE WEBSITE

PHP and MySQL

Laboratory Activity

Note: Some Codes utilized in this activity are excerpted from your classmate’s demo guide uploaded in the google classroom and GC. Might as well download it for your reference.

Step 1 – prepare your folder structure in your htdocs inside your e-commerce site folder.

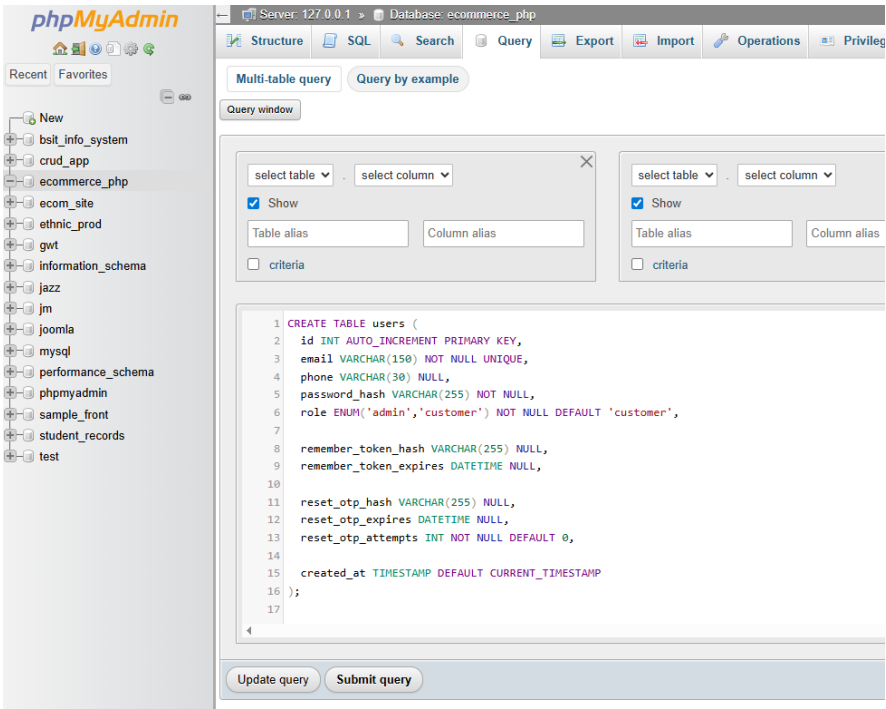
Htdocs> Yourecommerce_site > auth

Make sure you have the following PHP file inside auth folder.

- 1. db.php
- 2. login.php
- 3. logout.php
- 4. auth_check.php
- 5. forgot_password.php
- 6. verify_otp.php
- 7. reset_password.php
- 8. register.php

Step 2 – create a database and table

- 1. open phpMyAdmin
- 2. create a database – ecom_db
- 3. after creating your database, go to Query, copy and paste the given query below to create table users and its columns. Click submit Query.



QUERY

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  email VARCHAR(150) NOT NULL UNIQUE,  
  phone VARCHAR(30) NULL,  
  password_hash VARCHAR(255) NOT NULL,  
  role ENUM('admin','customer') NOT NULL DEFAULT 'customer',  
  remember_token_hash VARCHAR(255) NULL,  
  remember_token_expires DATETIME NULL,  
  reset_otp_hash VARCHAR(255) NULL,  
  reset_otp_expires DATETIME NULL,  
  reset_otp_attempts INT NOT NULL DEFAULT 0,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
remember_token_hash VARCHAR(255) NULL,  
remember_token_expires DATETIME NULL,  
  
reset_otp_hash VARCHAR(255) NULL,  
reset_otp_expires DATETIME NULL,  
reset_otp_attempts INT NOT NULL DEFAULT 0,  
  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Step 3 – Insert 1 admin user with hashed password.

- 1. Create a temporary file just to generate hash – one time use only.
- 2. In your ecommerce folder under auth folder create,

hash_password.php file.

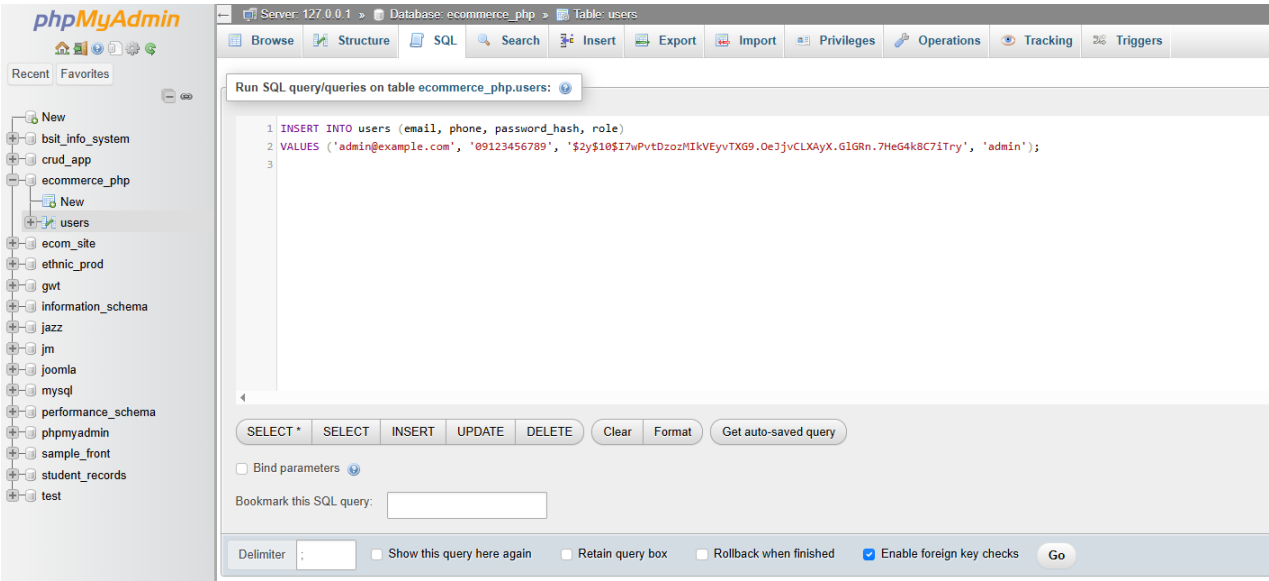
Then paste this syntax.

```
<?php  
echo password_hash("administrator2025", PASSWORD_DEFAULT);
```

- 3. Run in the browser. Copy the output hash.
- 4. Insert admin into DB. In phpMyAdmin SQL tab, paste the following. In the ‘**PASTE YOUR HASHED PASSWORD HERE**’ section, paste the copied hash in your browser recently.

```
INSERT INTO users (email, phone, password_hash, role)  
VALUES ('admin@example.com', '09123456789', 'PASTE YOUR HASHED PASSWORD HERE', 'admin');
```

- 5. Click Go.



Step 4 – Go to your db.php under your auth folder – we will now connect this to your database.
In your db.php, paste the following connection.

Note: In the \$dbName, change the database name to your database name (created recently example ecom_db)

```
<?php
```

```
$dbHost = "localhost";
$dbUser = "root";
$dbPass = "";
$dbName = "malot_db"; // your database name. Do not copy my database name if we do not have the same
database name.
```

```
$conn = new mysqli($dbHost, $dbUser, $dbPass, $dbName);
if ($conn->connect_error) {
    die("DB connection failed: " . $conn->connect_error);
}
$conn->set_charset("utf8mb4");
```

Step 5 – Code auth_check.php – this protects pages + auto login.

This file does `session_start()`, checks if logged in. If not logged in, it tries remember me cookie, it also provides role (`require_role (admin)` or `require_role (customer)`).

Copy and paste the code sample code below if you did not yet create your own.

```
<?php
session_start();
require_once __DIR__ . "/db.php";
```

```
function redirect_to_login() {
    header("Location: ../auth/login.php");
    exit;
}
```

```
if (!isset($_SESSION["role"]) && isset($_COOKIE["remember"])) {  
    $raw = $_COOKIE["remember"];
```

```
if (strpos($raw, ":") !== false) {
    [$suid, $token] = explode(":", $raw, 2);
    $suid = (int)$suid;
```

```
if ($uid > 0 && $token) {
    $stmt = $conn->prepare("SELECT id, email, role, remember_token_hash, remember_token_expires
        FROM users WHERE id=? LIMIT 1");
    $stmt->bind_param("i", $uid);
    $stmt->execute();
    $user = $stmt->get_result()->fetch_assoc();
}
```

```
if ($user && $user["remember_token_hash"] && $user["remember_token_expires"]) {
    $now = new DateTime("now");
    $exp = new DateTime($user["remember_token_expires"]);
```

```
$tokenHash = hash("sha256", $token);
```

```
if (hash_equals($user["remember_token_hash"], $tokenHash) && $exp > $now) {  
    $_SESSION["user_id"] = (int)$user["id"];  
    $_SESSION["email"] = $user["email"];
```

```

        $_SESSION["role"] = $user["role"];
    } else {
        setcookie("remember", "", time() - 3600, "/");
    }
}
}
}
}
}
}

```

```

function require_role($role) {
    if (!isset($_SESSION["role"]) || $_SESSION["role"] !== $role) {
        redirect_to_login();
    }
}

```

Step 6 – Create login.php inside auth folder. Since we already created the login.php file last meeting, we will work on the code that does reads email and password, fetches the user by email from db, password_verify(), sets session, if remember-me checked, sets cookie token, and auto redirect based on role. Copy and paste the code provided below if you did not yet code your login. If already have design for login, then just proceed with logic and session set up.

Sample login code

```

<?php
session_start();
require_once __DIR__ . "/db.php";

$error = "";

// If already logged in
if (isset($_SESSION["role"])) {
    if ($_SESSION["role"] === "admin") {
        header("Location: /malot_site/administrator/index.php");
    } else {
        header("Location: /malot_site/index.php");
    }
    exit;
}

function set_remember_me($conn, $userId) {
    // Create random token
    $token = bin2hex(random_bytes(32));
    $tokenHash = hash("sha256", $token);
    $expires = (new DateTime("+30 days"))->format("Y-m-d H:i:s");

    $stmt = $conn->prepare("UPDATE users SET remember_token_hash=?, remember_token_expires=? WHERE id=?");
    $stmt->bind_param("ssi", $tokenHash, $expires, $userId);
    $stmt->execute();
}

```

```

setcookie(
    "remember",
    $userId . ":" . $token,
    [
        "expires" => time() + (30 * 24 * 60 * 60),
        "path" => "/",
        "httponly" => true,
        "samesite" => "Lax",
        "secure" => false
    ]
);
}

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $email = strtolower(trim($_POST["email"] ?? ""));
    $password = $_POST["password"] ?? "";
    $remember = isset($_POST["remember"]);

    if (!$email || !$password) {
        $error = "Please fill in all fields.";
    } else {
        $stmt = $conn->prepare("SELECT id, email, password_hash, role FROM users WHERE email=? LIMIT 1");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $user = $stmt->get_result()->fetch_assoc();

        if ($user && password_verify($password, $user["password_hash"])) {

            $_SESSION["user_id"] = (int)$user["id"];
            $_SESSION["email"] = $user["email"];
            $_SESSION["role"] = $user["role"];

            if ($remember) {
                set_remember_me($conn, (int)$user["id"]);
            }

            if ($user["role"] === "admin") {
                header("Location: /malot_site/administrator/index.php");
            } else {
                header("Location: /malot_site/index.php");
            }
            exit;

        } else {
            $error = "Invalid email or password.";
        }
    }
}
?>

```

<!DOCTYPE html>

```
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Sign In</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>

<body class="bg-light">

<div class="container">
  <div class="row justify-content-center align-items-center vh-100">
    <div class="col-md-4">

      <div class="card shadow p-4">

        <h4 class="text-center mb-2">Sign In</h4>
        <p class="text-center text-muted small mb-3">
          Enter your credentials to access your account
        </p>

        <?php if($error): ?>
          <div class="alert alert-danger py-2">
            <?php echo htmlspecialchars($error); ?>
          </div>
        <?php endif; ?>

        <form method="POST" novalidate>

          <div class="mb-3">
            <input class="form-control" name="email" type="email" placeholder="Email Address" required>
          </div>

          <div class="mb-3">
            <input class="form-control" name="password" type="password" placeholder="Password" required>
          </div>

          <div class="d-flex align-items-center justify-content-between mb-3">

            <div class="form-check">
              <input class="form-check-input" type="checkbox" id="remember" name="remember">
              <label class="form-check-label small" for="remember">
                Remember me
              </label>
            </div>

            <a class="small" href="/malot_site/auth/forgot_password.php">
              Forgot password?
            </a>

          </div>

        </form>

      </div>

    </div>
  </div>
</div>
```

```

<button class="btn btn-primary w-100" type="submit">
  Sign In
</button>

<div class="text-center mt-3 small">
  Don't have an account?
  <a href="/malot_site/auth/register.php">Register Here</a>
</div>

</form>

</div>

</div>
</div>
</div>

</body>
</html>

```

*****THIS TIME, YOU MAY ACCESS YOUR LOGIN.PHP IN YOUR BROWSER. To check if it is accessible.
SAMPLE LINK: http://localhost/malot_site/auth/login.php
Note: change malot_site with your ecommerce name folder.

Step 7 – create auth/logout.php (logout with clear cookie)
This step let us create a logout php file that clears session, clears remember cookie, optionally clears token hash in DB and redirects to login. However, this is still up to you or requirement. But for this activity, we will apply this one.

Sample Code:

```

<?php
session_start();
require_once __DIR__ . "/db.php";

if (isset($_SESSION["user_id"])) {
    $uid = (int)$_SESSION["user_id"];
    $stmt = $conn->prepare("UPDATE users SET remember_token_hash=NULL, remember_token_expires=NULL
WHERE id=?");
    $stmt->bind_param("i", $uid);
    $stmt->execute();
}

setcookie("remember", "", time() - 3600, "/");

session_unset();
session_destroy();

```

```
header("Location: /malot_site/auth/login.php");
exit;
```

Step 8 – protect your admin pages

This is applicable to all pages outside auth. Instead of using session in every page, we will use the auth_check.php. why this happen? since files under auth folder applied or has session already so there are authentication and validation in place. We do not need to use session. Instead, put the below code to each page which serves as a session.

Example: in your add_category.php
Paste this before your html boilerplate.

Note:This is for the administrator php pages only.

```
<?php
require_once __DIR__ . '/../../auth/auth_check.php';
require_role("admin");
?>
```

Or

Note:This is for the customer php pages only.

```
<?php
require_once __DIR__ . "/auth/auth_check.php";
require_role("customer");
?>
```

Step 9 – create forgot password via OTP files

now that login/logout/protection works, build reset flow.

1. This forgot_password.php file will ask for phone number, finds user by phone, generates (OTP – 6 DIGITS), saves hash + expiry in DB, sends OTP, redirects to verify page.
2. Verify_otp.php file checks OTP input, compares to hash in DB, checks expiry and attempts, if correct, allow reset password.
3. Reset_password.php file asks for new password then confirmation, hashes it using password_hash () – applying your classmates' demo, updates DB, clears reset fields and remember token, then returns user to login.

Step 10 – test

****Checking of output****

Prepare 1/8 sheet write your name, the date today, and auth activity.