**Bài tập thực hành 03**

Lập trình Cơ sở dữ liệu với PHP

**Câu 1**

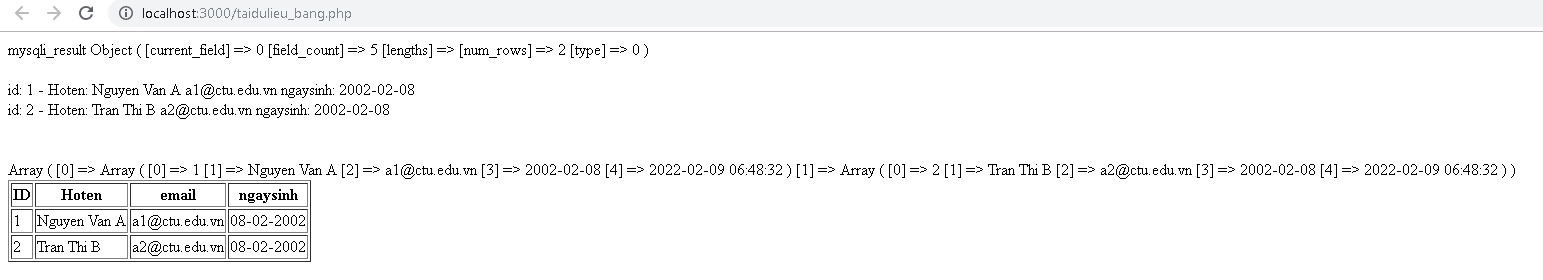
## Tạo chuỗi kết nối đến mysql

## 

## Tạo chuỗi kết nối đến CSDL

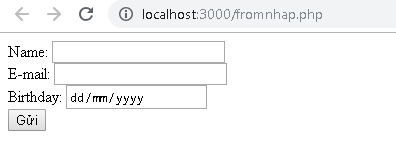
## 

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

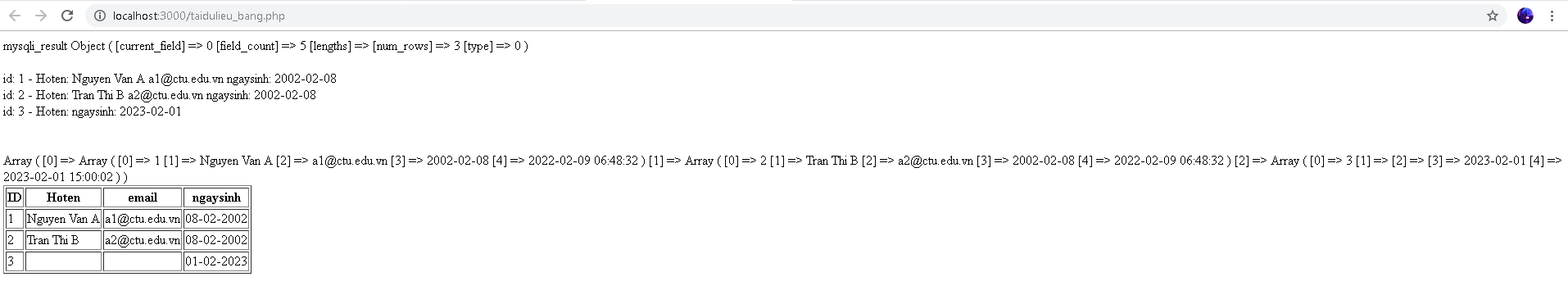


## Add: Form để thêm dữ liệu vào CSDL và chuyển đến trang thể hiện dữ liệu

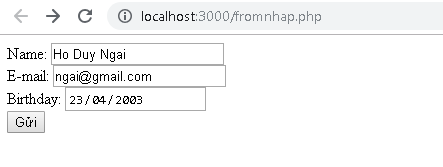
Tạo form giao diện để nhập: formnhap.php

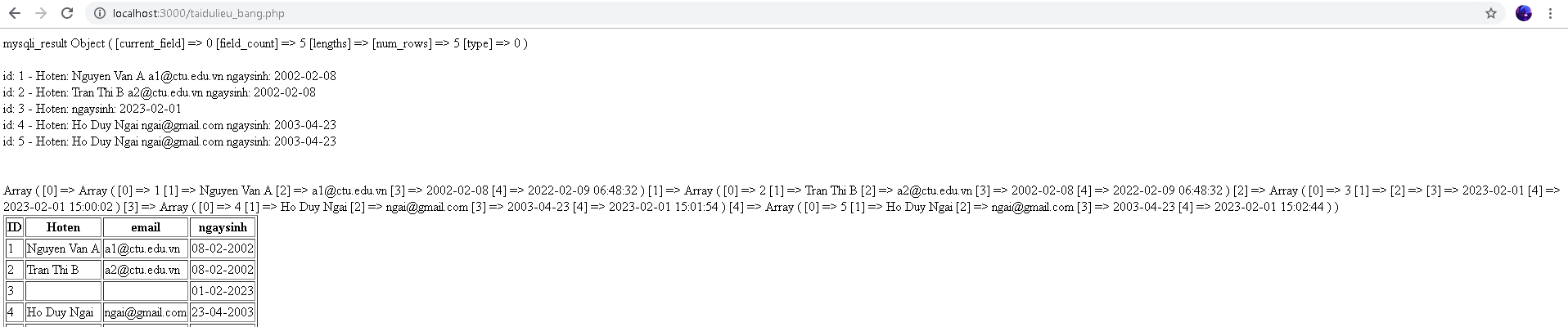


Tạo tập tin php xử lý hành động lưu: luu.php



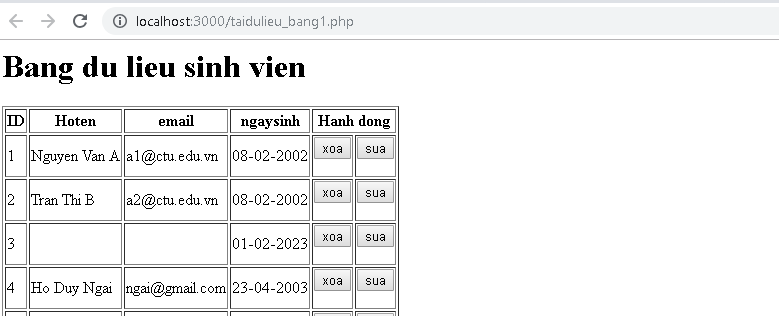
Các bạn chạy formnhap.php trên trình duyệt và thấy kết quả sau:

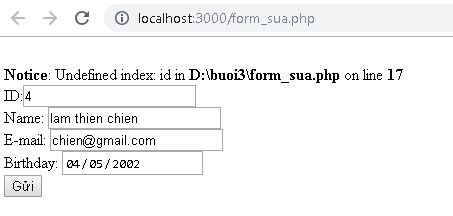




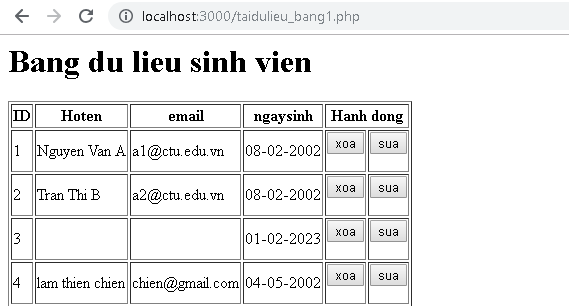
## Edit: Chỉnh sửa dữ liệu từ danh sách

Code để thực hiện giao diện trên trong taidulieu\_bang1.php

Khi chúng ta chọn sửa dữ liệu, tập tin sau (form\_sua.php) sẽ thực thi



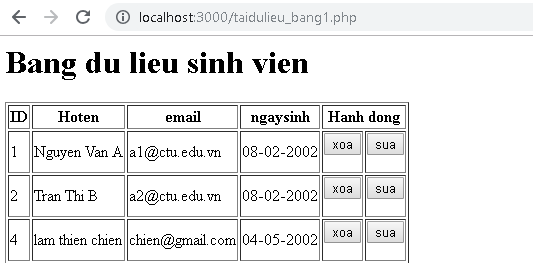
Khi bấm nút submit sẽ gọi qua tập tin sua.php sau:



Sau khi thực hiện sửa xong, sẽ tải trở lại file taidulieu\_bang1.php

## Delete: Xóa dòng dữ liệu trong bảng

Giữ nguyên tập tin như taidulieu\_bang1.php, chúng ta tạo tập tin xoa.php để xử lý khi người dùng bấm vào nút xóa tương ứng với dòng dữ liệu trong bảng



Câu 2:

-SQL:

&amp;lt;?php

$

$serverName = &amp;quot;serverName\sqlexpress&amp;quot;; //serverName\instanceName

$

$connectionInfo = array(&amp;quot;Database&amp;quot;=&amp;gt;&amp;quot;databaseName&amp;quot;,

&amp;quot;UID&amp;quot;=&amp;gt;&amp;quot;username&amp;quot;,

&amp;quot;PWD&amp;quot;=&amp;gt;&amp;quot;password&amp;quot;);

$

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

if($conn) {

echo &amp;quot;Connection established.&amp;lt;br /&amp;gt;&amp;quot;;

}else{

echo &amp;quot;Connection could not be established.&amp;lt;br /&amp;gt;&amp;quot;;

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

}

}

sqlsrv\_close

}

sqlsrv\_

sqlsrv\_close($conn);

?&amp;gt;

-Oracle:

&amp;lt;?php

$conn = oci\_connect(&amp;quot;username&amp;quot;, &amp;quot;password&amp;quot;,

&amp;quot;//localhost/xe&amp;quot;);

if (!$conn) {

$m = oci\_error();

echo $m[&amp;#39;message&amp;#39;], &amp;quot;\n&amp;quot;;

exit;

}

echo &amp;quot;Connected to Oracle!&amp;quot;;

oci\_close

oci\_

oci\_close($conn);

?

?&amp;gt;

``

-Sqlite:

&amp;lt;?php

$db = new SQLite3(&amp;#39;mydb.db&amp;#39;);

if (!$db) {

die(&amp;#39;Connect Error (&amp;#39; . $db-&amp;gt;lastErrorCode() . &amp;#39;) &amp;#39; . $db-

&amp;gt;lastErrorMsg());

}

echo &amp;#39;Connected successfully to SQLite database&amp;#39;;

$

$db-&amp;gt;close();

?&amp;gt;

Câu 3:

Class mysqli trong PHP được sử dụng để kết nối và thực hiện các thao tác trên một cơ sở dữ

liệu MySQL. Dưới đây là một số phương thức chính trong class mysqli:

constructor: Được sử dụng để thiết lập kết nối với cơ sở dữ liệu. Nó có 4 tham số: $host,

$user, $password, và $dbname.

query: Được sử dụng để thực hiện một truy vấn SQL trên cơ sở dữ liệu. Nó có một tham số,

là truy vấn SQL.

close: Được sử dụng để đóng kết nối với cơ sở dữ liệu.

Câu 4:

Phương thức die() trong PHP là một hàm dùng để dừng chạy script và xuất ra một thông báo

lỗi. Nó thường được sử dụng khi một kết nối đến cơ sở dữ liệu không thành công hoặc khi

xảy ra lỗi khác trong quá trình thực thi.

Câu 5:

Các phương thức khác trong mysqli có chức năng tương tự như:

fetch\_array: trả về một mảng kết hợp cả giá trị theo tên cột và số thứ tự.

fetch\_row: trả về một mảng giá trị theo số thứ tự cột.

fetch\_object: trả về một đối tượng tương đương với một dòng dữ liệu.

Câu 6 : Cú pháp của hàm header:

            header ($string, $replace = true, $http\_response\_code = null) {}

Trong đó:

$string: Chuỗi khai báo cho kết quả trả về từ Server, chuỗi này đóng vai trò quan trọng nhất vì nội dung của nó sẽ quyết định header sẽ làm gì.

$replace: Tham số này mặc định là true, nghĩa là định dạng của chuỗi $string sẽ được replace chứ không phải khai báo mới (trường hợp khai báo nhiều header).

$http\_response\_code: Mã code trả về từ Server. VD 404 là not found 301 là chuyển hướng có chủ đích

Header dùng để điều hướng, chuyển hướng trang

Câu 7 :

CREATE TABLE `major` (

`id` int(11) NOT NULL,

`name` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `student` (

`id` int(6) UNSIGNED NOT NULL,

`fullname` varchar(50) NOT NULL,

`email` varchar(50) DEFAULT NULL,

`Birthday` date DEFAULT NULL,

`reg\_date` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`major\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `student` (`id`, `fullname`, `email`, `Birthday`, `reg\_date`, `major\_id`) VALUES

(1, 'Nguyen Thi M', 'a1@ctu.edu.vn', '2002-02-08', '2022-02-08 23:48:32', 0),

(2, 'Tran Thi B', 'a2@ctu.edu.vn', '2002-02-08', '2022-02-08 23:48:32', 0);

ALTER TABLE `major`

ADD PRIMARY KEY (`id`);

ALTER TABLE `student`

ADD PRIMARY KEY (`id`),

ADD KEY `major\_id` (`major\_id`);

ALTER TABLE `student`

MODIFY `id` int(6) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

ALTER TABLE `major`

ADD CONSTRAINT `major\_ibfk\_1` FOREIGN KEY (`id`) REFERENCES `student` (`major\_id`);

COMMIT;

