Phỏng vấn thực tập PHP

1,OOP

**1. Hướng đối tượng là gì?**

-Đây là một kỹ thuật lập trình cho phép tạo ra các đối tượng trong code trừu tượng hóa các đối tượng thực tế trong cuộc sống

**2. Class và Object ?**

- Class là tập hợp những đối tượng có cùng điểm chung nào đấy, được tập hợp lại với nhau

-Object: Đối tượng cụ thể được thiết kế từ lớp

**3. Tính chất OOP**

+ Tính đóng gói:

-Cho phép kiểm soát quyền truy cập (và thay đổi) giá trị của thuộc tính hoặc quyền gọi phương thức của đối tượng (hoặc lớp) và đối tượng (hoặc lớp) con.

- Là để phục vụ cho việc đọc mã của người khác được dễ dàng hơn,

Ví dụ trong 1 lớp, nhìn thấy các phương thức private thì họ sẽ biết ngay là phương thức này dùng để thực thi, tính toán cho các phương thức khác, không được dùng để gọi ra ngoài, nên ko cần quan tâm tới nó, hơn nữa nó ngăn chặn việc truy cập, sửa đổi các hàm private, protect ở ngoài phạm vi lớp theo chủ đích của người lập trình

+Tính kế thừa:

-Trong lập trình hướng đối tượng cho phép một lớp (class) có thể kế thừa các thuộc tính và phương thức từ các lớp khác đã được định nghĩa.

+Tính trừu tượng:

-Giúp giảm sự phức tạp thông qua việc tập trung vào các đặc điểm trọng yếu hơn là đi sâu vào chi tiết.

-PHP có abstract class và interface để trừu tượng hóa các đối tượng

+Tính đa hình:

-Thể hiện qua việc có thể định nghĩa một đặc tính, hoặc phương thức cho một loạt các đối tượng gần giống nhau. Nhưng khi thực hiện thì các đối tượng khác nhau sẽ có cách thực hiện khác nhau và cho ra kết quả khác nhau.

-Trong lập trình hướng đối tượng cho phép các lớp con có thể viết lại (override) các thuộc tính hoặc phương thức từ lớp cha.

**4. Sự khác biệt giữa Abtract và Interface?**

- Abstract class: là một class cha cho tất cả các class có cùng bản chất. Bản chất ở đây được hiểu là kiểu, loại, nhiệm vụ của class.  Hai class cùng hiện thực một interface có thể hoàn toàn khác nhau về bản chất. Hiểu đơn giản như một thằng con (child class) chỉ có thể là con của một thằng cha, có tính cách giống cha (abstract class) nó

- Interface: là một chức năng mà bạn có thể thêm vào bất kì class nào. Từ chức năng ở đây không đồng nghĩa với phương thức (hoặc hàm). Interface có thể bao gồm nhiều hàm/phương thức và tất cả chúng cùng phục vụ cho một chức năng. Nếu một class implements nhiều interface mà các interface có những phương thức cùng tên thì sẽ không lỗi nếu các phương thức đó truyền vào số lượng biến bằng nhau. Còn nếu các phương thức đó khác số lượng biến truyền vào thì sẽ sinh ra lỗi "Fatal error".

|  |  |  |
| --- | --- | --- |
|  | Interface: | Abstract class |
|  | Một class có thể hiện thực nhiều interface.(tạm coi là thừa kế) | Không hỗ trợ đa thừa kế |
|  | Mọi phương thức, property đều mặc định là public. | Có thể xác định modifier. |
|  | Mọi phương thức, property của interface cần được hiện thực trong class. | Không cần thiết. |
|  | Không | Có |

**5. Thế nào là một phương thức static,Phân biệt static::method(),Self::method();**

### - Phương thức static.

+ Phương thức static là phương thức có thể truy cập mà không cần khởi tạo một đối tượng của class.

+ Phương thức static gắn liền với class hơn là với object.

+ Có địa chỉ xác định và không thay đổi địa chỉ trên vùng nhớ (tĩnh).

- Khi chương trình chạy, nó sẽ được sinh ra đầu tiên trước tất cả các truy nhập tới nó và tồn tại cho tới khi chương trình kết thúc.

- Phương thức static không thể gọi phương thức hoặc thuộc tính non-static. Nhưng phương thức non-static có thể gọi phương thức hoặc thuộc tính static.

Phương thức static có thể gọi ngay cả khi chưa khởi tạo object, do đó nếu phương thức static gọi đến một phương thức non-static thì khi chưa khởi tạo object, sẽ không có biến $this (là đại diện của object) để gọi đến phương thức 'non-static`.

Đương nhiên phương thức non-static luôn luôn có thể gọi đến phương thức static vì phương thức static đã tồn tại ngay từ khi chạy chương trình, khi object chưa được khởi tạo.

**\*static, self, this**

self và static là đại diện của class, còn $this là đại diện của object.

self: đại diện cho class khai báo nó.

static: đại diện cho class gọi đến nó.

**6. Thế nào là trait ?**

- Trait là một module giúp cho chúng ta có thể sử dụng lại các phương thức được khai báo trong trait vào các class khác nhau một cách đơn giản hơn là kế thừa như trước.

- Trait có chức năng gom lại các phương thức và thuộc tính mà chúng ta muốn sử dụng lại nhiều lần.

Traits như một abstract class ( đều không thể khởi tạo được) nhưng không hoàn toàn giống nhau.

Các phương thức trong Trait có thể bị override lại trong class sử dụng nó.

**7.** **Thế nào là namespace?**

-Namespace là một cơ chế cho phép chúng ta phân nhóm các thực thể như class, object, function thành những nhóm riêng biệt, mỗi nhóm đó được đặt cho một cái tên.

-Việc dùng Namespace sẽ để tránh xung đột khi sử dụng nhiều thư viện, trong code có các class với tên giống nhau.

-Namespace có thể hiểu đơn giản như là thư mục trên các hệ điều hành, để phân cấp và phân biệt các thực thể có tên giống nhau.

**8. Thế nào là magic function?**

-Magic functions là các hàm đặc biệt được tự động thực thi khi thực hiện một số hành động nhất định lên đối tượng của class đó.

Magic functions luôn có dạng \_\_<method\_name>.

-Ưu điểm của Magic functions:

* Giúp cho chúng ta tùy biến được các hành vi.
* Nó giúp cho chúng ta có thể khởi tạo một đối tượng theo cách mình muốn.

-Nhược điểm: Một magic functions có tốc độ chậm hơn các phương thức bình thường-Một số Magic functions:

* \_\_construct(): Thực hiện khi khởi tạo đối tượng.
* \_\_destruct(): Thực hiện khi hủy bỏ đối tượng.
* \_\_get(), \_\_set(), \_\_unset(): Khi cố gắng truy cập, thay đổi, xóa thuộc tính không có sẵn trong đối tượng.
* \_\_call(), \_\_callStatic(): Khi gọi hàm không có sẵn.
* \_\_toString(): Thực hiện khi được sử dụng như string.
* \_\_invoke(): Khi đối tượng được sử dụng như một hàm.
* \_\_sleep(), \_\_awake(): Khi gọi hàm serialize, unserialize lên đối tượng.
* \_\_clone(): Khi clone đối tượng.
* \_\_set\_state(), \_\_debugInfo(): Khi gọi var\_export, var\_dump lên đối tượng

### 9. ****Từ khóa final là gì? Chúng được sử dụng như thế nào trong PHP?****

Từ khóa final trong PHP được sử dụng để đánh dấu một lớp hoặc một hàm là cuối cùng.  
  
Nếu một lớp được đánh dấu là final, nó không thể được mở rộng (extends) để sử dụng các thuộc tính hoặc phương thức của nó.  
  
Nó ngăn chặn các lớp con của nó ghi đè một phương thức.  
  
Nếu chỉ có một hàm được đánh dấu là final, thì nó không thể được mở rộng.  
  
Một khai báo final có thể được thực hiện bằng cách thêm tiền tố vào tên lớp hoặc tên hàm với từ khóa **final**.

### 2.PHP

### 1. ****Bạn hiểu như thế nào về echo và print trong PHP?****

* **echo**không không trả về bất kỳ giá trị nào trong khi **print** trả về giá trị 1 có thể được sử dụng trong các biểu thức khác.
* **echo**có thể có nhiều đối số trong khi **print**chỉ có thể chấp nhận 1 đối số tại cùng một thời điểm.

**2. Theo bạn, sự khác nhau của toán tử & và && trong PHP là gì?**

-Toán từ & và && trong PHP đều là phép toán AND, tuy nhiên toán tử một dấu & áp dụng theo kiểu bit, nói dễ hiểu hơn một dấu & là phép AND thao tác trên các bit ví dụ 1 & 0 thì ra 0. Phép toán hai dấu & thì chỉ áp dụng cho kiểu boolean True và False.

# **3**. **Cookie và session có gì khác nhau? Người ta nói bản thân của session là cookie là đúng hay sai?**

**COOKIE** là một tập tin nhỏ được server nhúng vào máy tính của người dùng. Nếu lần đầu tiên trình duyệt truy cập vào website nó sẽ gửi một COOKIE đến trình duyệt của người dùng, và mỗi khi người dùng tiếp tục yêu cầu một trang web từ website này thì COOKIE với các thông tin thu nhập từ phía người dùng trên website này sẽ được sẽ gửi trả về server của website.

**SESSION** được hiểu là khoảng thời gian mà người sử dụng giao tiếp với 1 ứng dụng. SESSION được lưu trữ hoàn toàn trên server, do vậy tính bảo mật cao hơn cookie, các website hiện này thường dùng session để lưu thông tin của người dùng khi họ đăng nhập.

Chu kỳ sống của SESSION do webserver qui định, ta có thể điều chỉnh chu kỳ này khi cấu hình webserver, tại server sẽ có 1 PHPSESSID tương ứng được tạo ra, các PHPSESSID sẽ được lưu trong một tập tin văn bản ở tại vị trí được qui định trong file php.ini ở dòng session.save\_path.

-Người ta nói bản thân của **session là cookie**, về cơ bản SESSION và COOKIE đều là các tập tin lưu trữ lại thông tin của người dùng website, tuy nhiên dựa trên khái miệm, cách sử dụng, ứng dụng của chúng là khác biệt vì thế tôi cho rằng chúng là khác biệt.

# **4. Cho biết sự khác nhau giữa serialize và json\_encode? Lý giải theo cách bạn hiểu?**

* **json\_encode**: là phương thức dùng để chuyển đổi một mảng(Array) hoặc Object thành string(chuỗi) dữ liệu JSON để sử dụng khi lưu trữ trong database, và để chuyển đổi ngược lại ta dùng **json\_decode** để chuyển đổi một chuỗi dữ liệu JSON sang dạng mảng(Array) hoặc object để sử dụng trong code PHP.
* **serialize(array)**: là phương thức dùng để chuyển một mảng(Array) hoặc Object thành string(chuỗi) dữ liệu theo chuẩn của PHP để ta có thể lưu trữ hoặc truyền tải, khi muốn chuyển nó lại thành lại một mảng hoặc phương thức ta dùng phương thức **unserialize().**

# **5. Trong PHP để gộp mảng ta dùng hàm gì? Để tách mảng ta dùng hàm gì?**

Trong PHP cung cấp hàm **array\_merge($array1, $array2, $array3,…)** dùng để nối các mảng thành một mảng duy nhất và hàm **array\_slice(array,start,[length],[preserve])** dùng để tách mảng thành một mảng nhỏ hơn.

# **6. Để chuyển mảng thành chuỗi ta dùng hàm gì? Để tách chuỗi thành mảng ta dùng hàm gì?**

Trong PHP cung cấp hàm **implode(separator,array)** dùng để chuyển mảng thành một chuỗi và hàm **explode(separator,string,[limit])** dùng để chuyển một chuỗi thành mảng. Ngoài ra ta có **hàm join($ky\_tu,$array) giống hàm implode**

# **7. Mảng tuần tự là gì? Khác gì với bất tuần tự? Để duyệt mảng ta dùng vòng lặp nào?**

-Mảng tuần tự là mảng có key là chữ số bắt đầu từ 0 và tăng dần.

-Nó khác với mảng bất tuần tự là các phần tử trong nó KEY được sắp xếp tăng dần từ 0 còn mảng bất từ tự thì không.

-Để duyệt mảng tả có thể dùng bất cứ vùng lập nào? Tuy nhiên chuẩn nhất là dùng vòng lập foreach, vì vòng lập này được nhà phát triển PHP xây dựng riêng cho việc duyệt mảng, nêu nó dễ sử dụng hơn, tốc độ xử lý của nó nhanh hơn.

# **8. Phân biệt $\_POST và $\_GET trong php?**

* Cả **$\_GET** và **$\_POST** đều được dùng để gửi dữ liệu lên server.
* **$\_GET**: Gửi dữ liệu lên server thông qua URL, nên thông tin dữ liệu hiển thị lên url vì thế bảo mật kém, dữ liệu gửi lên bị giới hạn 1024 ký tự.
* **$\_POST**: Gửi dữ liệu lên server dưới dạng ẩn thông qua HTTP Header vì thế nó có tính bảo mật cao hơn so với GET, dữ liệu gửi lên không bị giới hạn.
* Tuy nhiên tốc độ thực thi xử lý của Post chậm hơn Get.

# **9. Hằng trong PHP khác gì so với biến? Nếu 1 hằng được định nghĩa 2 lần, thì liệu có bị lỗi không?**

Hằng và biến trong PHP là khái niệm khác nhau:

* Đầu tiên khác nhau là cách khai báo:
  + Biến thì chỉ cần dùng ký tự $ để gắn hoặc lấy giá trị.
  + Hằng dùng hàm define() gắn giá trị và dùng constant() để lấy giá trị.
* Điểm khác biệt lớn nhất là hằng là không thể thay đổi trong suốt quá trình chạy chương trình, biến thì có. Nêu 1 hằng được định nghĩa 2 lần thì chương trình vẫn chạy được bình thường tuy nhiên hằng chỉ có giá trị của lần định nghĩa đầu tiên.

**1. PHP là gì?**

-Đây là từ viết tắt của “**Personal Home Page**”

-[**PHP**](https://www.bitdegree.org/learn/learn-php) là một ngôn ngữ kịch bản mã nguồn mở.

-PHP được sử dụng phổ biến nhất để phát triển web. Ngôn ngữ được yêu thích và đánh giá cao vì nó được nhúng dễ dàng vào HTML.

**2. PHP ‘PEAR’ là gì?**

-PHP **PEAR**là một phần mở rộng của chính ngôn ngữ PHP. Đây là từ viết tắt của cụm “PHP Extension and Application Repository”.

**3. PHP được sử dụng làm gì?**

* Nó có thể tạo, ghi, đóng tệp hệ thống.
* Nó có thể mã hóa dữ liệu hoặc ngăn mọi người truy cập vào các phần của trang web.
* Nó được sử dụng để chạy xung quanh với cookie - bạn sửa đổi và đặt chúng theo nhu cầu của bạn.

**4. Có bao nhiêu phiên bản PHP?**

-Hiện tại, có **bốn** phiên bản PHP - **PHP 5.6; PHP 7; PHP 7.1 và PHP 7.2.**

**5. PHP có phải là ngôn ngữ lập trình hướng đối tượng không?**

-Phải. PHP 5

**6. Khi nào bạn sẽ sử dụng 'phương pháp cuối cùng?**

- **Phương pháp cuối cùng** được sử dụng trên một phương thức với mục đích làm cho nó không còn quá mức.

**7. PHP là thừa kế đa hay đơn?**

**- PHP mang tính kế thừa duy nhất.**

**8. Cho phép sử dụng "image" trong PHP như thế nào?**

Để sử dụng chức năng hình ảnh trong PHP, bạn cần tải xuống thư viện **GD Library** - một công cụ vẽ đồ họa cho phép bạn thay đổi thông tin dữ liệu của hình ảnh.

**9.Bạn kết nối cơ sở dữ liệu MySQL database vào PHP như thế nào?**

connection mysql\_connect(Locallhost, tên\_người dùng, pass, namedatabase, client\_flag);

**11.Bạn có thể tạo ra bao nhiêu đối tượng trong PHP?**

- Số lượng không giới hạn

**12.Giải thích ‘\_construct()’ và ‘\_destruct()’?**

### \_construct(): Tạo một đối tượng mới trong một lớp.Điều này cho phép đối tượng sao chép các tham số và cài đặt của lớp và tự động gán chúng cho chính nó.

**Destruct():** Nó sẽ tự động xóa mọi cài đặt của đối tượng mới

**13.Sự khác biệt giữa ‘==’ và ‘===’ là gì?**

==: So sánh giá trị

===: So sánh giá trị và kiểu dữ liệu

**14.Bạn có thể đặt thời gian thực hiện tối đa trong tập lệnh PHP không?**

-Bạn có thể, và có hai cách để làm điều đó. Nhưng tại sao bạn lại muốn làm điều đó ngay từ đầu?

-Khi bạn chạy mã, bạn có thể gặp phải một lỗi cảnh báo rằng bạn đã vượt quá **“giới hạn tối đa thời gian thực hiện” (maximum execution time).** Điều này sẽ ngăn mã của bạn chạy tiếp, dẫn đến làm hỏng quá trình.

Để tránh điều này, bạn cần thực hiện một lệnh ngay từ đầu mã của mình. Một cách khác để làm điều này là thực hiện cùng một lệnh trong tệp **.ini.**

**15.Khi nào sử dụng ‘@’?**

-Lệnh “@” được sử dụng để tránh các vấn đề trong mã của bạn bằng cách đơn giản yêu cầu PHP giải quyết chúng cho bạn.

-Những gì bạn sẽ làm chỉ đơn giản là thêm Lệnh “@” ở đầu một dòng mã có thể gây ra thông báo lỗi. Sau khi được thêm vào, “@” sẽ truyền bất kỳ lỗi nào mà dòng mã gây ra cho PHP.

**16.Làm thế nào để tìm ra độ dài của một mảng**?

-Count(tên mảng)

**17.Làm thế nào các đối tượng được truyền trong PHP - theo tham chiếu hay theo giá trị?**

- Trong PHP, các đối tượng chỉ có thể được truyền **theo giá trị (by value).**

**18. So sánh hàm ‘include()’ và ‘require()’ thông qua cách chúng xử lý các vấn đề phát sinh trong code?**

- **include() :** lỗi vẫn chạy tiếp.

**- require() :** lỗi dừng luôn chương trình

**19. Làm thế nào để trích xuất dữ liệu PHP ra Excel?**

- Sử dụng thư viện PHPExcel

**20.Liệt kê các loại lỗi trong PHP**

-Có **ba** loại lỗi chính trong PHP - **notices,warnings**và **fatal**.

**Notices: thường gặp nhất, ví dụ: lỗi tìm phần tử x, mà phần tử x bị null.**

**Warnings:**  là lỗi cấp hai, ví dụ: lỗi include thiếu file

**Fatal: lỗi nghiêm trong, ví dụ lỗi require, dừng chương trình luôn**

### ****6. Sự khác nhau giữ emty() và isset()****

 -Empty() check biến có giá trị là rỗng hoặc mảng rỗng,null,0.

-Isset() thì chỉ check sự tồn tại của biến, và biến có giá trị null không .

Empty bao quát hơn isset

### ****7. Sẽ hỏi những câu hỏi về các hàm có sẵn trong php ví dụ như:****

**-Tìm một phần tử tồn tại trong mảng ta dùng hàm gì ?** (in\_array)

**-Hàm array\_merge() và array\_combine() có gì khác nhau ?**

**-Đếm số phần tử trong mảng dùng hàm nào count();**

**-Thêm một hoặc nhiều phần tử vào cuối mảng và trả về số lượng phần tử của mảng ta dùng hàm gì? array\_push()**

**-Hàm nào  sẽ chuyển một chuỗi $string thành một mảng các phần tử ?explode ();**

🡺Nếu có câu nào mà bạn 0 ko trả lời dc thì mình khuyên bạn nên trả lời: là những hàm này em cũng ko nhớ, vì nó có rất nhiều, nhưng trong quá trình làm chủ yếu là tư duy logic, đến khi cần hàm nào thì mình lên google search,

3. MYQL

### Câu 20: Trong SQL có mấy loại Join?

Gợi ý câu trả lời:

“Trong SQL hiện nay, có các loại Join như sau:

+ Inner Join - được sử dụng trong trường hợp Join ở hai bảng khớp nhau khiến kết quả trả về là các bảng ghi.

+ Outer Join - Được sử để lấy  các bản ghi có mặt trong cả hai bảng, thường chia làm 2 loại là Full Outer Join và Half Outer Join (Left hoặc Right).

+ Cross Join - dùng để kết hợp bản ghi của bảng A và B lại với nhau để tạo thành một tích đề.”

### ****1. Website và web app có gì giống và khác ?****

Có thể tạm trả lời như sau: Web app thực chất là web site nhưng được gọi riêng để chỉ những website có những chức năng và ứng dụng riêng biệt

như quản lí bán hàng, quản lí kho...

Còn website là những web có tính tương tác thấp hơn nhiều so với webapp có thể là những trang web tĩnh, không có nhu cầu cập nhật dữ liệu...

### ****9: Biết cắt layout từ photoshop ko, và cắt cho họ xem****

đây được xem là câu hỏi rất hay gặp ở 1 số công ty, để làm được điều này bạn cần học vững kiến thức về html, css, ngoài ra kĩ năng sử dụng bootstrap tốt sẽ giúp bạn ghi điểm hơn.

### ****10: $$ có nghĩa là gì****

gắn tên cho biến:

ví dụ $a = "hello";

thì ta có

$$a = $hello

### ****11: Khi tắt cookie trên trình duyệt thì có ảnh hưởng tới session không:****

-không thể sử dụng được cả cookie và session khi tắt cookie, vì bản chất là session khi hoạt động cũng sẽ tạo một cookie trên trình duyệt để lưu dữ liệu, nên khi tắt cookie trên trình duyệt thì session cũng không sử dụng được

### ****12: Tại sao chúng tôi phải lựa chọn bạn vào công ty chúng tôi****

-trả lời: vậy công ty các anh có cái gì để tôi phải vào đây xin ứng tuyển... đùa chứ cấu này mình nghĩ là họ đang hỏi về điểm mạnh của bạn đấy, bạn hãy show hết tất cả những điểm tốt của mình ra, những kinh nghiệm mà bạn đã có trong quá trình học tập, cũng như là làm đồ án, nhưng nhớ là chém bão vừa vừa thôi nhé, chứ không là mấy anh phỏng vấn "bay" mất đó

**13: Có người yêu chưa, biết nhậu ko, biết chơi đá (banh) không…**

### ****2. Những lỗi bảo mật thường gặp trên website****

### ****3: khi website bị chậm bạn phải làm gì****

Tìm Nguyên nhân lỗi do đâu và tùy vào lỗi sẽ có hướng giải quyết khác nhau,

Thường có thể sẽ là do các nguyên nhân chính sau

* **Code không tối ưu, câu truy vấn sql chưa tối** ưu => giải pháp, phải kiểm tra xem code chỗ nào chưa tối ưu, tiến hành sữa lỗi và fix lại bug, giảm độ kồng kền của code, giảm dung lượng ảnh, video trên sever
* **Hosting cấu hình thấp, không được tối ưu, quá tải do nhiều người dùng**=> nâng cấp cấu hình host. Nhưng khi lượng request lên đến 1-2 triệu thì không thể nâng cấp server được nữa, **mà phải thiết kế sao cho nhiều server chạy cùng lúc**. Giả sử ta có 10 sever để phục vụ cho người dùng vậy làm sao để có thể điểu khiển nó, để thực hiện điều ấy ta  cần dùng **Cân  bằng tải với Load Balancer**, Người dùng sẽ không trực tiếp truy cập tới server, mà chỉ truy cập  tới load balancer. LB sẽ điều tiết, cân bằng lượng tải trên 10 server này.

### Khoảng cách sever và máy test xa => đăng kí dịch vụ cdn

### Không sử dụng cache cũng khiến website nặng nề

### Website bị tấn công  DDOS

### ****6: Làm thế nào để website có thể phục vụ hàng triệu người dùng****

Thường thì sẽ nâng cấp sever, băng thông, ram, bộ nhớ... khi lượt truy cập tăng lên nhưng khi lượng request lên đến 1-2 triệu thì không thể nâng cấp server được nữa, mà phải thiết kế sao cho nhiều server chạy cùng lúc.

Giả sử ta có 10 sever để phục vụ cho người dùng vậy làm sao để có thể điểu khiển nó, đẻ thực hiện điều ấy ta cần dùng **Cân bằng tải với Load Balancer**, Người dùng sẽ không trực tiếp truy cập tới server, mà chỉ truy cập tới load balancer. LB sẽ điều tiết, cân bằng lượng tải trên 10 server này.

**Phân tán dữ liệu với Content Delivery Network (CDN)**: CDN là một mạng lưới các server được phân bố trên nhiều khu vực.Ví dụ bạn có 1 sever để bên mỹ, mỗi lần bạn truy cập qua bên sver đó sẽ rất lâu vì khoản cách xa, nhưng khi sử dụng CDN bạn sẽ kết nối vs sever gần nhất ở VN, tốc độ sẽ nhanh hơn nhiều

**Caching**: Cache là một kĩ thuật để tăng tốc độ đọc dữ liệu, bằng cách lưu dữ liệu sẵn vào cache server để tăng tốc độ đọc ở những lần sau.

**-Các Phương pháp thiết kế hướng đối tượng (SOLID)**

**-Tìm hiểu các quy tắc trong PSR2**

### Kể tên những mẫu Design Pattern mà bạn biết?

Gợi ý câu trả lời:

“Một số mẫu  Design Pattern trong lập trình PHP hiện nay như:

+ Creational patterns - các mẫu tạo lập: Builder, Factory method pattern, Abstract Factory pattern, Singleton, Prototype.

+ Structural patterns - các mẫu cấu trúc: Composite, Bridge, Adapter, Flyweight, Decorator, Facade, Proxy.

+ Behavioral patterns - các mẫu ứng xử: Memento, State, Chain of responsibility, Interpreter, Command, Iterator, Observer, Mediator, Template method, State, Visitor, Strategy.”

### Nêu quy trình viết API trong lập trình PHP?

Gợi ý câu trả lời:

“Trong một quy trình viết API của lập trình PHP hiện nay cần tiến hành qua các bước như sau:

Bước 1: Thực hiện nhập dữ liệu.

Bước 2: Tiến hành xử lý Request khi có được dữ liệu cần thiết và Endpoint.

Bước 3: Trả về Response theo mã HTTP tương ứng với hàm Header.”

# **Quy Trình thiết kế website chuyên nghiệp**

## **bước 1: Thu thập thông tin từ khách hang**

## **Bước 2  lập kế hoach:**

## **Bước 3: thiết kế**

3.1. Tạo basic design

3.2. Thiết kế Database, tạo detail design

## **Bước 4: Phát triển (coding)**

## **Bước 5: Kiểm thử và chỉnh sửa**

### Bước 5.1: tạo test case

### 

### Bước 5.2: chạy test case

Bước 6: Bảo trì website

HOME

## Best Open Source PHP Server Hosts(Máy chủ PHP)

There are many open-source PHP server packages available on the internet. The names of the packs are often acronym of the tools they contain.

Here are the Best PHP Servers:

* [XAMPP](https://serverguy.com/servers/php-servers/#xampp)
* [LAMP](https://serverguy.com/servers/php-servers/#lamp)
* [LEMP](https://serverguy.com/servers/php-servers/#lemp)
* [MAMP](https://serverguy.com/servers/php-servers/#mamp)
* [WAMP](https://serverguy.com/servers/php-servers/#wamp)
* [AMPSS](https://serverguy.com/servers/php-servers/#ampss)
* [EasyPHP](https://serverguy.com/servers/php-servers/#easyphp)

**What is Solution Stack?**

-là một tập hợp các chương trình khác nhau cần thiết để tạo ra một nền tảng hoàn chỉnh.

-Xampp,Lamp…

* **Vd:Xampp**
* X- Cross-Platform
* A- Apache Server
* M- MariaDB
* P- PHP
* P- Pearl

## **Four Essential tools for PHP Servers are:4 công cụ cần thiết của máy chủ PHP**

* Operating System
* Web Server
* DBMS
* PHP
* Một ứng dụng chạy được cần có 4 thứ trên

### Operating System

Hệ điều hành là phần mềm hệ thống quản lý phần cứng máy tính và các phần mềm khác nhau.

Hệ điều hành giúp bạn giao tiếp với máy tính. Con người không thể sử dụng bất kỳ máy tính nào không có bất kỳ hệ điều hành nào.

* Windows
* Linux
* Mac OS

### Web Server.

### Máy chủ Web là một chương trình sử dụng HTTP (Giao thức truyền siêu văn bản) để cung cấp các tệp tạo thành trang Web cho người dùng.

* Internet Information Service
* Apache HTTP Server
* LIGHTTPD
* NGINX
* Node.js

### DBMS

### MariaDB và MySQL là các Hệ quản trị cơ sở dữ liệu phổ biến nhất.

### PHP

### -Ngôn ngữ lâp trình

### Hỏi đáp Người Phỏng Vấn

**Về công việc**

-Anh có thể nói rõ hơn về công việc tại vị trí tôi được tuyển dụng vào làm không?

-Với thời gian làm việc như em thì thưc tập trong bao lâu em mới có thể đủ năng lực để vào làm dự án cho công ty.

-Với vị trí này thì em có thể hưởng quyền lợi gì vậy ạ, ví dụ như trợ cấp, bảo hiểm…

**Về công ty**

-Công ty có bao nhiêu người vậy ạ?

-Bô phận e ưng tuyển có bao nhiêu người?

**Về người phỏng vấn:**

-Anh chị đang làm ở vị trí gì vậy ạ?

-Anh đã làm ở đây lâu chưa?

-Với công việc hiện tại anh thấy như thế nào ạ?

-Anh có nhân xét như thế nào về bộ phận em đang ứng tuyển ạ