**Tài liệu hướng dẫn lập trình Module IPU**

1. Hướng dẫn các thành phần của Module
   1. Model

Là file php để đại diện cho 1 bảng trong cơ sở dữ liệu, nó sẽ giúp cho việc lấy dữ liệu, thêm, sửa, xóa nhanh hơn.

* 1. View

Là các file hiển thị nội dung, nó có dạng .blade.php -> Là template của Laravel

* 1. Controller

Là các file xử lý các request từ trình duyệt hoặc gọi Ajax gửi tới

* 1. Router

Định nghĩa trong file web.php -> ghép yêu cầu tới hệ thống tới các Controller tương ứng

* 1. Assets: JS/CSS

Là các file css định nghĩa style hiển thị, file js lập trình mã html và gửi/nhận dữ liệu tới máy chủ.

Mã javascript hoặc css có thể viết trong file view blade.php.

1. Xây dựng chức năng cụ thể Trung kế IP
   1. Tổng quan cách hoạt động

A diagram of a computer program

Description automatically generated

Cơ bản cách hoạt động của 1 chức năng là:

* Tải trang theo url được định nghĩa trong router web.php
* Hàm trong Controller được gọi, lấy view hiển thị ra
* View đã được hiển thị thì các hàm javascript sẽ được gọi và thực thi để hiển thị dữ liệu thông qua hàm gọi ajax.
* Các hàm gọi ajax được Controller đón nhận và xử lý sau đó trả về dữ liệu.
* Các hàm javascript tiếp tục bắt các sự kiện trên view và gửi dữ liệu thông qua hàm gọi ajax => Việc này thực hiện nhiều lần theo các sự kiện được lập trình.
  1. Tạo Model

File: IpuTrungkeIp.php trong thư mục App

class IpuTrungkeIp extends Model

{

    protected $table = "ipu\_trungke\_ip";

    protected $fillable = ['user\_id', 'tentrungke', 'diachiketnoi', 'taikhoan', 'matkhau', 'chuanbaohieu','thu1','thu2','thu3','thu4','thu5','outboundproxy', 'created\_at', 'updated\_at'];

}

Biến: $table định nghĩa bảng lưu trữ trong CSDL

Biến: $fillable định nghĩa các trường thông tin được phép truyền vào form để thêm/sửa. Cách làm: Điền tất cả các trường có trong DB loại trừ trường id tự sinh của bảng.

* 1. Tạo View

Tạo view ipu\_trungke\_ip\_e1.blade.php trong thư mục resources/views trong đây có mã html hiển thị form input và javascript để xử lý view

Tham khảo view này để xây dựng các form khác tương tự, sử dụng <fieldset> và <legend> để dựng view giống với thiết kế tại ứng dụng trên Winform.

* 1. Tạo Controller hiển thị View

Trong Controller IPUController, tạo hàm ipu\_trungke\_ip\_e1 và truyền vào dữ liệu ban đầu, sau đó gọi tới view

* 1. Tạo Router

Trong web.php tạo router liên kết tới hàm view vừa tạo

Route::get('/ipu\_trungke\_ip\_e1', [IPUController::class, 'ipu\_trungke\_ip\_e1']);

* 1. Lập trình xử lý sự kiện Ajax trên View

Trong file view ipu\_trungke\_ip\_e1.blade.php chúng ta sẽ lập trình để xử lý:

* Tải danh sách trung kế ip hiện có, sử dụng thư viện jqxgrid để hiển thị và bắt sự kiện chọn bản ghi để sửa, xóa
* Thêm trung kế IP
* Sửa trung kế IP
* Xóa trung kế IP
* Bắt sự kiện thay đổi chế độ làm việc
* Xem chi tiết trong file phần code js này
  1. Lập trình xử lý sự kiện Ajax trên Controller

Chúng đã đã định nghĩa trong file web.php 2 router:

* Route::get('/publicajaxipu', [IPUController::class, 'publicajaxipu']) => Sử dụng để gọi lấy thông tin mà không cần đăng nhập
* Route::post('/ajaxipu', [IPUController::class, 'ajaxipu']) => Sử dụng để thêm/sửa/xóa thông tin (Cần phải đăng nhập)

Tương ứng với 2 route chúng ta có 2 hàm xử lý:

* Publicajaxipu và ajaxipu: Tại đây sẽ xử lý dữ liệu đầu vào và phân nhánh các câu lệnh cập nhật vào DB như: thêm/sửa/xóa hoặc lấy dữ liệu.
* Xem chi tiết trong file.

**Để thực hiện xây dựng cho các chức năng khác của module IPU thì cần thực hiện lặp lại các bước từ 2-7, bổ sung các file, các trường thông tin và hàm xử lý theo bảng dữ liệu mới.**

**Lưu ý: Các bảng dữ liệu làm việc đều bắt đầu bằng “ipu\_” để phân biệt với module khác.**