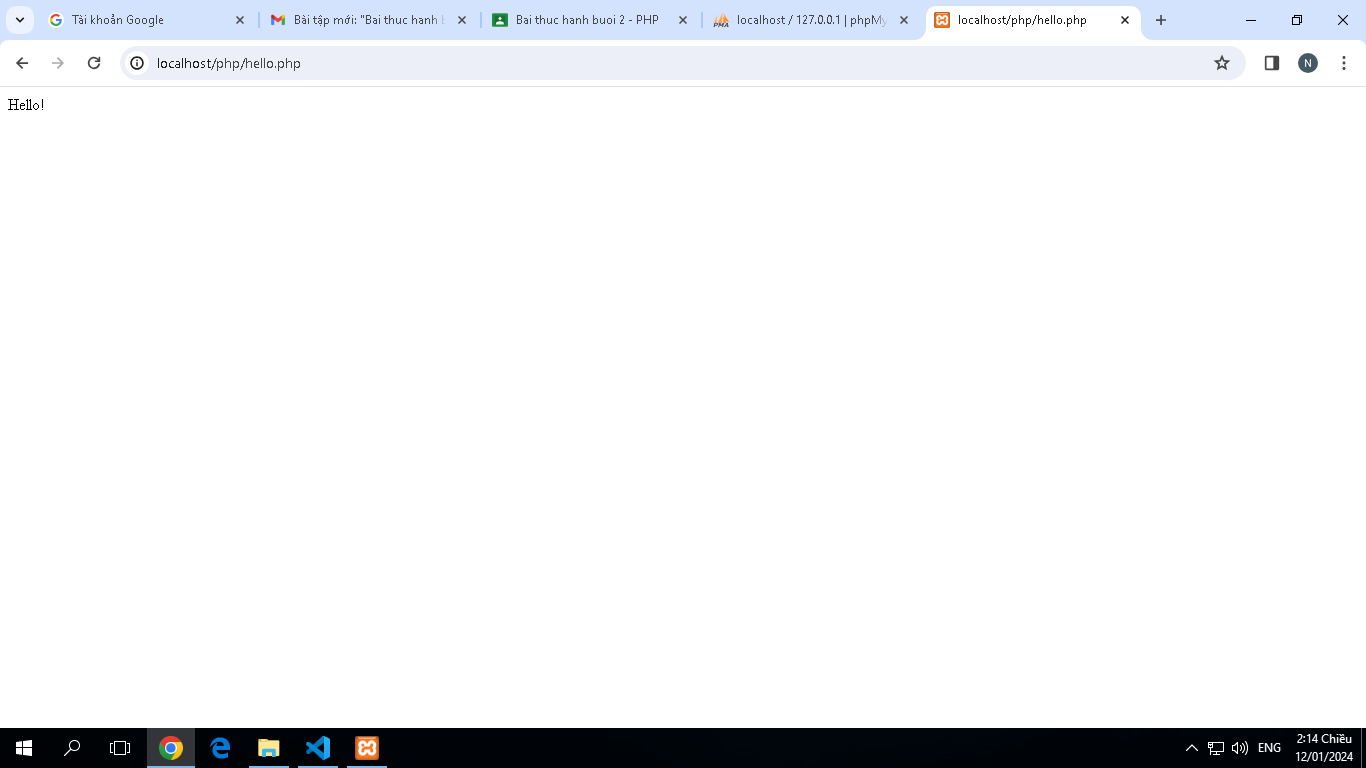
**Bài tập thực hành 02**

**Làm quen với ngôn ngữ PHP**

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

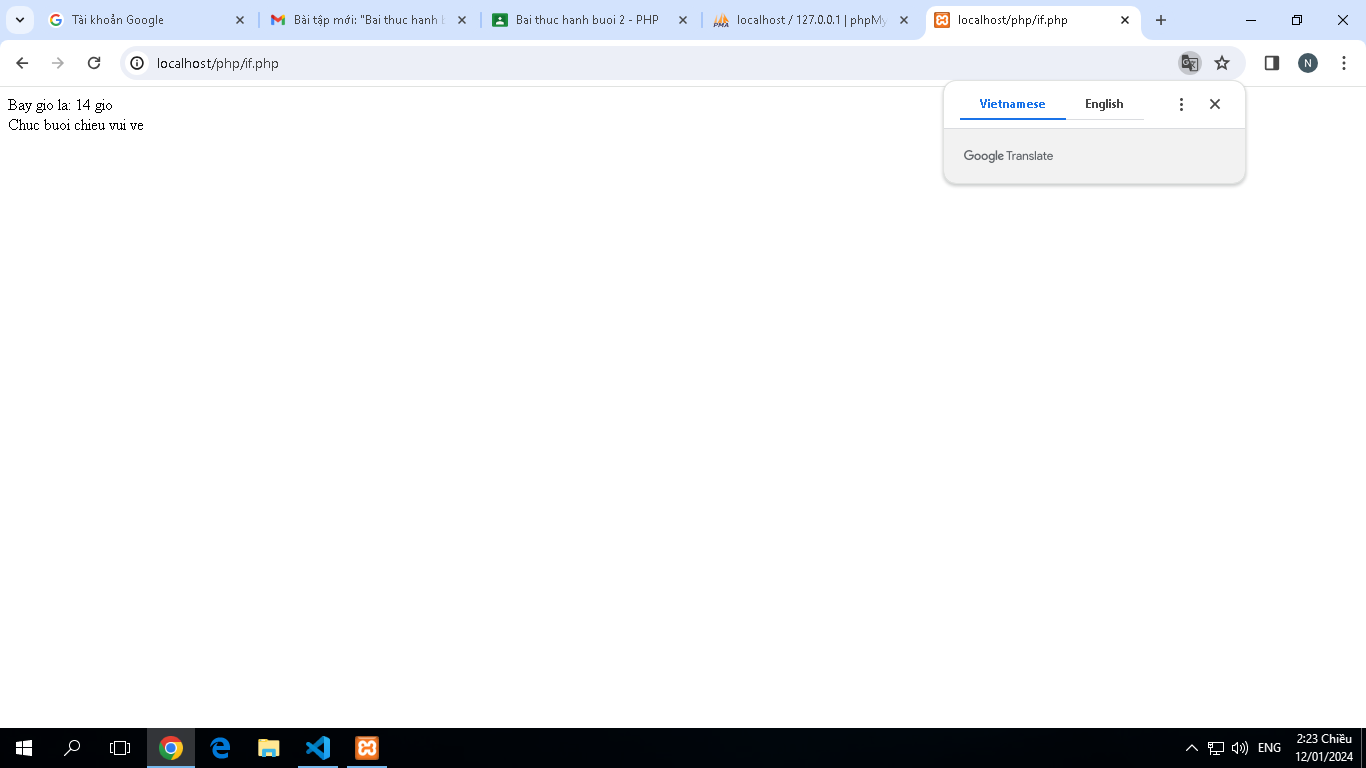
Tạo file php đầu tiên

tạo tập tin hello.php hiển thị chữ “Hello!”

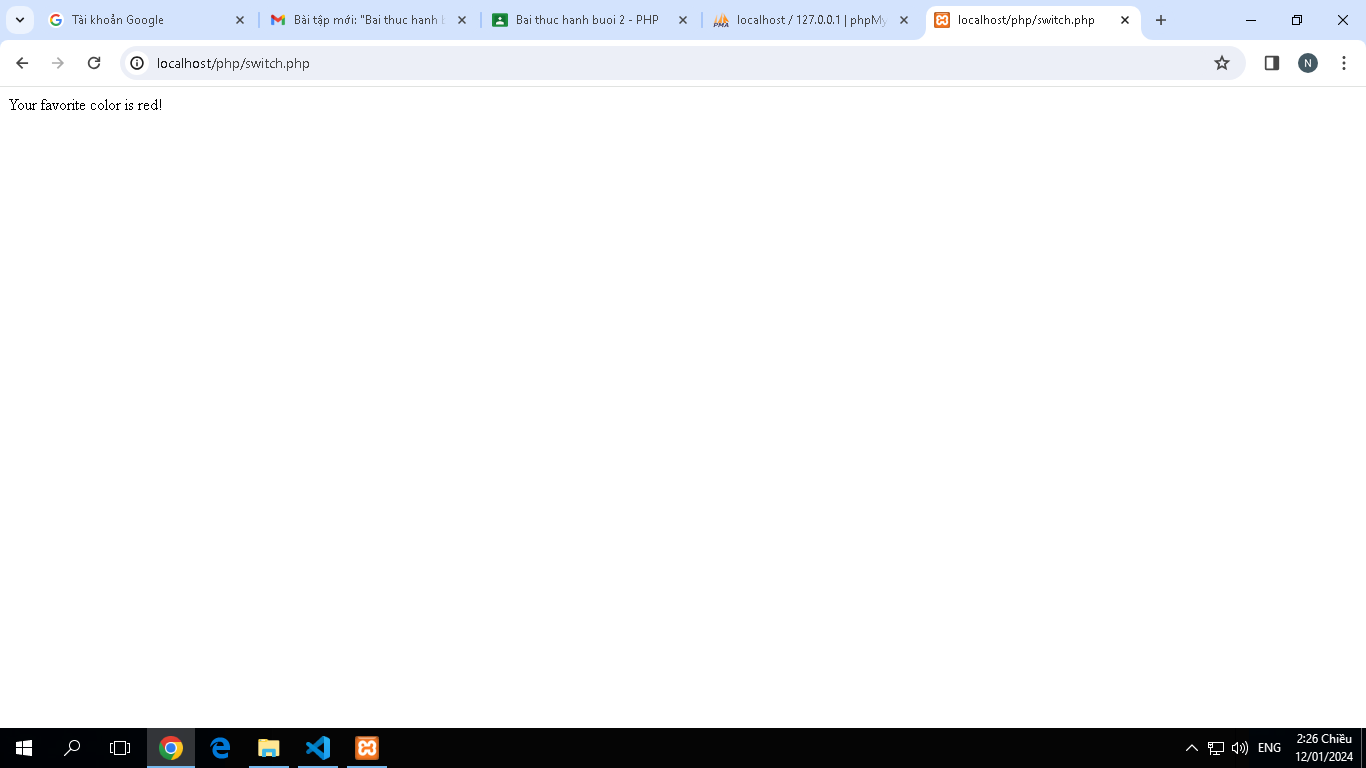


Cấu trúc lập trình trong php

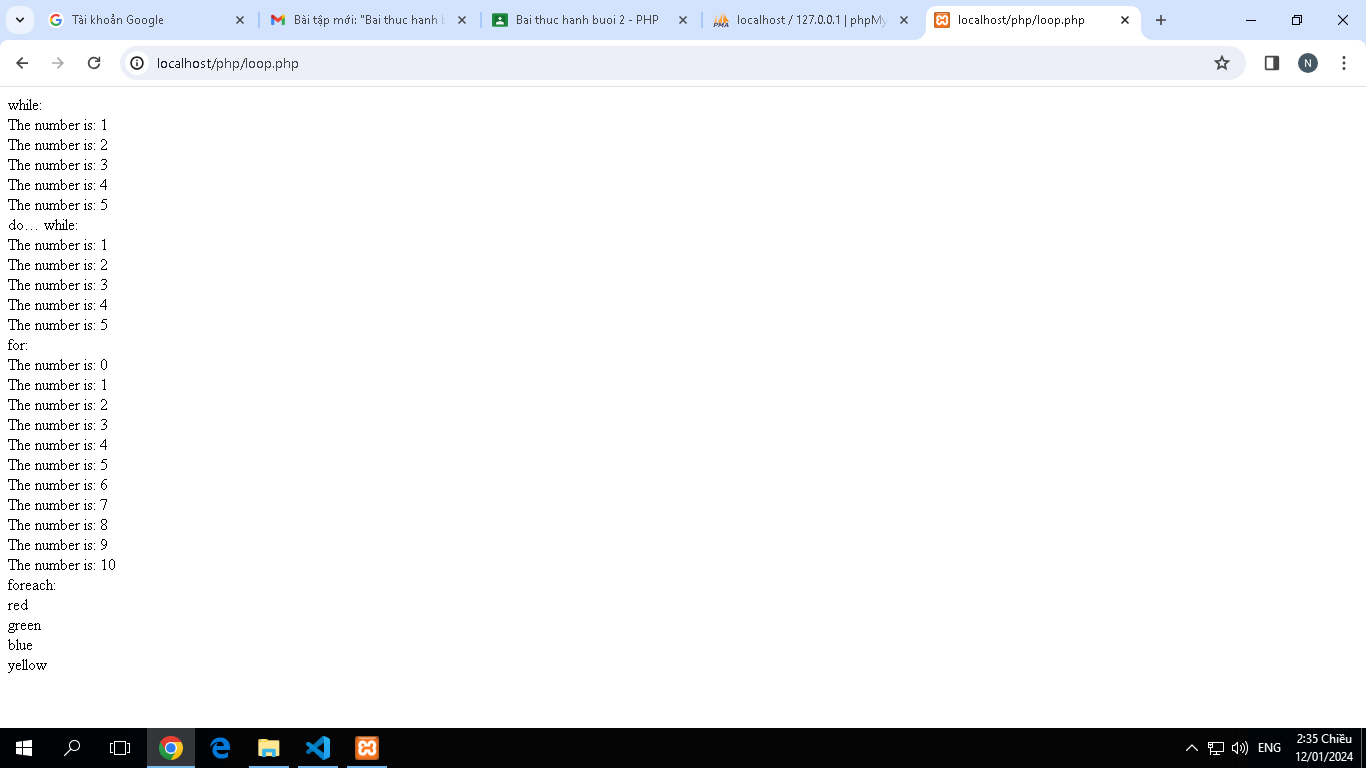
Tạo if.php có nội dung sau:



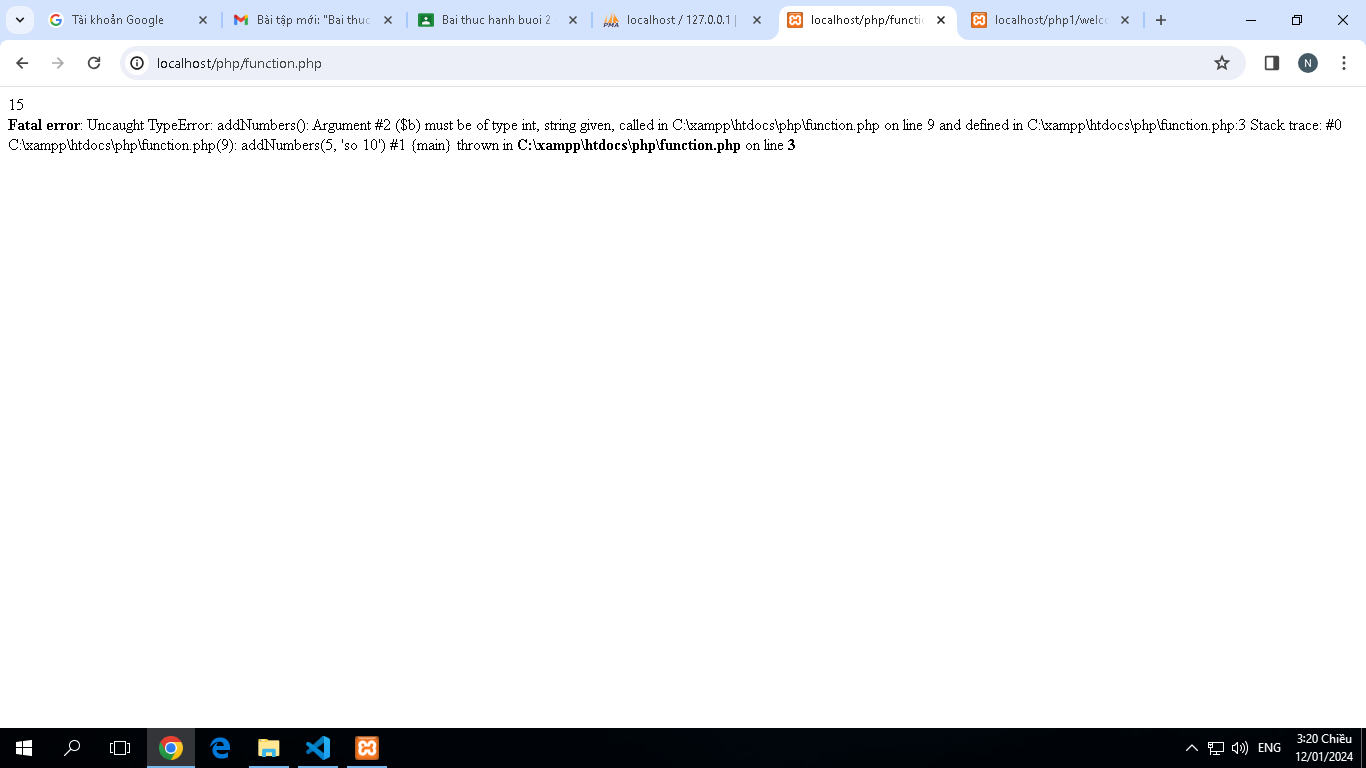
Tạo switch.php có nội dung sau:



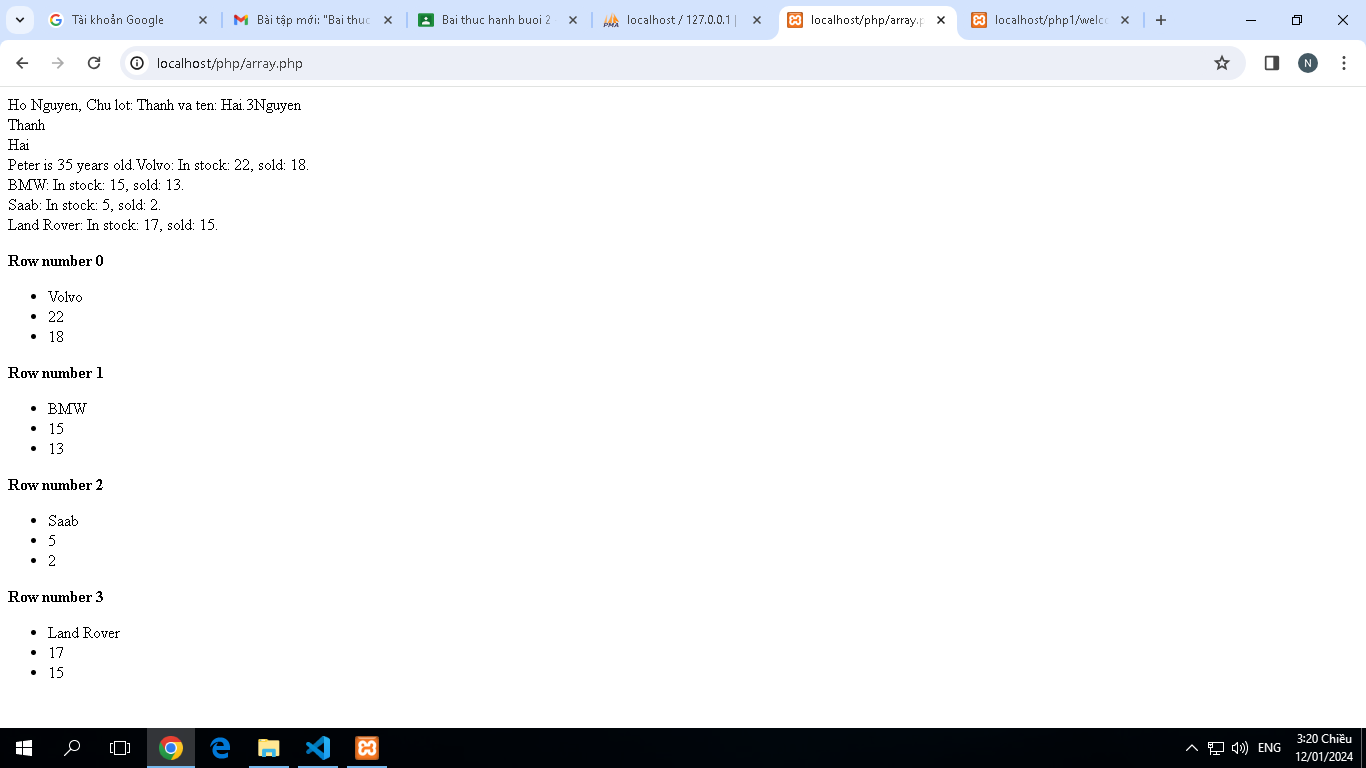
Tạo file loop.php viết các lệnh sau và quan sát kết quả:



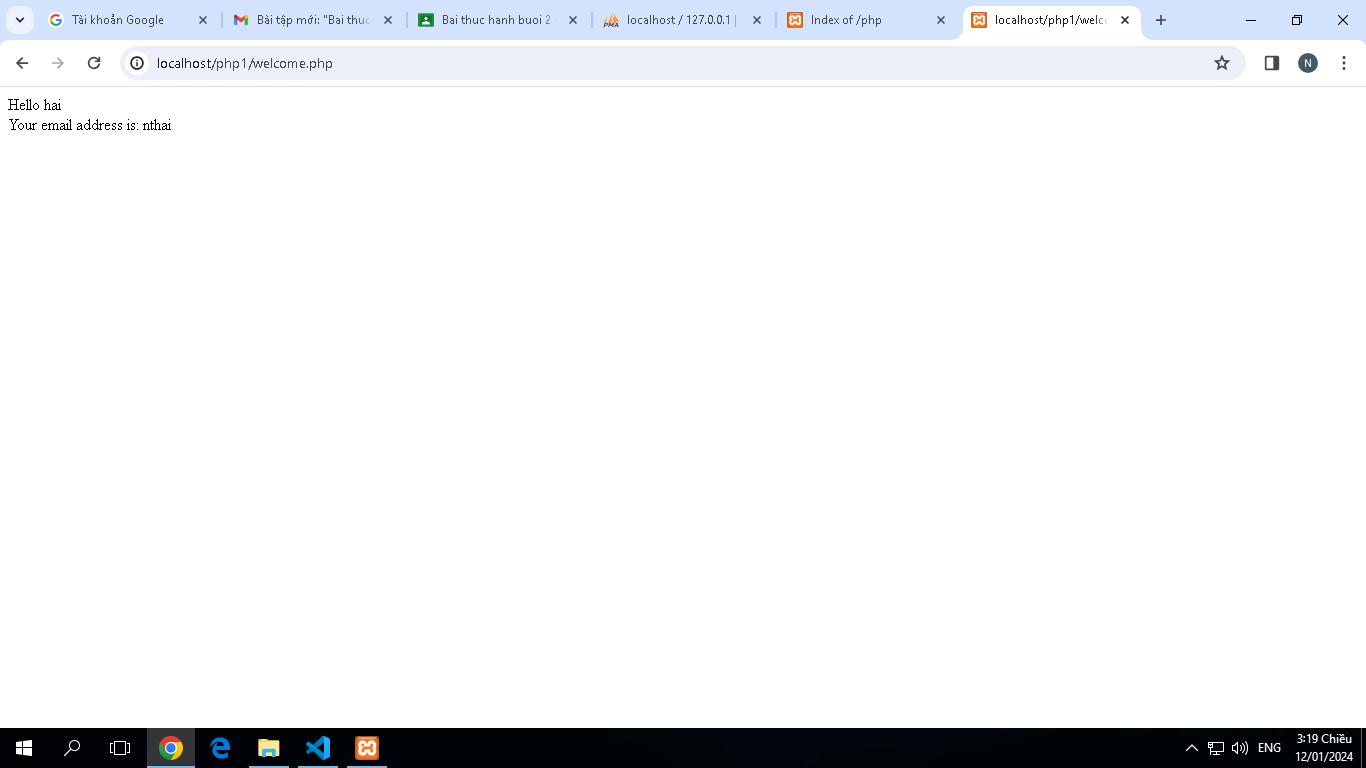
Tạo tập tin function.php với lệnh sau để trả về kết quả cộng 2 biến:



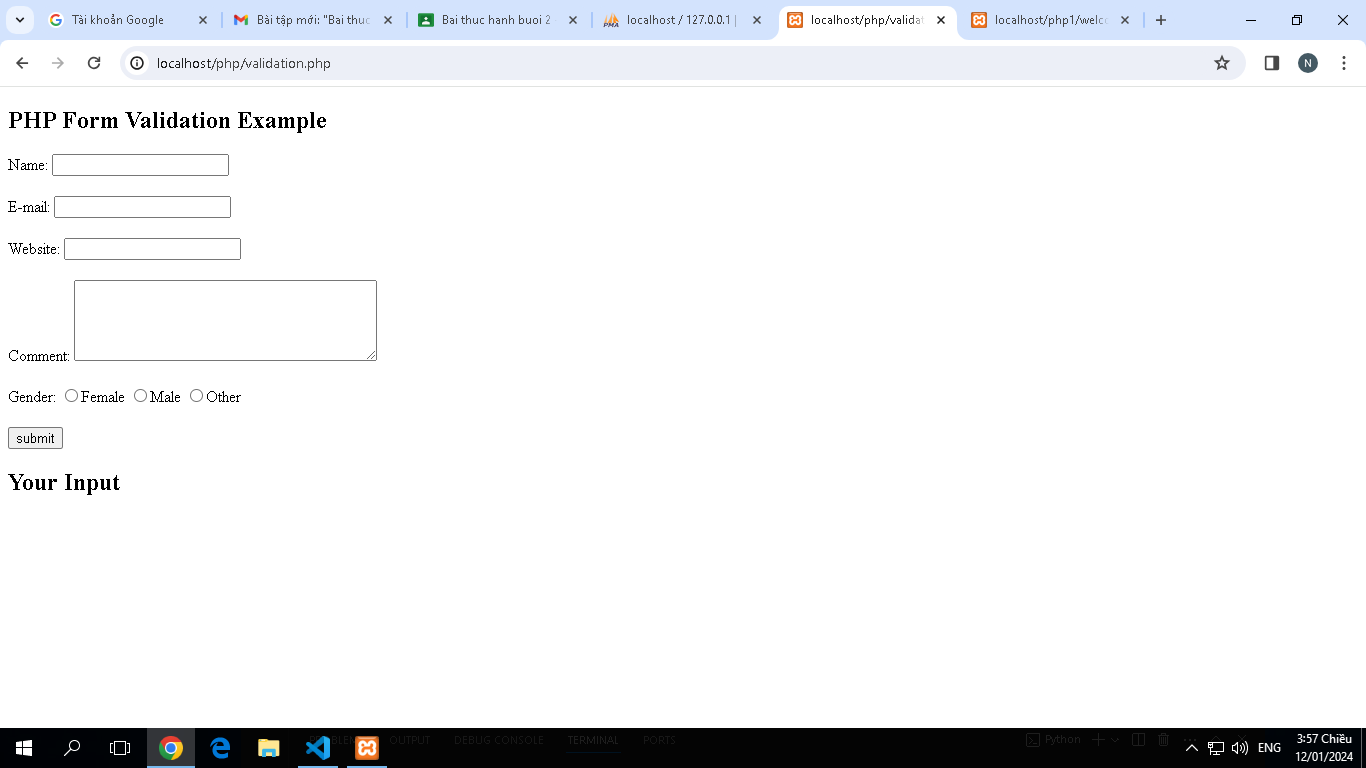
Tạo tập tin array.php với các lệnh sau:



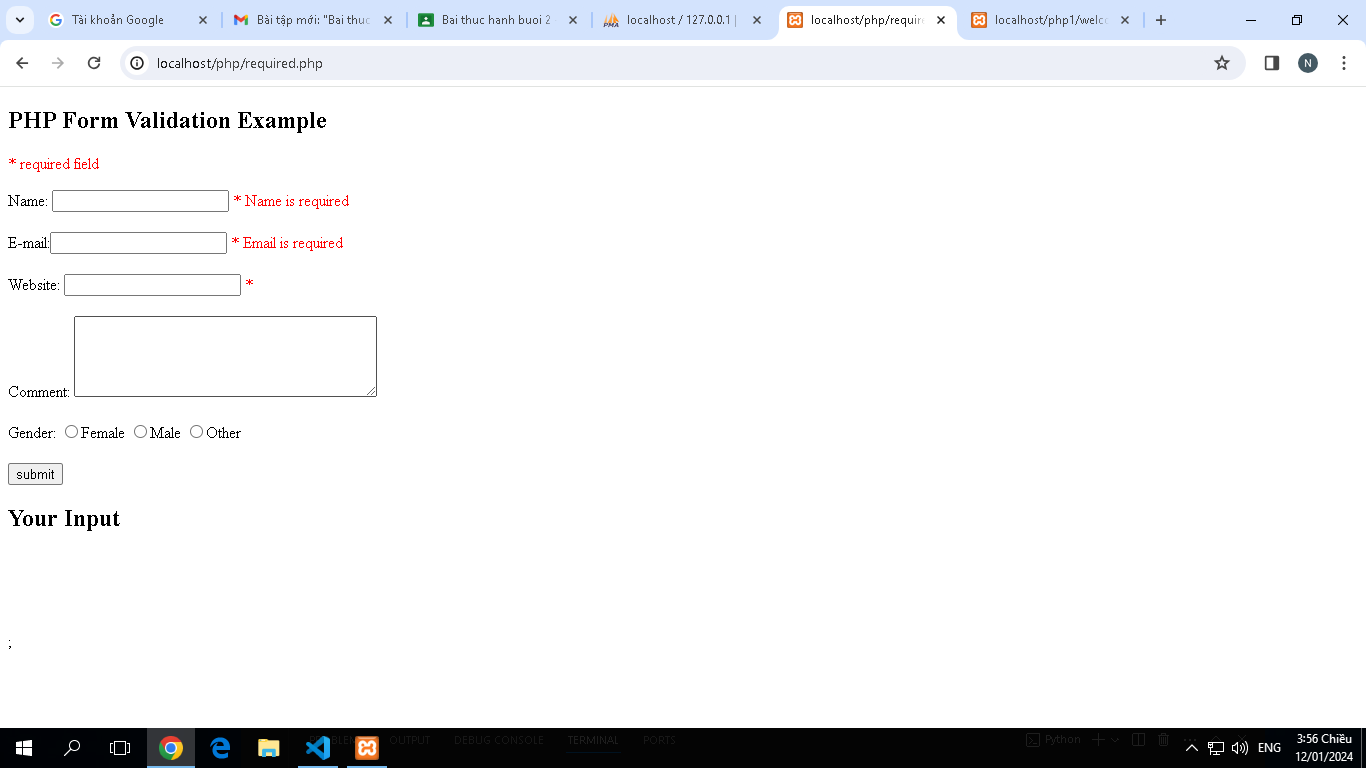
Tạo tập tin welcome.php + Tạo tập tin nhap.php



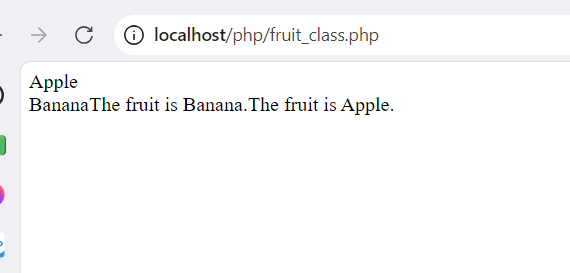
Tạo tập tin sau: validation.php:



Tạo tập tin required.php:



Tạo 1 tập tin fruit\_class.php:



**Câu 2: Hãy cho biết cách khai báo biến, và gán trị cho biến trong PHP. Liệt kê các chú ý khi đặt tên biến. Cho các ví dụ về khai báo biến, và gán giá trị kiểu chuỗi, kiểu số, ngày tháng.**

- Khai báo sử dụng dấu đôla **$** theo sau là tên của biến.

- Cách gán giá trị thì sau khi đặt tên biến thì sử dụng dấu bằng để gán cho nó 1 giá trị nào đó. Nếu là chuỗi thì đặt trong dấu “ ”.

- Các chú ý khi đặt tên biến:

+ Tên biến là một tập hợp gồm một hoặc nhiều ký tự.

+ Tên biến tuyệt đối không được phép chứa các ký tự đặt biệt (Ví dụ như: @, #, !, %, ^, &, ....)

+ Tên biến không được bắt đầu bằng một chữ số.

\*\*các ví dụ:

- Khai báo biến: $tên biến;

- Kiểu chuỗi: $name = "Nhật Thiên Lan";

- Kiểu số: $tuoi = 20;

- Ngày tháng: $datetime = date(‘y-m-d H:i:s’);

**Câu 3: Liệt kê ký hiệu để biết bắt đầu và kết thúc khối lệnh PHP.**

bắt đầu: <?php

kết thúc: ?>

**Câu 4: Liệt kê các ký hiệu có thể được dùng để tạo comment (ghi chú) trong khối lệnh PHP.**

Có 2: // hoặc /\*\*/

**Câu 5: Cho biết có bao nhiêu loại khối lệnh để thực hiện vòng lặp. Cho biết sự khác biệt của foreach cho với các loại khối lệnh vòng lặp khác. Cho biết cách dùng của lệnh “continue” và lệnh “break” trong các vòng lặp.**

## **Vòng lặp:** for, while, **foreach, do-while.**

## **Sự khác biệt của foreach:** Câu lệnh foreach được sử dụng để lặp qua các mảng hoặc các đối tượng có thể lặp được.

**Cách dùng lệnh “continue”:** lệnh **continue** được sử dụng để bỏ qua phần còn lại của vòng lặp hiện tại và tiếp tục với lần lặp tiếp theo. Lệnh **continue** thường được sử dụng trong cấu trúc vòng lặp để bỏ qua một số dòng mã lệnh và chuyển sang lần lặp kế tiếp.

**Cách dùng lệnh “break”:** Câu lệnh break nằm bên trong khối câu lệnh, Nó cung cấp cho bạn quyền kiểm soát và bất cứ khi nào bạn cần thoát khỏi vòng lặp, bạn có thể thoát ra ngoài ngay khi cần. Sau khi thoát khỏi vòng lặp, câu lệnh của vòng lặp sẽ được thực thi.

**Câu 6: Viết 1 tập tin php tên giaithua.php, định nghĩa hàm giai thừa và chạy thử với 10!**

<?php

// Định nghĩa hàm giai thừa

function tinhGiaiThua($n) {

    if ($n == 0 || $n == 1) {

        return 1;

    } else {

        return $n \* tinhGiaiThua($n - 1);

    }

}

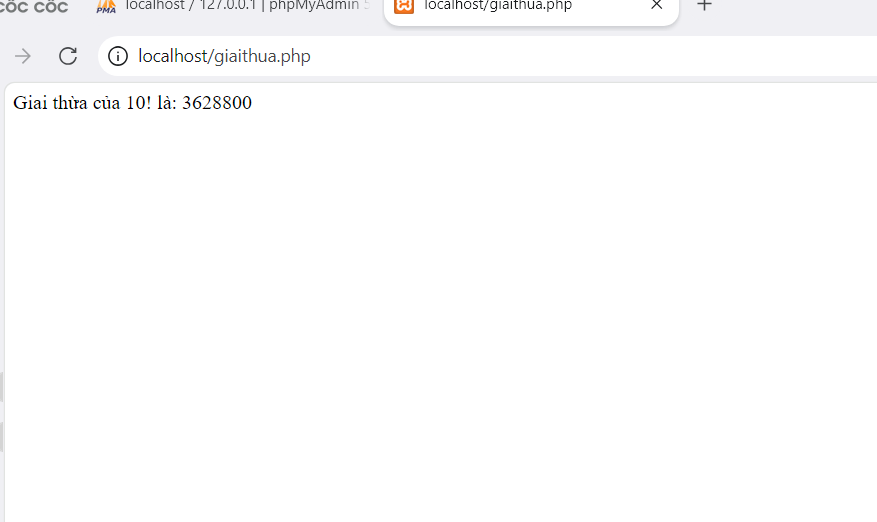
// Chạy thử hàm giai thừa với 10!

$ketQua = tinhGiaiThua(10);

// Hiển thị kết quả

echo "Giai thừa của 10! là: " . $ketQua;

?>



**Câu 7: Viết 1 tập tin cong2mang.php viết hàm xử lý để cộng được 2 mảng có cùng độ dài, nếu 2 mảng không cùng độ dài thì thông báo lỗi. Chạy thử hàm với 2 mảng sau:**

**a={344,224,223,7737,9922,-828}, b={-344,-324,123,773,-9922,828} và cho biết kết quả hiển thị.**

<?php

// Hàm cộng 2 mảng có cùng độ dài

function congHaiMang($a, $b) {

    // Kiểm tra độ dài của 2 mảng

    if (count($a) != count($b)) {

        die("Lỗi: Hai mảng không có cùng độ dài.");

    }

    // Khởi tạo mảng kết quả

    $ketQua = array();

    // Thực hiện cộng từng phần tử tương ứng của hai mảng

    for ($i = 0; $i < count($a); $i++) {

        $ketQua[$i] = $a[$i] + $b[$i];

    }

    return $ketQua;

}

// Mảng a

$a = array(344, 224, 223, 7737, 9922, -828);

// Mảng b

$b = array(-344, -324, 123, 773, -9922, 828);

// Gọi hàm cộng 2 mảng

$ketQuaCong = congHaiMang($a, $b);

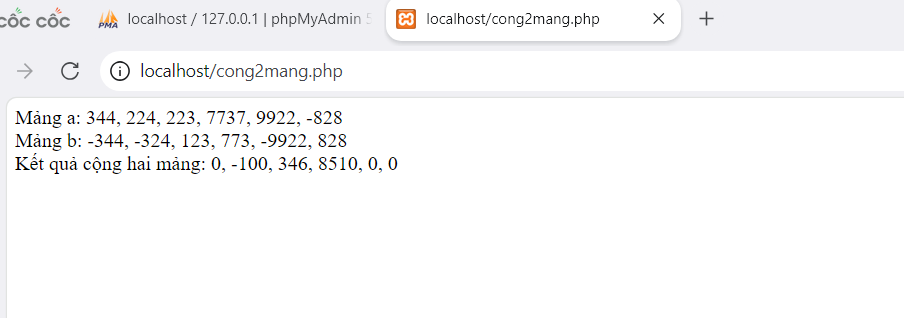
// Hiển thị kết quả

echo "Mảng a: " . implode(', ', $a) . "<br>";

echo "Mảng b: " . implode(', ', $b) . "<br>";

echo "Kết quả cộng hai mảng: " . implode(', ', $ketQuaCong);

?>



**Câu 8: Chạy ví dụ tập tin validation.php, dựa vào link 2 , bạn hãy giải thích những bước diễn ra quá trình này.**

- Đầu tiên là tạo ra các biến rỗng.

**-** Tiếp theo biến toàn cục **$\_SERVER["REQUEST\_METHOD"]** để kiểm tra loại yêu cầu HTTP mà trang web nhận được.

- Các biến như $name, $email, $website, $comment, và $gender được gán giá trị từ dữ liệu được gửi đi từ form sau khi được xử lý qua hàm test\_input.

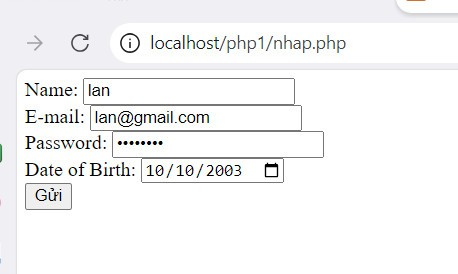
- Hàm này được sử dụng để xử lý dữ liệu đầu vào. Nó thực hiện các bước như **trim** (loại bỏ khoảng trắng ở đầu và cuối chuỗi), **stripslashes** (loại bỏ các dấu gạch chéo), và **htmlspecialchars** (chuyển đổi các ký tự đặc biệt thành các entity HTML).

- Tiếp theo sử dụng phương thức POST để gửi dữ liệu đến chính trang web hiện tại.

- tiếp theo là các dòng để nhập dữ liệu vào sau đó nhập submit để gửi dữ liệu đi.

- cuối cùng là hiển thị kết quả ra.

**Câu 9: Hãy sửa lại tập tin nhap.php, thêm 1 ô để nhập mật khẩu (các ký tự nhập vào sẽ biến thành dấu \*), và 1 ô để nhập ngày tháng năm sinh của bạn. Sau khi nhập xong hiển thị tên, email và mật khẩu, ngày tháng năm sinh đã nhập. Chụp màn hình sau khi bạn đã nhập dữ liệu của chính bạn, chụp màn hình hiển thị dữ liệu nhập.**





**Câu 10: Viết tập tin sinhvien\_class.php để khai báo 1 lớp sinhvien với các thuộc tính: mssv, hoten, ngaysinh với kiểu giá trị phù hợp. Viết các phương thức xây dựng, hủy, gán trị, trả về giá trị của các thuộc tính, viết hàm tính tuổi. Khai báo 1 biến sv có kiểu class\_sinhvien, gán các giá trị cần thiết với thông tin của bạn và thử tính tuổi dựa vào hàm vừa mới xây dựng trong class.**