# Front-End – VueJS

* Dùng theme demo1
* Những File đã thay đổi hoặc tạo mới:
  + src/view/layout/aside/Menu.vue
  + src/view/pages/Dashboard.vue
  + src/view/content/widgets/mixed/Widget1.vue
  + src/view/content/widgets/list/Widget1.vue
  + src/view/content/widgets/list/Widget3.vue
  + src/view/pages/QuanLyNhanVien.vue
  + src/router.js
* **Màn hình hiển thị tra cứu:**
  + File Dashboard.vue
    - mixed/Widget1.vue
    - list/Widget1.vue
* **Màn hình quản lý nhân viên:**
  + File QuanLyNhanVien.vue
    - list/Widget3.vue
  + Note:
    - Sửa cái react của treeview: khi bấm vào thì nó có animation các thứ... (so sánh vs template)
* *Sau khi Design xong:*
  + Học cách dùng Template (Xem Vid hướng dẫn) để tạo được trang mới và gắn Link đến trang mới ở thanh sidebar (router...) → Chuyển Màn hình 2 sang trang mới
  + Trau chuốt thêm Style (Bootstrap các thứ...) cho đẹp (nếu có thời gian)

# Front-End – HTML

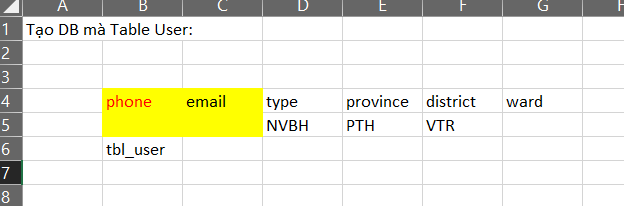
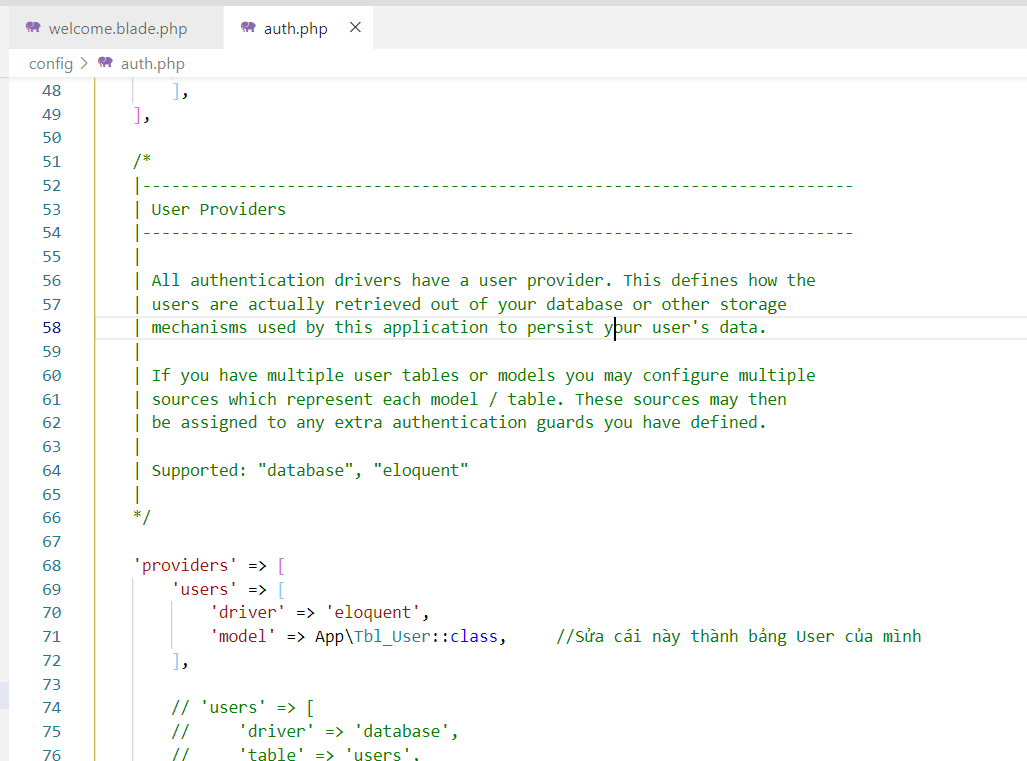
*[Chuyển sang HTML vì Vue của mình không chạy đc]*

* **Mở Server:**
  + Vào html/tools và chạy CMD
  + Gõ: gulp localhost --demo1
  + Server sẽ chạy ở localhost/8080
* Những File đã thay đổi / tạo mới:
  + index.html
    - line 135: Link đến 2 màn hình
  + HienThiTraCuu.html
    - line 135: Link đến 2 màn hình
    - line 4761: các thành phần trong màn hình
  + QuanLyNhanVien.html
    - line 135: Link đến 2 màn hình
    - line 4761: các thành phần trong màn hình
    - line 6494: link đến Script
  + src/js/QuanLyNhanVienScript.js

# Back-End – Laravel

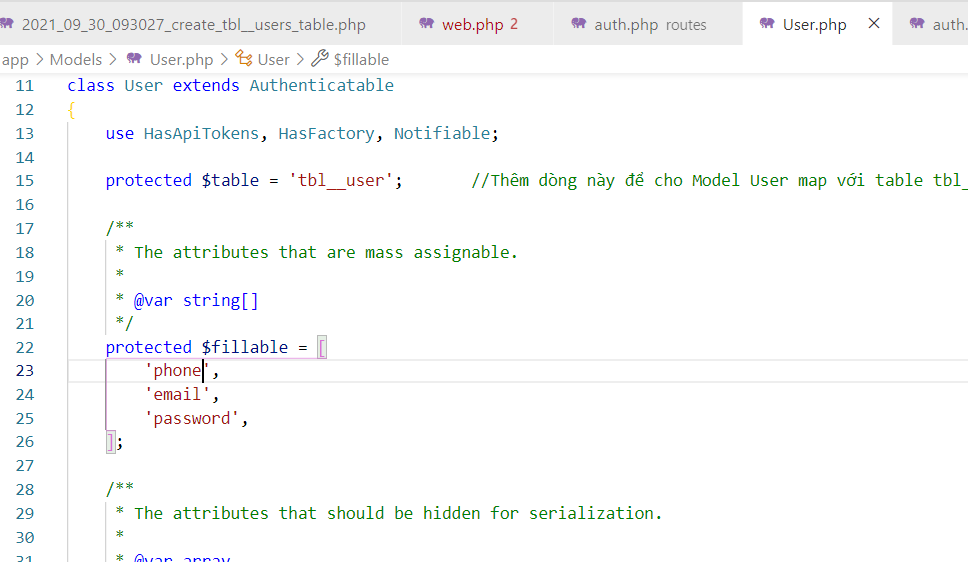
*[Cho vào Project Laravel]*

[Laravel 8.] – Project: **nvbh**

* Tạo Table Users
  + 
  + Và Password
* Tạo tính năng Login
  + <https://viblo.asia/p/tim-hieu-ve-authentication-trong-laravel-Ljy5VoG3Kra>
  + Sửa:
    - 
  + Với đoạn này:
    - 
    - → php artisan make:auth chỉ dành cho Laravel version 5.7 → Làm theo kiểu version 8.0: Starter Kit
      * <https://laravel.com/docs/8.x/starter-kits>

**SỬA LỖI:** với table Tbl\_User thì không cần tạo Model mới, mà chỉ cần khai báo protected $table = tbl\_User trong Model User là được!

* Sửa lỗi bảng Tbl\_User: Thêm Password
  + Và insert thêm 1 giá trị (hàng) vào bảng
* Sửa các thuộc tính trong $fillable cho giống với bảng mới

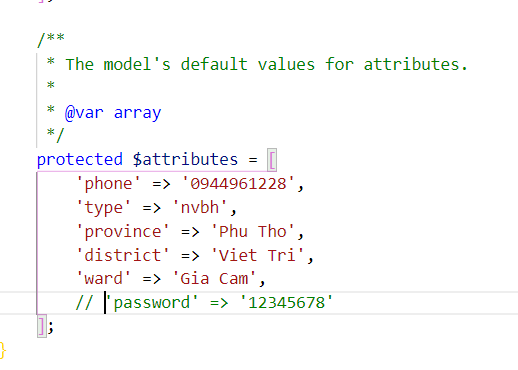


* LỖI: sao phải có default value cho Password?

LỖI: đăng nhập không được

* + → Sửa:
    - 

Password chuyển từ hidden sang fillable

* + - 

Không còn cần default cho password

* Thêm các trường dữ liệu mới để nhập khi Register, và bỏ Default value cho chúng đi
  + 
  + **→ LỖI: không hiển thị lên view khi vào /register**
    - (Không cần)
* Khi vào trang hienThi và traCuu thì chỗ góc phải trên màn hình phải hiển thị tên người dùng
  + Chưa có

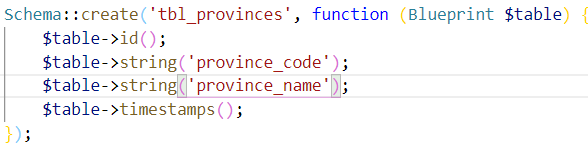
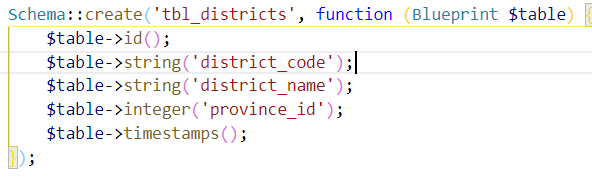
# Backend: Tạo Database:

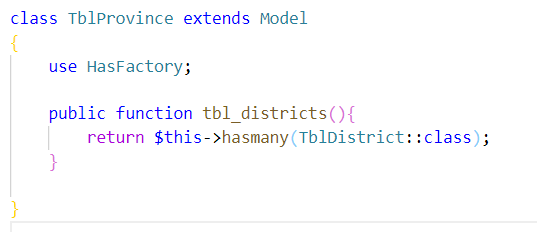
Theo cấu trúc trong file Danh mục nhân viên

Code để hiển thị ra View: /quanLy

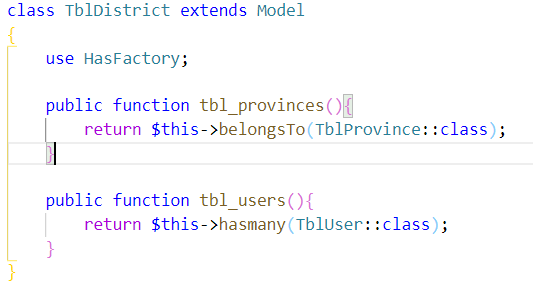
→ Sử dụng Laravel Nested-set để hiển thị / lưu trữ dữ liệu theo cấu trúc treeview

<https://www.codecheef.org/article/laravel-treeview-structure-and-display-hierarchical-data-example>

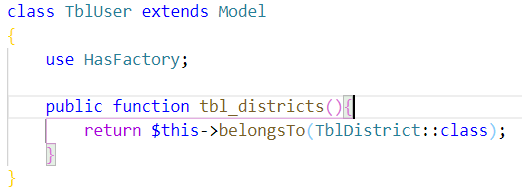
* Install laravel-nestedset
  + Mở GitBash và gõ composer require kalnoy/nestedset
* Create Model & Database migration
  + Xóa bảng tbl\_\_user để tạo bảng mới phù hợp hơn
  + **Thay đổi**: Không thể làm Model theo kiểu của trang web hướng dẫn đc vì ko phù hợp – nó là nested collumn trong cùng 1 table →
  + Tạo Model:
    - TblProvince
      * 
    - TblDistrict
      * 
    - TblUser
      * 
  + Tạo các Relationship giữa các Model:
    - app/Models/TblProvince.php :

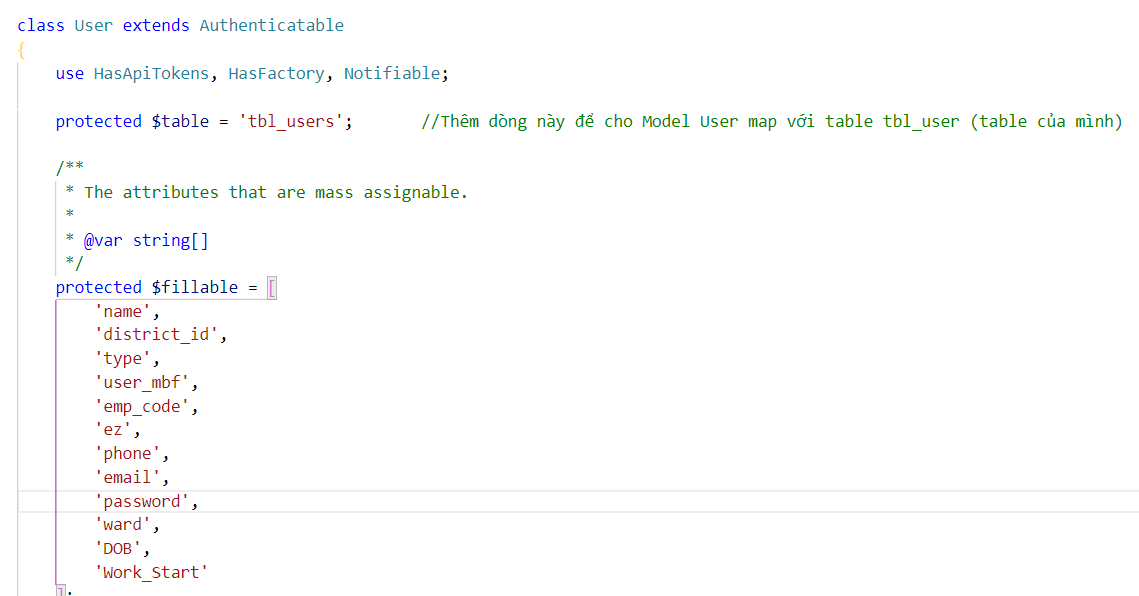


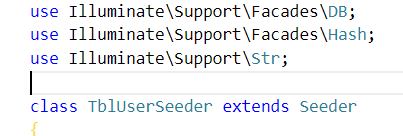
* + - app/Models/TblDistrict.php :

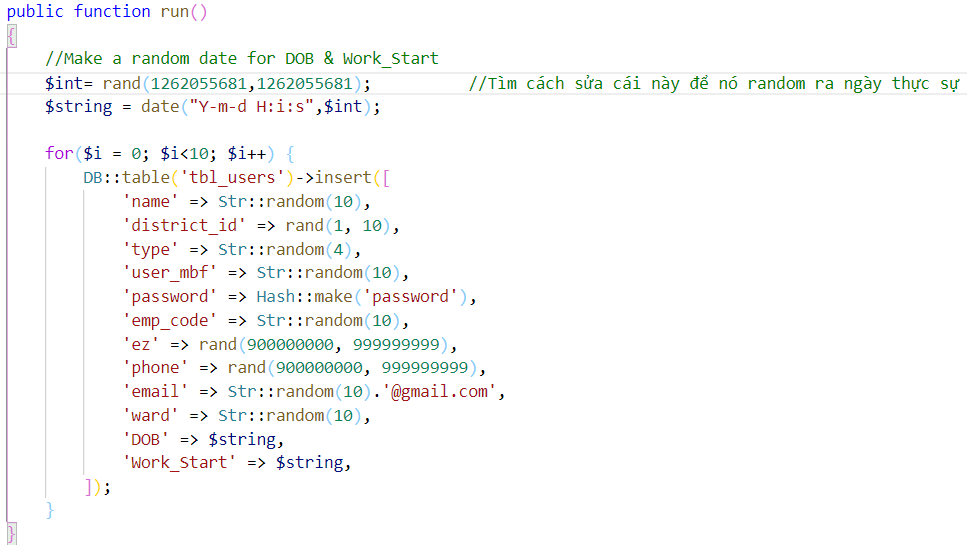


* + - app/Models/TblUser.php :

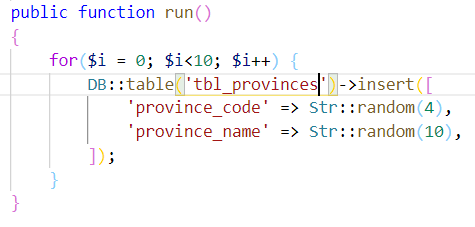


* + Sửa lại Model User.php :
  + 
* Tạo **Seed**:
  + Lệnh Artisan:
    - php artisan make:seeder TblProvinceSeeder
    - ...
    - php artisan db:seed
    - php artisan migrate:fresh --seed (drop hết các table, migrate lại và seed lại)
  + **TblUserSeeder**:

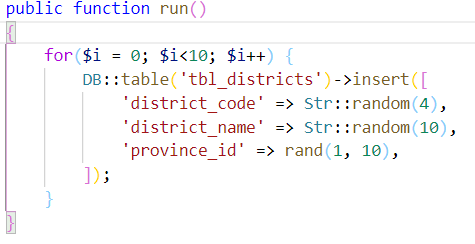




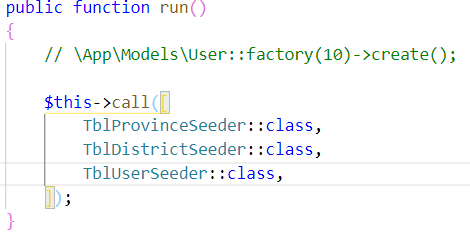
* + **TblProvinceSeeder**:



* + **TblDistrictSeeder**:



* + **DatabaseSeedeer**:



* **Đẩy dữ liệu lên View:**
  + Tạo *Class mới* để lưu trữ cấu trúc dữ liệu Tree
    - Tạo Folder CustomClass trong app/
    - Tạo các Class lưu trữ từng Level trong Tree:
      * *Company*:

<?php

namespace App\CustomClass;

class Company

{

    //Thuộc tính của Class phải có tên là text và children để đúng format hiển thị dữ liệu được

    public $text;

    public $children;

    //Getter, Setter

    function set\_province($children) {

        $this->children = $children;

    }

    function set\_text($text) {

        $this->text = $text;

    }

    function get\_province() {

        return $this->children;

    }

    //Định nghĩa hàm toArray() để dùng trong đoạn Script ở blade

    public function toArray()

    {

        return get\_object\_vars($this);

    }

}

* + - * *District, Province, Employee*: tương tự (đặt là Employee vì User bị trùng)
        + *Employee* thì chỉ có text thôi vì nó ko có con nữa
  + Tạo *Controller*:
    - Lệnh **Artisan**: php artisan make:controller MainController
    - Function showTraCuu() để hiển thị view HienThiTraCuu.blade.php

/\*\*

     \* Hiển thị màn hình Hiển thị tra cứu

     \*

     \*

     \* @return \Illuminate\View\View

     \*/

    public function showTraCuu()

    {

        return view('HienThiTraCuu');

    }

* + - Function **showQuanLy()** để hiển thị view QuanLyNhanVien.blade.php

public function showQuanLy()

    {

        //Tạo Tree

        $myTree = new Company;

        $myTree->set\_text('Công Ty');

        //Lấy Province từ Database

        $tblProvinces = TblProvince::all();

        //Lặp với từng Province: với mỗi Province thì tìm những District tương ứng của nó

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

            $provinces[$i] = new Province;  //Tạo Array provinces

            $provinces[$i]->set\_text($tblProvinces[$i]->province\_name);     //Set text cho nó là tên province

            //Lấy District từ Database: những District có province\_id = id của province này

            $tblDistricts = TblDistrict::where('province\_id', $tblProvinces[$i]->id)->get();

            //Lặp với từng District: với mỗi District thì tìm những User tương ứng của nó

            for($j = 0; $j < count($tblDistricts); $j++) {

                $districts[$j] = new District;  //Tạo Array districts

                $districts[$j]->set\_text($tblDistricts[$j]->district\_name); //Set text cho nó là tên district

                //Lấy User từ Database: những User có district\_id = id của district này

                $tblUsers = TblUser::where('district\_id', $tblDistricts[$j]->id)->get();

                //Lặp với từng User: với mỗi User thì cho vào 1 phần tử của Array và set text

                for($k = 0; $k < count($tblUsers); $k++) {

                    //Đặt tên Class là Employee vì User có rồi -> bị trùng

                    $employees[$k] = new Employee; //Tạo Array employees

                    $employees[$k]->set\_text($tblUsers[$k]->name);  //Set text cho nó là tên user

                }

                $districts[$j]->set\_user($employees);   //Set user cho District

            }

            $provinces[$i]->set\_district($districts);   //Set district cho Province

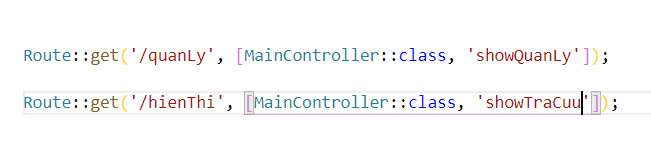
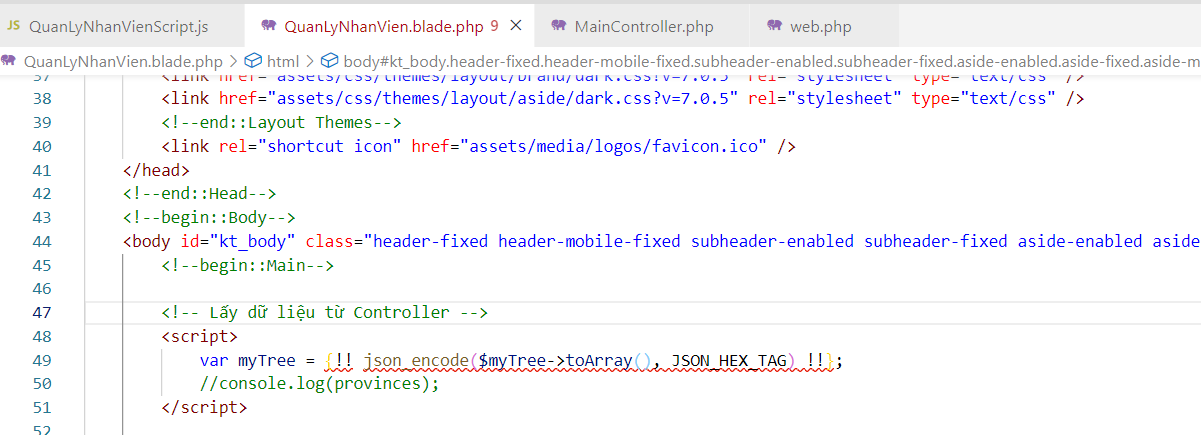
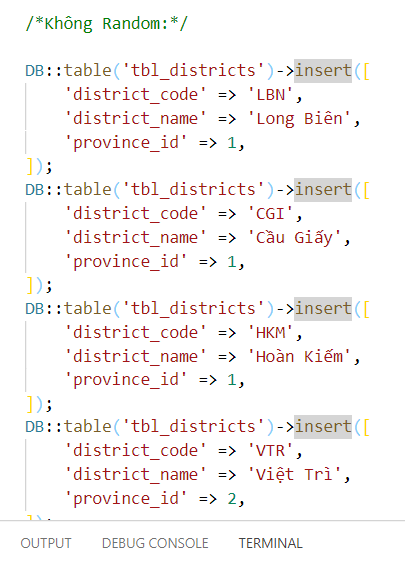
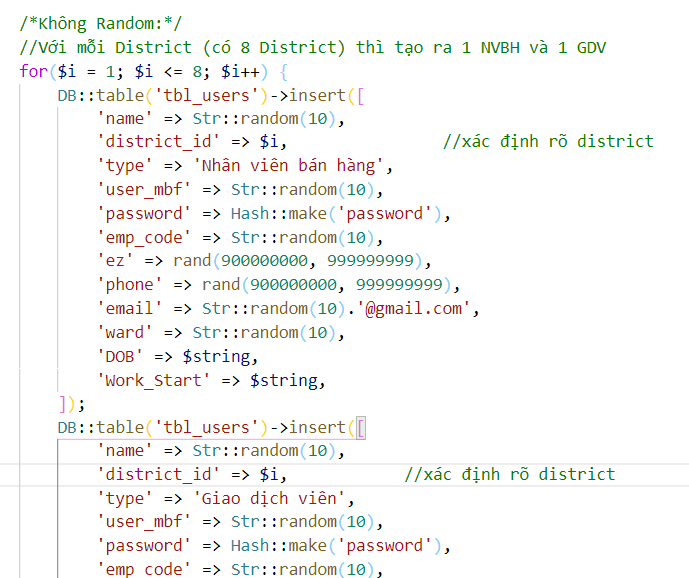
        }

        $myTree->set\_province($provinces);  //Set province cho Tree

        //Đẩy lên View

        return view('QuanLyNhanVien', compact('myTree'));

    }

* + *Route*: routes/web.php : Thêm đoạn này
    - 
  + *View*: Thêm đoạn Script để lấy dữ liệu từ Controller và chuyển sang JSON:
    - 
  + *JavaScript*: Lấy dữ liệu từ View để hiển thị lên
    - 
* **Sửa lại Seed**: Không Random nữa cho dễ nhìn:
  + 
  + 
  + 
* Kết quả: **Thành Công**

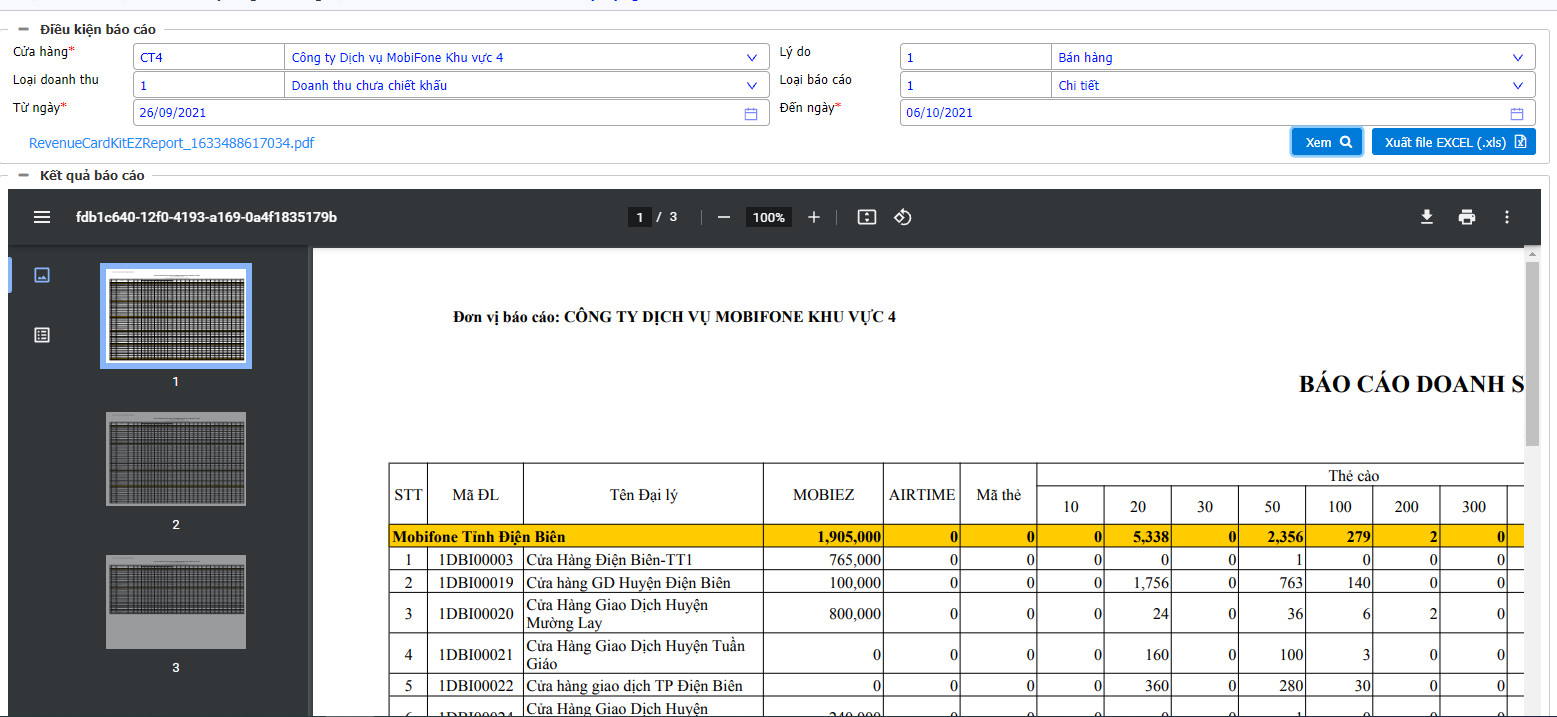


* **Fix Bug**: Khi các tỉnh trước có nhiều District hơn thì các tỉnh sau bị thừa → Thêm dòng này để Clear array trước khi tiếp tục vòng lặp
  + 

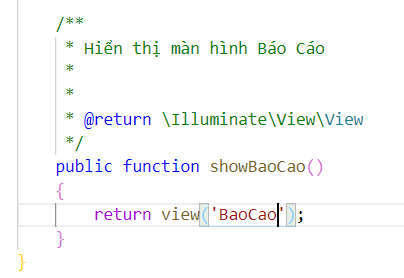
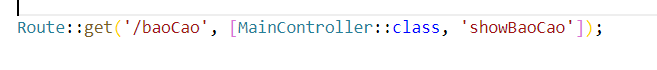
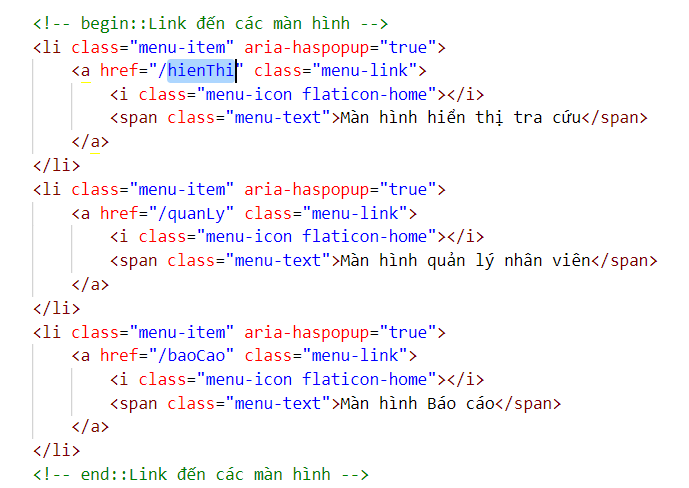
# NEW FUNCTION: Tạo chức năng mới

## **Requirement:**

View + Controller + DB: Xem báo cáo và tải Excel

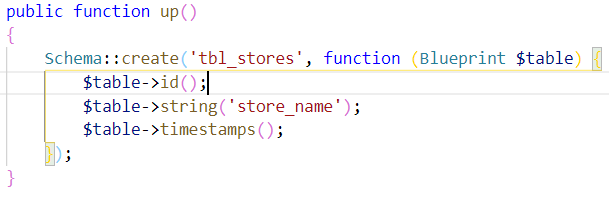
* View:
  + 
* Chọn gồm: **Tỉnh, ngày/tháng**, đánh giá, Loại báo cáo...
* Seed: 5000 dòng
* Nội dung:
  + Mỗi Attribute trong Table ở Database sẽ là 1 cái lựa chọn trong Form → Sau khi chọn xong hết và bấm [Xem] → Submit → Câu lệnh SELECT lấy ra những hàng dữ liệu thỏa mãn → Tự chuyển thành File PDF hiển thị bên dưới: Mỗi hàng trong Table ở DB sẽ là 1 hàng trong Bảng → Nếu bấm xuất File Excel: xuất ra Excel và cho phép Download về
* Sử dụng: Theme
  + *Form*
  + *KTDataTable*

## **Development:**

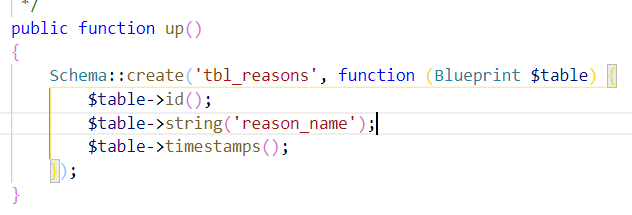
* ***View***: Design HTML
  + Dùng Project demo1\_HTML
  + Tạo File BaoCao.html từ file layout/general/empty-page.html trong **Theme**
  + Thêm Link đến màn hình mới
    - index.html [Line 174]
    - HienThiTraCuu.html [Line 174]
    - QuanLyNhanVien.html [Line 174]
    - BaoCao.html [Line 125]
  + Sửa Title
    - HienThiTraCuu.html [Line 19]
    - QuanLyNhanVien.html [Line 19]
    - BaoCao.html [Line 19]
  + BaoCao.html :
    - Line 40 : Link đến CSS
    - Line 4424 : Sửa BreadCrum
    - Line 4521 : *NỘI DUNG PAGE*
    - Line 5923 : Comment cái Sticky Toolbar lại cho đỡ vướng
    - Line 6279 : Link đến JavaScript
  + dist/BaoCaoScript.js
    - File tạo mới
  + dist/BaoCaoStyle.css
    - File tạo mới
* **[View]**: Chuyển sang Project Laravel:
  + Copy File BaoCaoScript.js sang public/assets/js
  + Copy File BaoCaoStyle.css sang public/assets/css
  + Copy File test.pdf sang public/assets
  + Tạo File BaoCao.blade.php
    - Copy nội dung sang
    - Sửa link đến CSS và JS
    - Sửa link đến File PDF
  + Tạo function link đến View trong app/Http/Controllers/MainController.php
    - 
  + Tạo Route đến function này trong routes/web.php
    - 
  + Sửa và thêm Link đến các màn hình của các Page View (hienThi, quanLy, baoCao)
    - 
  + Sửa Title các View
* ***Database***: Tạo Model, Seed
  + Model **TBLReport**: (Bảng chính – Báo cáo)
    - ID
    - Mã ĐL : dealerCode
    - Tên Đại lý : dealerName
    - MOBIEZ : mobiez
    - AIRTIME : airtime
    - Mã Thẻ : cardCode
    - Thẻ cào – 10 : card10
    - Thẻ cào – 20 : card20
    - Thẻ cào – 30 : card30
    - Thẻ cào – 50 : card50
    - Thẻ cào – 100 : card100
    - Thẻ cào – 200 : card200
    - Thẻ cào – 300 : card300
    - Thẻ cào – 500 : card500
    - Tỉnh : province\_id [Link đến **TblProvince**]
    - Cửa hàng : store\_id [Link đến **TblStore**]
    - Lý do : reason\_id [Link đến **TblReason**]
    - Loại doanh thu : revenueType\_id [Link đến **TblRevenueType**]
    - Loại báo cáo : reportType\_id [Link đến **TblReportType**]
    - Đánh giá : evaluation
    - Ngày : timestamps



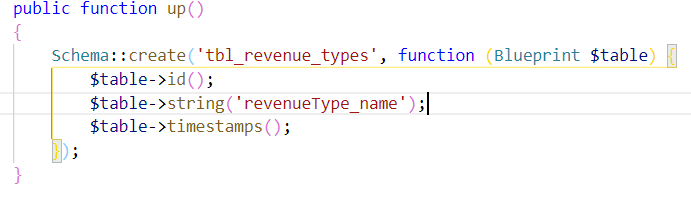
* + Model **TblProvince**: *(Đã có)*
  + Model **TblStore**:
    - ID
    - Tên cửa hàng : store\_name



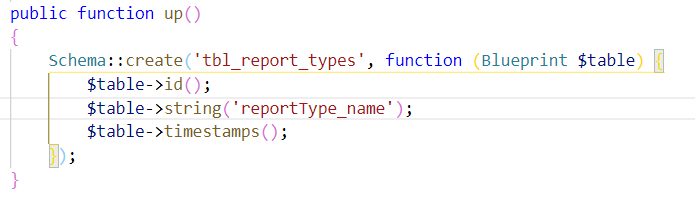
* + Model **TblReason**:
    - ID
    - Tên lý do : reason\_name

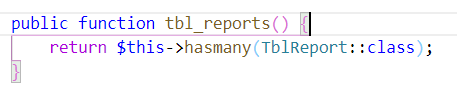


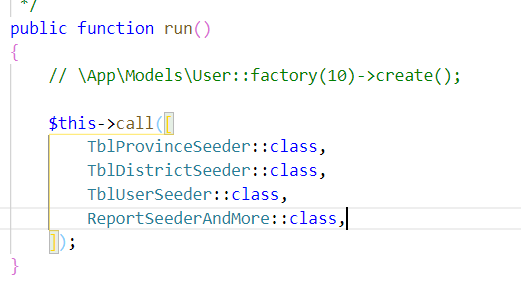
* + Model **TblRevenueType**:
    - ID
    - Tên loại doanh thu : revenueType\_name

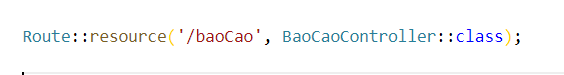
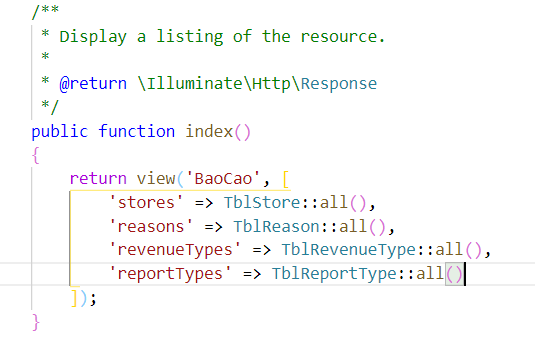
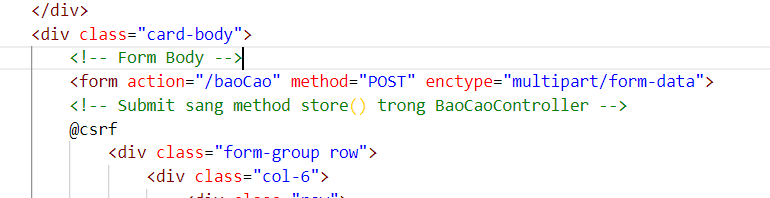
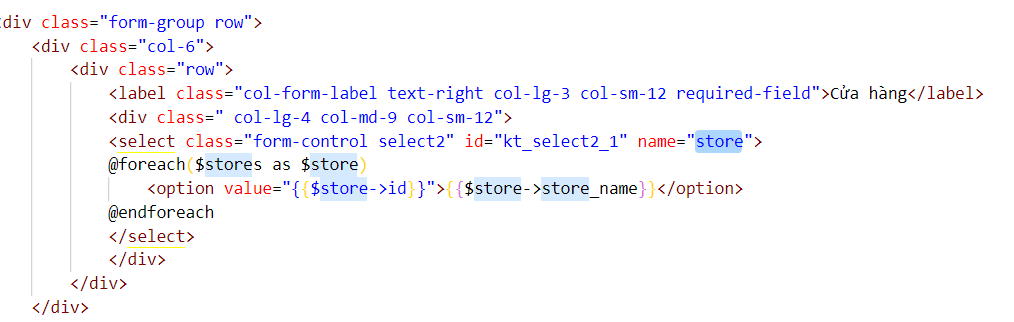
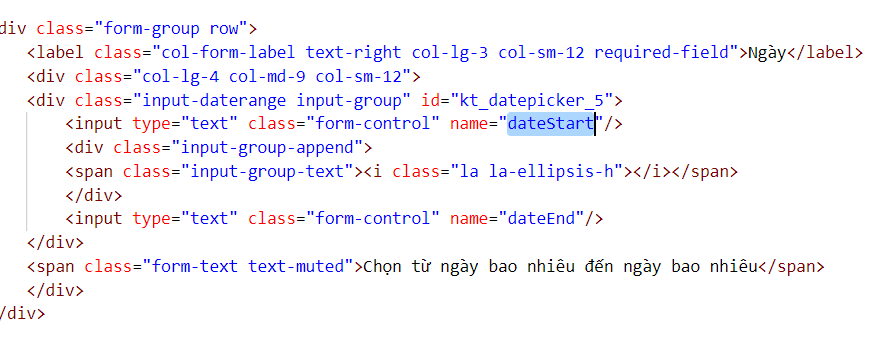


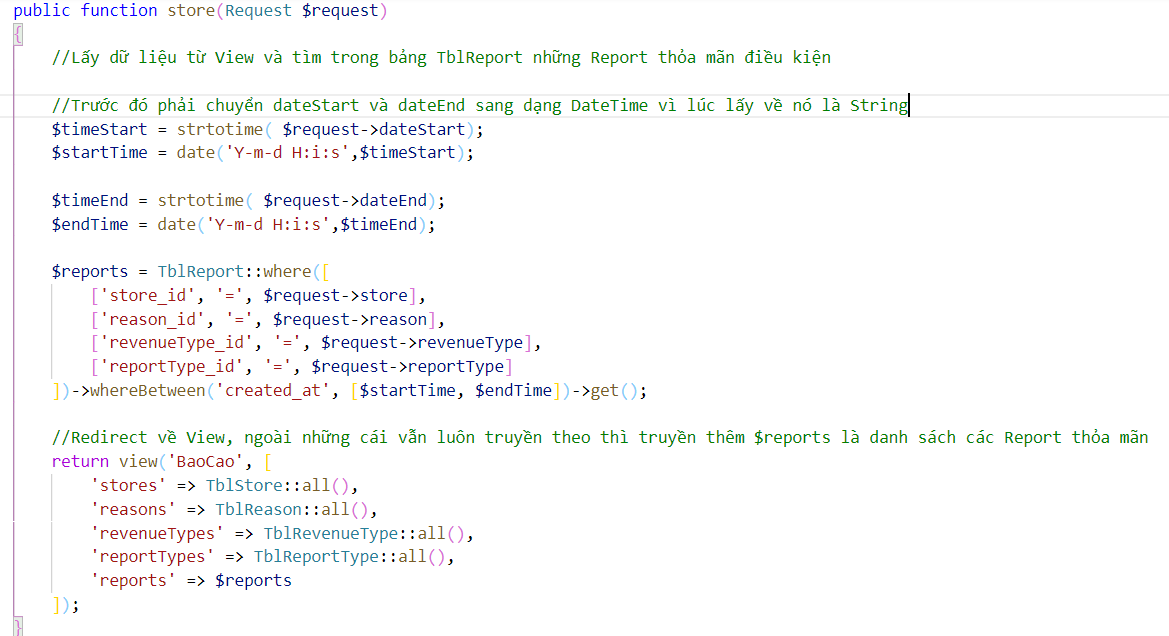
* + Model **TblReportType**:
    - ID
    - Tên loại báo cáo : reportType\_name

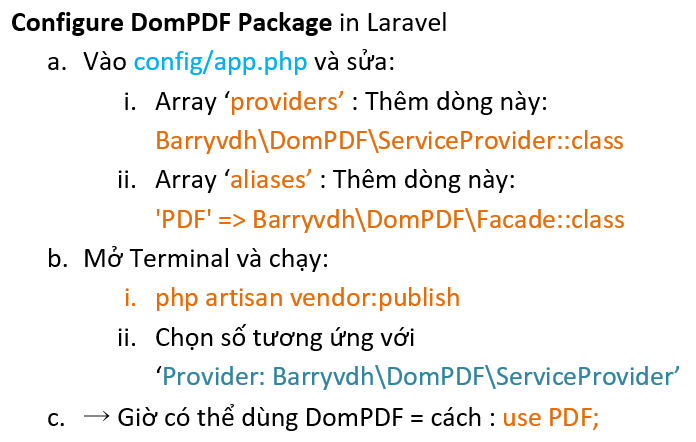
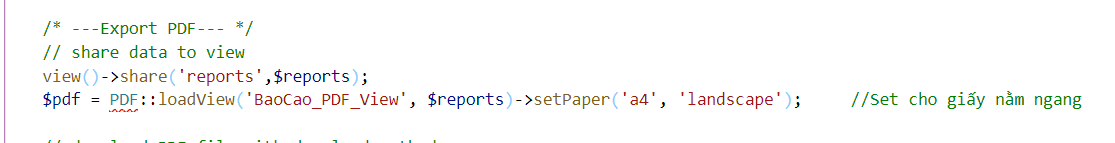
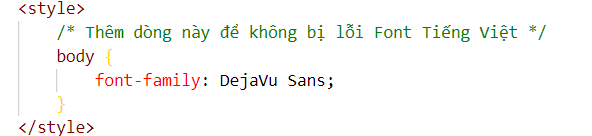
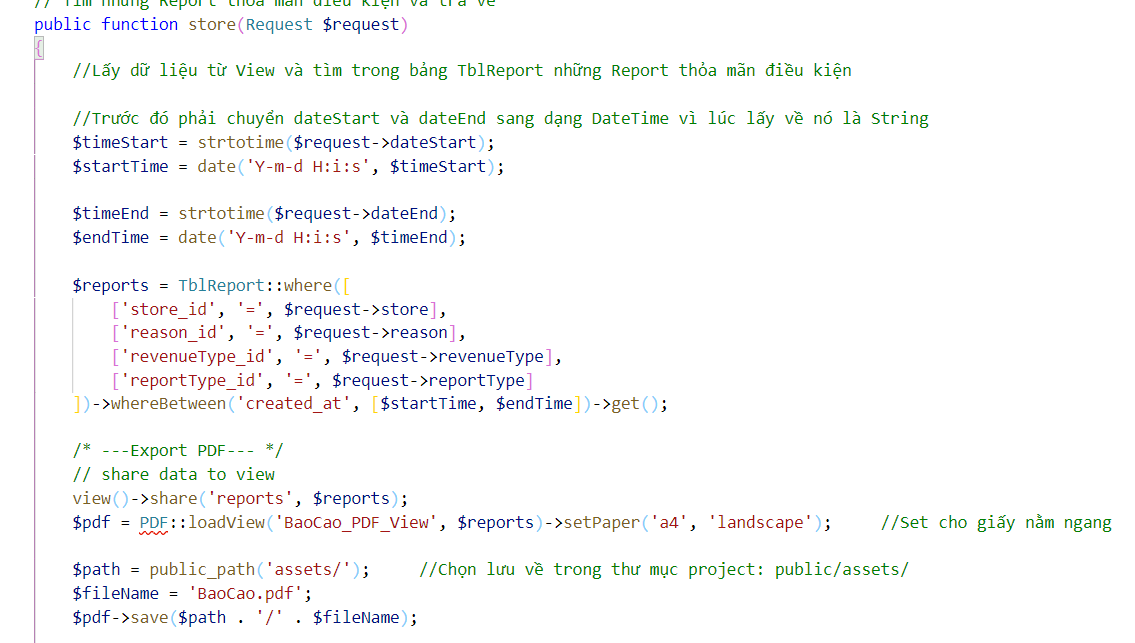
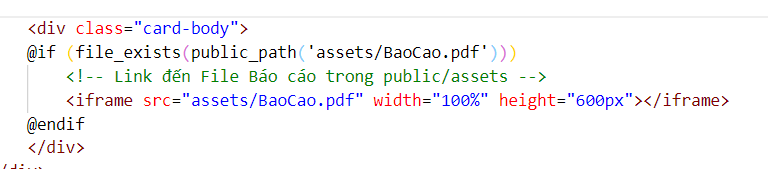
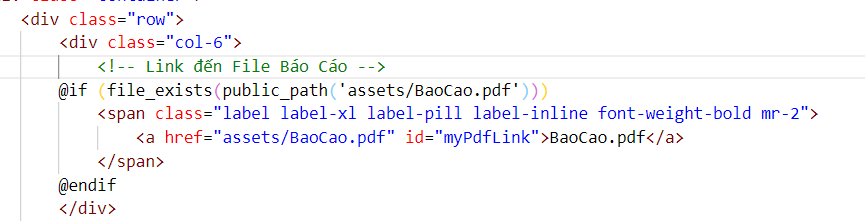
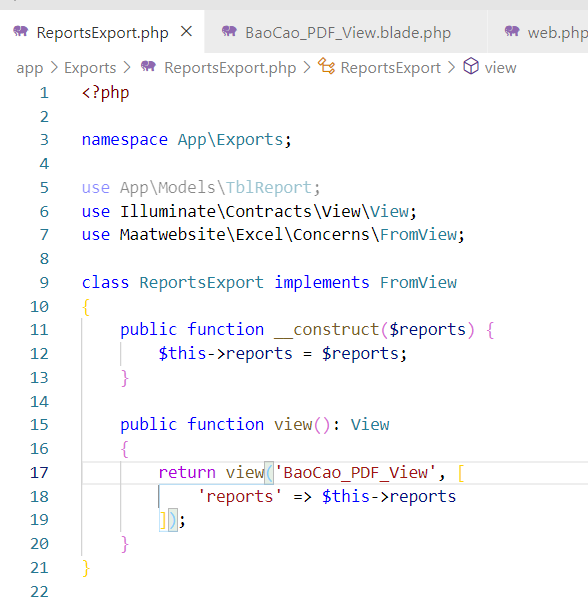
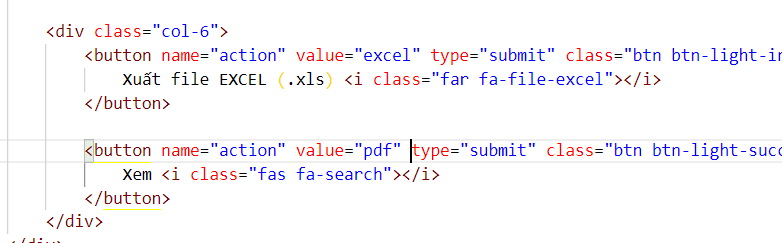
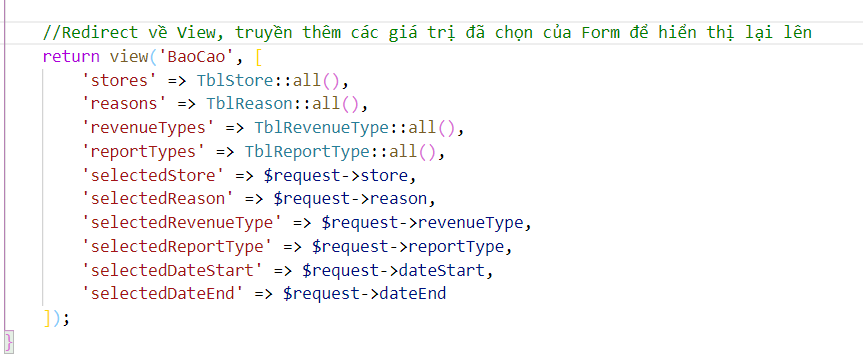
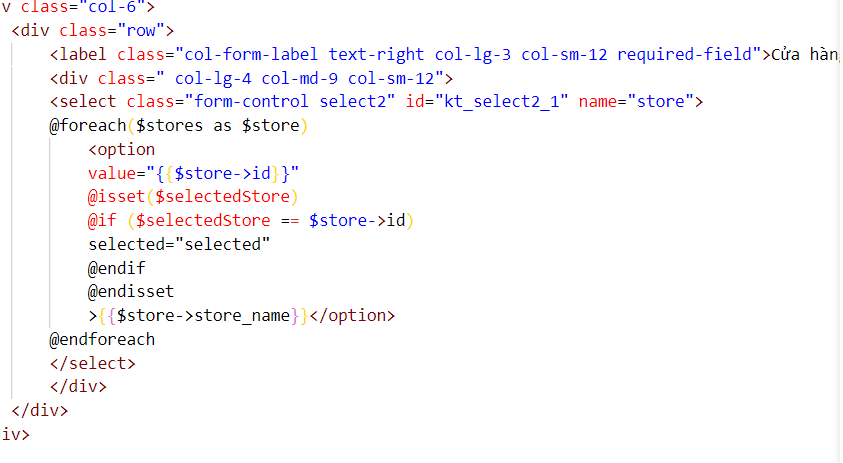
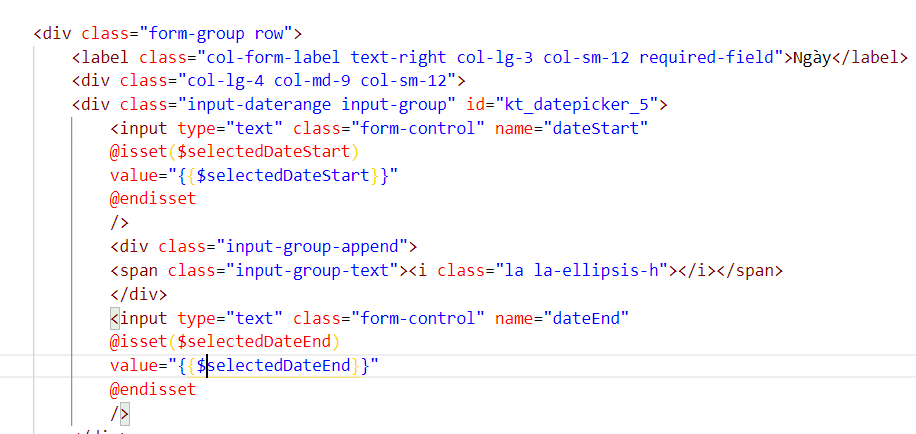
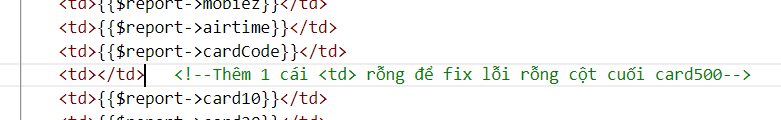


* + **Quan hệ**:
    - Các Model**: TblProvince, TblStore, TblReason, TblRevenueType, TblReportType:**
      * 
    - TblReport:
      * 
  + **Seed** : database/seeders/**ReportSeederAndMore**.php → Class này sẽ chứa Seed của tất cả các Model trên kia
    - Thêm lệnh gọi Seeder trong database/seeders/DatabaseSeeder.php



* + - Note: Seed cả ‘created\_at’ để sau này còn search theo ngày
* ***Controller***:
  + Tạo Controller mới để sử dụng cho Tính năng này:
    - Tạo app/Http/Controllers/**BaoCaoController**.php
    - Tạo Route cho nó trong routes/web.php
      * 
      * *(Resource Route với cú pháp mới của Laravel 8.)*
    - Xóa hàm *showBaoCao*() trong **MainController** cũng như Route của nó
  + BaoCaoController – *index*():
    - 
  + Sửa view để lấy dữ liệu từ Controller:
    - 
    - ...
  + **[Tính năng Search]:**
    - Form Submit:
      * 
    - Sửa attribute “name” trong các <input> để lấy dữ liệu về:
      * 
      * 
    - *store*() trong BaoCaoController:



* Xuất dữ liệu ra File **PDF**:
  + GitBash ở C:\Windows\System32\cmd.exe
    - composer require barryvdh/laravel-dompdf
  + 
  + Tạo File BaoCao\_PDF\_View.blade.php để lưu nội dung của File PDF
    - 
  + Set cho giấy nằm ngang
    - 
  + Chữa lỗi Font Tiếng Việt
    - 
  + Set cho File PDF tải về ở trong thư mục Project luôn, sau đó Redirect lại trang /baoCao : (Hàm store() trong BaoCaoController)
    - 
    - 
  + BaoCao.blade.php: Thêm phần kiểm tra xem nếu File PDF chưa tồn tại thì ko hiển thị → Tránh bị lỗi 404
    - 
  + Tương tự với Link trong Component [Điều kiện báo cáo]:
    - 
* Xuất dữ liệu ra File **Excel**:
  + Sửa File **php.ini** (Ở XAMPP Control Panel → Apache → Config)
    - Bỏ Comment extension gd
  + Install **Laravel-Excel**:
    - Như trong file Word **Laravel Export Excel**
  + Chạy câu lệnh **Terminal**:
    - php artisan make:export ReportsExport --model=TblReport
    - → Tạo file ReportsExport trong app\Exports
  + Code trong **ReportsExport**
    - 
    - Nhận về biến $reports và Xuất File Excel dựa theo *BaoCao\_PDF\_View* với $reports
  + **BaoCao.blade.php**
    - Giờ button [Xem] và [Xuất file EXCEL] đều Submit Form → trả về hàm store() trong Controller, NHƯNG có name là action và value khác nhau → Trong Controller Dựa vào đó để xử lý dữ liệu: quyết định xem nên Export PDF hay Export EXCEL
    - 
  + **BaoCaoController**:
    - Sửa:
    - 
* Thêm tính năng: *Lưu lại các giá trị đã chọn sau khi Submit*:
  + Khi return về view sau khi export pdf/excel: Gửi thêm dữ liệu về giá trị đã chọn:
    - 
  + Trên View: Hiển thị giá trị đã chọn:
    - 
    - 
* Sửa lỗi khi Export thì cột cuối (Card500) bị rỗng: Thêm 1 <td> rỗng là OKE
  + 
* **Thành Công!**

# ADD FUNCTION: NewsApp – Port1090

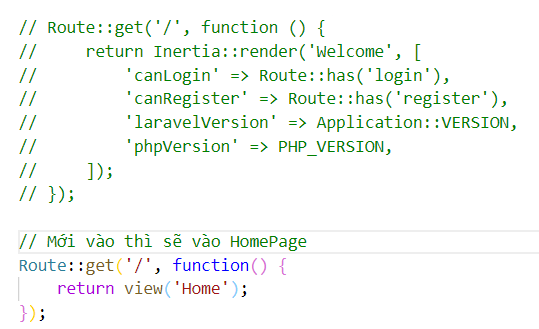
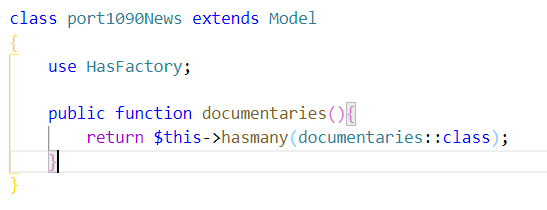
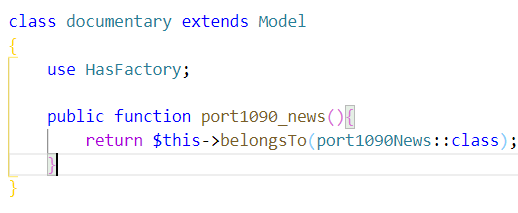
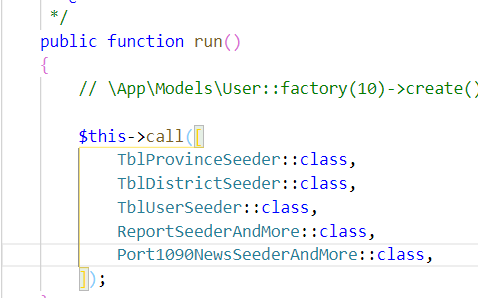
**[Requirement]** :

* Lấy chức năng Port1090 (CRUD, và cả Công văn) trong Project newsApp → Thêm vào *Project Mobifone*
* Design lại Front-End theo Theme cho đẹp

**[Development]** :

* **Plan**:
  + Tạo 1 trang Home Page để chứa link đến tất cả các trang (đơn giản thôi)

Sửa Route để khi mở localhost:8000 → Redirect đến trang Home này thay vì trang Welcome

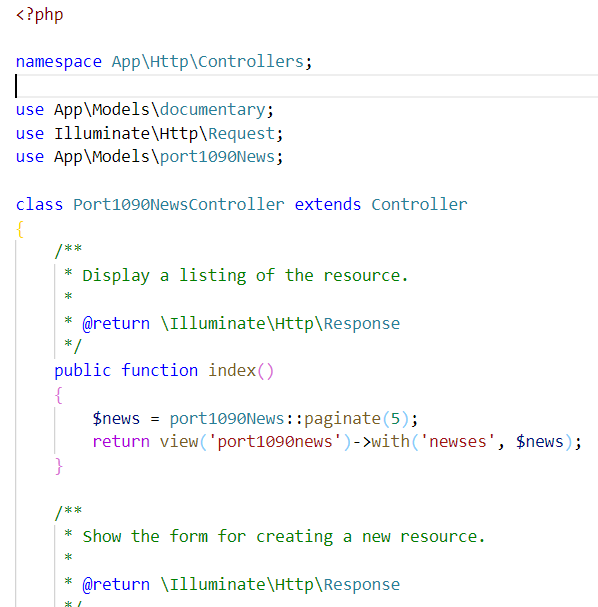
* + Port1090:
    - Tạo Model Database theo như Database cũ
    - Cứ chuyển tất cả các File (kể cả View) gốc sang đã, sao cho nó chạy được rồi chỉnh sửa lại View cho đẹp sau
* **HomePage**:
  + Tạo trang: resources/views/Home.blade.php (Có phần khung của 1 trang trắng trong Theme để dùng Style)
    - 
  + Sửa Route:
    - 
  + Code trang Home:
    - 
* **Model - DB**:
  + Model *port1090News*:
    - 
    - 
  + Model *documentary*:
    - 
    - 
  + Để làm Seed:
    - Tạo Folder images trong public và copy ảnh noimage.png vào
    - Tạo Folder files trong public và copy file NOFILE.pdf vào
  + Tạo Seed: Port1090NewsSeederAndMore
    - 
  + Gọi đến Seed trong DatabaseSeeder
    - 
* **Route**:
  + Copy Route cho Port1090 sang routes/web.php, sửa lại Typo vì project mới là Laravel 8.0



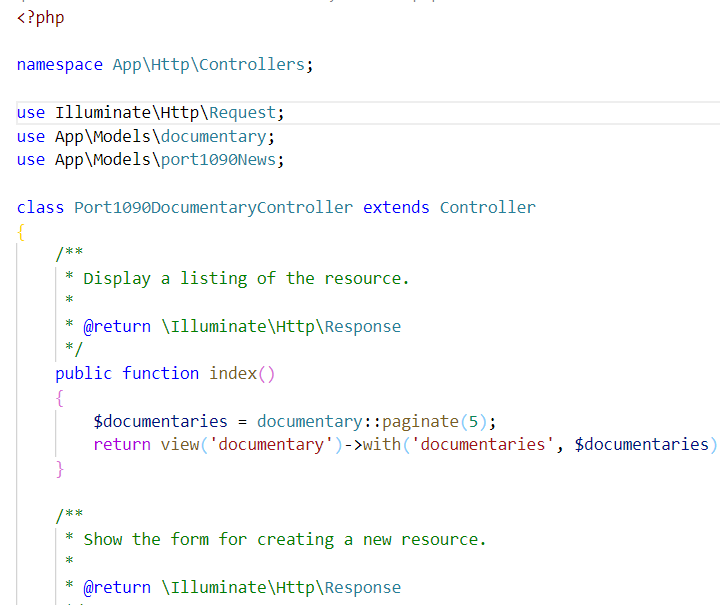
* **Controller**: Tạo các Controller (resource) và Copy nội dung sang:
  + *Port1090HomeController*:



* + *Port1090NewsController*:



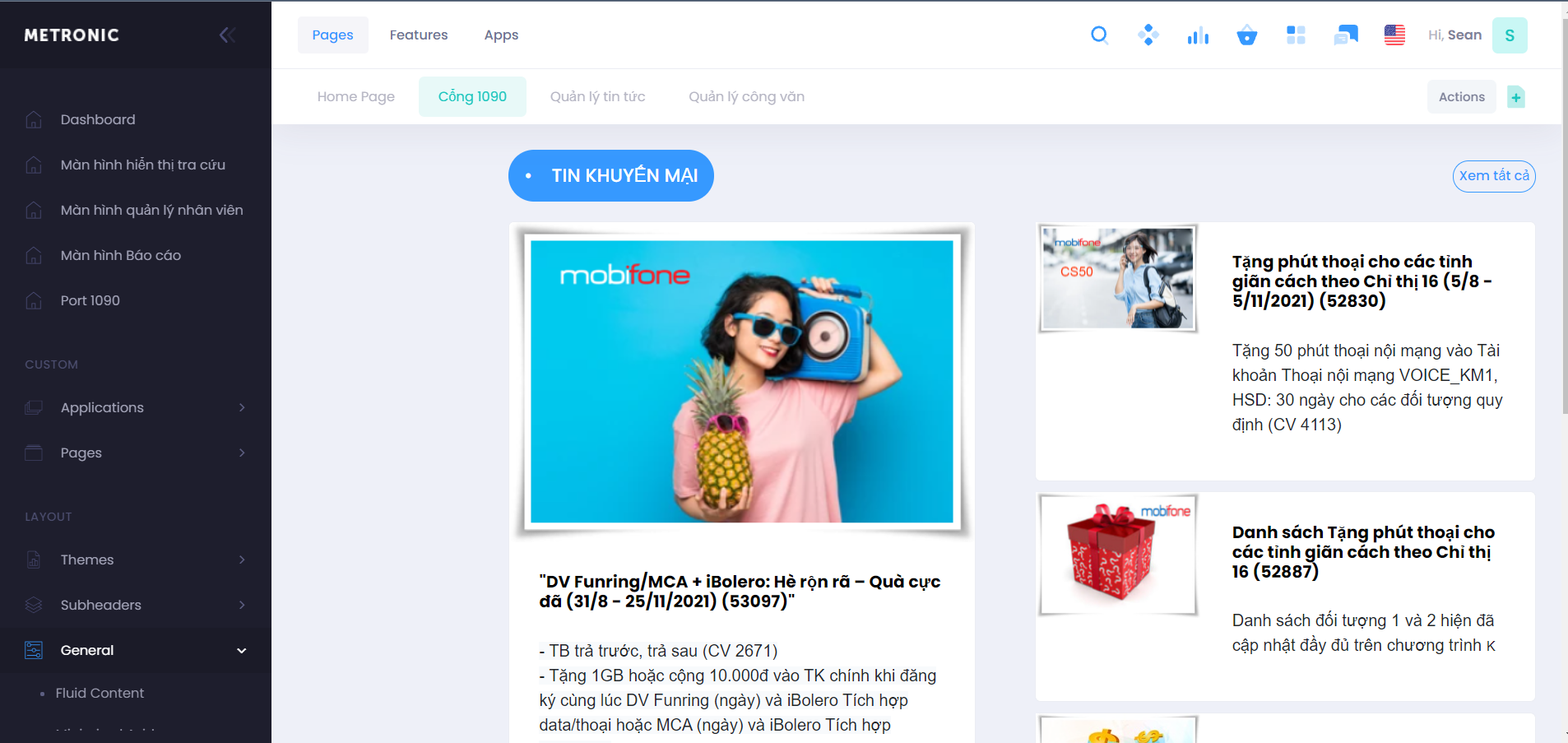
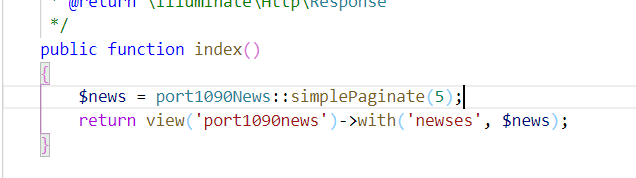
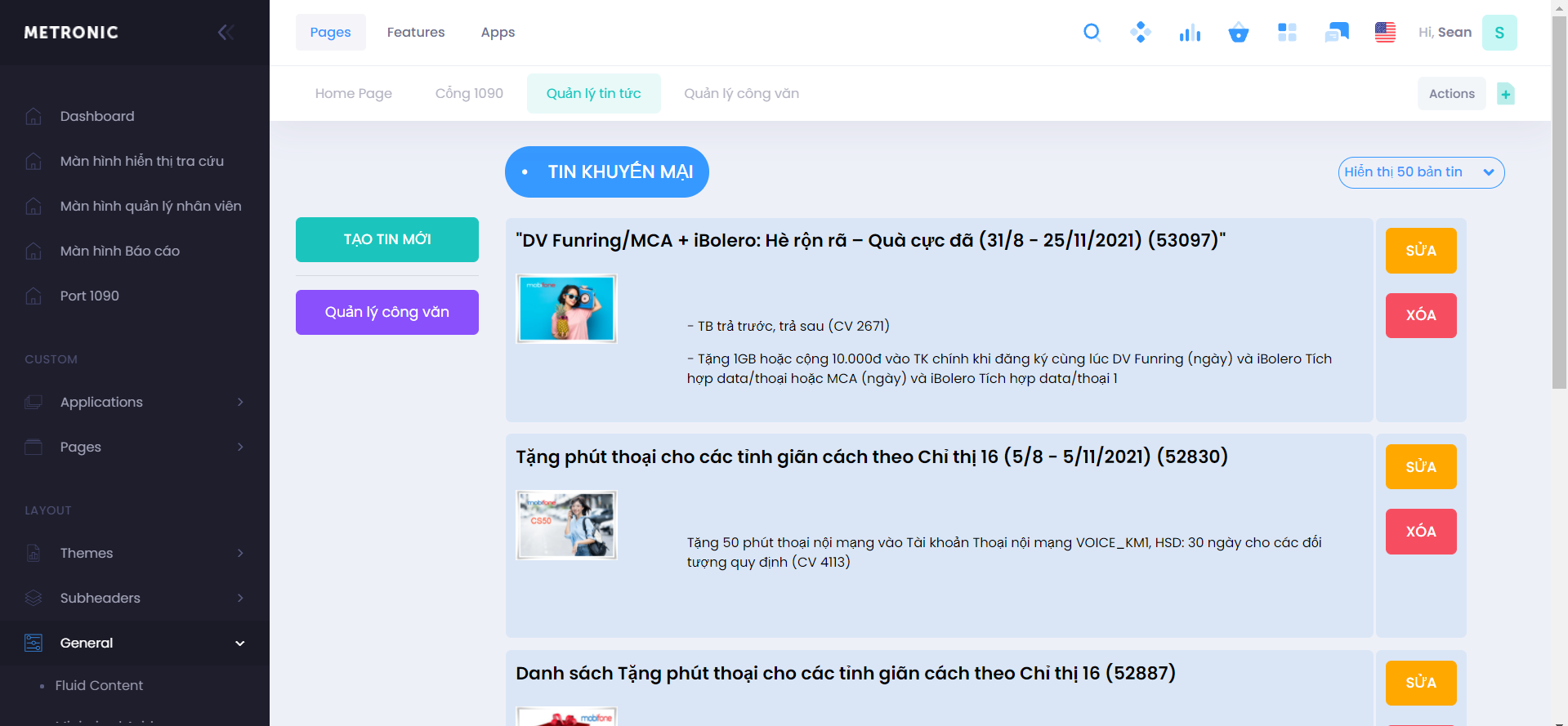
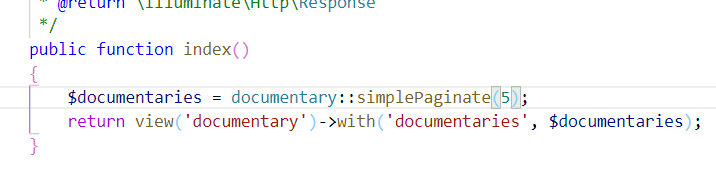
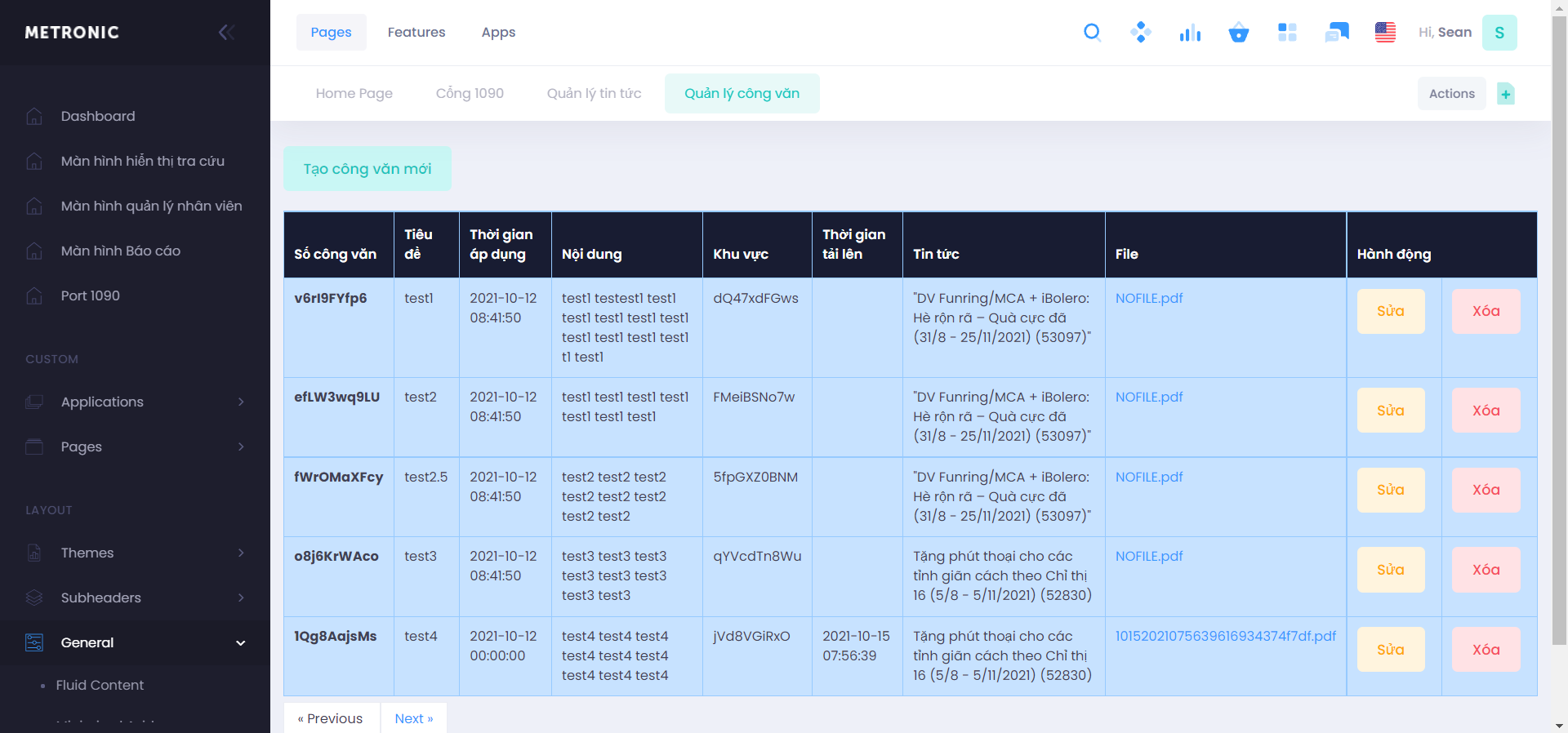
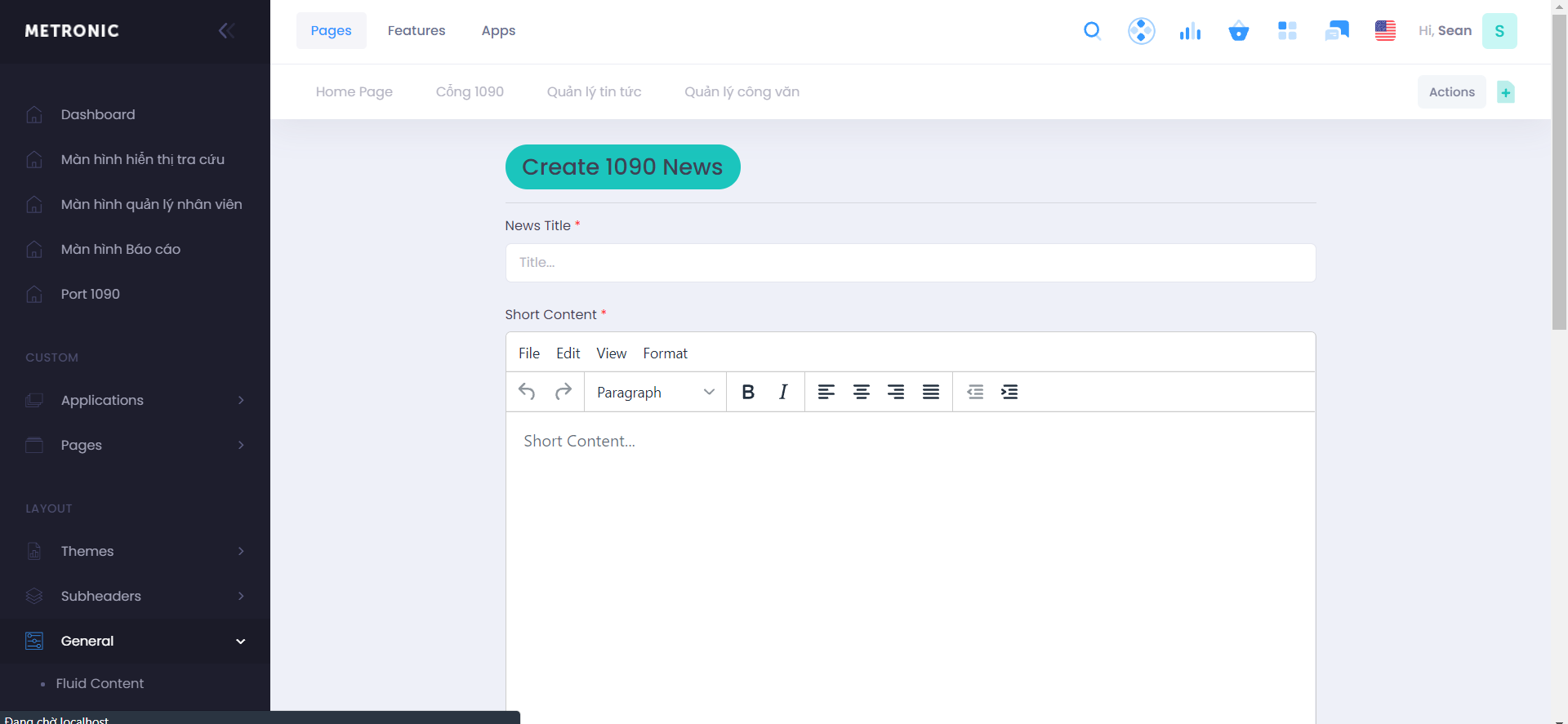
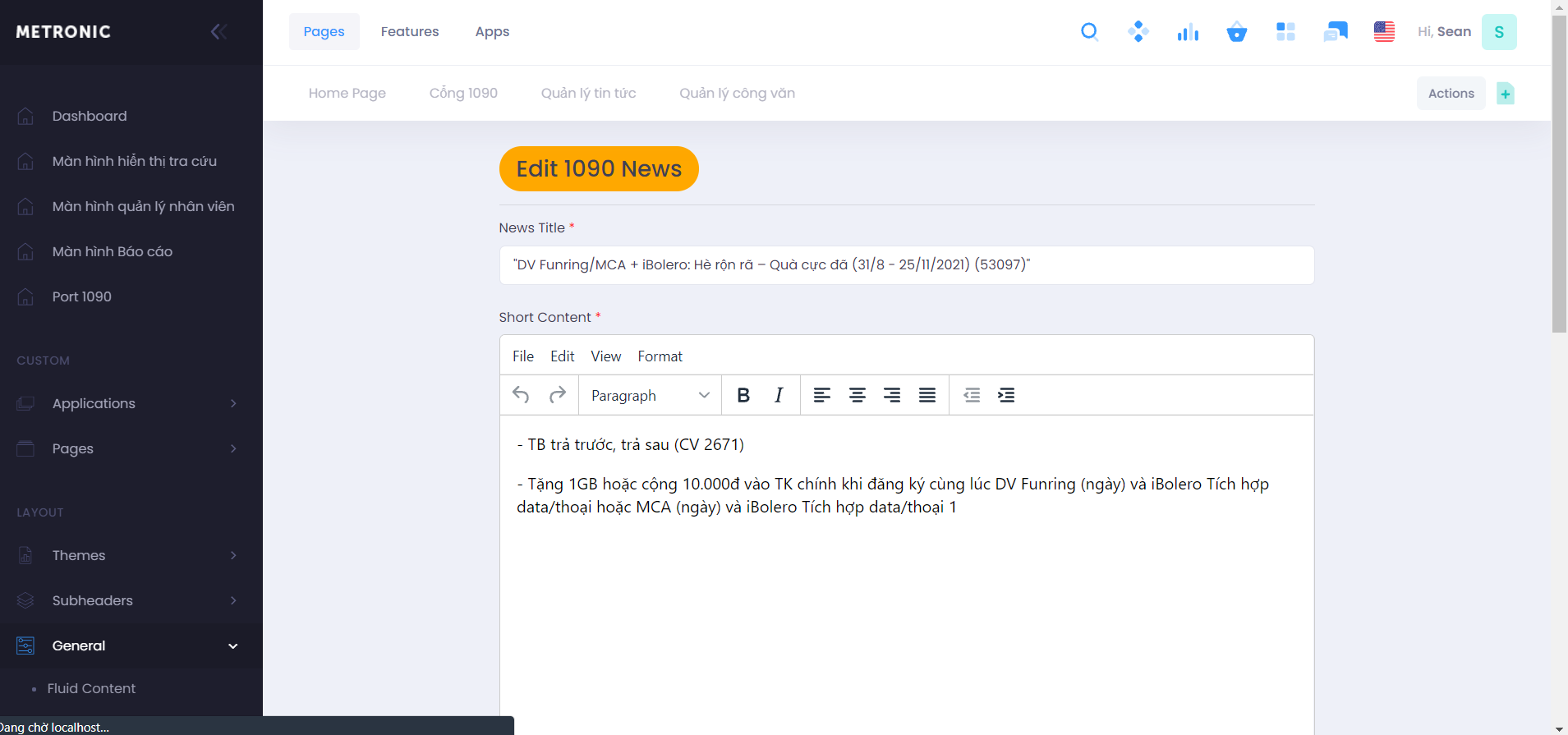
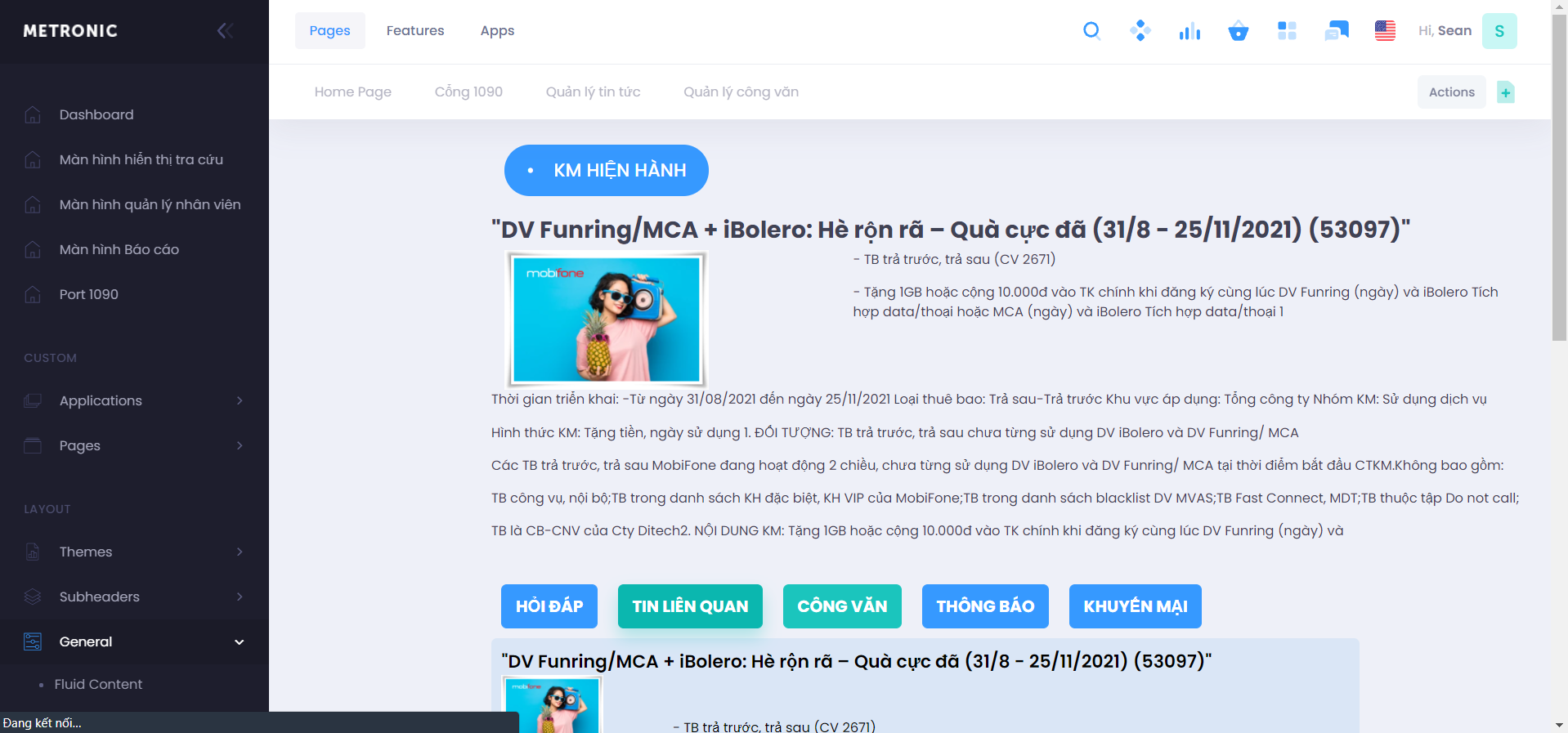
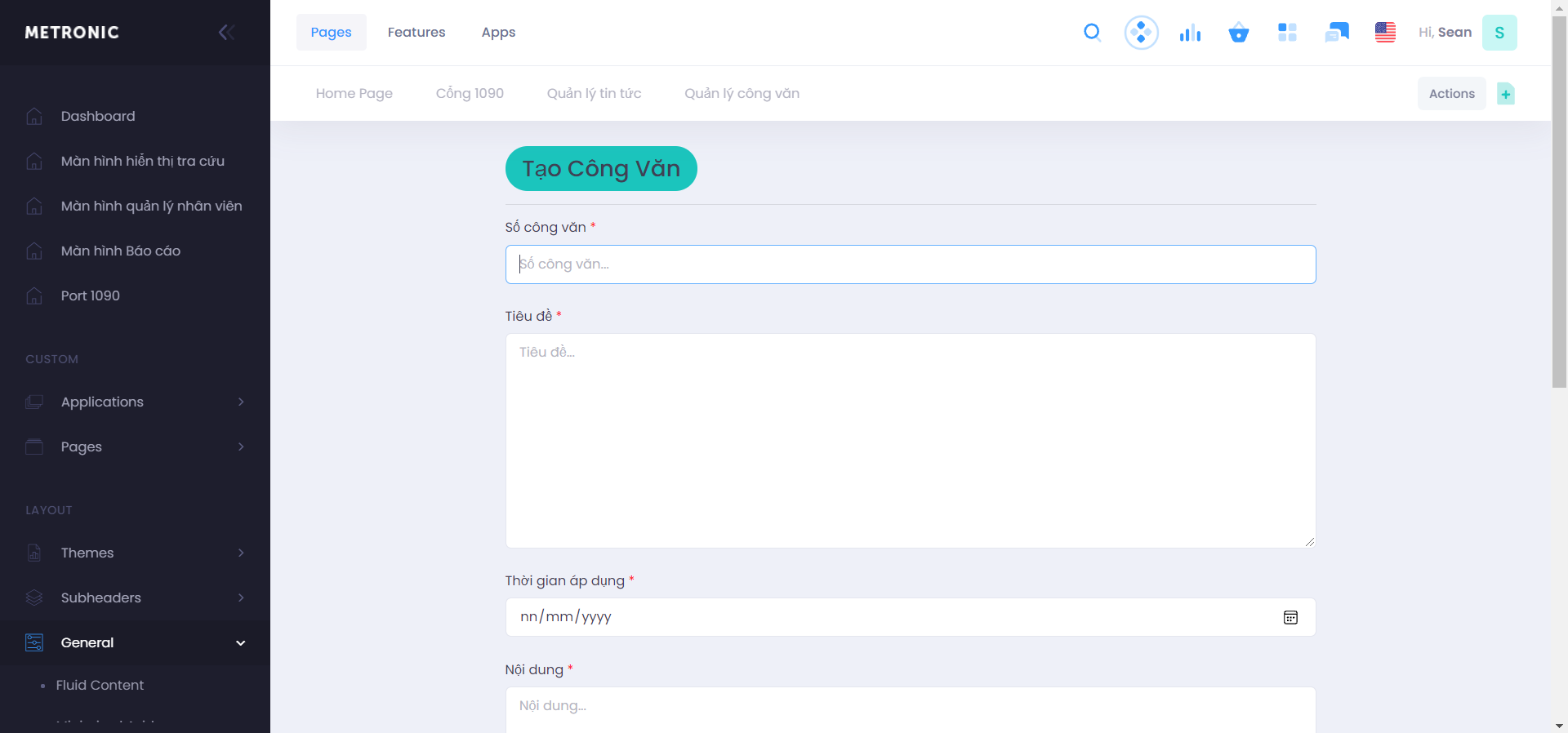
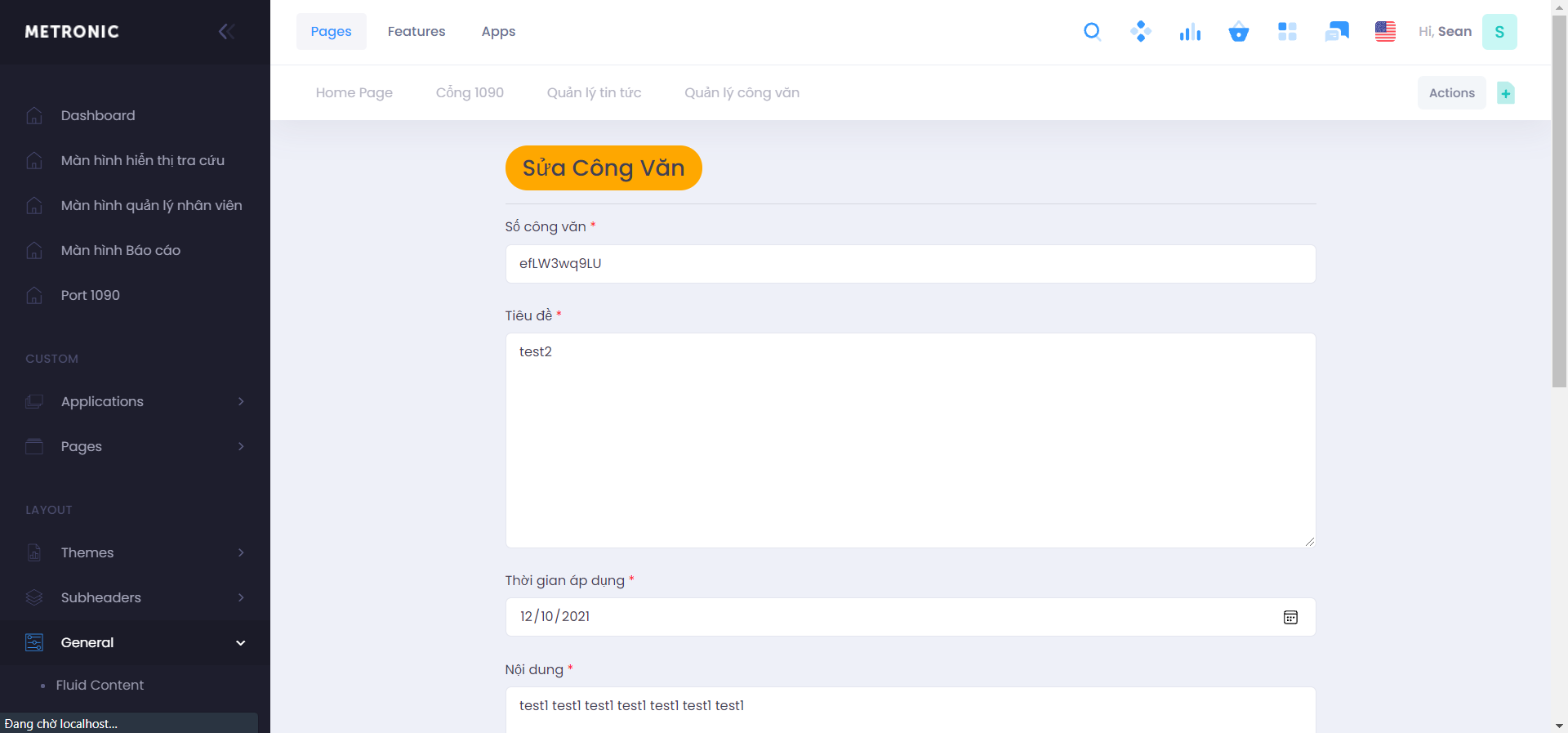
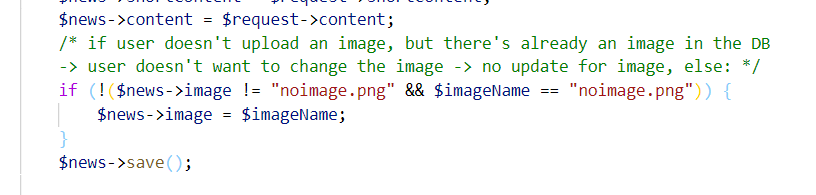
* + *Port1090DocumentaryController*:



* + Khi Copy sang phải sửa lại như:

use App\documentary; sửa thành use App\Models\documentary;

Vì sự khác biệt giữa 2 phiên bản Laravel

* **View**: Copy các File View sang
  + .blade.php :
    - create1090news
    - createDocumentary
    - documentary
    - edit1090news
    - editDocumentary
    - port1090home
    - port1090news
    - port1090newsDetail
  + .css :
    - custom
  + Sửa các trang blade : thay vì @extends(‘layouts.app’)... thì cho nó làm trang *HTML* thông thường, copy code từ trang app.blade.php của Project **newsApp**
    - Lỗi: Hỏng Style! → Thực ra cx ko quan trọng vì đằng nào mình cx sửa lại Front-End → Kệ nó
* **Sửa lại Front-End** các trang:
  + Cho vào Khung trang của Theme : Code – Line 4520
    - Thêm link đến custom.css
    - Sửa <title> các trang – cho giống như tên File
    - Thêm Link đến port1090home ở bên thanh Sidebar
      * Line 145
      * Thêm ở tất cả các trang
  + Ở mỗi trang: Thêm cái **Nav-bar** trên đầu:
    - Các *Link*:
      * Link về Home Page
      * Link đến trang port1090home : Cổng 1090
      * Link đến trang port1090news : Quản lý tin tức
      * Link đến trang documentary : Quản lý công văn
    - Chú ý để nav-link active cho phù hợp
    - Line 4431
  + Sửa lại Style từng trang
    - Sửa các trang và sửa custom.css
  + port1090home:
    - 
  + port1090news:
    - Sửa Lỗi paging
      * 
      * [*Port1090NewsController*] Chuyển sang simplePaginate() thì nó ko lỗi nữa
    - 
  + documentary:
    - Sửa lỗi Paging: Tương tự: [*Port1090DocumentaryController*]
      * 
    - 
  + create1090news:
    - 
  + edit1090news:
    - 
    - Sửa lỗi bắt buộc phải upload lại image: [*Port1090NewsController*]
      * 
  + port1090newsDetail
    - 
  + createDocumentary
    - 
  + editDocumentary
    - 
      * Sửa lỗi bắt buộc phải Upload lại File:
        + 
        + [*Port1090DocumentaryController*]
* Sửa lỗi Style Paging
  + <https://laravel.com/docs/8.x/pagination#using-bootstrap>
* Upload Code lên GitHub → Ghi thông tin thay đổi trên Commit thay vì phải chép lại ở đây