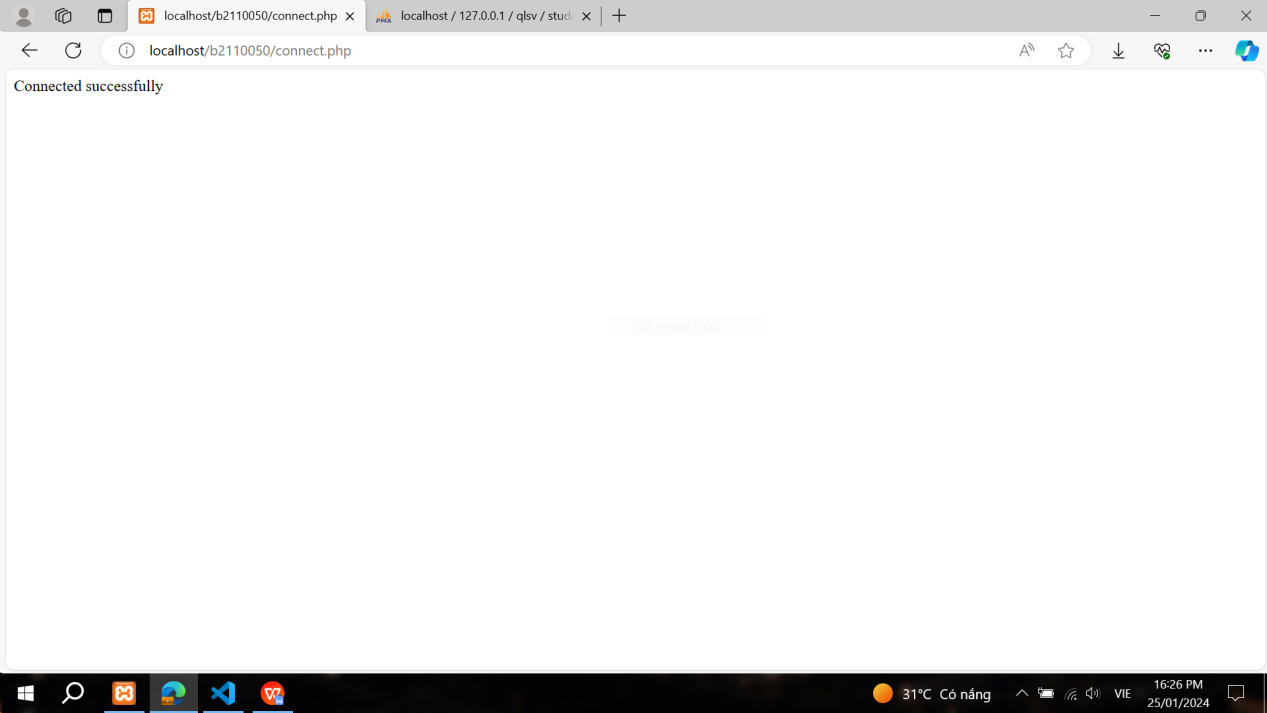
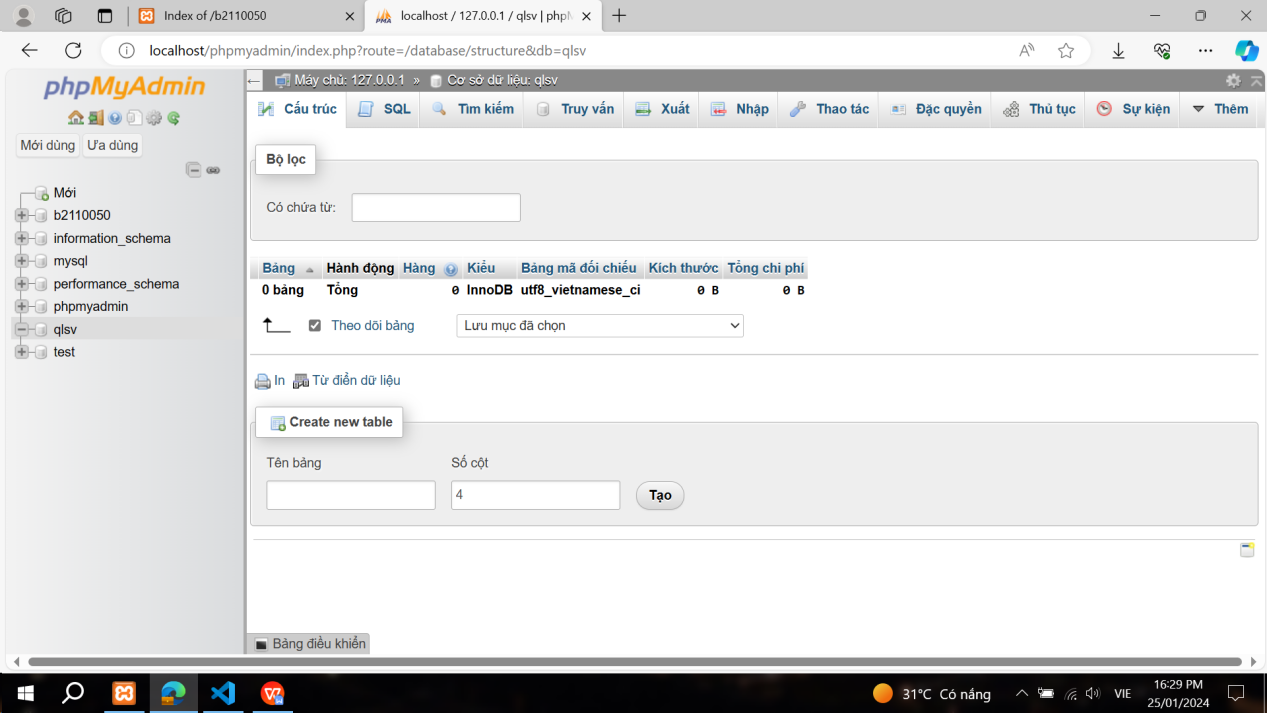
TH03: LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI PHP

Câu 1:

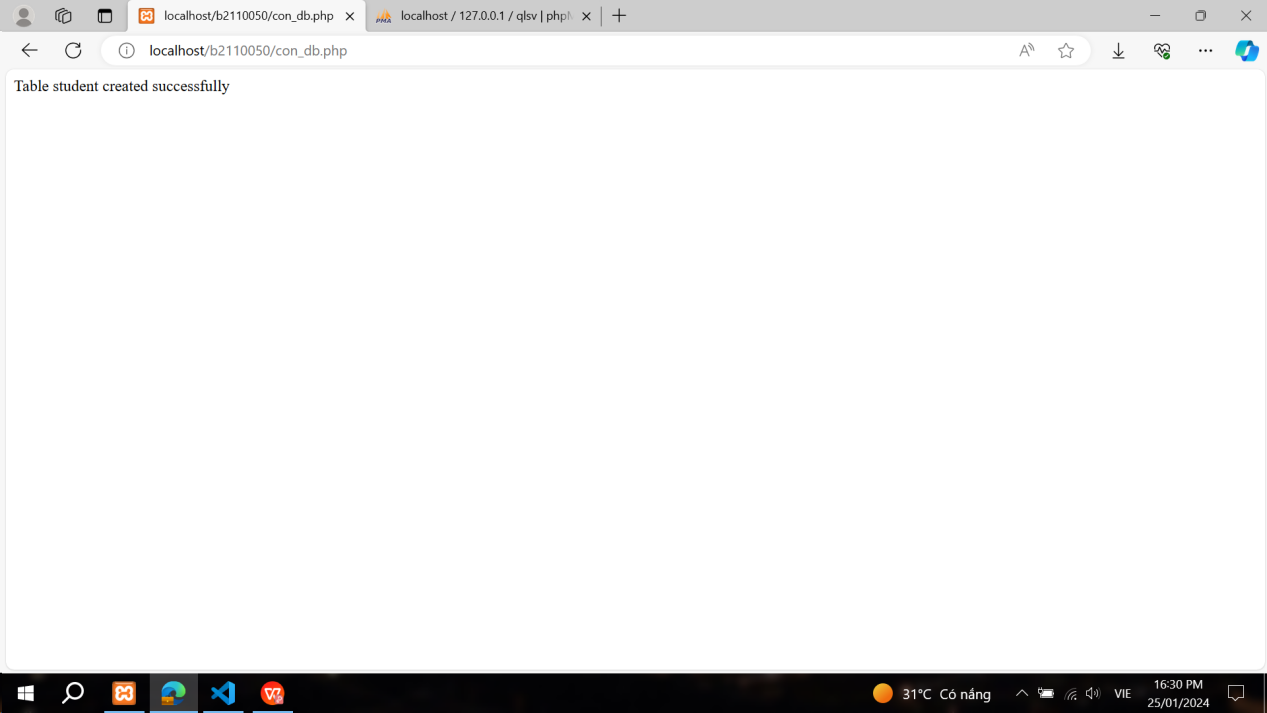
//Tạo tập tin connect.php

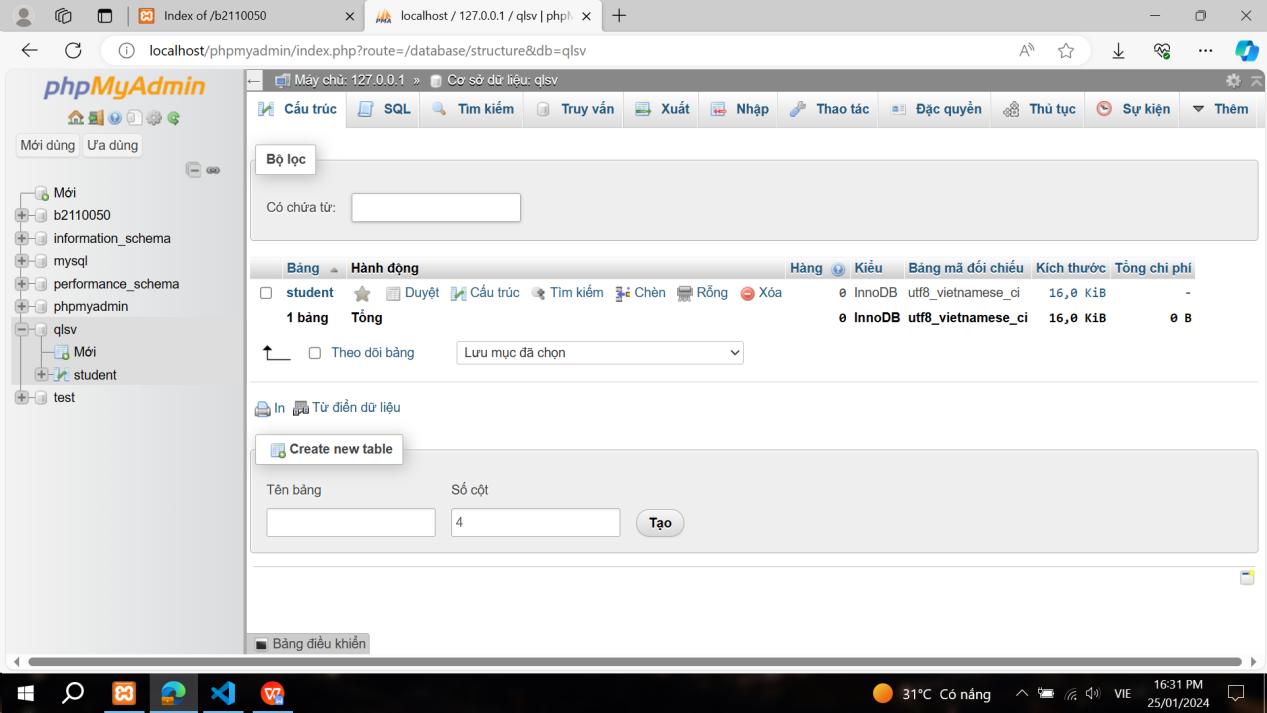


// tạo 1 csdl tên là qlsv

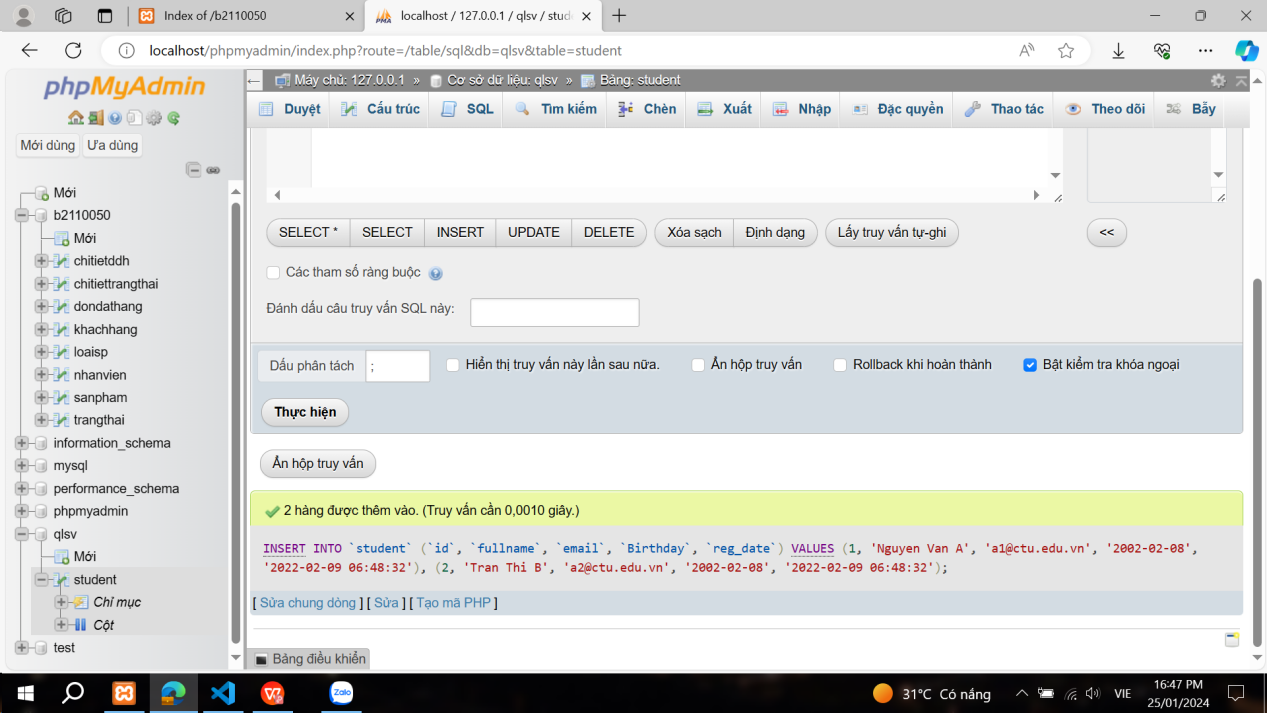


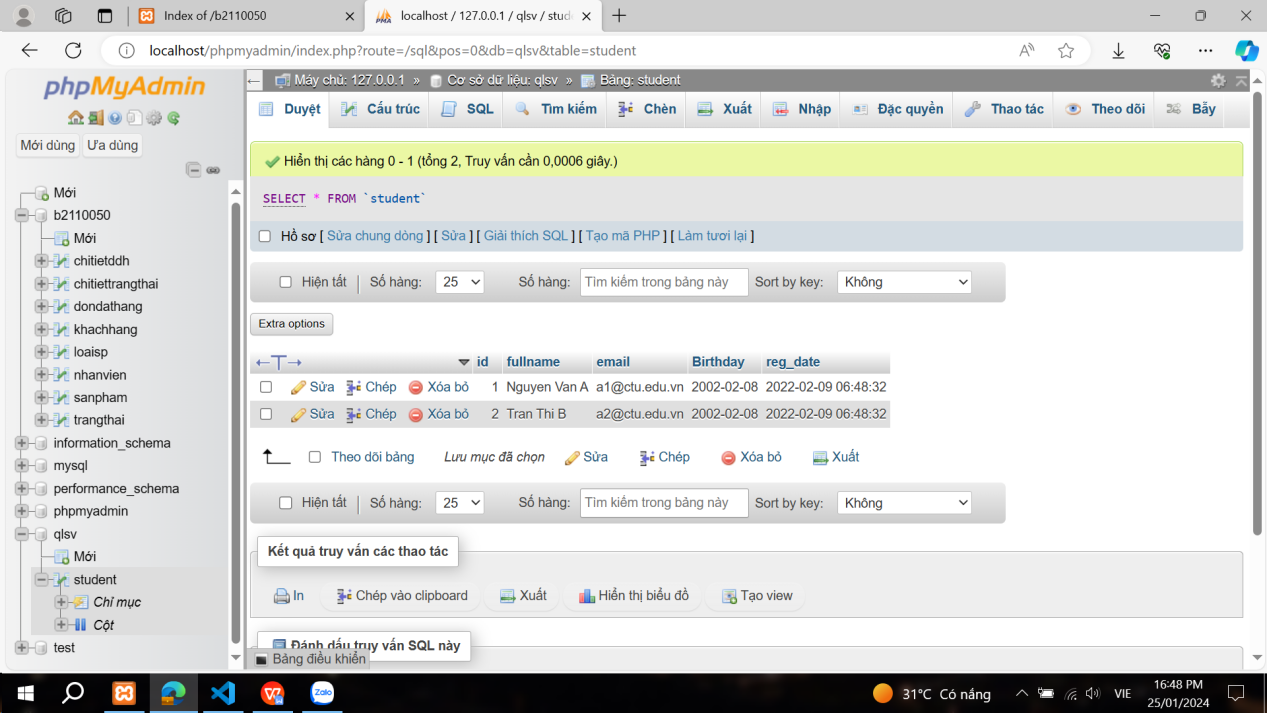
//Tạo 1 tập tin php con\_db.php



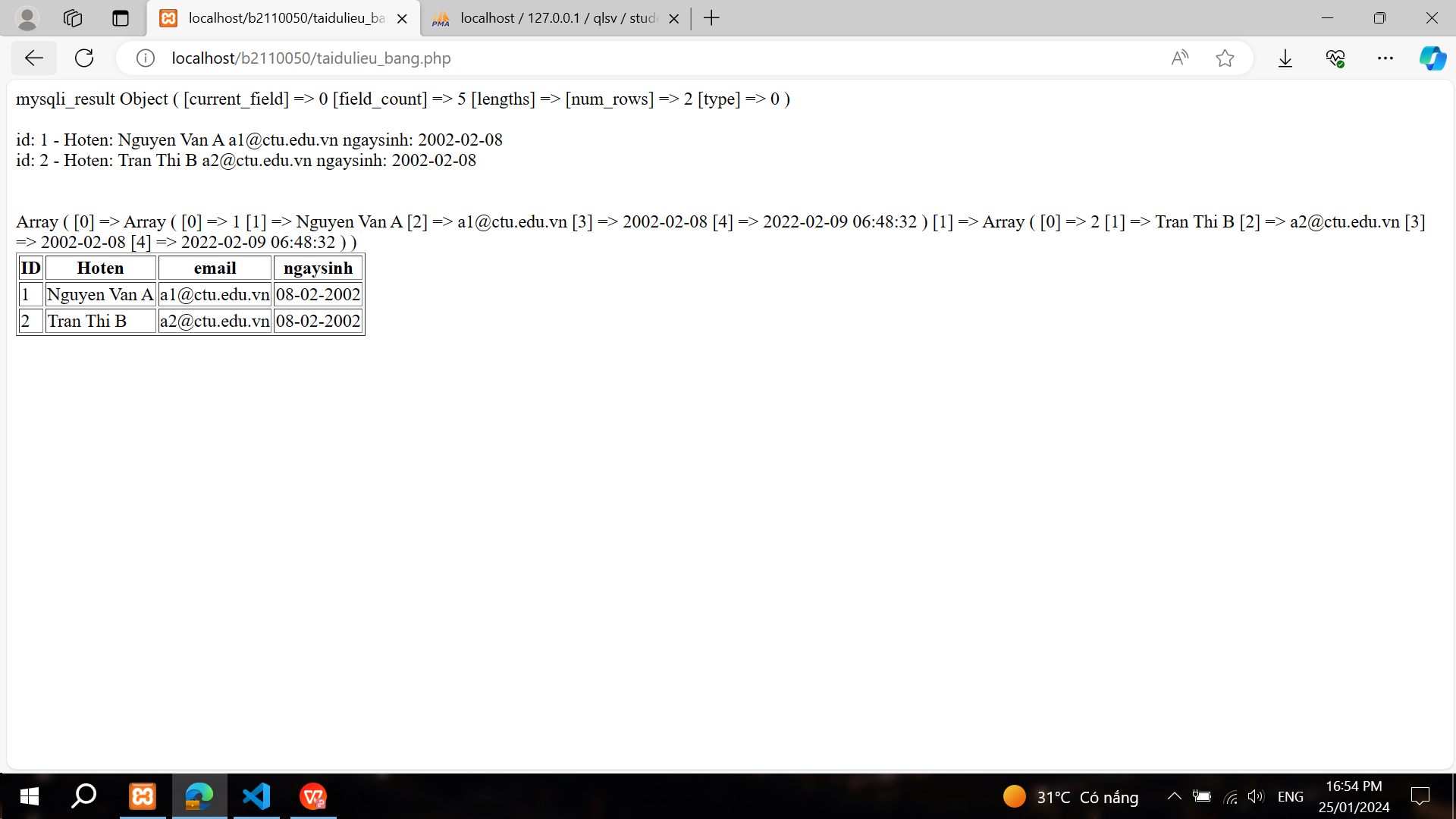


// Thêm dữ liệu vào bảng

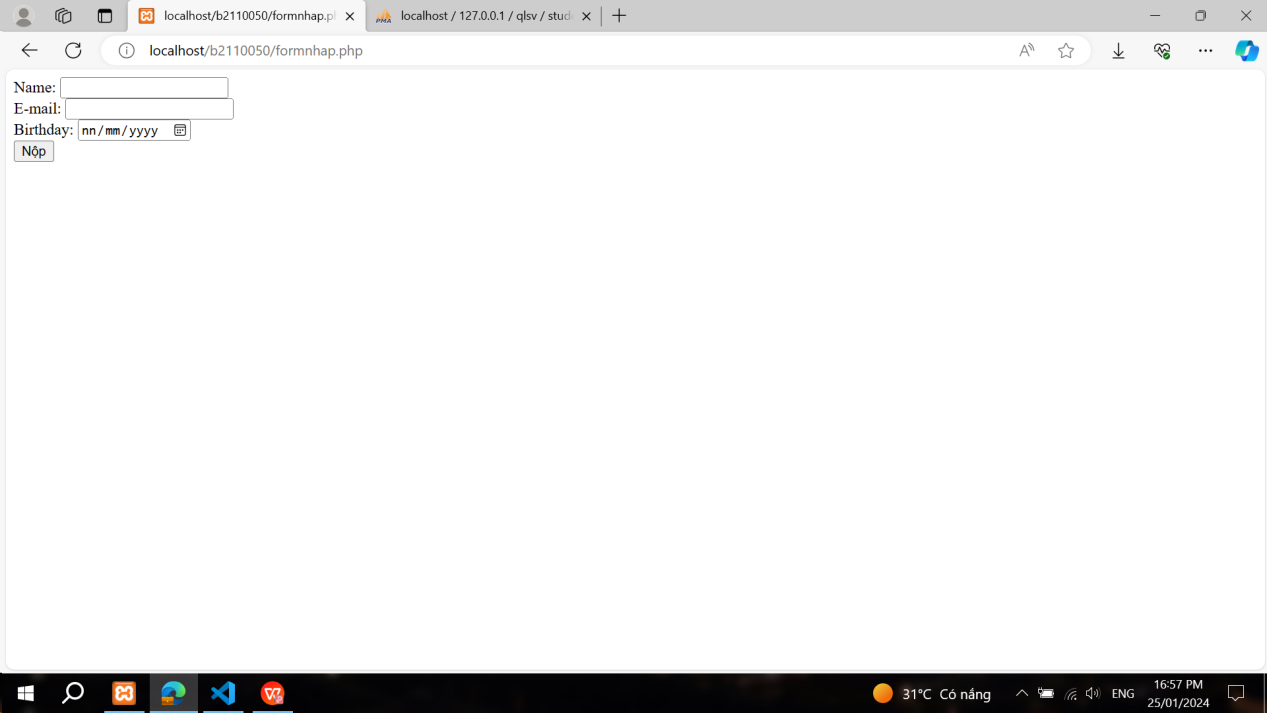




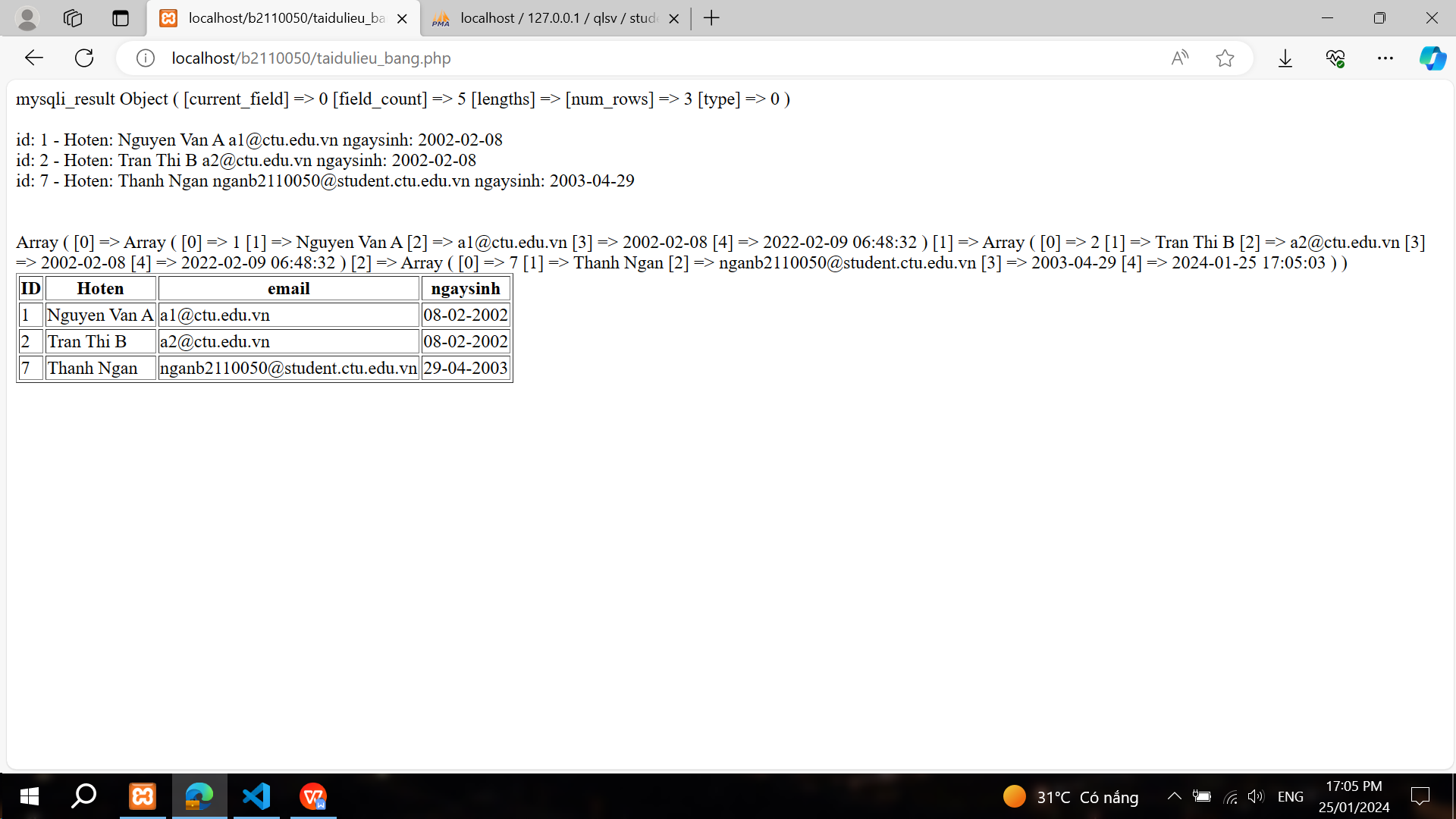
//Tạo tập tin taidulieu\_bang.php.



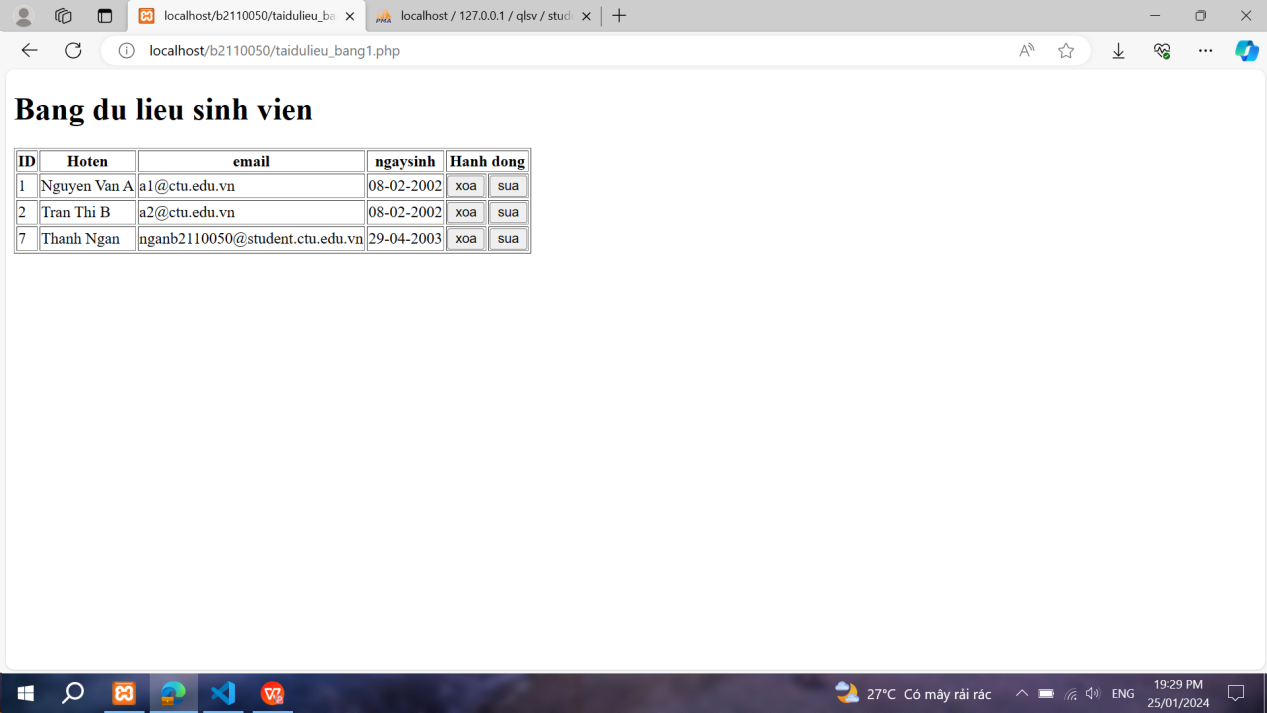
//Tạo form giao diện để nhập: formnhap.php



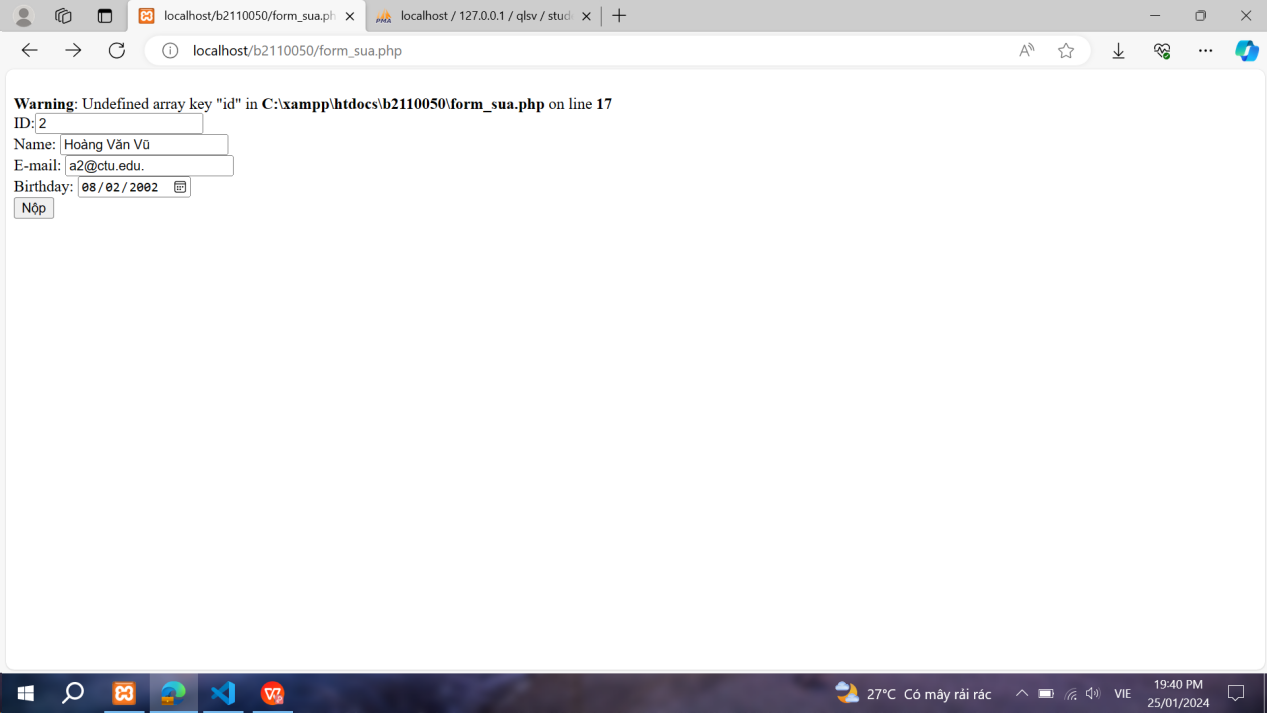
//Tạo tập tin php xử lý hành động lưu: luu.php

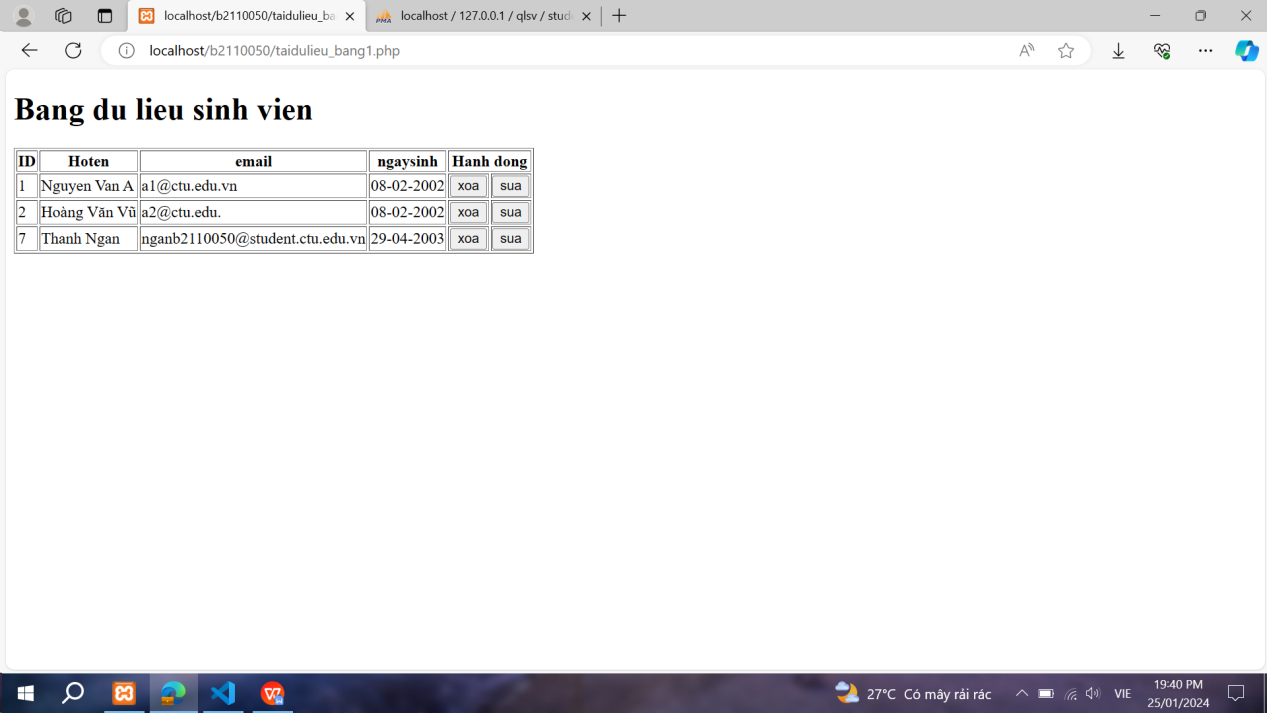


//Tạo tập tin taidulieu\_bang1.php



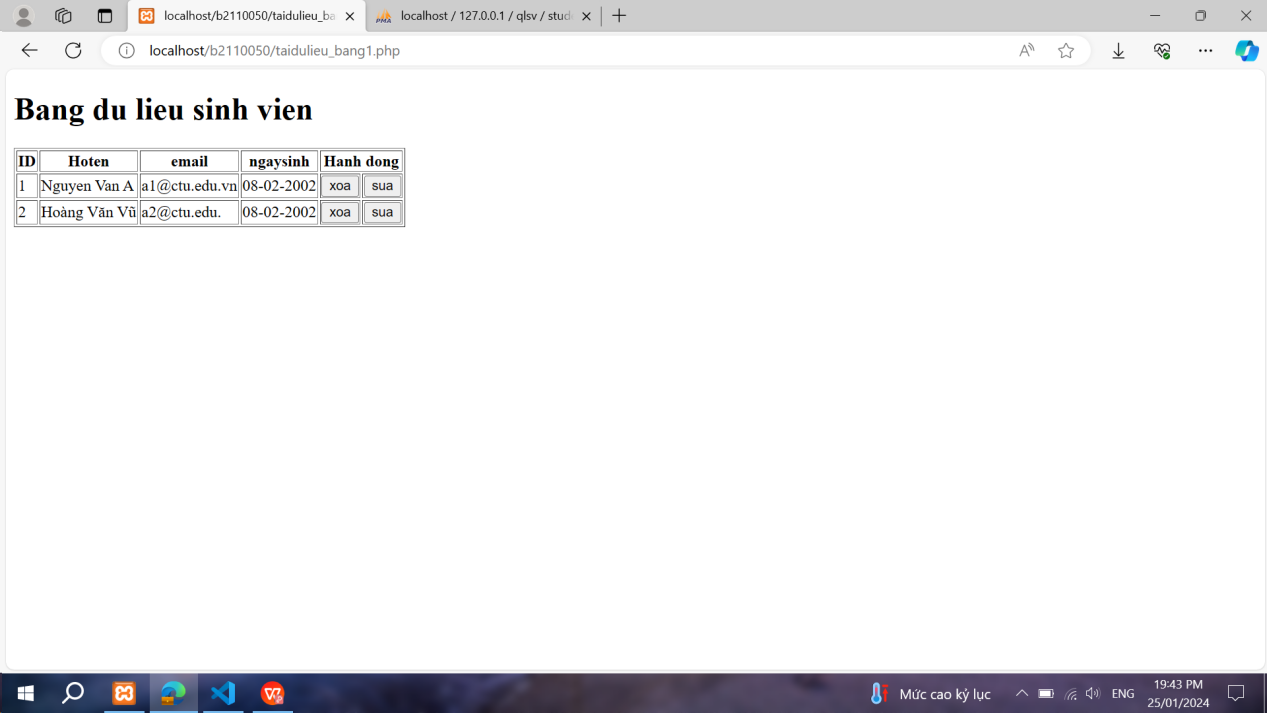
//Tao tap tin form\_sua.php





//tao tap tin xoa.php

Đã xóa id: 7



Câu 2:

- Oracle:

<?php

$servername = "your\_host";

$username = "your\_username";

$password = "your\_password";

$servicename = "your\_servicename";

// Tạo kết nối

$conn = oci\_connect($username, $password, "//{$servername}/{$servicename}");

// Kiểm tra kết nối

if (!$conn) {

$e = oci\_error();

trigger\_error(htmlentities($e['message'], ENT\_QUOTES), E\_USER\_ERROR);

}

// Thực hiện các thao tác cơ sở dữ liệu ở đây

// Đóng kết nối

oci\_close($conn);

?>

- SQL server:

<?php

$serverName = "your\_server\_name";

$connectionOptions = array(

"Database" => "your\_database",

"Uid" => "your\_username",

"PWD" => "your\_password"

);

// Tạo kết nối

$conn = sqlsrv\_connect($serverName, $connectionOptions);

// Kiểm tra kết nối

if (!$conn) {

die(print\_r(sqlsrv\_errors(), true));

}

// Thực hiện các thao tác cơ sở dữ liệu ở đây

// Đóng kết nối

sqlsrv\_close($conn);

?>

- SQLite:

<?php

$database = "your\_database.db";

// Tạo kết nối

$conn = new SQLite3($database);

// Kiểm tra kết nối

if (!$conn) {

die("Kết nối không thành công");

}

// Thực hiện các thao tác cơ sở dữ liệu ở đây

// Đóng kết nối

$conn->close();

?>

(tài liệu tham khảo: chatGPT).

Câu 3:

- Class mysqli trong PHP được sử dụng để thực hiện các thao tác cơ bản với MySQL.

- Các phương thức trong class mysqli:

+ mysqli::\_\_construct: Phương thức khởi tạo một đối tượng mysqli và tạo kết nối đến MySQL.

+ mysqli::query: Thực hiện một truy vấn SQL trên cơ sở dữ liệu.

+ mysqli::real\_escape\_string: Escapes một chuỗi để sử dụng trong truy vấn SQL để ngăn chặn tấn công SQL injection.

+ mysqli::prepare: Chuẩn bị một câu truy vấn SQL để sử dụng với các tham số.

+ mysqli\_stmt::bind\_param: Liên kết các biến với các tham số trong một câu truy vấn chuẩn bị.

+ mysqli\_stmt::execute: Thực thi một câu truy vấn chuẩn bị đã được liên kết với các tham số.

+ mysqli\_stmt::get\_result: Nhận kết quả của một câu truy vấn chuẩn bị dạng SELECT.

+ mysqli\_result::fetch\_assoc: Trả về một mảng kết hợp chứa các cột từ hàng kết quả.

+ mysqli::close: Đóng kết nối đến MySQL.

(Tài liệu tham khảo: chatGPT).

Câu 4:

- Phương thức die() trong PHP được sử dụng để dừng thực thi của chương trình và hiển thị một thông báo lỗi. Nó giống như phương thức exit() và có thể thay thế lẫn nhau. Phương thức này có thể nhận một chuỗi hoặc giá trị để hiển thị thông báo lỗi tương ứng.

- Trong ví dụ cụ thể của bài, phương thức die("Connection failed: " . $conn->connect\_error); được sử dụng khi kết nối đến MySQL không thành công. Nếu có lỗi kết nối, chương trình sẽ dừng lại và hiển thị thông báo lỗi "Connection failed" cùng với thông điệp chi tiết về lỗi được trả về từ MySQL ($conn->connect\_error).

Câu 5:

- Cách 4: Sử dụng “fetch\_row”

// Load dữ liệu mới vào biến result

$result = $conn->query($sql);

// Trình bày dữ liệu trong bảng HTML

echo "<table border='1'><tr><th>ID</th><th>Hoten</th><th>email</th><th>ngaysinh</th></tr>";

// Output data of each row

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

echo "<tr><td>" . $row[0] . "</td><td>" . $row[1] . "</td><td>" . $row[2] . "</td><td>" . $row[3] . "</td></tr>";

}

echo "</table>";

- Cách 5: Sử dụng “fetch\_array”

// Load dữ liệu mới vào biến result

$result = $conn->query($sql);

// Trình bày dữ liệu trong bảng HTML

echo "<table border='1'><tr><th>ID</th><th>Hoten</th><th>email</th><th>ngaysinh</th></tr>";

// Output data of each row

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

echo "<tr><td>" . $row['id'] . "</td><td>" . $row['fullname'] . "</td><td>" . $row['email'] . "</td><td>" . $row['Birthday'] . "</td></tr>";

}

echo "</table>";

- Cách 6: Sử dụng fetch\_object

// Load dữ liệu mới vào biến result

$result = $conn->query($sql);

// Trình bày dữ liệu trong bảng HTML

echo "<table border='1'><tr><th>ID</th><th>Hoten</th><th>email</th><th>ngaysinh</th></tr>";

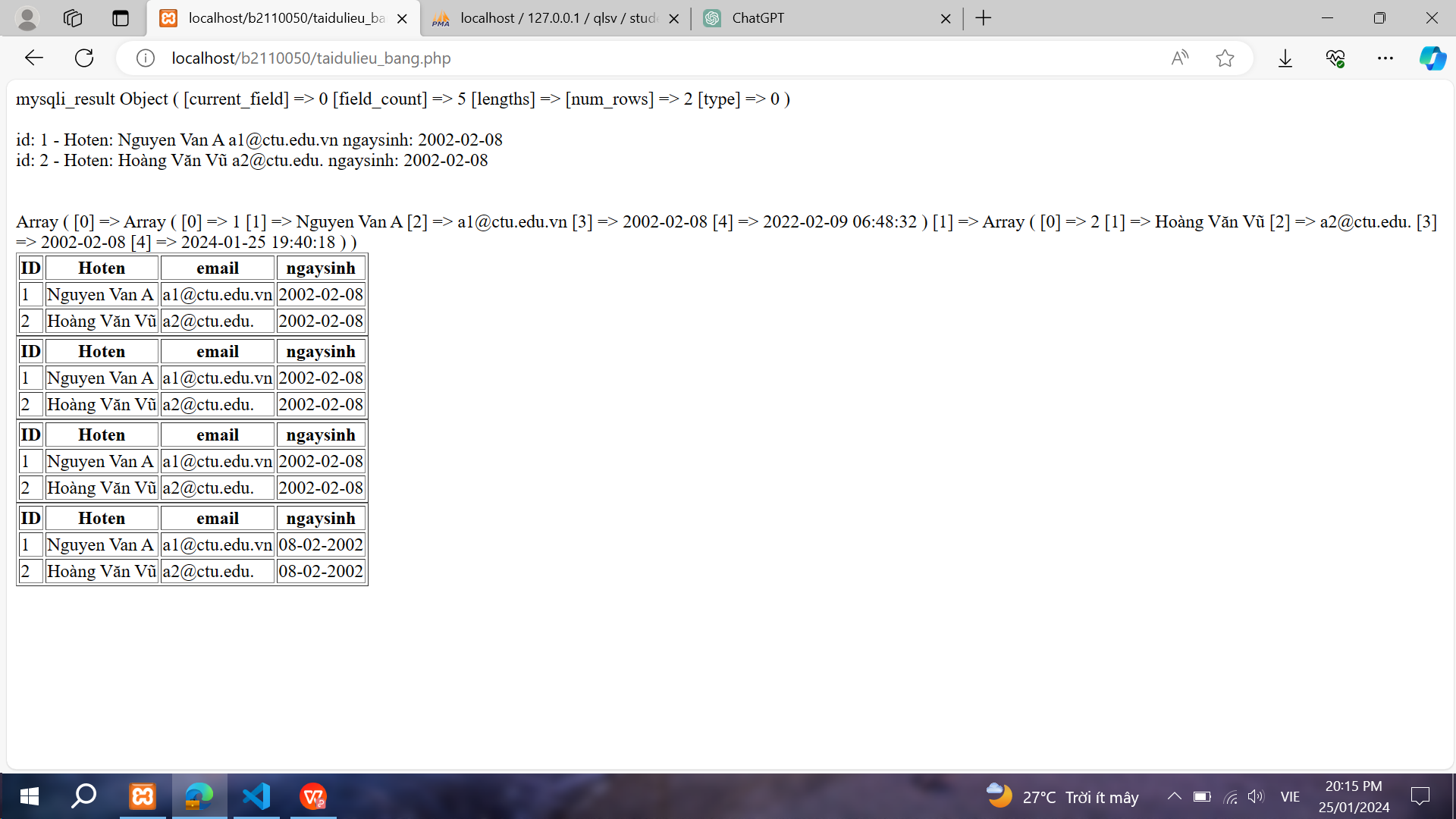
// Output data of each row

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

echo "<tr><td>" . $row->id . "</td><td>" . $row->fullname . "</td><td>" . $row->email . "</td><td>" . $row->Birthday . "</td></tr>";

}

echo "</table>";



Câu 6:

- Công dụng chính của header() là điều chỉnh thông tin liên quan đến HTTP, chẳng hạn như các tiêu đề trang, chuyển hướng trang, quản lý cookie và caching.

+ Chuyển Hướng Trang (HTTP Redirect):

<?php

header("Location: http://example.com/newpage.php");

exit();

?>

+ Quản Lý Cache:

<?php

// Chỉ định thời gian sống của trang (ví dụ: 1 giờ)

header("Cache-Control: max-age=3600");

?>

+ Đặt Tiêu Đề Trang:

<?php

header("Content-Type: text/html; charset=UTF-8");

?>

+ Xử Lý Cookie:

<?php

// Tạo một cookie

setcookie("user", "John Doe", time() + 3600, "/");

// Gửi một tiêu đề với thông báo cookie

header("Set-Cookie: user=John Doe; expires=" . gmdate('D, d M Y H:i:s \G\M\T', time() + 3600) . "; path=/");

?>

+ Xử Lý Định Dạng Trả Về (Content-Disposition):

<?php

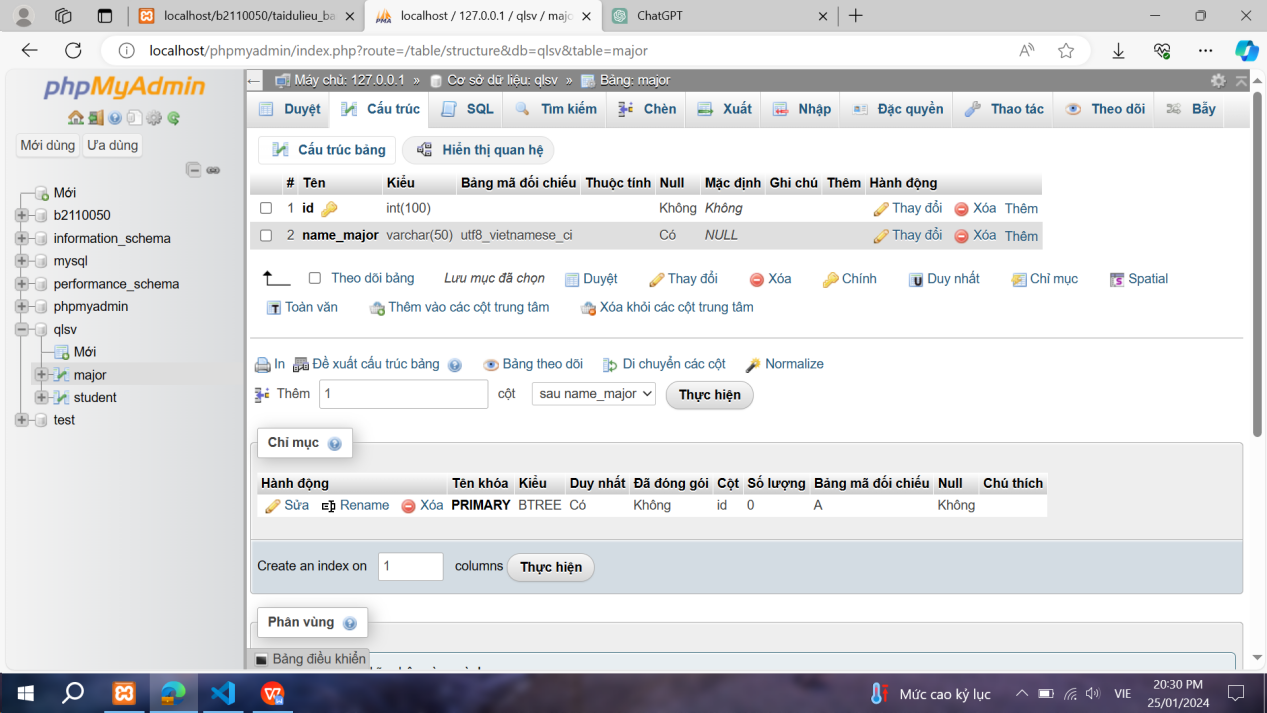
// Đặt tiêu đề để tải file về máy tính người dùng

header("Content-Disposition: attachment; filename=example.txt");

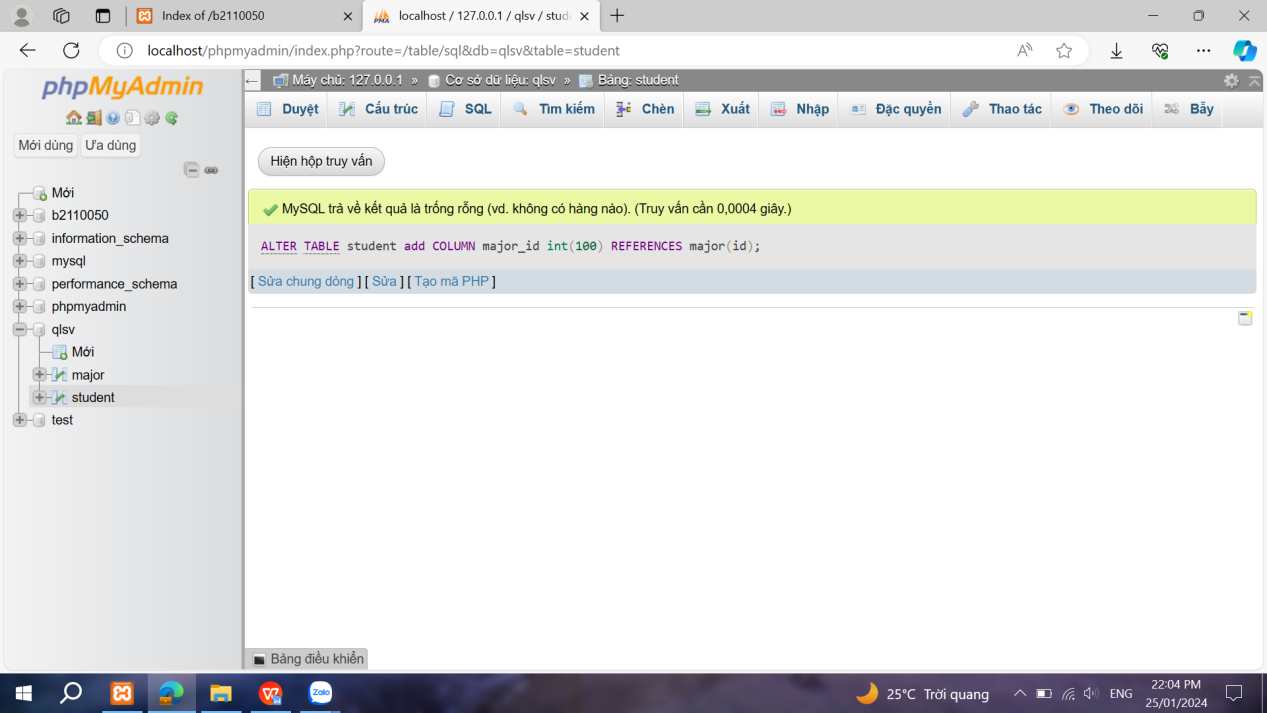
?>

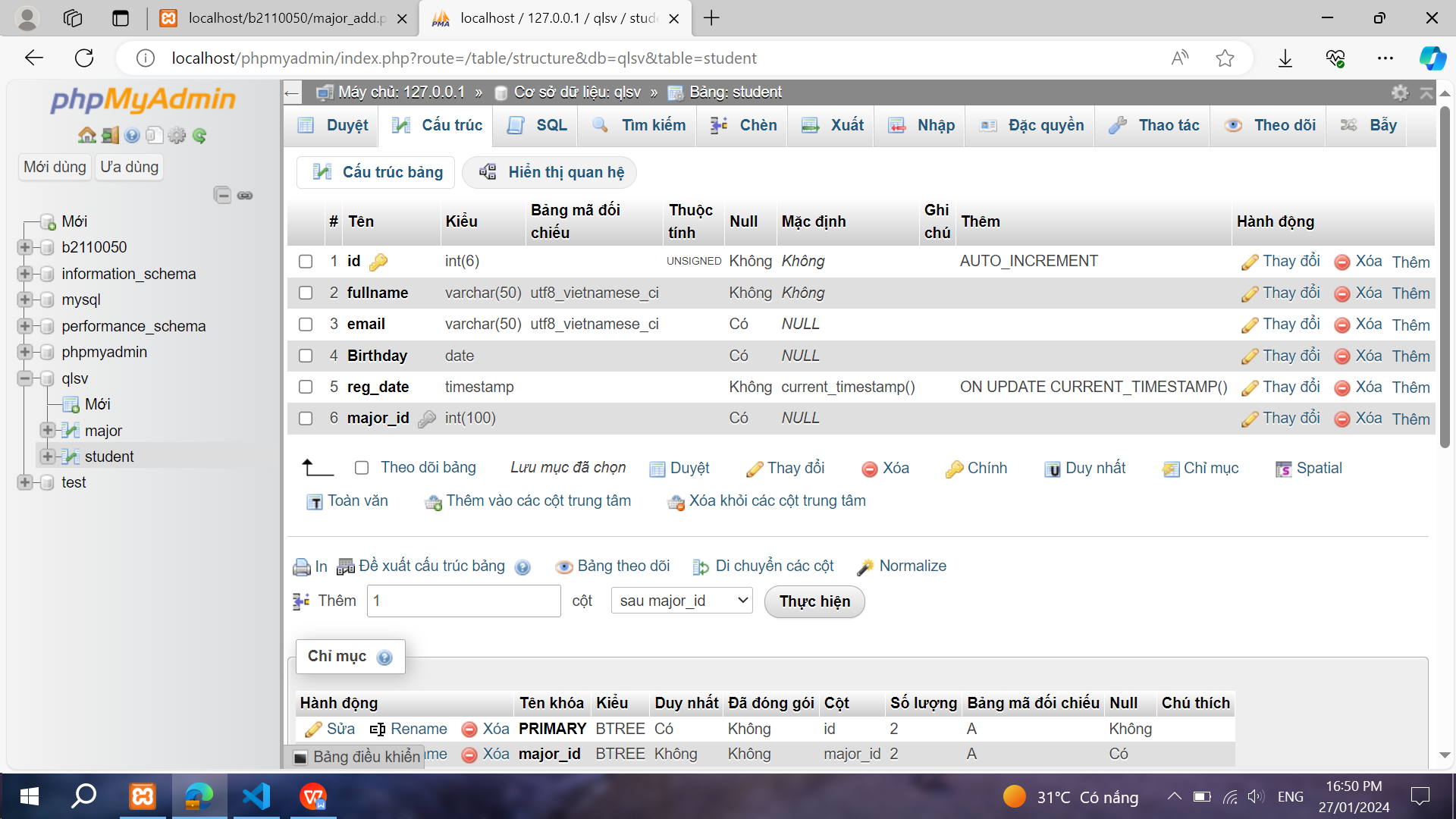
Câu 7:

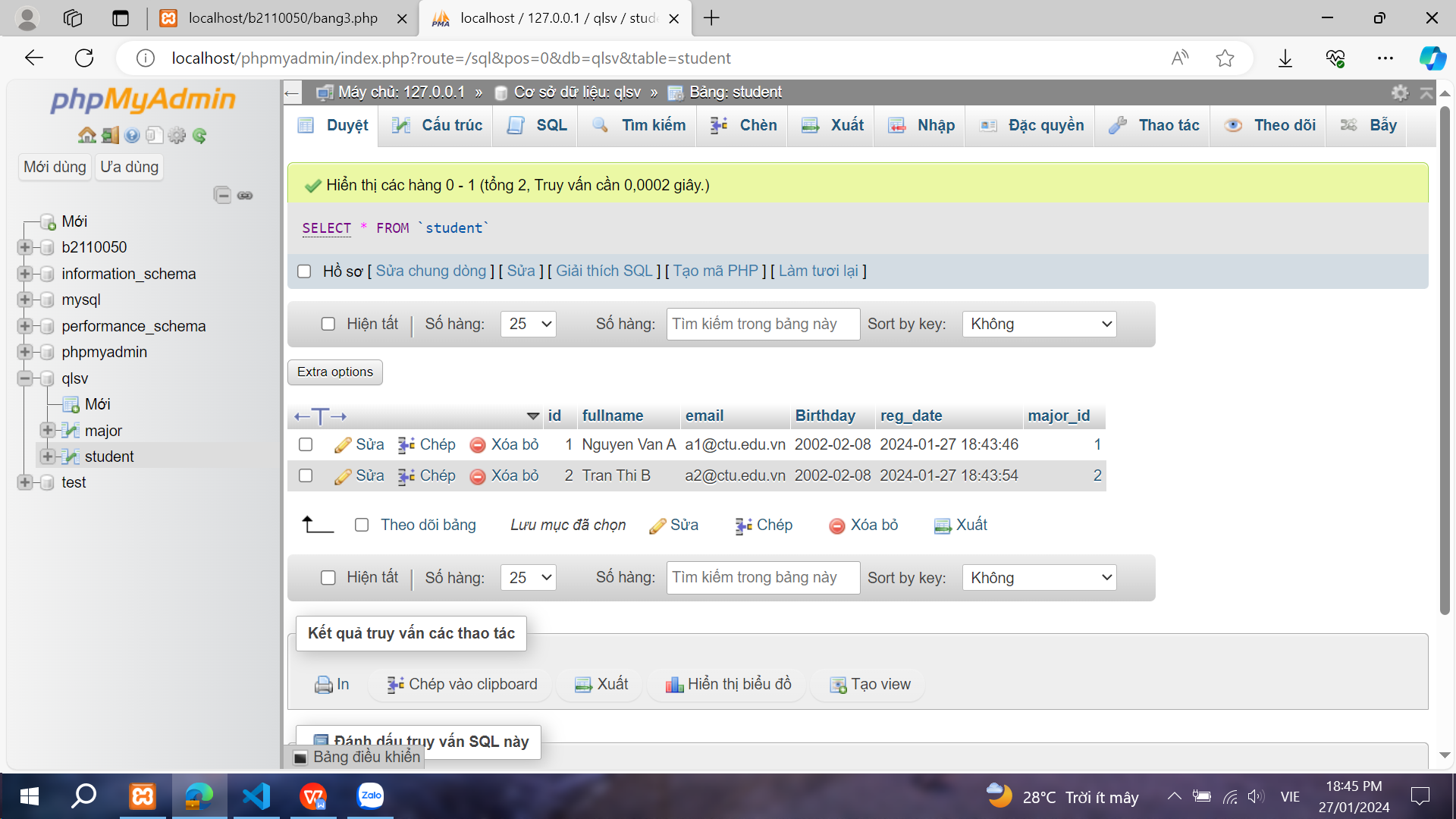
// tao bang major



//them khoa ngoai cho bang student

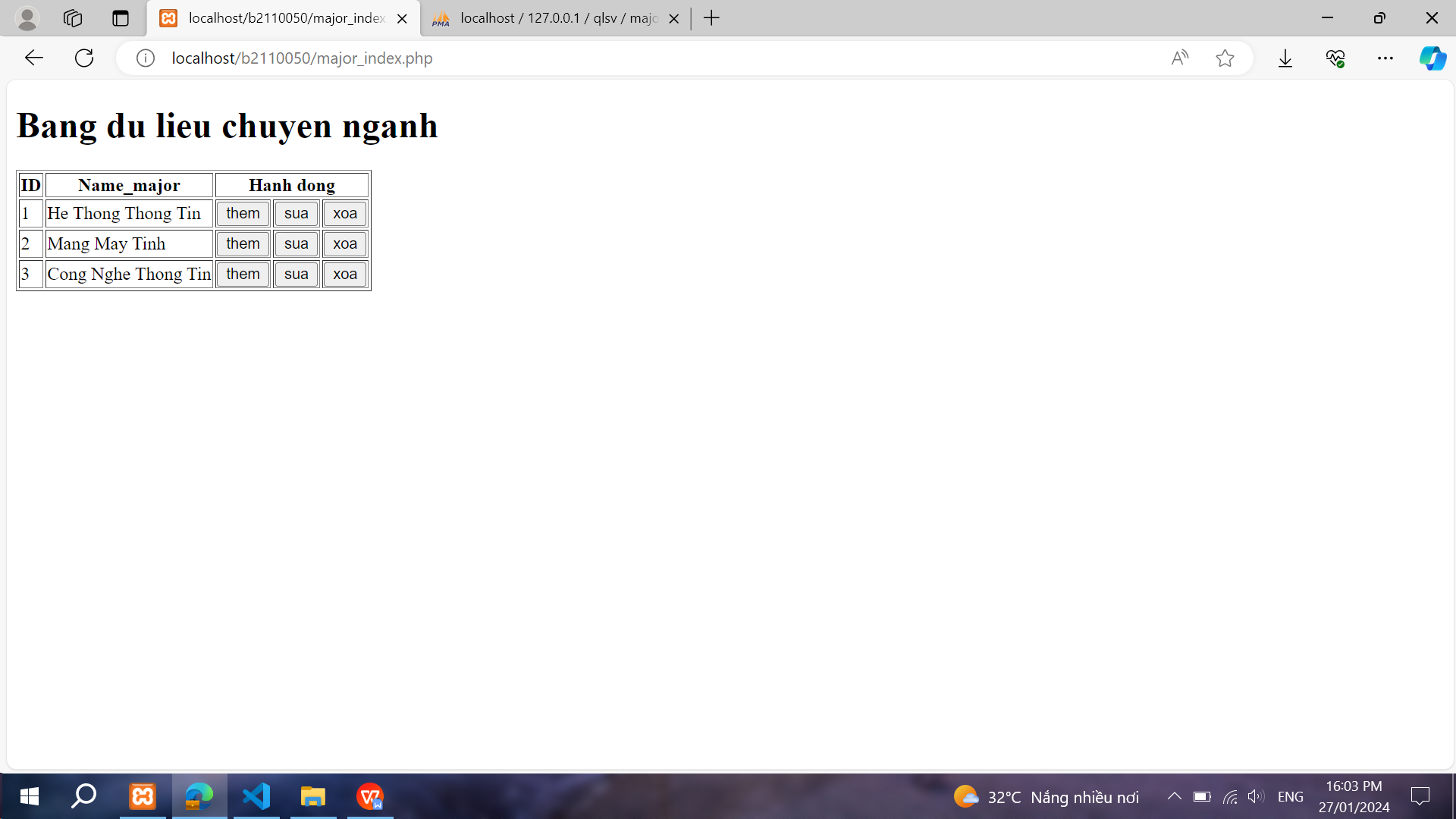




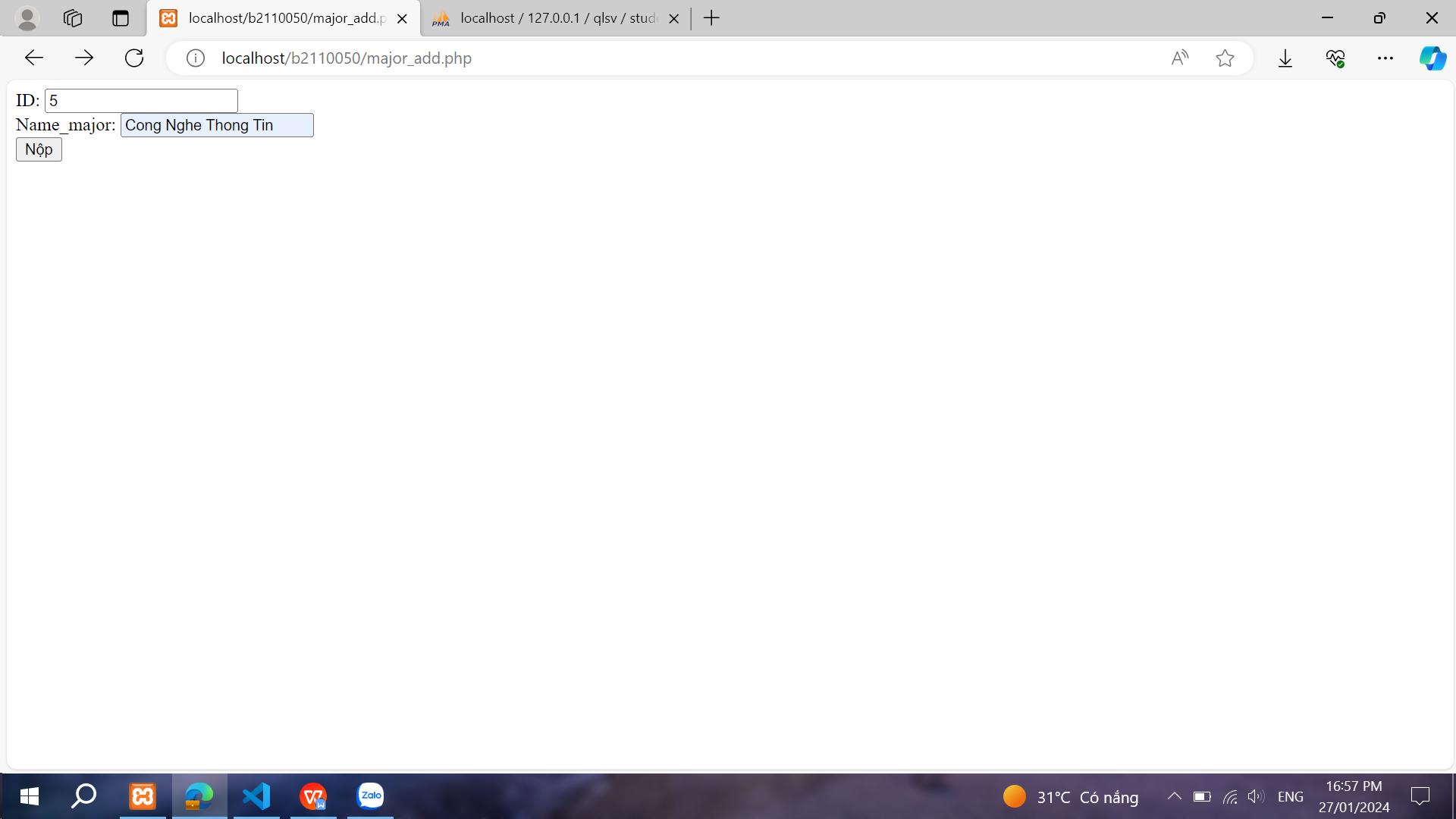


Câu 8:

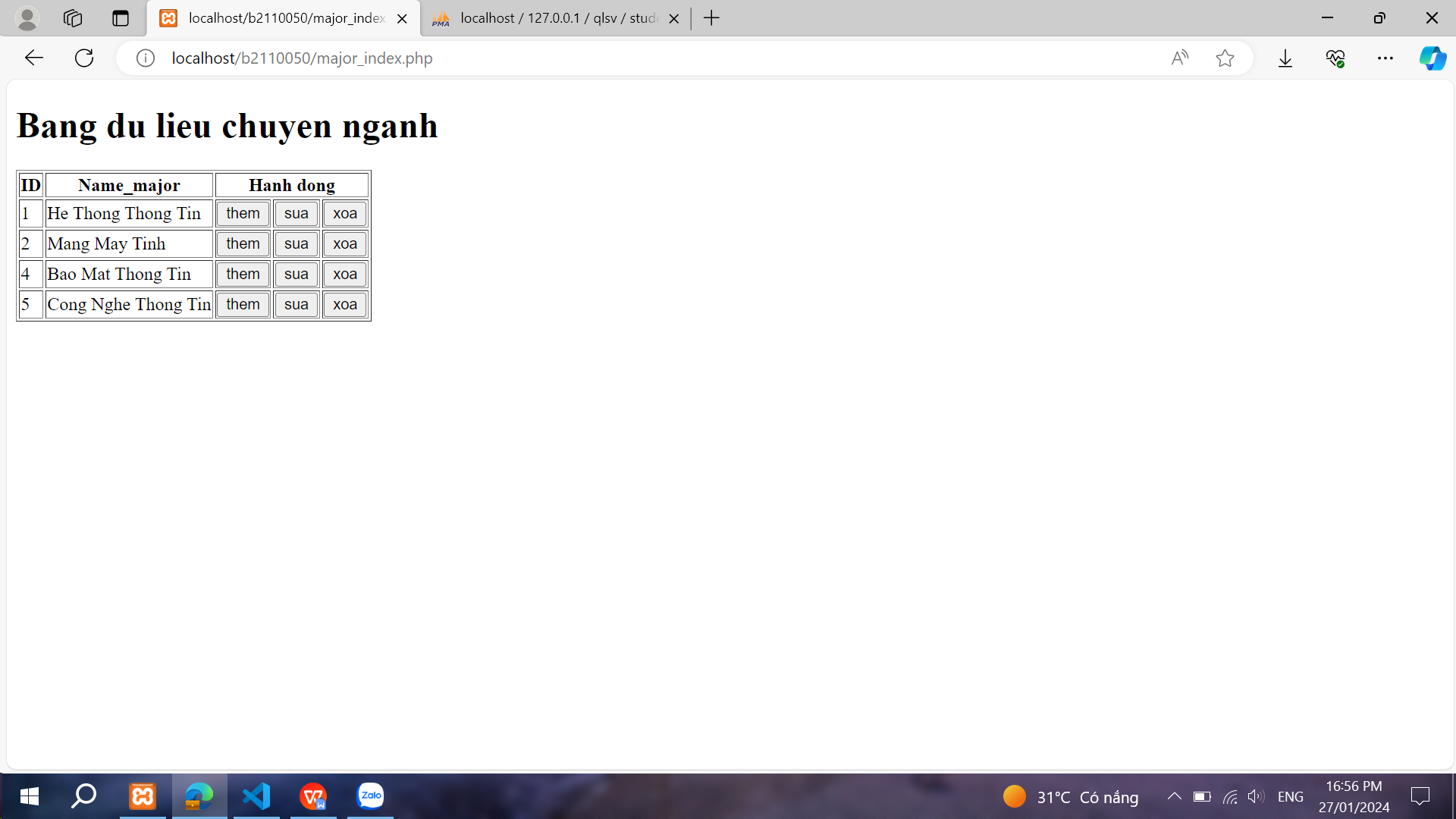
// tao tap tin major\_index.php



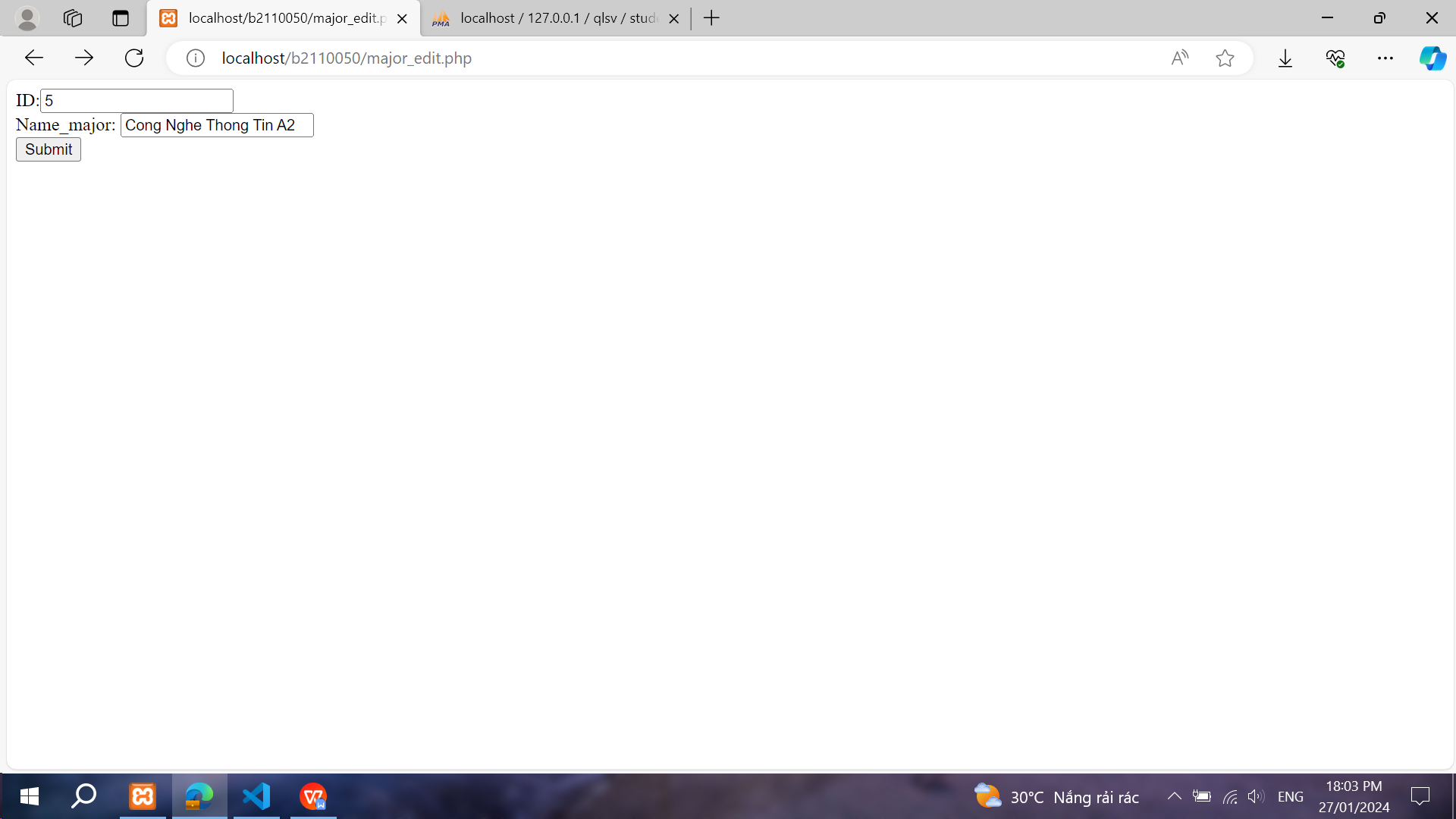
// tao major\_add.php



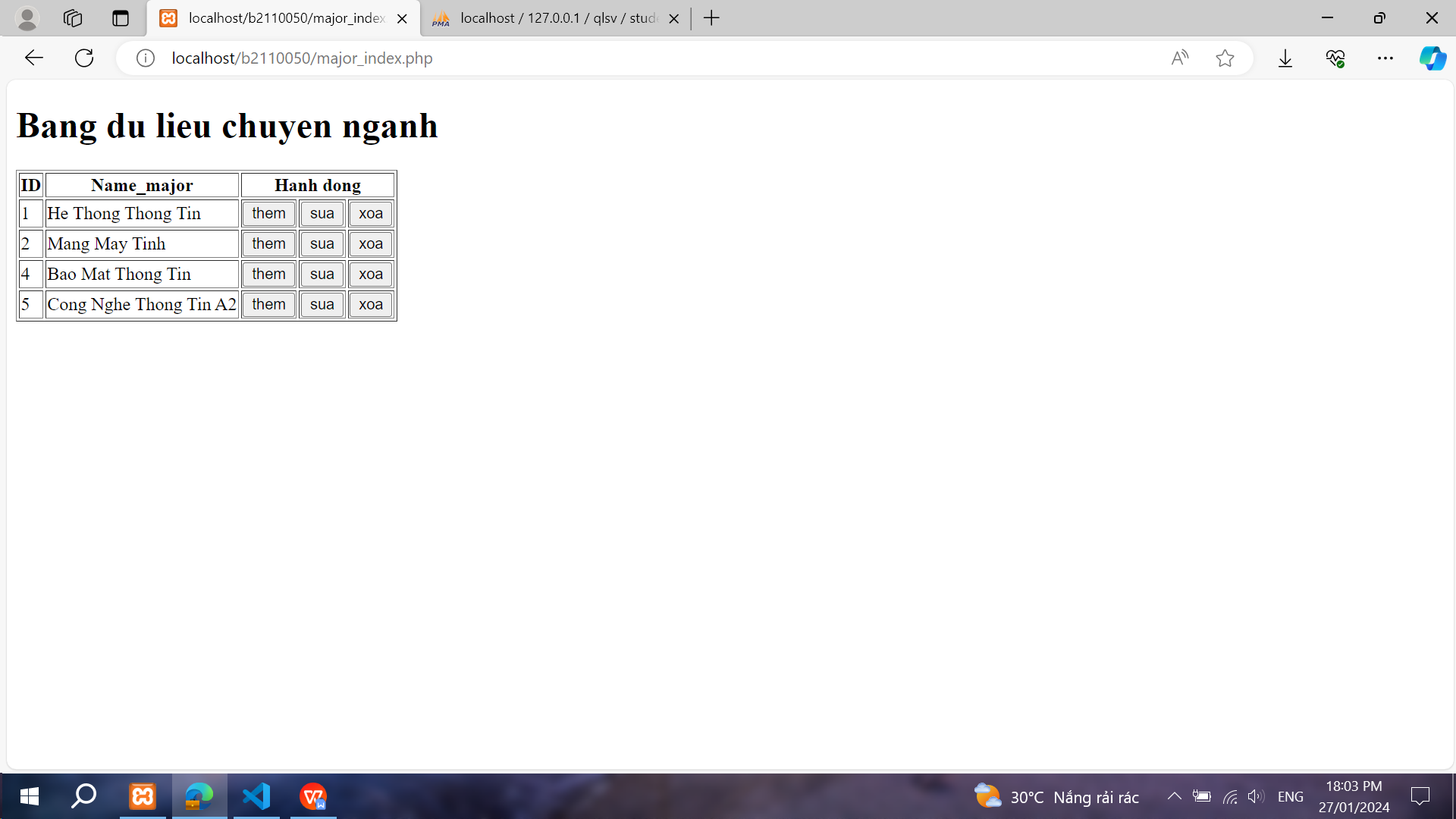
//tao major\_save.php



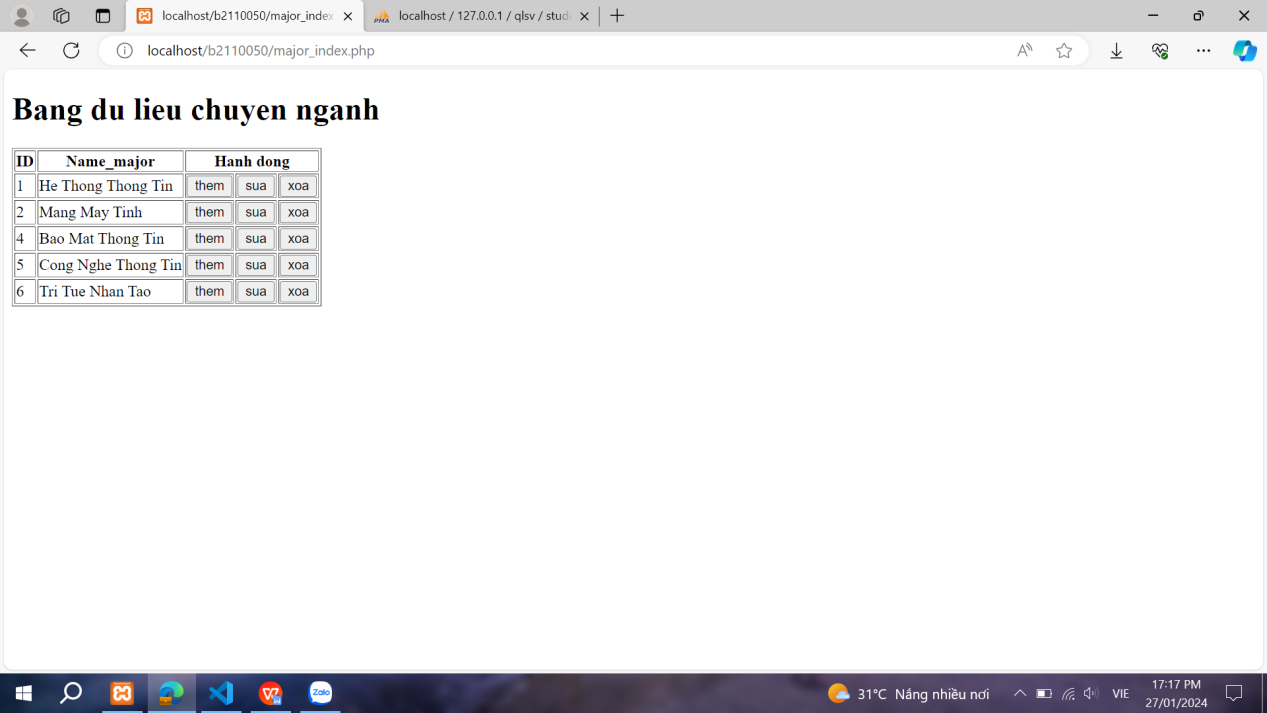
// tao major\_edit

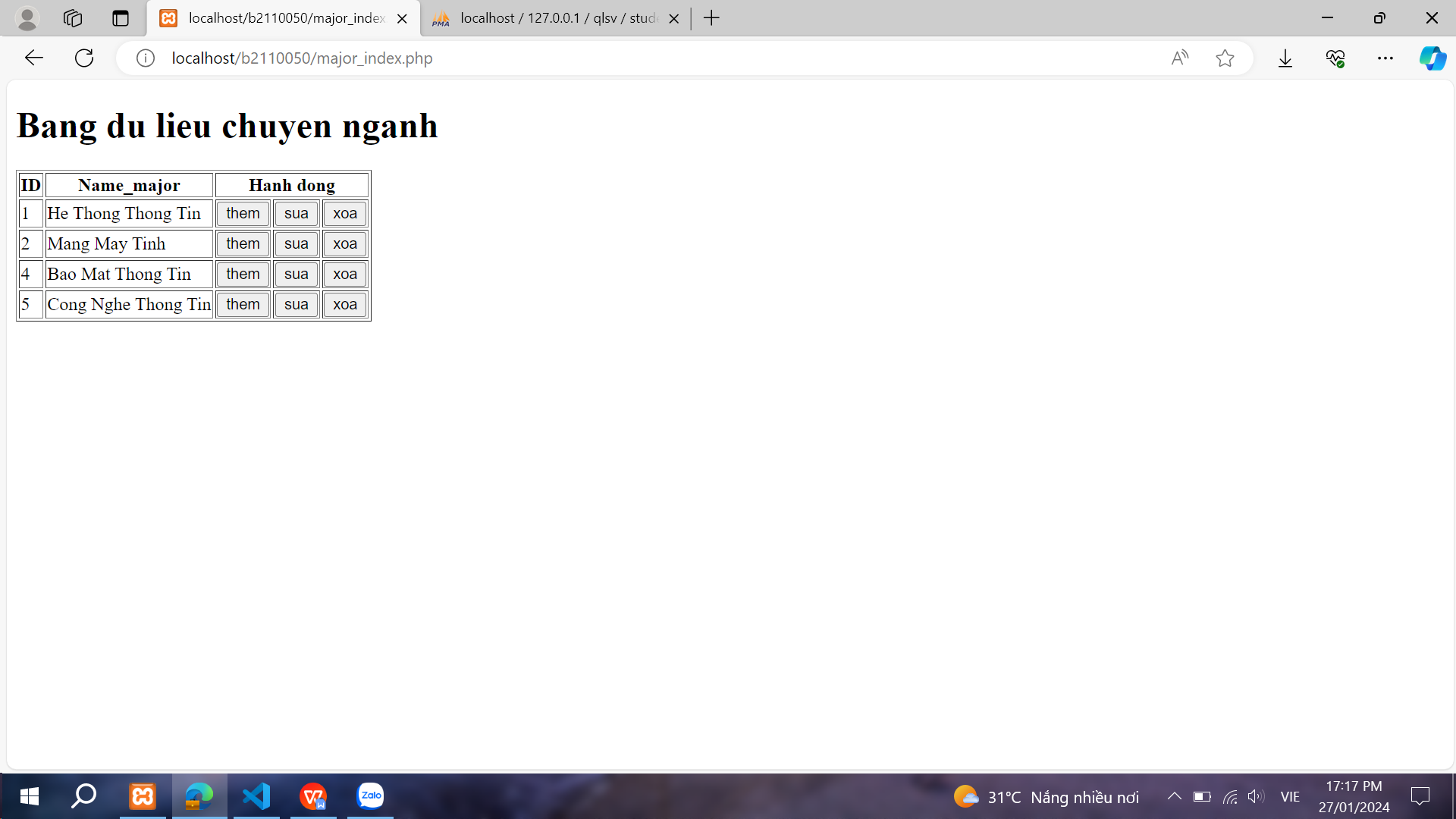


//tao major\_edit\_save

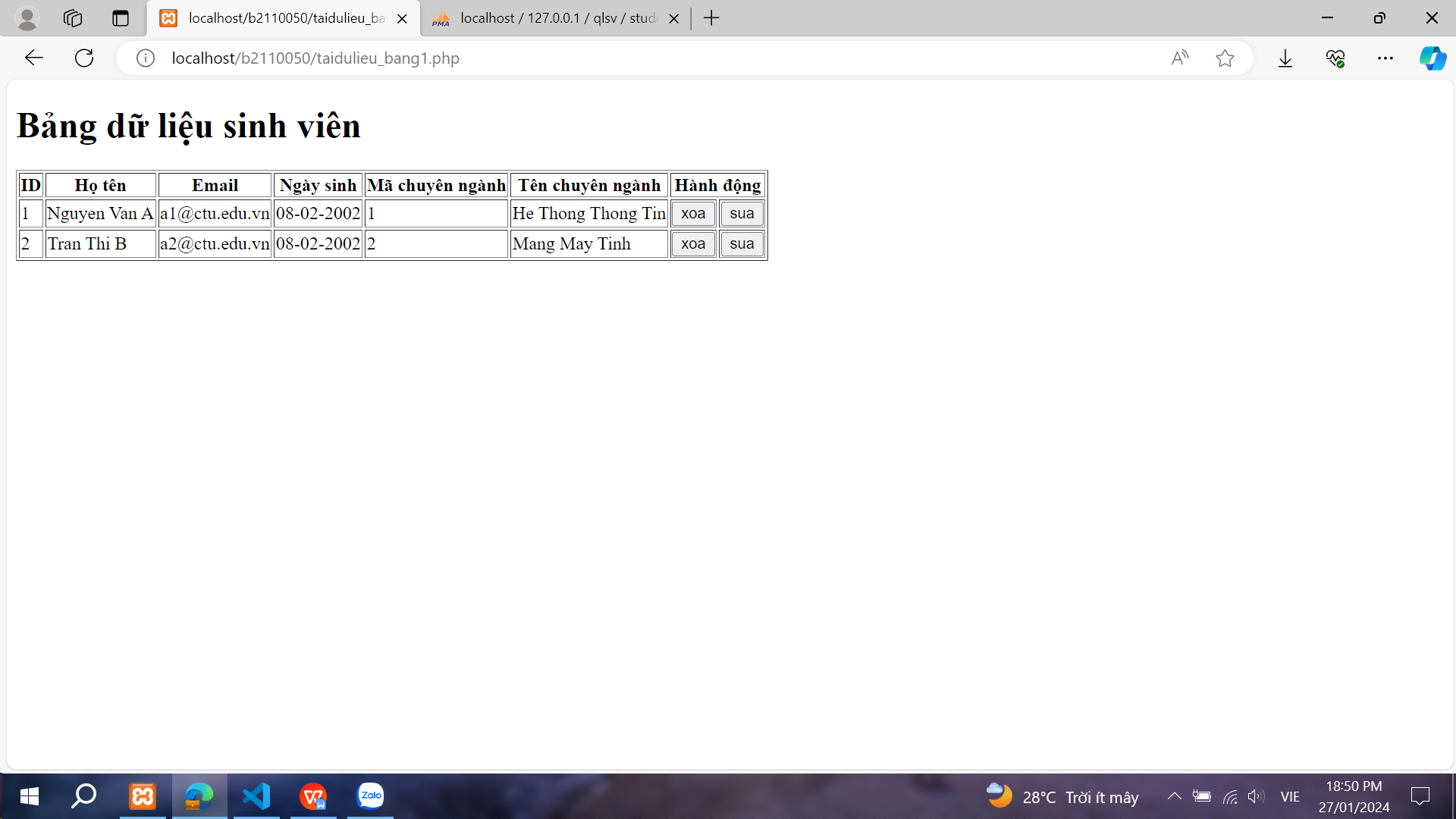


//tao major\_xoa.php -> da xoa du lieu co id=6





Câu 9:



Câu 10: