HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Lập trình Web

|  |
| --- |
| Tài liệu cho bài tập số 2  *Người thực hiện: Trần Ngọc Phúc*  *Mã sinh viên: B16DCCN271*  *Lớp sinh viên: D16CNPM03* |

MỤC LỤC

[I - Tạo thư mục theme sử dụng Stater theme 2](#_Toc23808235)

[II - Chuyển đổi giao diện tĩnh với Bootstrap thành giao diện động với Wordpress 3](#_Toc23808236)

[**1.** **Chuyển đổi phần CSS và Javascript** 3](#_Toc23808237)

[**2.** **Chuyển đổi phần HTML** 3](#_Toc23808238)

[**a.** **Chuyển đổi phần nội dung thẻ <head> và các đường dẫn** 3](#_Toc23808239)

[**b.** **Chuyển đổi phần thanh điều hướng** 3](#_Toc23808240)

[**3.** **Giúp việc tùy chỉnh giao diện dễ dàng với người dùng hơn thông qua các plugin** 4](#_Toc23808241)

[**a.** **Đơn giản hóa việc chỉnh sửa các trường bằng Advanced Custom Field (ACF)** 4](#_Toc23808242)

[**b.** **Tự tạo kiểu nội dung và quản lý với Custom Post Type UI (CPT UI)** 4](#_Toc23808243)

[**4.** **Một số vấn đề gặp phải khi chuyển đổi giao diện** 4](#_Toc23808244)

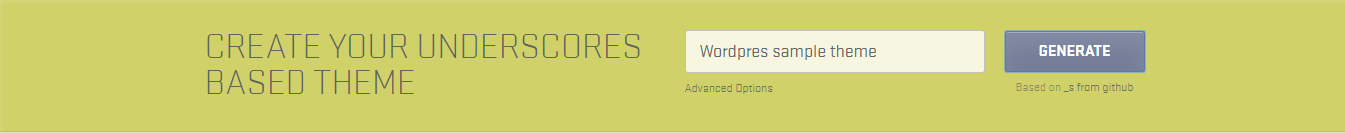
[**a.** **Tách nhỏ bố cục trang** 4](#_Toc23808245)

[**b.** **Ẩn tiêu đề trang** 5](#_Toc23808246)

Chuyển đổi giao diện tĩnh bằng Bootstrap thành giao diện động với Wordpress

# **I - Tạo thư mục theme sử dụng Stater theme**

Cấu trúc thư mục cần tuân theo một quy chuẩn là rất quan trọng khi sản phẩm được sử dụng để đưa lên các chợ trực tuyến hay thương mại hóa. Phần ví dụ này sẽ áp dụng một starter theme đó là [Underscores](https://underscores.me/), một starter theme Wordpress khá nổi tiếng.

Truy cập trang web underscores.me, nhập tên giao diện vào ô trống và nhấn Generate, một file nén chứa starter theme với tên được chỉ định sẽ được tải về. Giải nén nó và đưa vào thư mục chứa theme của Wordpress (`…\wp-content\themes`).

Tạo starter theme với Underscores

|  |  |
| --- | --- |
| Sau khi giải nén, ta sẽ được thư mục theme với cấu trúc như hình. Có thể thấy đầy đủ các file cần thiết cho một theme Wordpress đã được tạo sẵn.  Đồng thời, giao diện vừa tạo sẽ xuất hiện trong mục quản lý giao diện của Wordpress (‘ *Appearance > Themes* ’), tuy nhiên nó lại không có hình ảnh nào để nhận diện mà chỉ là một màu trắng, chính xác thứ được hiển thị ở đây là file hình ảnh `screenshot.png` có sẵn trong starter theme của Underscores. Chúng ta nên thay thế nó bằng một hình ảnh cụ thể để thuận tiện cho việc nhận diện trong quá trình sử dụng. Chỉ đơn giản là thay thế hình ảnh có sẵn bằng một file ảnh cũng với tên `screenshot` bất kỳ (thường là ảnh chụp lại màn hình giao diện trang chủ của theme).  Vậy là ta đã có được một theme wordpress với cấu trúc thư mục chuẩn. Giai đoạn tiếp theo sẽ là đưa các file cắt HTML/CSS với Bootstrap vào trong theme này, sẽ được nói ở phần thứ 2 của tài liệu. |  |

# **II - Chuyển đổi giao diện tĩnh với Bootstrap thành giao diện động với Wordpress**

## **Chuyển đổi phần CSS và Javascript**

Với các phần mã CSS và Javascript có thể đơn giản sao chép trực tiếp các đoạn mã đã viết trước đó vào các file hoặc thư mục tương ứng. Lưu ý, các phần mã này chưa thể hoạt động khi chúng chưa liên kết với HTML sẽ được nói tới ở phần sau.

## **Chuyển đổi phần HTML**

Đầu tiên cần nhận thấy phần Header và Footer của một website sẽ cố định trên các trang khác nhau, do đó, ta sẽ tiến hành tách riêng hai thành phần này từ file cắt Web trước đó và đưa vào các file `header.php` và `footer.php` tương ứng. Phần hướng dẫn sau đây sẽ tập trung vào phần header, các phần còn lại trong trang được xử lý hoàn toàn tương tự.

1. **Chuyển đổi phần nội dung thẻ <head> và các đường dẫn**

Tùy vào thiết kế mà phần header sẽ có các giới hạn khác nhau nhưng sẽ bao gồm phần nằm trong thẻ <head> và thường là phần chứa navbar trong thẻ <body>.

Sao chép toàn bộ nội dung thẻ <head> vào vị trí tương ứng trong file `header.php` ngay trên thẻ <?php wp\_head() ?> (wp\_head là một Action hook của Wordpress).

Tiếp theo, Wordpress không thể hiểu được các đường dẫn tĩnh ta đã truyền vào trước đó để liên kết các file CSS hay JavaScript. Do đó, việc cần làm là thay thế toàn bộ các đường dẫn tĩnh trước đó bằng cách sử dụng một hàm mà Wordpress có thể hiểu được đó là hàm `bloginfo($info);` . Các đường dẫn sau khi xử lý sẽ tương tự như hình bên dưới.

|  |  |
| --- | --- |
| ***Trước:*** | ***Sau:*** |

Từ giờ trở đi, tất cả các các liên kết tĩnh trong thư mục đều được xử lý tương tự.

1. **Chuyển đổi phần thanh điều hướng**

Phần chứa navbar cũng tương tự, ta thay thế đoạn mã của thẻ <header> trong file `header.php` bằng đoạn mã của mình. Để xử lý dữ liệu động cho thanh điều hướng, Wordpress sử dụng một hàm để làm điều đó, đó là `wp\_nav\_menu( array $args = array() );`. Hàm này sẽ tạo ra một div chứa trong đó là các menu item <li> được bọc trong một thẻ <ul>. Thay thế đoạn mã chứa nội dung các danh mục bằng hàm này, kết quả sẽ giống như hình dưới đây (có thể thay thế bằng các tham số khác).

|  |  |
| --- | --- |
| ***Trước:*** | ***Sau:*** |

Bằng cách này, chúng ta có thể thay đổi nội dung của thanh điều hướng một cách linh hoạt bằng cách truy cập vào ‘ *Appearance > Menus* ’ trong trang quản lý của Wordpress và thay đổi chúng từ đây.

## **Giúp việc tùy chỉnh giao diện dễ dàng với người dùng hơn thông qua các plugin**

### **Đơn giản hóa việc chỉnh sửa các trường bằng Advanced Custom Field (ACF)**

**Advanced Custom Field** là một trong các Plugin được giới thiệu trong tài liệu này. Đây là một plugin bên thứ ba mạnh mẽ giúp người tùy chỉnh dễ dàng các thành phần giao diện được hiển thị, nó thân thiện với người sử dụng hơn bằng cách sử dụng giao diện đồ họa được nhúng vào trình quản lý của Wordpress, giúp họ dễ dàng thay đổi các thành phần giao diện mà không cần phải can thiệp vào mã nguồn.

### **Tự tạo kiểu nội dung và quản lý với Custom Post Type UI (CPT UI)**

**Custom Post Type UI** là một plugin với chức năng tạo và quản lý các kiểu dữ liệu tùy chỉnh (*custom post type*) trong Wordpress, phân loại (*taxonomy*) chúng thay vì các kiểu mặc định và phương thức phân loại đi kèm mà Wordpress cung cấp. Các thao tác cũng rất đơn giản.

Khi kết hợp plugin này với ACF ở phần trước, đem lại một giải pháp toàn diện giúp cho sản phẩm đến gần hơn với người dùng, đặc biệt là khi người dùng không có nhiều kiến thức về mã nguồn nhưng vẫn có thể thao tác chỉnh sửa bố cục, thành phần trong trang Web một cách dễ dàng.

## **Một số vấn đề gặp phải khi chuyển đổi giao diện**

1. **Tách nhỏ bố cục trang**

Thông thường, trong một website ngoài các thành phần chung là header và footer, việc tái sử dụng một số khối nội dung đặc biệt cũng thường xuyên xảy ra. Do đó, để thuận tiện chỉnh sửa và tinh gọn mã nguồn chúng ta nên tách tất cả các thành phần này ra thành các file để xử lý riêng biệt. Wordpress hỗ trợ điều đó, và thực tế là các giao diện hiện nay đều được xây dựng theo cách tương tự. Trong Wordpress các thành phần được tách ra như vậy được gọi là các *template-parts*.

Trước tiên, tạo một file `\*.php`, file này chứa nội dung của thành phần muốn tách riêng. Sau đó, thêm đoạn mã với hàm ` get\_template\_part() ` vào bất cứ đâu trên trang để hiển thị nó. Ví dụ, đoạn mã sau sẽ thêm template part với slug là `content` và tên là `author` vào trang:

<?php get\_template\_part('content', 'author') ?>

|  |  |
| --- | --- |
| Và tên file trong trường hợp này sẽ có dạng như sau: |  |

1. **Ẩn tiêu đề trang**

Khi tạo theme, Wordpress sẽ tự động chèn thêm vào một số thành phần, đôi khi không thực sự cần thiết cũng như làm hỏng bố cục giao diện. Tiêu đề của trang là một trong số đó.

Có 2 giải pháp cho vấn đề này đó là xóa bằng cách can thiệp trực tiếp vào mã nguồn và xóa đoạn mã đó đi, tuy nhiên trước khi làm như thế, chúng ta nên biết mục đích tại sao Wordpress lại thêm nó vào trong mã nguồn. Để có câu trả lời cho điều đó, chúng ta sẽ đi nói về giải pháp thứ 2 trước, sử dụng plugin để ẩn tiêu đề khỏi khung nhìn trình duyệt, việc làm này đảm bảo việc toàn vẹn mã nguồn. Vậy tại sao phải đảm bảo toàn vẹn mã nguồn trong trường hợp này? Ở phần trước, chúng ta đã tách phần header chứa thẻ <head> thường dùng để chứa tiêu đề trang trong đó ra để dùng chung cho tất cả các trang trong website, do đó, khi mỗi trang nội dung yêu cầu tiêu đề khác nhau thì cần có một cách thêm tiêu đề khác thay vì sử dụng header; thứ hai, tiêu đề trang là thứ mà các công cụ tìm kiếm sử dụng để tìm ra các trang web, do đó, việc xóa đi nó khiến việc tiếp cận trang web bằng các công cụ tìm kiếm là không thể, thực tế hiện nay, nếu bạn không thể tìm thấy một trang web bằng công cụ tìm kiếm như Google hay Bing… gần như đồng nghĩa với việc trang web đó không tồn tại.

Tóm lại, tùy vào trường hợp, mà có thể sử dụng một trong hai cách trên. Nếu bạn hiểu về mã nguồn cũng như thấy việc nó xuất hiện trong các kết quả tìm kiếm không quan trọng, thì chỉ việc xóa nó đi; nhưng nếu bạn không muốn hoặc không thể tìm thấy đoạn mã đó thì cách đơn giản và an toàn nhất là sử dụng plugin để ẩn nó.

Plugin được sử dụng để ẩn tiêu đề trang: **Hide Title**

|  |  |
| --- | --- |
| **Sau khi tải và kích hoạt thành công như các plugin khác, trên giao diện chỉnh sửa của mỗi trang (page) sẽ có thêm mục Hide Title với hộp chọn như trong hình. Tích để ẩn đi thành phần tiêu đề trên trang đó.** | Mục chỉnh sửa mới xuất hiện với lựa chọn ẩn tiêu đề trang |