**Xử lí đăng ký**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["submit"])) {

    $tendn = $\_POST["username"];

    $matkhau = $\_POST["password"];

    $email = $\_POST["email"];

    $number\_phone  = $\_POST["number\_phone"];

    $address = $\_POST["address"];

*// Mã hóa mật khẩu trước khi lưu*

    $hashed\_password = password\_hash($matkhau, PASSWORD\_DEFAULT);

*// Kiểm tra xem các trường có rỗng không*

*// if (empty($tendn) || empty($hashed\_password) || empty($email) || empty($number\_phone) || empty($address)) {*

*//     echo "Vui lòng điền đầy đủ thông tin.";*

*//     header("Location: ../pages/dangky.php");*

*//     exit();*

*// }*

*// Kiểm tra username đã tồn tại chưa*

    $check\_sql = "SELECT \* FROM user WHERE username = '$tendn'";

    $check\_res = $newShop->runQuery($check\_sql);

    if ($check\_res->num\_rows > 0) {

*// Username đã tồn tại*

        echo "Tên đăng nhập đã tồn tại, vui lòng chọn tên khác.";

        header("Location: ../pages/dangky.php");

*// Bạn có thể redirect hoặc xử lý hiển thị thông báo khác*

    } else {

*// Chèn dữ liệu mới vào database*

        $insert\_sql = "INSERT INTO user (username, password, email, number\_phone, address) VALUES ('$tendn', '$hashed\_password', '$email', '$number\_phone', '$address')";

        $insert\_res = $newShop->runQuery($insert\_sql);

        if ($insert\_res) {

            $\_SESSION["username"] = $tendn;

            header("Location: ../pages/dangnhap.php");

            exit();

        } else {

            echo "Đăng ký thất bại, vui lòng thử lại.";

            header("Location: ../pages/dangky.php");

        }

    }

}

*// Dùng prepare*

*/\**

*<?php*

*session\_start();*

*require\_once "../conf/shopDB.php";*

*$newShop = new shopDB("localhost", "root", "", "shopDB");*

*$newShop->connectDB();*

*if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["submit"])) {*

*$tendn = $\_POST["username"];*

*$matkhau = $\_POST["password"];*

*$email  = $\_POST["email"];*

*// Kiểm tra username đã tồn tại chưa — dùng prepared statement*

*$check\_sql = "SELECT \* FROM user WHERE username = ?";*

*$stmt = $newShop->prepareAndExecute($check\_sql, "s", [$tendn]);*

*$stmt->store\_result(); // để kiểm tra num\_rows*

*if ($stmt->num\_rows > 0) {*

*// Username đã tồn tại*

*$stmt->close();*

*echo "Tên đăng nhập đã tồn tại, vui lòng chọn tên khác.";*

*header("Location: ../pages/dangky.php");*

*exit();*

*} else {*

*$stmt->close();*

*// Mã hóa mật khẩu trước khi lưu*

*$hashed\_password = password\_hash($matkhau, PASSWORD\_DEFAULT);*

*// Chèn dữ liệu mới — dùng prepared statement*

*$insert\_sql = "INSERT INTO user (username, password, email) VALUES (?, ?, ?)";*

*$insert\_stmt = $newShop->prepareAndExecute($insert\_sql, "sss", [$tendn, $hashed\_password, $email]);*

*if ($insert\_stmt) {*

*$\_SESSION["username"] = $tendn;*

*$insert\_stmt->close();*

*header("Location: ../pages/dangnhap.php");*

*exit();*

*} else {*

*echo "Đăng ký thất bại, vui lòng thử lại.";*

*header("Location: ../pages/dangky.php");*

*exit();*

*}*

*}*

*}*

*\*/*

**Xử lí đăng nhập**

<?php

ini\_set('session.gc\_maxlifetime', 10800);

*// Set cookie session tồn tại trong 3 tiếng*

session\_set\_cookie\_params(10800);

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

if (isset($\_POST["submitdn"]) && $\_SERVER["REQUEST\_METHOD"] === "POST") {

    $tendn = $\_POST["txttendn"];

    $matkhau = $\_POST["pmk"];

    $sql = "select \* from user where username = '$tendn'";

    $result = $newShop->runQuery($sql);

    $row = $result->fetch\_assoc();

    if ($row) {

        if (password\_verify($matkhau, $row["password"])) {

            $\_SESSION["username"] = $row["username"];

            header("Location: ../index.php");

        } else {

            echo "

            <script>

            alert ('Sai mật khẩu');

            window.location.href = '../pages/dangnhap.php';

            </script>

            ";

        }

    }

}

*// Dùng prepare*

*/\**

*<?php*

*session\_start();*

*require\_once "../conf/shopDB.php";*

*$newShop = new shopDB("localhost", "root", "", "shopDB");*

*$newShop->connectDB();*

*if (isset($\_POST["submitdn"]) && $\_SERVER["REQUEST\_METHOD"] === "POST") {*

*$tendn   = $\_POST["txttendn"];*

*$matkhau = $\_POST["pmk"];*

*// Dùng prepared statement để lấy thông tin user*

*$sql = "SELECT \* FROM user WHERE username = ?";*

*$stmt = $newShop->prepareAndExecute($sql, "s", [$tendn]);*

*$result = $stmt->get\_result();*

*$row = $result->fetch\_assoc();*

*if ($row) {*

*// So sánh mật khẩu nhập với mật khẩu đã hash trong database*

*if (password\_verify($matkhau, $row["password"])) {*

*$\_SESSION["username"] = $row["username"];*

*$stmt->close();*

*header("Location: ../index.php");*

*exit();*

*} else {*

*echo "Sai mật khẩu!";*

*}*

*} else {*

*echo "Tài khoản không tồn tại!";*

*}*

*$stmt->close();*

*}*

*?>*

*\*/*

**Xử lí đăng xuất**

<?php

session\_start(); *// Bắt buộc có trước khi unset/destroy session*

session\_unset();    *// Xóa hết các biến session*

session\_destroy();  *// Hủy session*

header("Location: ../pages/dangnhap.php");

exit();

**Xử lí sửa sp**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

*// Nhận dữ liệu từ form*

$id = $\_POST["product\_id"];

$ten = $\_POST["product\_name"];

$gia = $\_POST["price"];

*// Thư mục ảnh*

$target\_dir = "../img/";

*// Lấy tên ảnh cũ trong database*

$sqlSelect = "SELECT imageURL FROM product WHERE product\_id = $id";

$res = $newShop->runQuery($sqlSelect);

$row = $res->fetch\_assoc();

$anh\_cu = $row["imageURL"];

*// Nếu người dùng chọn ảnh mới*

if (!empty($\_FILES["image"]["name"])) {

    $hinh = basename($\_FILES["image"]["name"]);

    $target\_file = $target\_dir . $hinh;

*// Xoá ảnh cũ nếu tồn tại*

    if (file\_exists($target\_dir . $anh\_cu) && !empty($anh\_cu)) {

        unlink($target\_dir . $anh\_cu);

    }

*// Upload ảnh mới*

    if (move\_uploaded\_file($\_FILES["image"]["tmp\_name"], $target\_file)) {

*// Update dữ liệu sản phẩm với ảnh mới*

        $sqlUpdate = "UPDATE product SET product\_name = '$ten', price = '$gia', imageURL = '$hinh' WHERE product\_id = $id";

    } else {

        echo "Lỗi upload ảnh!";

        exit();

    }

} else {

*// Nếu không chọn ảnh mới, chỉ update tên và giá*

    $sqlUpdate = "UPDATE product SET product\_name = '$ten', price = '$gia' WHERE product\_id = $id";

}

*// Chạy query update*

$newShop->runQuery($sqlUpdate);

*// Chuyển về trang chỉnh sửa*

header("Location: ../pages/edit\_product.php?id=" . $id);

exit();

**Xử lí thanh toán**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

if (!isset($\_SESSION['username'])) {

    die("Bạn chưa đăng nhập.");

}

*// Kiểm tra submit form*

if (isset($\_POST['submit\_payment'])) {

    $username = $\_SESSION['username'];

    echo $username;

*// Lấy user\_id từ bảng user theo username*

    $sqlUser = "SELECT user\_id FROM user WHERE username = ?";

    $stmtUser = $newShop->prepareAndExecute($sqlUser, "s", [$username]);

    $userData = $stmtUser->get\_result()->fetch\_assoc();

    if (!$userData) {

        die("Không tìm thấy thông tin người dùng.");

    }

    $user\_id = $userData['user\_id'];

*// Lấy thông tin địa chỉ giao hàng*

    if (isset($\_POST['radio']) && $\_POST['radio'] === 'khac') {

*// Địa chỉ mới*

        $customer\_name = $\_POST['hovaten\_moi'] ?? '';

        $customer\_numberphone = $\_POST['sodienthoai\_moi'] ?? '';

        $customer\_address = $\_POST['diachi\_moi'] ?? '';

    } else {

*// Địa chỉ mặc định lấy từ bảng user*

        $sqlUserInfo = "SELECT username, number\_phone, address FROM user WHERE username = ?";

        $stmtUserInfo = $newShop->prepareAndExecute($sqlUserInfo, "s", [$username]);

        $userInfo = $stmtUserInfo->get\_result()->fetch\_assoc();

        $customer\_name = $userInfo['username'];

        $customer\_numberphone = $userInfo['number\_phone'];

        $customer\_address = $userInfo['address'];

    }

*// Kiểm tra giỏ hàng*

    if (!isset($\_SESSION['cart']) || empty($\_SESSION['cart'])) {

        die("Giỏ hàng của bạn đang trống.");

    }

*// Tính tổng tiền*

    $total\_amount = 0;

    foreach ($\_SESSION['cart'] as $item) {

        $total\_amount += $item['price'] \* $item['quantity'];

    }

*// Phương thức thanh toán*

    $payment\_method = $\_POST['payment\_method'] ?? 'cod';

*// Thêm vào bảng orders*

    $order\_date = date("Y-m-d H:i:s");

    $status = "Đang chờ";

    $sqlInsertOrder = "INSERT INTO `order` (user\_id, order\_date, total\_amount, status, Customer\_name, Customer\_numberphone, Customer\_address, payment\_method) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

    $stmtInsertOrder = $newShop->prepareAndExecute($sqlInsertOrder, "isdsssss", [

        $user\_id,

        $order\_date,

        $total\_amount,

        $status,

        $customer\_name,

        $customer\_numberphone,

        $customer\_address,

        $payment\_method

    ]);

    if ($stmtInsertOrder) {

        $order\_id = $newShop->getConnection()->insert\_id; *// Lấy order\_id vừa insert*

*// Thêm chi tiết đơn hàng*

        $sqlInsertDetail = "INSERT INTO order\_detail (order\_id, product\_id, quantity, price) VALUES (?, ?, ?, ?)";

        $stmtDetail = $newShop->getConnection()->prepare($sqlInsertDetail);

        foreach ($\_SESSION['cart'] as $item) {

            $product\_id = $item['product\_id'];

            $quantity = $item['quantity'];

            $price = $item['price'];

            $stmtDetail->bind\_param("iiid", $order\_id, $product\_id, $quantity, $price);

            $stmtDetail->execute();

        }

        $stmtDetail->close();

*// Xóa giỏ hàng*

        unset($\_SESSION['cart']);

        echo "<p>Thanh toán thành công! Đơn hàng của bạn đã được lưu.</p>";

        echo '<a href="../index.php">Quay lại trang chủ</a>';

    } else {

        echo "<p>Đã có lỗi xảy ra khi xử lý đơn hàng. Vui lòng thử lại.</p>";

    }

} else {

    echo "<p>Phương thức không hợp lệ.</p>";

}

**Xử lí thêm vào giỏ hàng**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

if ($\_SERVER["REQUEST\_METHOD"] === "POST" && isset($\_POST['submit'])) {

    $product\_id = (int)$\_POST['product\_id'];

    $quantity = (int)$\_POST['quantity'];

*// Lấy thông tin sản phẩm từ database để đảm bảo giá và tên chuẩn*

    $sql = "SELECT \* FROM product WHERE product\_id = ?";

    $params = [$product\_id];

    $stmt = $newShop->prepareAndExecute($sql, "i", $params);

    $product = $stmt->get\_result()->fetch\_assoc();

    if (!$product) {

        echo "Sản phẩm không tồn tại.";

        exit;

    }

*// Tạo mảng giỏ hàng nếu chưa có*

    if (!isset($\_SESSION['cart'])) {

        $\_SESSION['cart'] = [];

    }

*// Kiểm tra sản phẩm đã có trong giỏ chưa*

    $found = false;

    foreach ($\_SESSION['cart'] as &$item) {

        if ($item['product\_id'] == $product\_id) {

            $item['quantity'] += $quantity;

            $found = true;

            break;

        }

    }

    unset($item); *// Xoá tham chiếu*

*// Nếu chưa có thì thêm mới*

    if (!$found) {

        $\_SESSION['cart'][] = [

            'product\_id' => $product\_id,

            'product\_name' => $product['product\_name'],

            'price' => $product['price'],

            'imageURL' => $product['imageURL'],

            'quantity' => $quantity

        ];

    }

*// Sau khi thêm thành công:*

    $\_SESSION['success\_message'] = "Đã thêm sản phẩm <strong>" . htmlspecialchars($product['product\_name']) . "</strong> vào giỏ hàng!";

*// Chuyển lại trang chi tiết sản phẩm*

    header("Location: ../pages/chitietsanpham.php?product\_id=" . $product\_id);

    exit;

}

**Xử lí thêm sản phẩm mới**

<?php

session\_start();

require\_once "../conf/shopDB.php";

*// Kết nối database*

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

*// Nhận dữ liệu từ form*

$tensp = $\_POST["product\_name"];

$gia = $\_POST["price"];

$loai = $\_POST["category\_id"];

$nsx  = $\_POST["manufacturer\_id"];

*// Xử lý ảnh upload*

$ext = strtolower(pathinfo($\_FILES["anh"]["name"], PATHINFO\_EXTENSION));

*// Hàm tạo chuỗi random*

function generateRandomString($length = 10)

{

    return substr(str\_shuffle("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"), 0, $length);

}

*// Tạo tên ảnh mới random hoàn toàn*

$newImageName = "product\_" . generateRandomString(10) . "." . $ext;

*// Thư mục lưu ảnh*

$target\_dir  = "../img/";

$target\_file = $target\_dir . $newImageName;

*// Di chuyển ảnh vào thư mục*

if (move\_uploaded\_file($\_FILES["anh"]["tmp\_name"], $target\_file)) {

*// Thêm sản phẩm vào database*

    $sql = "INSERT INTO product (product\_name, price, category\_id, manufacturer\_id, imageURL) VALUES (?, ?, ?, ?, ?)";

    $params = [$tensp, $gia, $loai, $nsx, $newImageName];

    $newShop->prepareAndExecute($sql, "sdiis", $params);

*// Chuyển hướng về trang chủ*

    header("Location: ../index.php");

    exit();

} else {

    header("Location: ../index.php");

    exit();

}

**Xử lí tìm kiếm**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

if (isset($\_POST['txttimkiem'])) {

    $limit = 5;

    $keyword = "%" . $\_POST['txttimkiem'] . "%";

    $sql = "SELECT \* FROM product WHERE product\_name LIKE ? ORDER BY product\_id DESC";

    $params = [$keyword];

    $stmt = $newShop->prepareAndExecute($sql, "s", $params);

    $res = $stmt->get\_result();

    $resultHTML = [];

    if ($res->num\_rows > 0) {

        while ($product = $res->fetch\_assoc()) {

            $resultHTML[] = $product;

        }

    }

*// Tính tổng số trang*

    $totalProduct = count($resultHTML);

    $totalPage = ceil($totalProduct / $limit);

*// Lưu vào session*

    $\_SESSION['ketquatimkiem'] = $resultHTML;

    $\_SESSION['timkiem\_totalPage'] = $totalPage;

    $\_SESSION['txttimkiem'] = $\_POST['txttimkiem'];

}

*// Quay lại index*

header("Location: ../index.php");

exit();

**Xử lí xóa sản phẩm**

<?php

session\_start();

require\_once "../conf/shopDB.php";

$newShop = new shopDB("localhost", "root", "", "shopDB");

$newShop->connectDB();

*// Nhận dữ liệu từ form*

$idsp = $\_POST["product\_id"];

$sql  = "delete from product where product\_id = '$idsp'";

$newShop->runQuery($sql);

header("Location: ../index.php");

**Class shopDB**

<?php

class shopDB

{

    public $svname;

    public $usname;

    public $pass;

    public $dbname;

    public $conn;

*// Hàm khởi tjao nhận giá trị để kết cơ sở dữ liệu*

    function \_\_construct($svname, $usname, $pass, $dbname)

    {

*$this*->svname = $svname;

*$this*->usname = $usname;

*$this*->pass = $pass;

*$this*->dbname = $dbname;

    }

*// Destruct tự động đóng khi đối tượng bị hủy*

    function \_\_destruct()

    {

        if (*$this*->conn) {

*$this*->conn->close();

        }

    }

*// Check status để đăng nhập*

    function isLogined()

    {

        if (!$\_SESSION["username"]) {

            echo "<script>

            confirm ('Xin vui lòng đăng nhập');

            window.location.href = '../pages/dangnhap.php';

            </script>";

        }

    }

*// Hàm kết nối với database*

    function connectDB()

    {

*$this*->conn = new mysqli(*$this*->svname, *$this*->usname, *$this*->pass, *$this*->dbname);

        if (!*$this*->conn) {

            die("Lỗi kết nối: " . mysqli\_connect\_error());

        } else {

            echo "Kết nối thành công " . "<br>";

        }

    }

*// Hàm đóng kết nối*

    function closeDB()

    {

*$this*->conn->close();

    }

*// Hàm trả về kết quả truy vấn*

    function runQuery($query)

    {

        $result = *$this*->conn->query($query);

        return $result;

    }

*// Hàm truy vấn nhiều nhiều cái*

    function runnMultipleQuery($query)

    {

        $res = *$this*->conn->multi\_query($query);

        return $res;

    }

    public function getConnection()

    {

        return *$this*->conn;

    }

*// Hàm prepareAndExecute dùng để truy vấn tránh sql injection*

    function prepareAndExecute($query, $type, $params)

    {

        $stmt = *$this*->conn->prepare($query);

        if (!$stmt) {

            die("Lỗi truy vấn: " . *$this*->conn->error);

        }

        $stmt->bind\_param($type, ...$params);

        $stmt->execute();

        return $stmt;

    }

*// Hàm lấy tất cả dữ liệu của query dưới dạng mảng*

    function fetchAll($query)

    {

        $result = *$this*->conn->query($query);

        $data = [];

        if (!$result) {

            die("Lỗi truy vấn: " . *$this*->conn->error);

        } else {

            while ($row = $result->fetch\_assoc()) {

                $data[] = $row;

            }

        }

        return $data;

    }

*// Hàm lấy số bản ghi bị ảnh hưởng sau thao tác INSERT/UPDATE/DELETE*

    function affectedRows()

    {

        return *$this*->conn->affected\_rows;

    }

}

*// Thêm nhiều dữ liệu (dùng hàm runnMultipleQuery)*

*// $sql = "INSERT INTO MyGuests (firstname, lastname, email)*

*// VALUES ('John', 'Doe', 'john@example.com');";*

*// $sql .= "INSERT INTO MyGuests (firstname, lastname, email)*

*// VALUES ('Mary', 'Moe', 'mary@example.com');";*

*// $sql .= "INSERT INTO MyGuests (firstname, lastname, email)*

*// VALUES ('Julie', 'Dooley', 'julie@example.com')";*

Class shopDB nhưng dùng PDO

<?php

class shopDB

{

    public $svname;

    public $usname;

    public $pass;

    public $dbname;

    public $conn;

*// Hàm khởi tạo nhận giá trị để kết cơ sở dữ liệu*

    function \_\_construct($svname, $usname, $pass, $dbname)

    {

*$this*->svname = $svname;

*$this*->usname = $usname;

*$this*->pass = $pass;

*$this*->dbname = $dbname;

    }

*// Destruct tự động đóng khi đối tượng bị hủy*

    function \_\_destruct()

    {

*$this*->conn = null;

    }

*// Check status để đăng nhập*

    function isLogined()

    {

        if (!isset($\_SESSION["username"])) {

            echo "<script>

            confirm('Xin vui lòng đăng nhập');

            window.location.href = '../pages/dangnhap.php';

            </script>";

        }

    }

*// Hàm kết nối với database*

    function connectDB()

    {

        try {

            $dsn = "mysql:host={*$this*->svname};dbname={*$this*->dbname};charset=utf8";

*$this*->conn = new PDO($dsn, *$this*->usname, *$this*->pass);

*// Thiết lập chế độ báo lỗi*

*$this*->conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

            echo "Kết nối thành công <br>";

        } catch (PDOException $e) {

            die("Lỗi kết nối: " . $e->getMessage());

        }

    }

*// Hàm đóng kết nối*

    function closeDB()

    {

*$this*->conn = null;

    }

*// Hàm trả về kết quả truy vấn*

    function runQuery($query)

    {

        try {

            return *$this*->conn->query($query);

        } catch (PDOException $e) {

            die("Lỗi truy vấn: " . $e->getMessage());

        }

    }

*// Hàm thực thi nhiều câu query (tách từng câu ra rồi chạy lần lượt)*

    function runnMultipleQuery($query)

    {

        try {

            $queries = explode(";", $query);

            foreach ($queries as $q) {

                if (trim($q) != '') {

*$this*->conn->exec($q);

                }

            }

            return true;

        } catch (PDOException $e) {

            die("Lỗi thực thi truy vấn: " . $e->getMessage());

        }

    }

    public function getConnection()

    {

        return *$this*->conn;

    }

*// Hàm prepareAndExecute dùng để tránh sql injection*

    function prepareAndExecute($query, $params)

    {

        try {

            $stmt = *$this*->conn->prepare($query);

            $stmt->execute($params);

            return $stmt;

        } catch (PDOException $e) {

            die("Lỗi truy vấn: " . $e->getMessage());

        }

    }

*// Hàm lấy tất cả dữ liệu của query dưới dạng mảng*

    function fetchAll($query)

    {

        try {

            $stmt = *$this*->conn->query($query);

            return $stmt->fetchAll(PDO::FETCH\_ASSOC);

        } catch (PDOException $e) {

            die("Lỗi truy vấn: " . $e->getMessage());

        }

    }

*// Hàm lấy số bản ghi bị ảnh hưởng sau thao tác INSERT/UPDATE/DELETE*

    function affectedRows($stmt)

    {

        return $stmt->rowCount();

    }

}