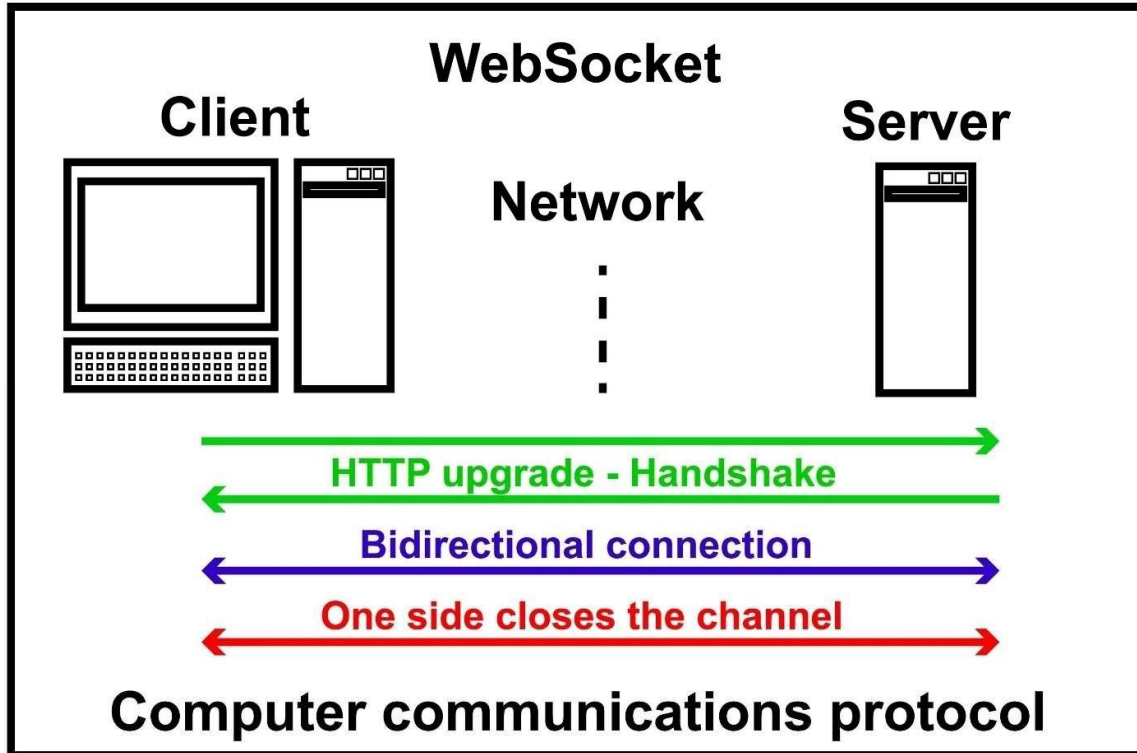


**CHƯƠNG 1: TÌM HIỂU VỀ CÔNG NGHỆ WEB <sup>4</sup>**

**1. Lý thuyết Cốt lõi (Khái niệm)**

Chương 1 giới thiệu về cấu trúc và nguyên tắc làm việc của WWW<sup>5</sup>. Khái niệm cốt lõi bạn phải nắm là Mô hình Client-Server và luồng hoạt động của một yêu cầu HTTP:

1. Web Client (Trình duyệt): Máy của bạn (Chrome, Firefox).
2. Web Server: Một máy tính "ở xa" chứa mã nguồn (PHP, HTML...) của trang web.
3. URL (Địa chỉ): Định danh duy nhất của tài nguyên (ví dụ: <https://www.tlu.edu.vn/>).
4. DNS (Hệ thống tên miền): Giống như "danh bạ điện thoại" của Internet, phân giải tên miền (như [tlu.edu.vn](https://www.tlu.edu.vn/)) thành một địa chỉ IP (như 103.1.238.194) mà máy tính có thể hiểu<sup>6</sup>.
5. Giao thức HTTP: Ngôn ngữ (quy tắc) giao tiếp giữa Client và Server<sup>7</sup>.



**2. Nhiệm vụ Thực hành (BẮT BUỘC)**

Chương này không code PHP, nhưng chúng ta sẽ "thực hành" với công cụ quan trọng nhất của lập trình viên web: Developer Tools (F12).

Kịch bản: Bạn sẽ "khám nghiệm" (autopsy) chính trang web của Trường Đại học Thủy Lợi <sup>8</sup> để xem Client và Server "nói chuyện" với nhau như thế nào, dựa trên lý thuyết HTTP bạn vừa học<sup>9</sup>.

"Code" Khởi đầu (Starter "Code"):

Không có code. "Starter" của bạn là URL: <https://www.tlu.edu.vn/> Các bước

(Tương đương // TODO):



“Tại sao các trang web hiện nay không đóng gói toàn bộ giao diện, mã JavaScript và hình ảnh vào một file duy nhất, mà lại để trình duyệt tải từng tài nguyên riêng lẻ dẫn đến rất nhiều request trong Network?”

## **5. Kết nối Đánh giá (Rất quan trọng)**

Kỹ năng "Đọc Tab Network" bạn vừa thực hành là kỹ năng gỡ lỗi (debug) số 1 của lập trình viên web.

Khi làm Bài tập PHP (20%)<sup>10</sup> hoặc Bài tập lớn (50%)<sup>11</sup> sau này, nếu ứng dụng của bạn báo lỗi (ví dụ: Lỗi 500, 404, hoặc dữ liệu không hiển thị), tab "Network" là nơi đầu tiên bạn phải kiểm tra để xem Web Server (PHP/Laravel) đã thực sự trả về lỗi gì.

### 1. Lý thuyết Cốt lõi (Khái niệm)

Chương này là nền móng. PHP là ngôn ngữ chạy phía Server. Bạn viết code, server thi hành, và chỉ trả về kết quả (thường là HTML) cho trình duyệt. Trình duyệt không bao giờ thấy code PHP của bạn.

Các cú pháp cốt lõi bạn phải nhớ:

- Khối lệnh PHP: Mọi code PHP phải nằm trong `<?php ... ?>`.
- Biến (2.1): Luôn bắt đầu bằng dấu đô-la (\$). Ví dụ: `$ten = "Tuấn";`, `$tuoi = 20;`. Biến PHP không cần khai báo let hay var như JavaScript.
- In ra màn hình: Dùng echo hoặc print. Ví dụ: `echo "Chào bạn $ten";`.
- Nối chuỗi: Dùng dấu chấm (.), không phải dấu cộng (+) như JavaScript. Ví dụ: `echo "Tuổi của bạn là: " . $tuoi;`.
- Cấu trúc điều khiển (2.2): Cú pháp if...else, for, while, switch giống hệt C, Java, và JavaScript.
- Hàm (2.3): Dùng từ khóa function. Ví dụ: `function tinhTong($a, $b) { return $a + $b; }.`
- Kết thúc dòng: Mọi câu lệnh PHP phải kết thúc bằng dấu chấm phẩy (;).

### 2. Nhiệm vụ Thực hành (BẮT BUỘC)

Kịch bản: Bạn sẽ tạo một tệp `sinhvien.php`. Tệp này sẽ khai báo thông tin cơ bản của một sinh viên, sau đó dùng logic if/else để xếp loại và dùng vòng lặp for để in ra một thông điệp.

Code Khởi đầu (Starter Code):

Tạo 1 tệp `chapter2.php` (hoặc tên bất kỳ) trong thư mục `htdocs` của XAMPP và dán code sau: PHP

```
<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

    <title>PHT Chương 2 - PHP Căn Bản</title>

</head>

<body>

    <h1>Kết quả PHP Căn Bản</h1>

    <?php

        // BẮT ĐẦU CODE PHP CỦA BẠN TẠI ĐÂY
```

```
// TODO 1: Khai báo 3 biến

// $ho_ten = "Nguyễn Văn A"; (Thay bằng tên của bạn)

// $diem_tb = 7.5; (Thay bằng điểm bạn muốn)

// $co_di_hoc_chuyen_can = true; (hoặc false)


// TODO 2: In ra thông tin sinh viên

// Dùng lệnh echo để in ra: "Họ tên: $ho_ten", "Điểm: $diem_tb"

// (Lưu ý: Phải in ra cả thẻ <br> để xuống dòng trong HTML)


// TODO 3: Viết cấu trúc IF/ELSE IF/ELSE (2.2)

// Dựa vào $diem_tb, in ra xếp loại:

// - Nếu $diem_tb >= 8.5 VÀ $co_di_hoc_chuyen_can == true => "Xếp loại: Giỏi"

// - Ngược lại, nếu $diem_tb >= 6.5 VÀ $co_di_hoc_chuyen_can == true => "Xếp loại: Khá"

// - Ngược lại, nếu $diem_tb >= 5.0 VÀ $co_di_hoc_chuyen_can == true => "Xếp loại: Trung bình"

// - Các trường hợp còn lại (bao gồm cả $co_di_hoc_chuyen_can == false) =>
"Xếp loại: Yếu (Cần cố gắng thêm!)" // Gợi ý:

Dùng toán tử && (AND)


// TODO 4: Viết 1 hàm đơn giản (2.3)

// Tên hàm: chaoMung()

// Hàm này không có tham số, chỉ cần `echo "Chúc mừng bạn đã hoàn thành PHT
Chương 2!"`


// TODO 5: Gọi hàm bạn vừa tạo

// Gợi ý: Gõ tên hàm và dấu ();


// KẾT THÚC CODE PHP CỦA BẠN TẠI ĐÂY

?>
```

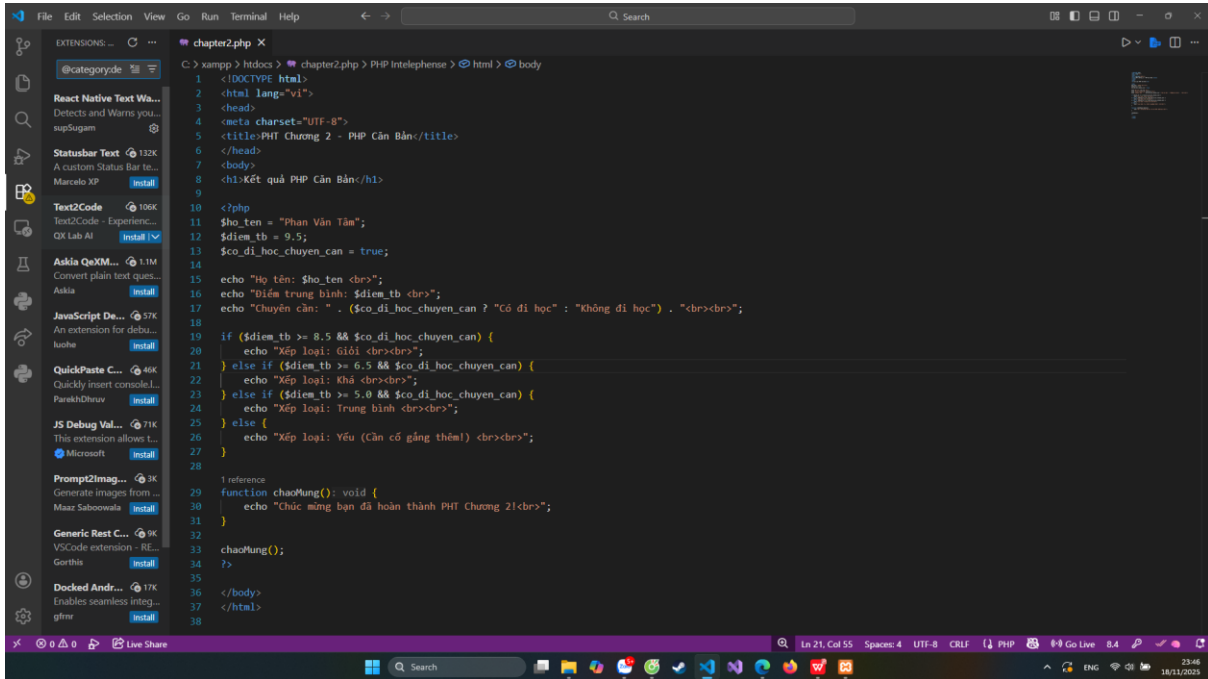
```
</body>
```

```
</html>
```

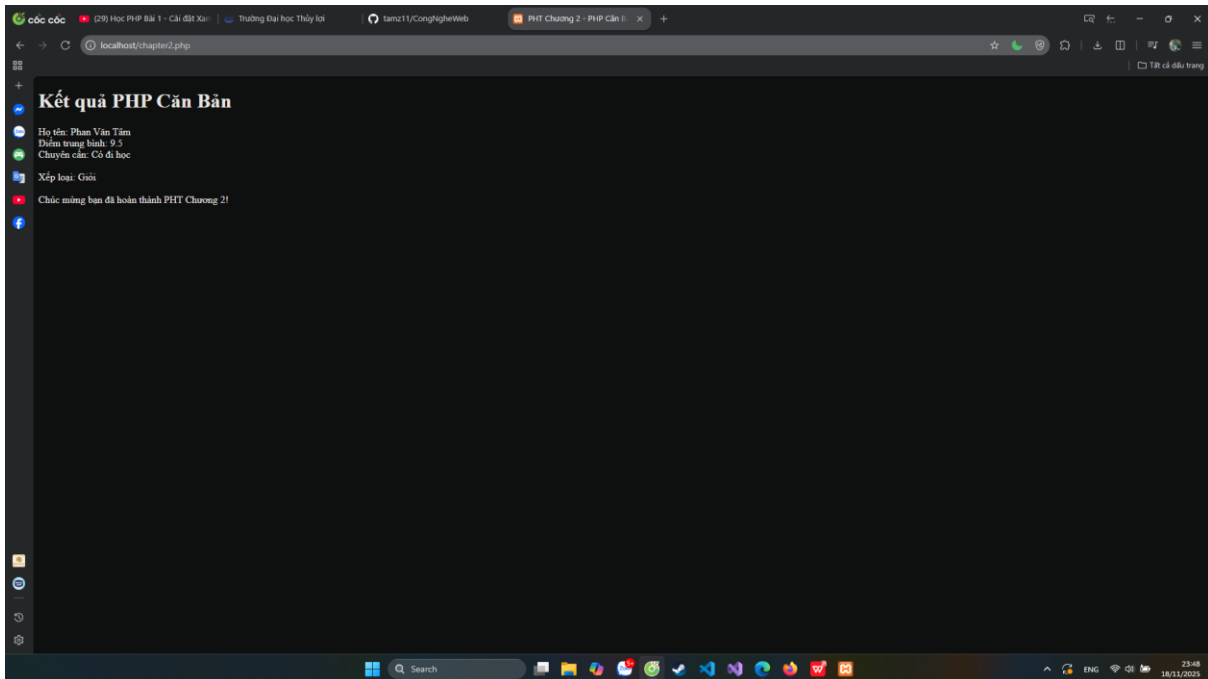
### 3. Yêu cầu Bằng chứng(Proof of Word

Bạn phải nộp lại 2 bằng chứng sau:

- Code đã hoàn thiện: Dán (paste) toàn bộ code của tệp chapter2.php mà bạn đã hoàn thiện.
- Ảnh chụp màn hình Kết quả (Trình duyệt Web): Chạy tệp PHP của bạn trên XAMPP (ví dụ: <http://localhost/chapter2.php>) và chụp ảnh màn hình trình duyệt hiển thị kết quả.



```
1 <!DOCTYPE html>
2 <html lang="vi">
3 <head>
4 <meta charset="UTF-8">
5 <title>PHI Chương 2 - PHP Căn Bản</title>
6 </head>
7 <body>
8 <h1>Kết quả PHP Căn Bản</h1>
9
10 <?php
11 $ho_ten = "Phan Văn Tâm";
12 $diem_tb = 9.5;
13 $co_di_hoc_chuyen_can = true;
14
15 echo "Họ tên: $ho_ten <br>";
16 echo "Điểm trung bình: $diem_tb <br>";
17 echo "Chuyên cần: " . ($co_di_hoc_chuyen_can ? "Có đi học" : "Không đi học") . " <br><br>";
18
19 if ($diem_tb >= 8.5 && $co_di_hoc_chuyen_can) {
20     echo "Xếp loại: Giỏi <br><br>";
21 } else if ($diem_tb >= 6.5 && $co_di_hoc_chuyen_can) {
22     echo "Xếp loại: Khá <br><br>";
23 } else if ($diem_tb >= 5.0 && $co_di_hoc_chuyen_can) {
24     echo "Xếp loại: Trung bình <br><br>";
25 } else {
26     echo "Xếp loại: Yếu (Cần cố gắng thêm) <br><br>";
27 }
28
29 function chàoMung(): void {
30     echo "Chúc mừng bạn đã hoàn thành PHI Chương 2!<br>";
31 }
32
33 chàoMung();
34 ?>
35
36 </body>
37 </html>
```



#### 4. Câu hỏi Phản biện (Bắt buộc)

Sau khi hoàn thành Phần 2 & 3, hãy đặt 01 câu hỏi tư duy.

(Gợi ý: "PHP là ngôn ngữ kịch bản. Vậy sự khác biệt chính giữa 'Lớp và Đối tượng' (2.4) trong PHP so với 'Lớp và Đối tượng' trong một ngôn ngữ biên dịch như Java là gì? Tại sao PHP lại cần đến OOP?").

Trong PHP, mỗi request đều chạy trong môi trường độc lập và toàn bộ biến, đối tượng, cũng như cấu trúc OOP sẽ bị giải phóng ngay sau khi kết thúc. Trong khi đó, các ngôn ngữ chạy lâu dài như Java có thể duy trì đối tượng trong bộ nhớ heap, tận dụng caching hoặc singleton thực sự. Vậy trong bối cảnh vòng đời ngắn của PHP, liệu việc thiết kế kiến trúc theo các mô hình OOP nặng như DDD (Domain Driven Design) hay dùng nhiều layer (Service, Repository, Factory, Interface...) có làm giảm hiệu suất tổng thể do phải khởi tạo lại toàn bộ cấu trúc mỗi request? Hay ngược lại, điều này giúp mã dễ bảo trì hơn và chi phí CPU là không đáng kể so với lợi ích?

#### 5. Kết nối Đánh giá (Rất quan trọng)

Việc làm chủ các cú pháp if/else, function, và echo trong PHT này là bắt buộc.

Đây là những kỹ năng nền tảng nhất để bạn có thể làm Bài tập trên lớp (Phần PHP), chiếm 20% tổng điểm, dự kiến vào Tuần 5. Nếu bạn không thể làm PHT này, bạn sẽ không thể lấy dữ liệu (\$\_POST - Chương 3) hay truy vấn CSDL (Chương 4), vì tất cả đều cần PHP căn bản.