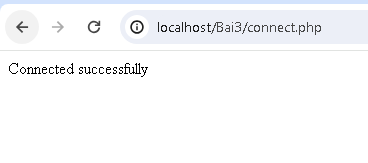
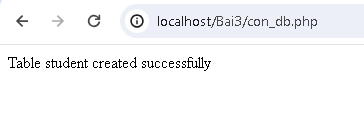
**1.Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.**

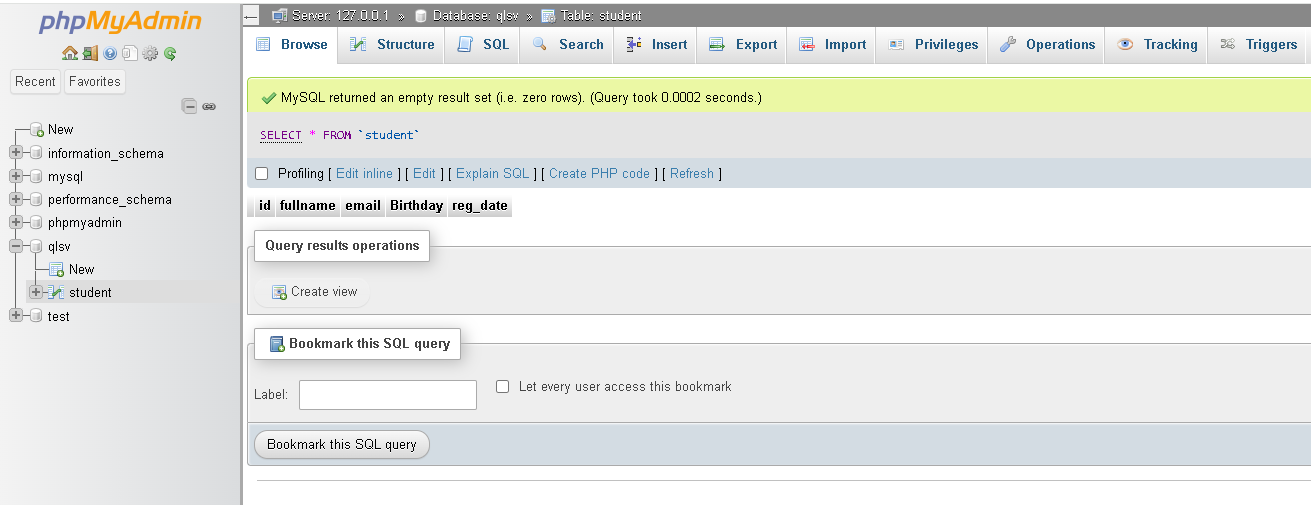
* Tạo chuỗi kết nối đến mysql

connect.php

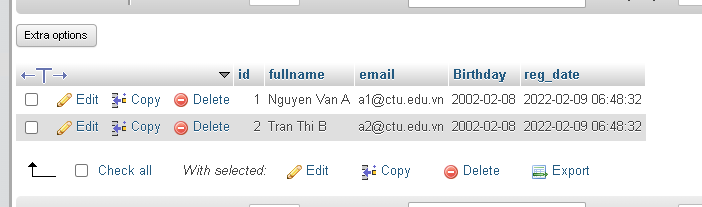


con\_db.php

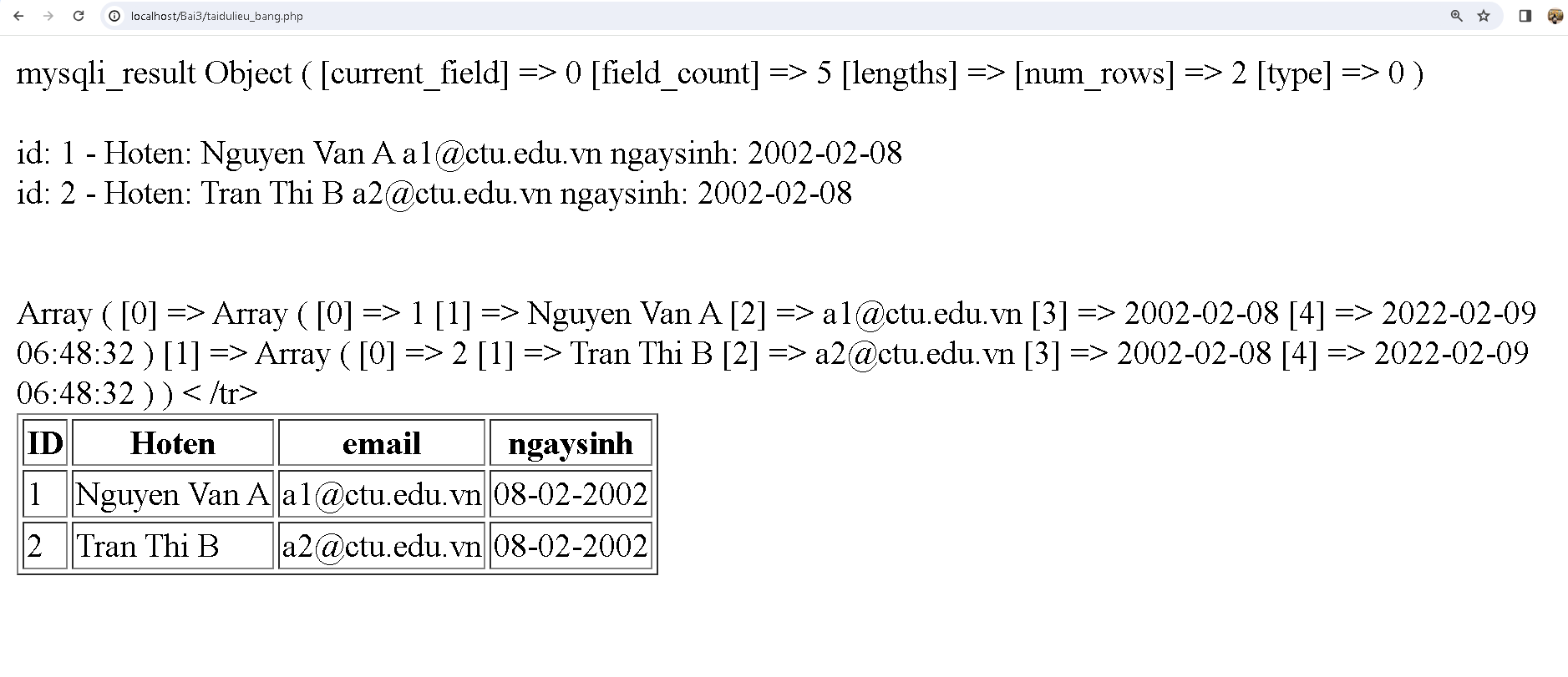




* Index: Tải dữ liệu từ bảng trong csdl đưa lên trình bày trên trang web

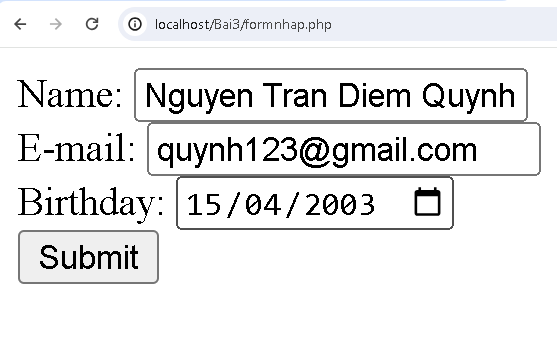


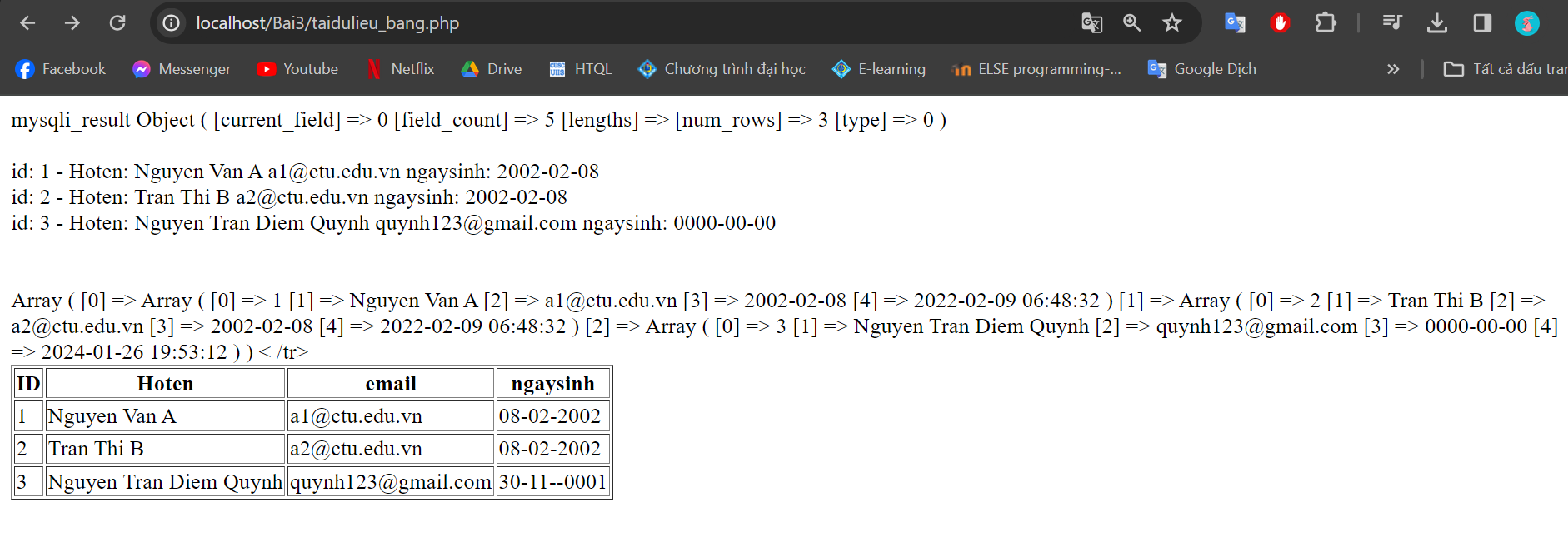
taidulieu\_bang.php



* Add: Form để thêm dữ liệu vào CSDL và chuyển đến trang thể hiện dữ liệu

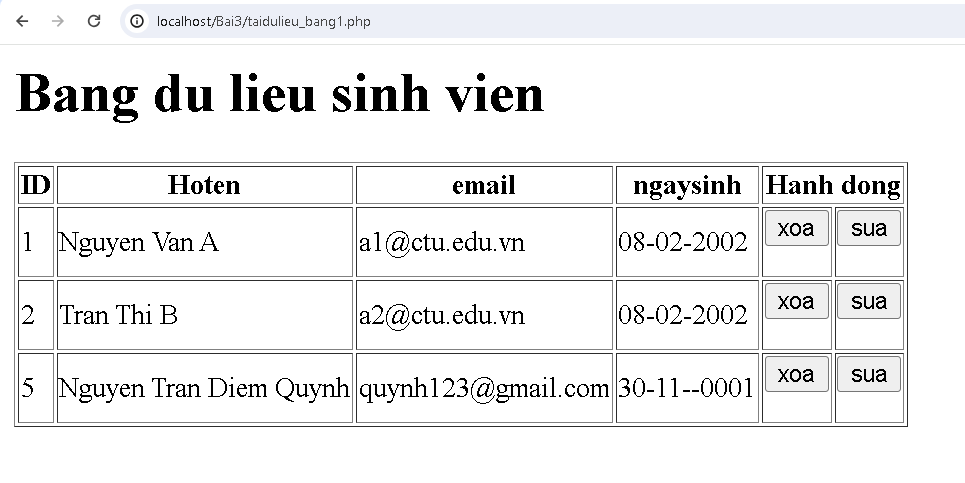
formnhap.php



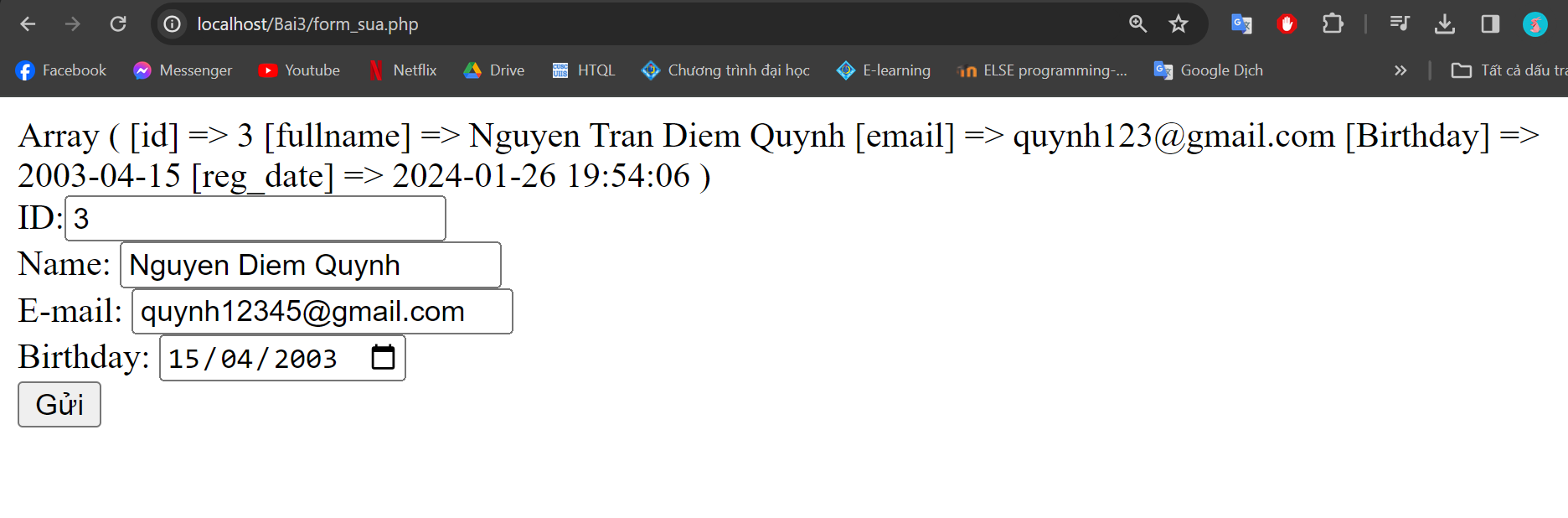


* Edit: Chỉnh sửa dữ liệu từ danh sách

tailieu\_bang1.php



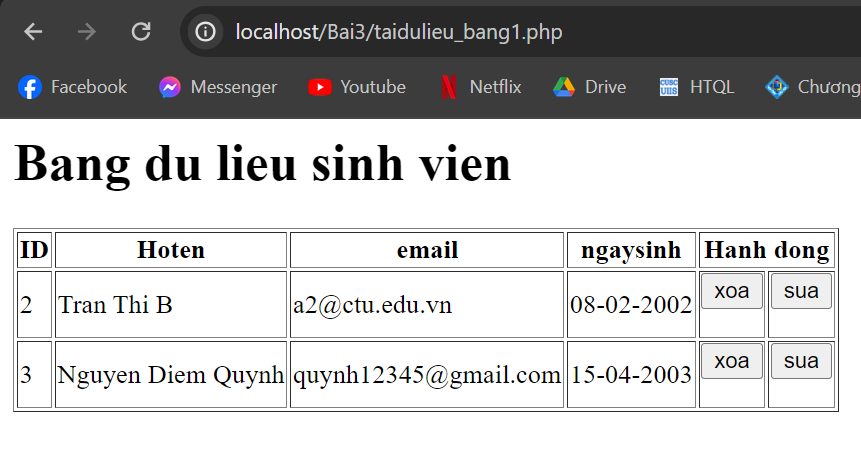
form\_sua.php





* Delete: Xóa dòng dữ liệu trong bảng

xoa.php



**2.Từ code kết nối với MySQL, bạn hãy tìm và trình bày code kết nối với các hệ quản trị CSDL khác như Oracle, SQL server, SQLite.**

* Code kết nối với Oracle:

<?php

$user='scott';

$pass='tiger';

$link=oci\_connect($user, $pass, 'ORCL');

if(!$link){

$error=oci\_error();

echo $error['Connection failed'];

exit();

}

echo "Connected successfully";

?>

* Code kết nối với SQL server:

<?php

function OpenConnection()

{

$serverName = "localhost";

$connectionOptions = array("Database"=>"AdventureWorks",

"Uid"=>"MyUser", "PWD"=>"MyPassword");

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

if($conn == false)

die(FormatErrors(sqlsrv\_errors()));

return $conn;

}

?>

* Code kết nối với SQLite: Chương trình PHP sau minh họa cách kết nối tới Database đang tồn tại. Nếu Database đó không tồn tại, thì nó sẽ được tạo, và cuối cùng một đối tượng Database sẽ được trả về.

<?php

class MyDB extends SQLite3

{

function \_\_construct()

{

$this->open('test.db');

}

}

$db = new MyDB();

if(!$db){

echo $db->lastErrorMsg();

} else {

echo "Opened database successfully\n";

}

?>

Nguồn: <https://fr.slideshare.net/myloveforyounqt/php-vi-oracle-database>

<https://topdev.vn/blog/huong-dan-ket-noi-php-voi-sql-server/>

<https://hoclaptrinh.vn/tutorial/hoc-sqlite-co-ban-va-nang-cao/su-dung-sqlite-voi-php>

**3.Cho biết class mysqli để hỗ trợ thực hiện những điều gì. Hãy liệt kê và mô tả các phương thức trong class mysqli (tham khảo ở 2 hoặc các tài liệu khác mà bạn tìm được).**

Trong PHP, class mysqli được sử dụng để tạo và quản lý kết nối với cơ sở dữ liệu MySQL. Nó cung cấp các phương thức và thuộc tính để thực hiện các thao tác cơ bản như thực hiện truy vấn, quản lý giao dịch, xử lý lỗi, và nhiều chức năng khác.

* Các phương thức trong class mysqli:

affected\_rows(): Trả về số lượng hàng bị ảnh hưởng trong thao tác MySQL trước đó.

autocommit(): Bật hoặc tắt chế độ tự động commit cho các thay đổi trong cơ sở dữ liệu.

begin\_transaction(): Bắt đầu một giao dịch.

change\_user(): Thay đổi người dùng của kết nối cơ sở dữ liệu được chỉ định.

character\_set\_name(): Trả về bảng mã ký tự mặc định cho kết nối cơ sở dữ liệu.

close(): Đóng kết nối cơ sở dữ liệu đã mở trước đó.

commit(): Xác nhận giao dịch hiện tại.

connect(): Mở một kết nối mới đến máy chủ MySQL.

connect\_errno(): Trả về mã lỗi từ lỗi kết nối cuối cùng.

connect\_error(): Trả về mô tả lỗi từ lỗi kết nối cuối cùng.

data\_seek(): Điều chỉnh con trỏ kết quả đến một hàng tùy ý trong tập kết quả.

debug(): Thực hiện các hoạt động gỡ lỗi.

dump\_debug\_info(): Ghi thông tin gỡ lỗi vào log.

errno(): Trả về mã lỗi cuối cùng cho cuộc gọi hàm gần đây nhất.

error(): Trả về mô tả lỗi cuối cùng cho cuộc gọi hàm gần đây nhất.

error\_list(): Trả về danh sách các lỗi cho cuộc gọi hàm gần đây nhất.

fetch\_all(): Truy xuất tất cả các hàng kết quả dưới dạng mảng kết hợp, mảng số hoặc cả hai.

fetch\_array(): Truy xuất một hàng kết quả dưới dạng mảng kết hợp, mảng số hoặc cả hai.

fetch\_assoc(): Truy xuất một hàng kết quả dưới dạng mảng kết hợp.

fetch\_field(): Trả về trường tiếp theo trong tập kết quả dưới dạng đối tượng.

fetch\_field\_direct(): Trả về meta-data cho một trường duy nhất trong tập kết quả dưới dạng đối tượng.

fetch\_fields(): Trả về một mảng đối tượng đại diện cho các trường trong tập kết quả.

fetch\_lengths(): Trả về độ dài của các cột của hàng hiện tại trong tập kết quả.

fetch\_object(): Trả về hàng hiện tại của tập kết quả dưới dạng đối tượng.

fetch\_row(): Truy xuất một hàng từ tập kết quả và trả về dưới dạng mảng đánh số.

field\_count(): Trả về số cột cho truy vấn gần đây nhất.

field\_seek(): Đặt con trỏ trường đến vị trí trường được chỉ định.

get\_charset(): Trả về một đối tượng bảng mã ký tự.

get\_client\_info(): Trả về phiên bản thư viện client MySQL.

get\_client\_stats(): Trả về thống kê về client cho từng quá trình.

get\_client\_version(): Trả về phiên bản thư viện client MySQL dưới dạng số nguyên.

get\_connection\_stats(): Trả về thống kê về kết nối của client.

get\_host\_info(): Trả về tên máy chủ MySQL và loại kết nối.

get\_proto\_info(): Trả về phiên bản giao thức MySQL.

get\_server\_info(): Trả về phiên bản máy chủ MySQL.

get\_server\_version(): Trả về phiên bản máy chủ MySQL dưới dạng số nguyên.

info(): Trả về thông tin về truy vấn thực hiện cuối cùng.

init(): Khởi tạo MySQLi và trả về một tài nguyên sử dụng cho real\_connect().

insert\_id(): Trả về ID được tạo tự động từ truy vấn cuối cùng.

kill(): Yêu cầu máy chủ để kết thúc một luồng MySQL.

more\_results(): Kiểm tra xem có kết quả nào khác từ một truy vấn đa kết quả hay không.

multi\_query(): Thực hiện một hoặc nhiều truy vấn trên cơ sở dữ liệu.

next\_result(): Chuẩn bị kết quả tiếp theo từ multi\_query().

options(): Thiết lập các tùy chọn kết nối bổ sung và ảnh hưởng đến hành vi của một kết nối.

ping(): Ping một kết nối đến máy chủ MySQL, hoặc thử kết nối lại nếu kết nối đã bị mất.

poll(): Kiểm tra kết nối.

prepare(): Chuẩn bị một câu lệnh SQL để thực thi.

query(): Thực hiện một truy vấn đối với cơ sở dữ liệu.

real\_connect(): Mở một kết nối mới đến máy chủ MySQL.

real\_escape\_string(): Escapes các ký tự đặc biệt trong một chuỗi để sử dụng trong một câu lệnh SQL.

real\_query(): Thực hiện một truy vấn SQL duy nhất.

reap\_async\_query(): Trả kết quả từ một truy vấn SQL bất đồng bộ.

refresh(): Làm mới/đổ bảng hoặc bộ đệm, hoặc đặt lại thông tin máy chủ sao lưu.

rollback(): Quay lại giao dịch hiện tại cho cơ sở dữ liệu.

select\_db(): Chọn cơ sở dữ liệu mặc định cho các truy vấn cơ sở dữ liệu.

set\_charset(): Thiết lập bảng mã ký tự mặc định cho client.

set\_local\_infile\_default(): Bỏ định nghĩa người dùng cho lệnh LOAD DATA LOCAL INFILE.

set\_local\_infile\_handler(): Đặt hàm gọi lại cho lệnh LOAD DATA LOCAL INFILE.

sqlstate(): Trả về mã lỗi SQLSTATE cho lỗi.

ssl\_set(): Sử dụng để thiết lập kết nối an toàn sử dụng SSL.

stat(): Trả về trạng thái hệ thống hiện tại.

stmt\_init(): Khởi tạo một câu lệnh và trả về một đối tượng để sử dụng với stmt\_prepare().

store\_result(): Chuyển một tập kết quả từ truy vấn cuối cùng.

thread\_id(): Trả về ID luồng cho kết nối hiện tại.

thread\_safe(): Trả về xem thư viện client có được biên dịch như là thread-safe hay không.

use\_result(): Bắt đầu việc truy xuất một tập kết quả từ truy vấn cuối cùng được thực hiện.

warning\_count(): Trả về số lượng cảnh báo từ truy vấn cuối cùng trong kết nối.

**4.Trong tập tin connect.php, Bạn hãy cho biết ý nghĩa của phương thức die() và cách dùng.**

* Ý nghĩa của phương thức die():

die() là một hàm trong PHP được sử dụng để kết thúc chương trình ngay lập tức và hiển thị một thông báo lỗi tùy chỉnh (nếu được cung cấp).

Nếu không có thông báo lỗi được cung cấp, die() chỉ đơn giản là kết thúc chương trình mà không cần thông báo.

* Cách sử dụng die():

die() thường được sử dụng trong các tình huống cần dừng chương trình ngay lập tức khi có một điều kiện không thỏa mãn hoặc có lỗi xảy ra.

Thông thường, die() được sử dụng để xử lý lỗi và cung cấp thông báo lỗi cho người phát triển hoặc người sử dụng cuối.

**5.Trong tập tin taidulieu\_bang.php có dùng phương thức fetch\_assoc và fetch\_all để lấy dữ liệu từ đối tượng $result. Ngoài 2 phương thức này, các bạn hãy liệt kê các phương thức khác trong mysqli có chức năng tương tự, và hãy chạy thử các phương thức này như các cách hiển thị dữ liệu cách 4, cách 5, cách 6... trong tập tin taidulieu\_bang.php.**

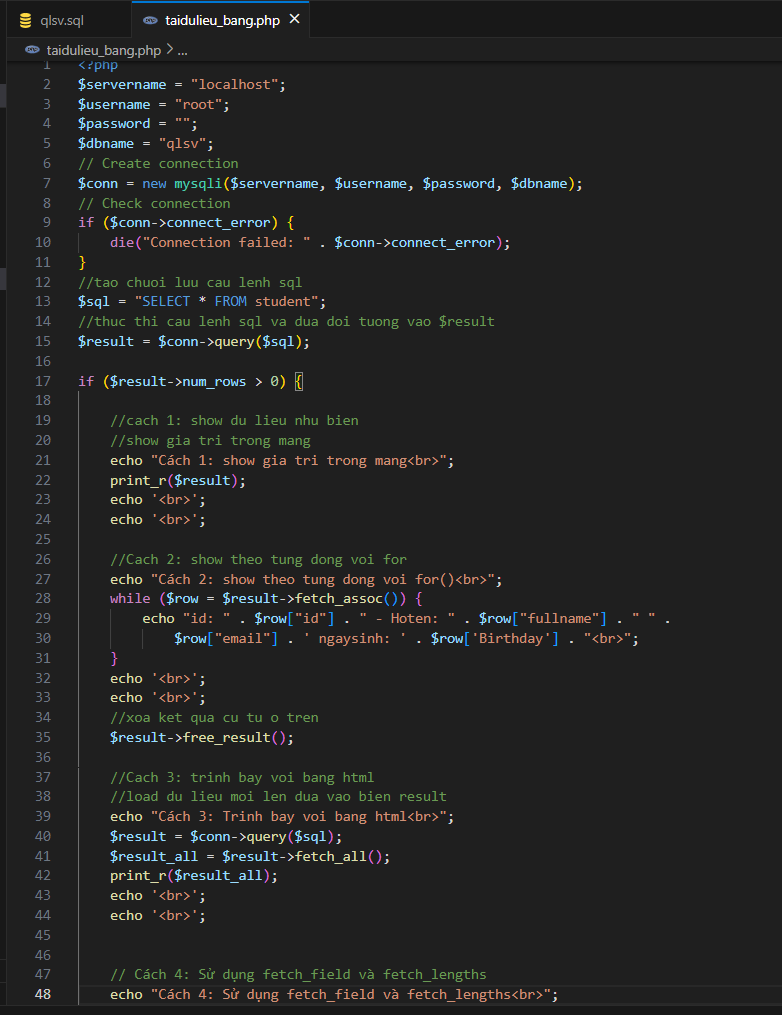
fetch\_array(): Trả về một mảng chứa cả các chỉ số và tên cột.

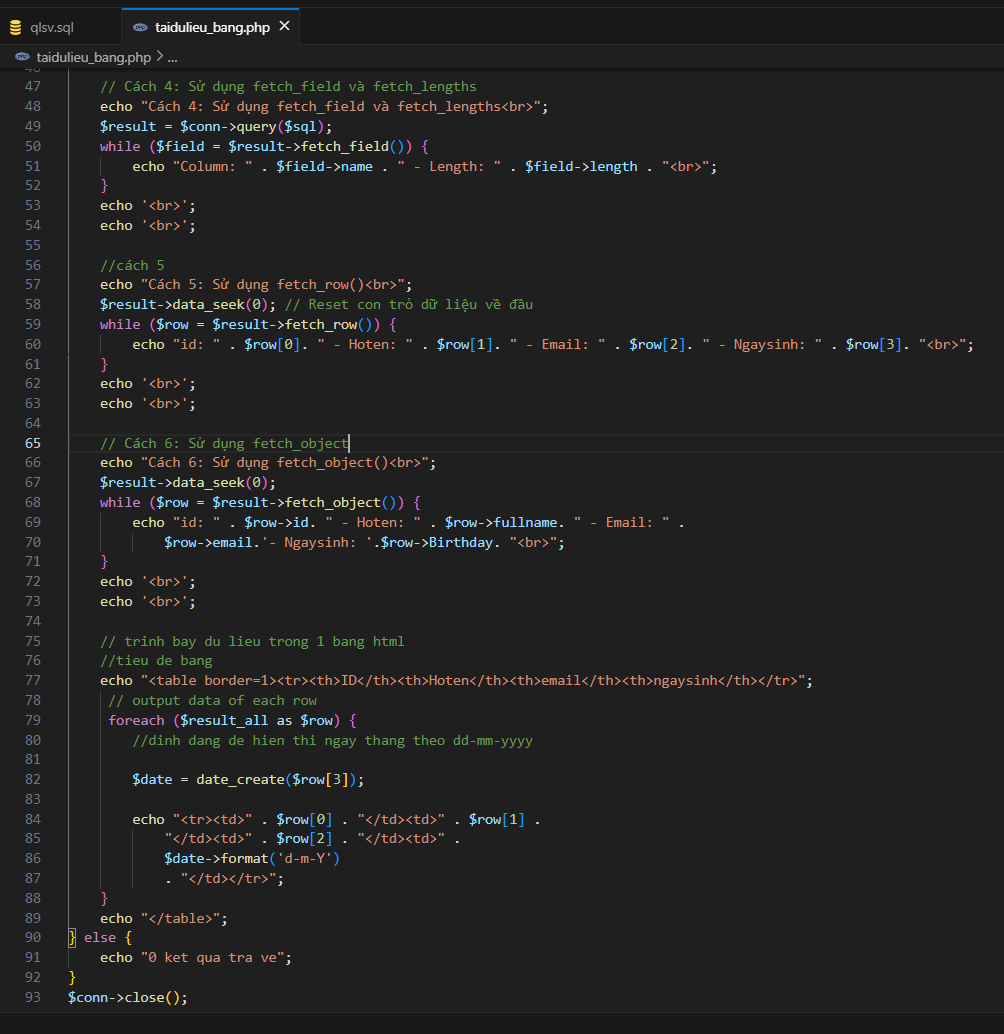
fetch\_row(): Trả về một mảng chỉ chứa các chỉ số của cột.

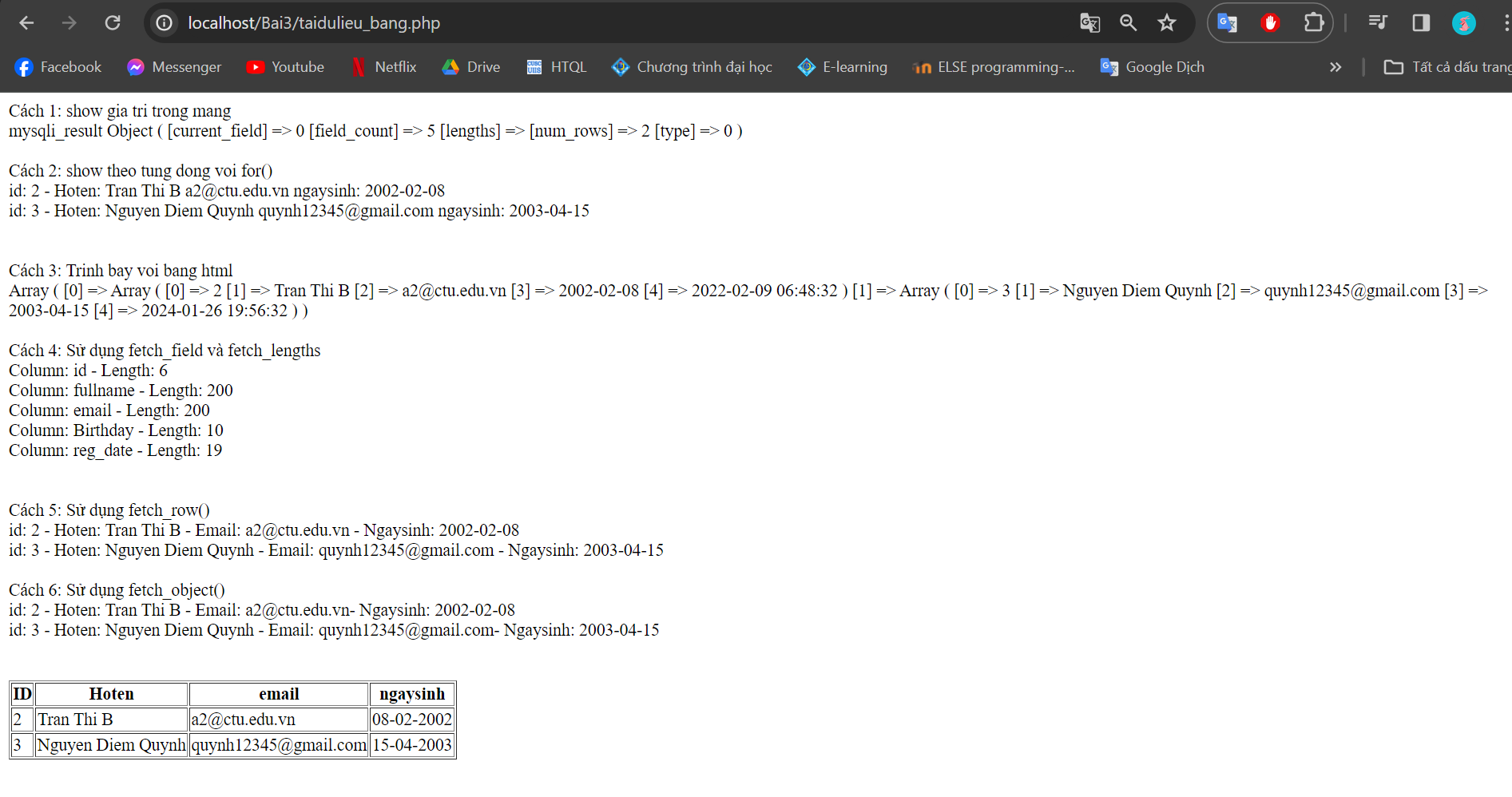
fetch\_object(): Trả về một đối tượng chứa các thuộc tính tương ứng với tên cột.

fetch\_field(): Trả về một đối tượng chứa thông tin về một cột.

fetch\_lengths(): Trả về độ dài của các giá trị trong hàng hiện tại.







**6.Phương thức header trong hàm luu.php để di chuyển đến trang khác sau khi chúng ta thực hiện xong các hành động nào đó (như di chuyển đến trang chủ khi xong hành động sửa/xóa,...).Hãy mô tả chi tiết cách sử dụng, và công dụng của header (Tham khảo 3).**

* Cách Sử Dụng:

header("Location: trang\_muon\_chuyen\_huong.php"): Chỉ định trang mà bạn muốn chuyển hướng đến. Đối số "Location" cung cấp URL hoặc đường dẫn tương đối của trang mới.

Lưu ý rằng việc sử dụng header("Location: ...") phải được gọi trước khi xuất bất kỳ đoạn mã HTML hoặc văn bản nào khác để tránh lỗi "headers already sent".

* Công Dụng:

Chuyển hướng người dùng đến trang khác sau khi đã thực hiện thành công một hành động như thêm, sửa, xóa dữ liệu.

Dùng để chuyển hướng người dùng khi đăng nhập thành công, hoặc khi xảy ra một sự kiện quan trọng.

Thực hiện các tác vụ tương tự trong quá trình xử lý form.

**7.Vào CSDL qlsv, bạn hãy viết lệnh tạo thêm 1 bảng sau để có thêm thông tin về chuyên ngành (kiểu dữ liệu các bạn tự cho sao cho phù hợp):**

**Tạo bảng major(id, name\_major)**

CREATE TABLE major (

id INT PRIMARY KEY,

name\_major VARCHAR(255) NOT NULL

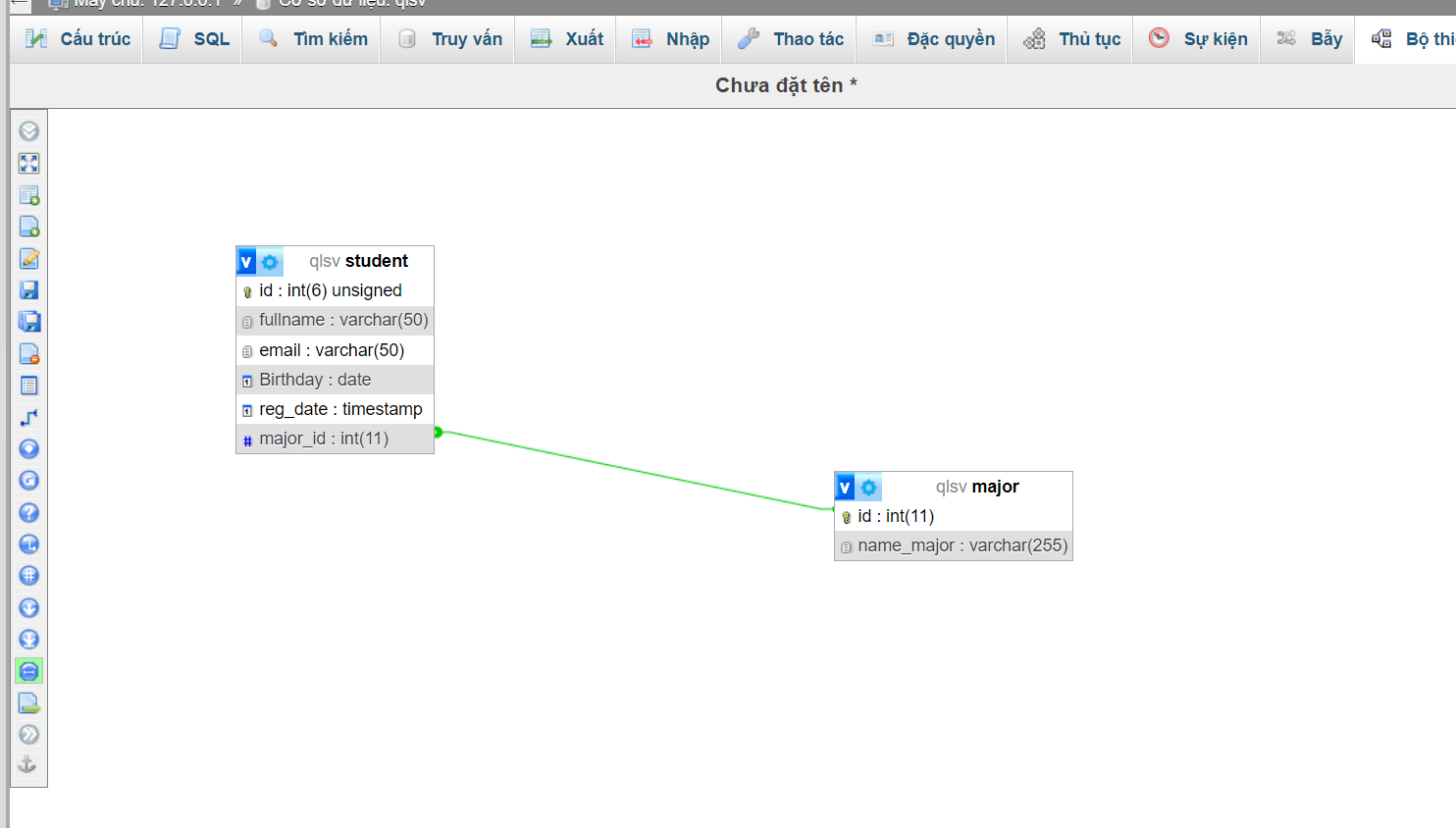
);

**Và thêm cột khóa ngoại major\_id vào bảng student:** student(id, fullname, email, birthday, major\_id)

ALTER TABLE student

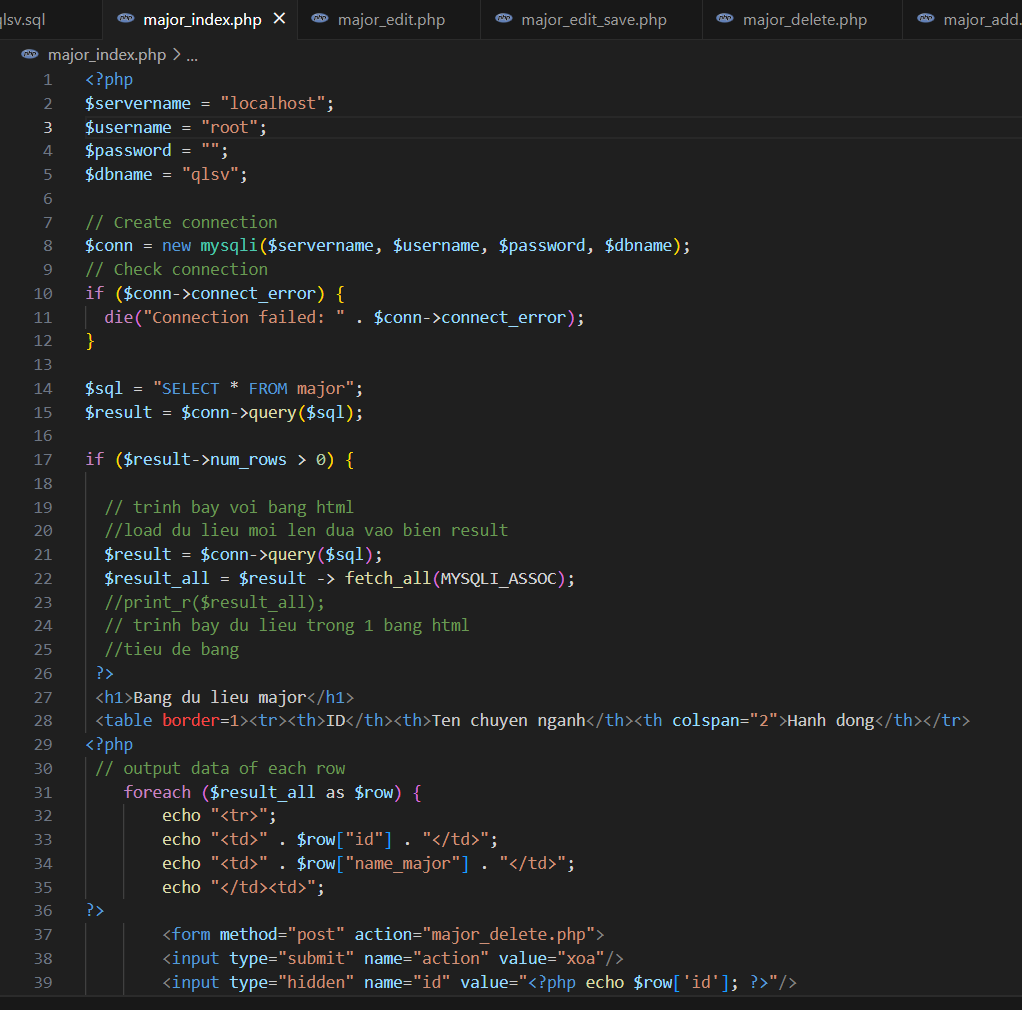
ADD COLUMN major\_id INT,

ADD CONSTRAINT fk\_major\_student FOREIGN KEY (major\_id) REFERENCES major(id);

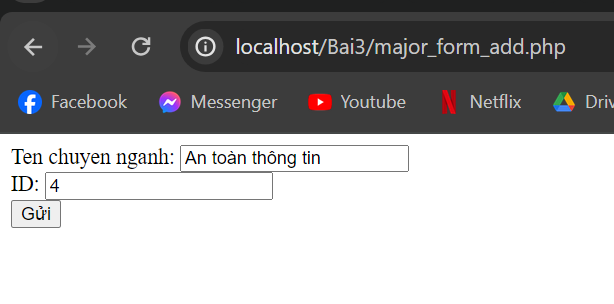
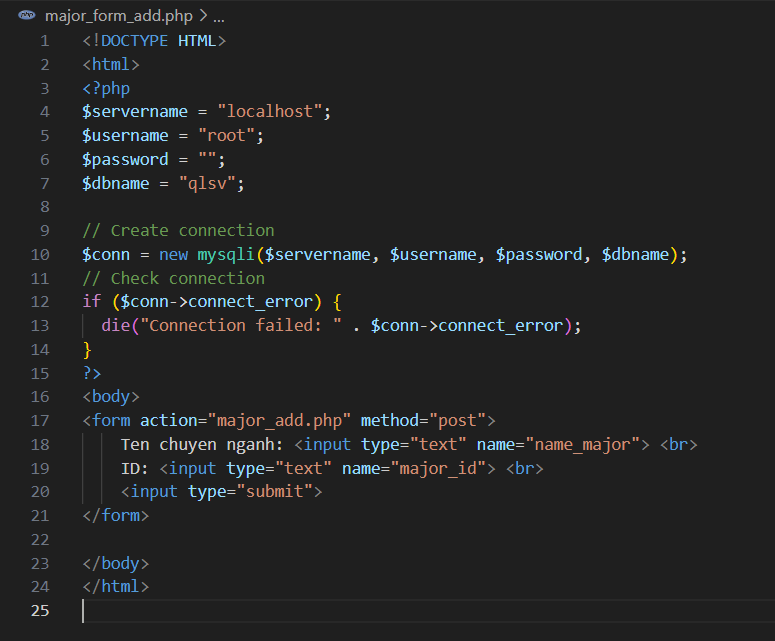


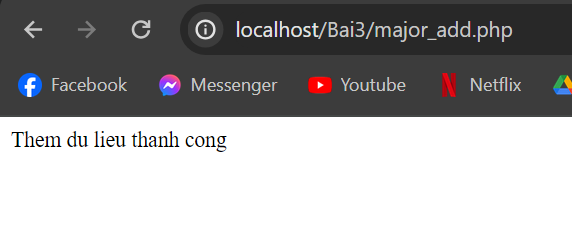
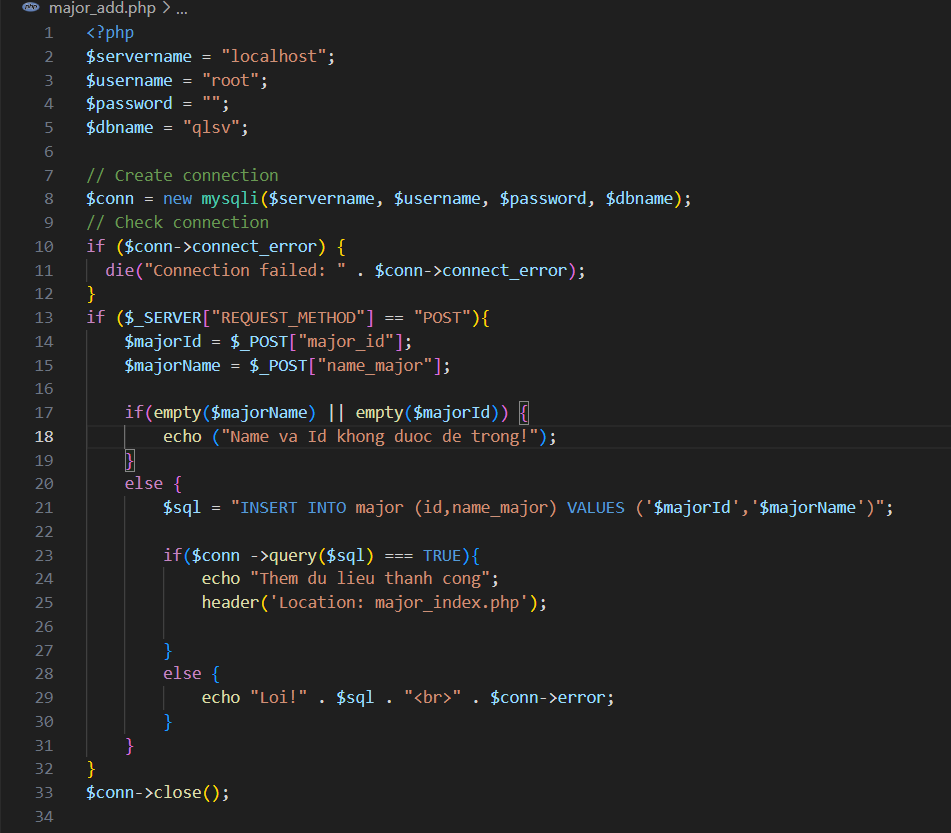
**8.Tạo các tập tin major\_index.php, major\_add.php, major\_edit (và major\_edit\_save.php để thực hiện hành động sửa trên csdl), major.xoa để có các trang hiển thị danh sách, thêm, sửa, xóa đối với bảng major. Gợi ý: Tham khảo taidulieu\_bang1.php, form\_sua.php, sua.php, xoa.php**

major\_index.php

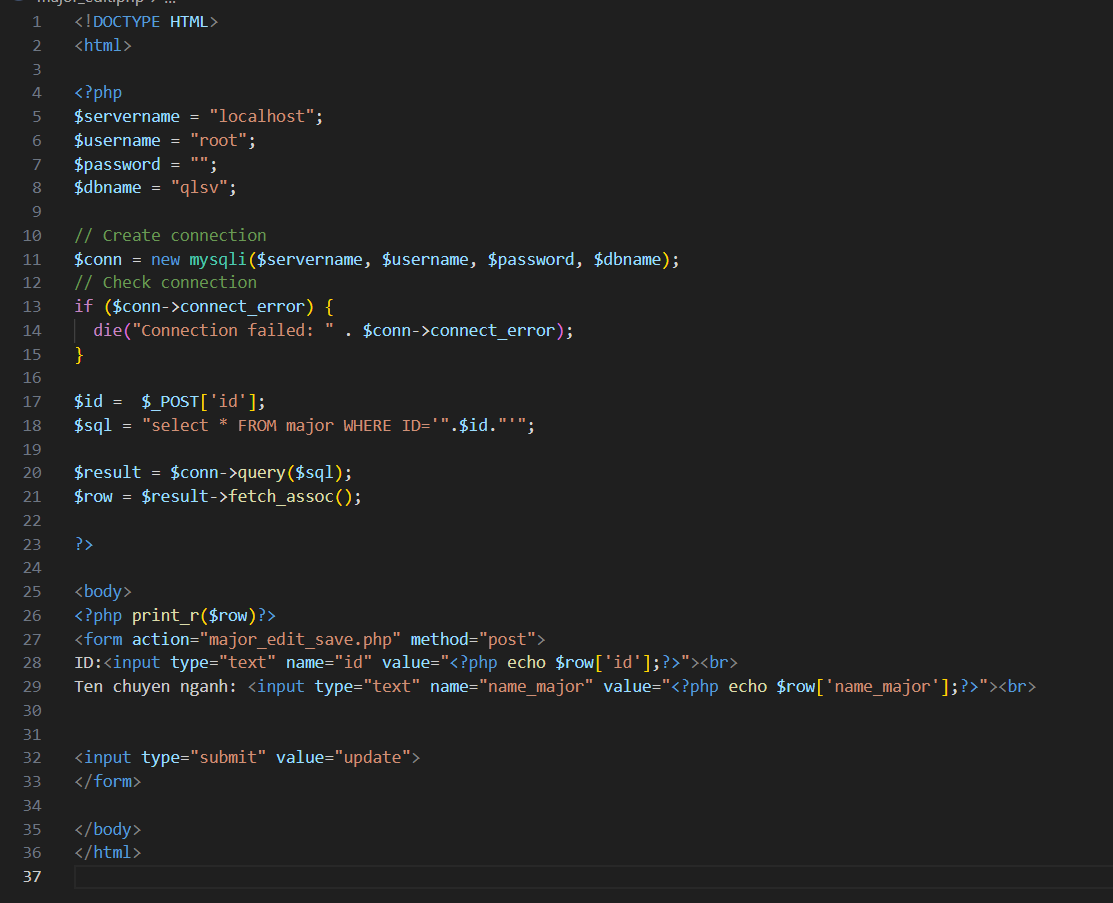
****

****

major\_form\_add.php

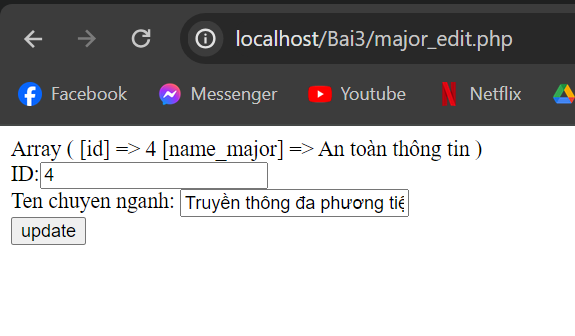
major\_add.php

major\_edit.php



major\_edit\_save.php





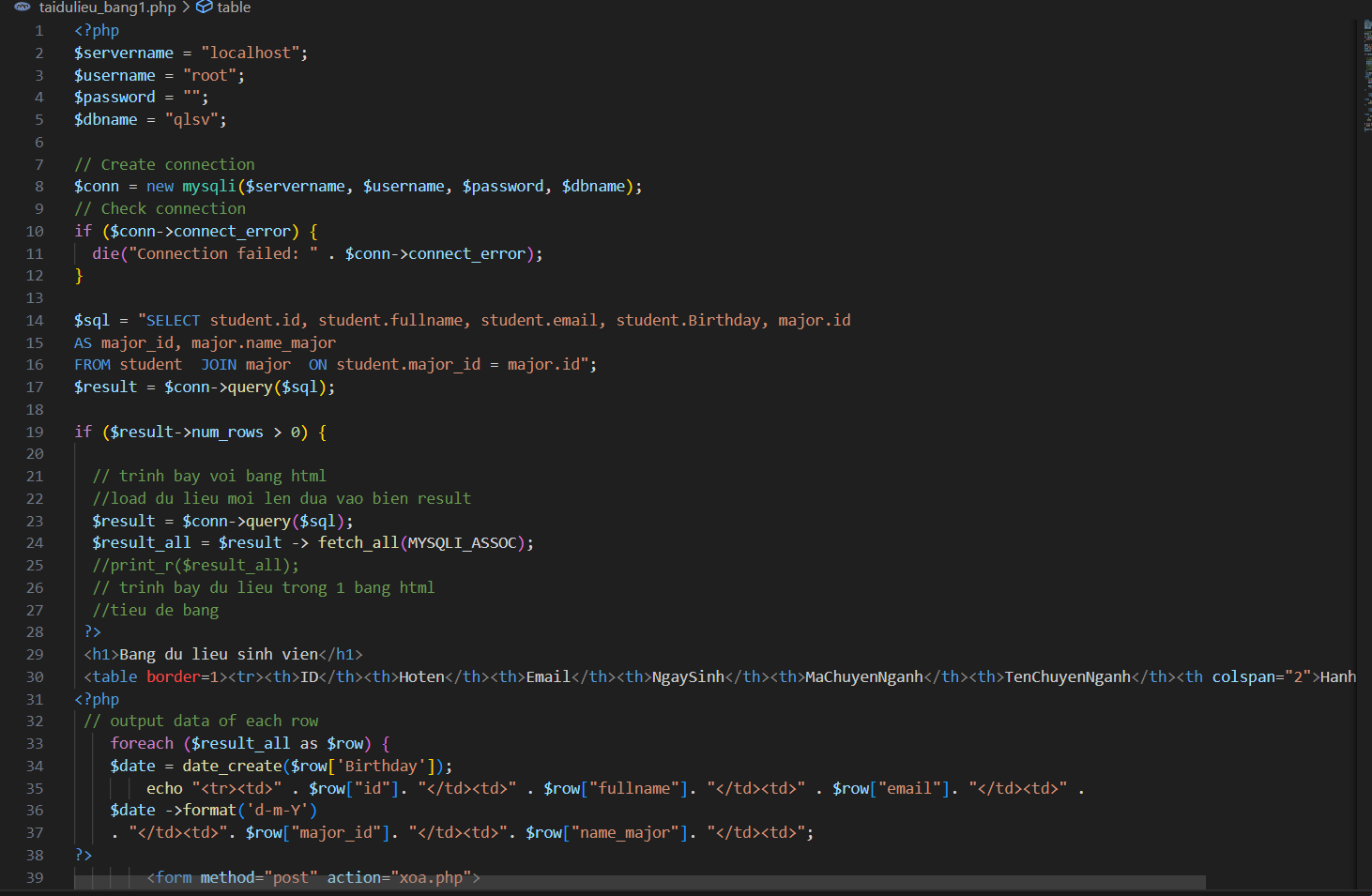
major\_delete.php

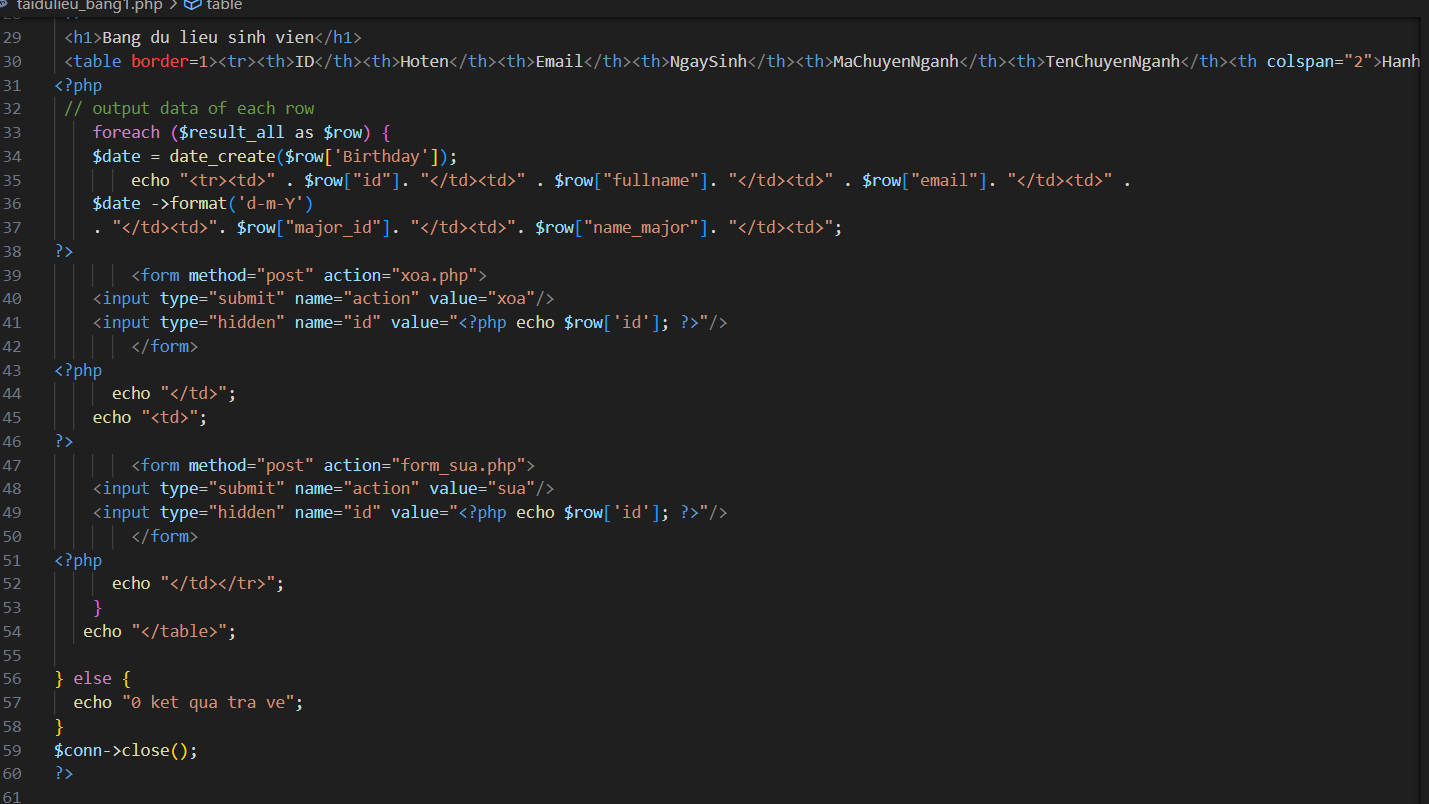






**9. Cập nhật lại taidulieu\_bang1.php thêm hiển thị mã chuyên ngành và tên chuyên ngành tương ứng của sinh viên. Gợi ý: các bạn sửa câu lệnh sql kết nối thêm bảng chuyên ngành.**

****

****

****

**10. Cập nhật tương ứng ở tập tin formnhap.php, form\_sua.php sau khi thêm phần nhập khóa ngoại dạng combobox lấy dữ liệu từ bảng major. Gợi ý: Select dữ liệu từ bảng major và đưa dạng list với tag option https://www.w3schools.com/tags/tag\_option.asp**