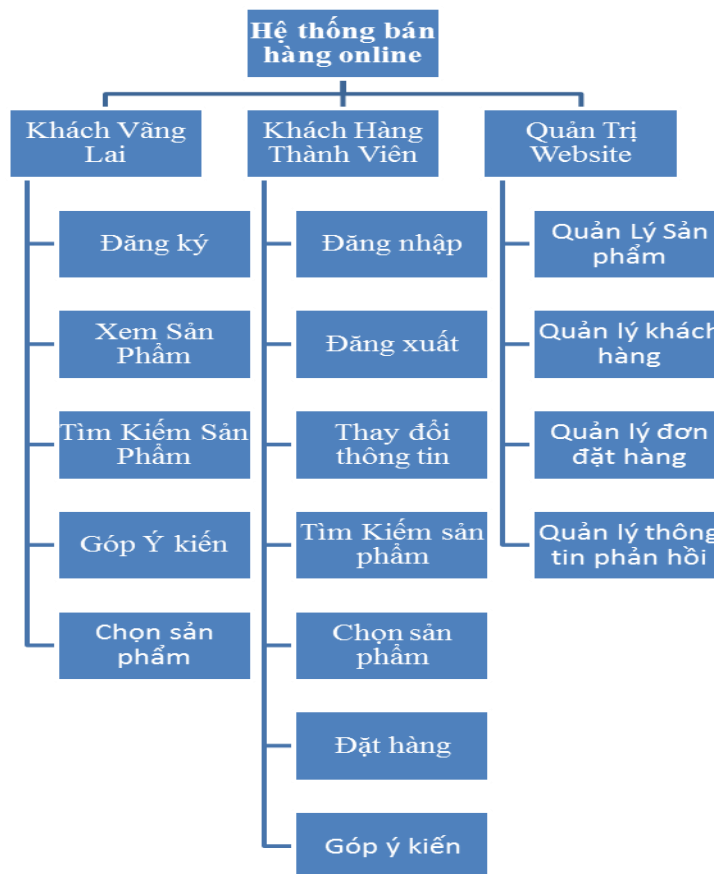


Buổi 01: Khảo sát, phân tích cấu trúc Database và tạo các trang cho đề án

1.1./ Mô hình phân rã chức năng



1.2/ Tạo thư mục tên DoAn vào ổ đĩa làm bài, Download tập tin Layout.rar chép vào thư mục DoAn và giải nén tập tin này.

1.3./ Tạo CSDL tên là db_banhang và import tập tin db_banhang.sql vào CSDL vừa tạo.

1.4./ Tạo Project laravel và cấu hình như sau:

Mở file .env trong Laravel và khai báo theo code sau:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_banhang
DB_USERNAME=root
DB_PASSWORD=
```

Gõ vào tên CSDL đã tạo ở trên

Mở file **Config\Database.php** và chỉnh **Charset** và **Collation** theo đúng csdl db_banhang

```
'mysql' => [
    'driver' => 'mysql',
    'url' => env('DATABASE_URL'),
    'host' => env('DB_HOST', '127.0.0.1'),
    'port' => env('DB_PORT', '3306'),
    'database' => env('DB_DATABASE', 'forge'),
    'username' => env('DB_USERNAME', 'forge'),
    'password' => env('DB_PASSWORD', ''),
    'unix_socket' => env('DB_SOCKET', ''),
    'charset' => 'utf8',
    'collation' => 'utf8_general_ci',
    'prefix' => '',
    'prefix_indexes' => true,
    'strict' => true,
```

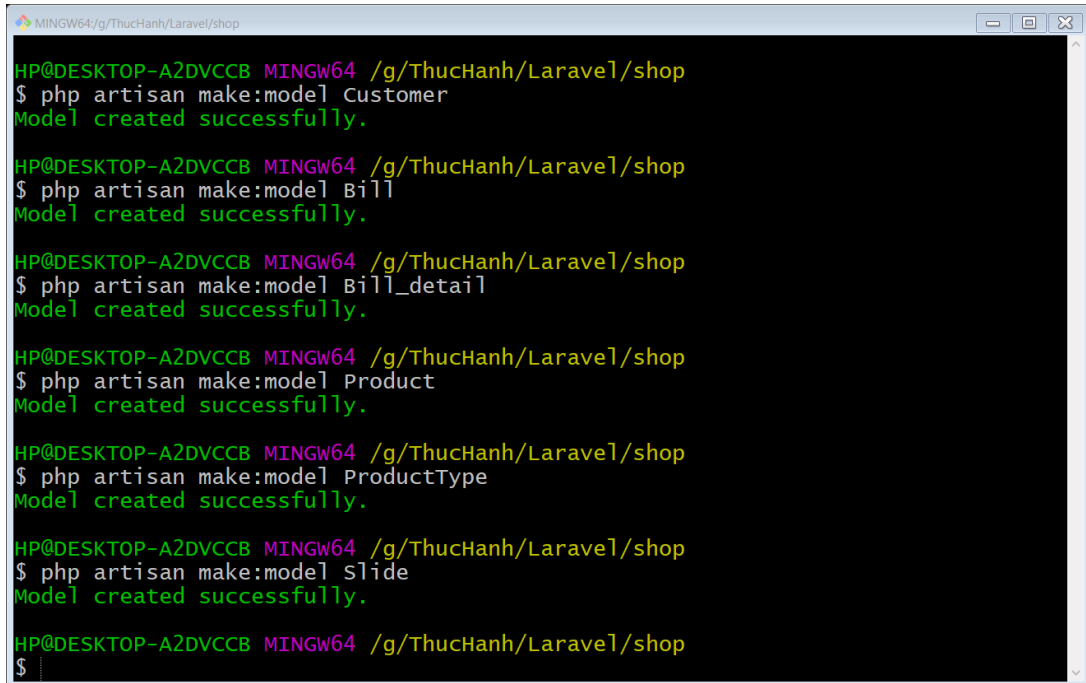
Mở file **Config/app.php** và khai báo múi giờ như sau:

```
'timezone' => 'asia/Ho_Chi_Minh',
```

1.5./ Tạo Model, liên kết các model

➤ Tạo Model bằng lệnh: Mở cửa sổ Cmd và gõ lệnh:

php artisan make:model <model_name>



```
HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model Customer
Model created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model Bill
Model created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model Bill_detail
Model created successfully.

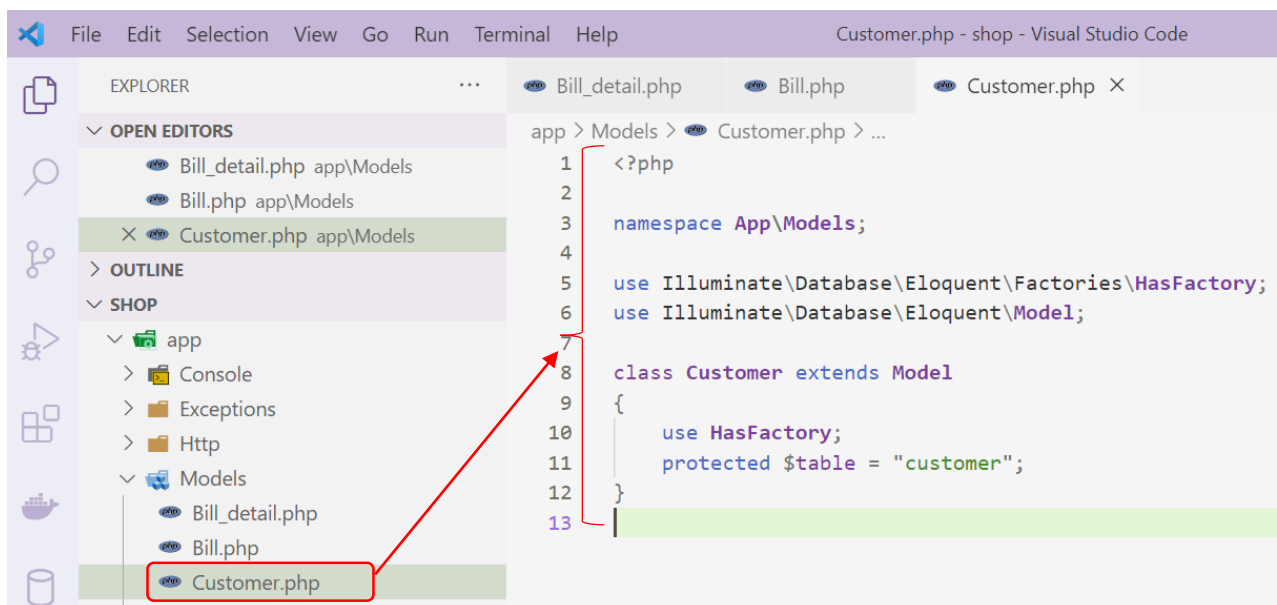
HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model Product
Model created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model ProductType
Model created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:model Slide
Model created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$
```

➤ Khai báo các Models gắn với các bảng tương ứng: **protected \$table = "table_name"** ;



Tương tự khai báo cho các Model khác

1.6/ Khai báo mối quan hệ cho các Models

- Một loại sản phẩm có nhiều sản phẩm và một sản phẩm chỉ thuộc 1 loại sản nào đó.
- Một sản phẩm được hiển thị ở nhiều chi tiết của đơn đặt hàng nào đó và 1 chi tiết chỉ có sản phẩm nào đó.
- Một đơn đặt hàng có nhiều chi tiết đặt hàng và 1 chi tiết chỉ thuộc 1 đơn đặt hàng nào đó.
- Một khách hàng có nhiều đơn đặt hàng và 1 đơn đặt hàng chỉ thuộc 1 khách hàng nào đó.

Thực hiện:

➤ Mở Model ProductType

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TypeProduct extends Model
{
    use HasFactory;
    protected $table = "type_products";

    public function Product()
    {
        return $this->hasMany(Product::class, "id_type", "id");
    }
}
```

➤ Mở Model Customer

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Customer extends Model
{
    use HasFactory;
    protected $table = "customer";

    public function bill()
    {
        return $this->hasMany(Bill::class, "id_customer", "id");
    }
}
```

➤ Mở Model Product

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;
    protected $table = "products";

    public function TypeProduct()
    {
        return $this->belongsTo(TypeProduct::class, "id_type", "id");
    }

    public function BillDetail()
    {
        return $this->hasMany(BillDetail::class, "id_product", "id");
    }
}
```

➤ Mở Model Bill_detail

```
<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class BillDetail extends Model
{
    use HasFactory;
    protected $table = "bill_detail";

    public function Product()
    {
        return $this->belongsTo(Product::class, "id_product", "id");
    }

    public function Bill()
    {
        return $this->belongsTo(Bill::class, "id_bill", "id");
    }
}
```

➤ Mở Model Bill

```
<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Bill extends Model
{
    use HasFactory;
    protected $table = "bills";

    public function BillDetail()
    {
        return $this->hasMany(BillDetail::class, "id_bill", "id");
    }
    public function Customer()
    {
        return $this->belongsTo(Customer::class, "id_customer", "id");
    }
}
```

1.7./ Tạo trang chung và Cấu Layout Trang Chủ

Bước 01: thực hiện tiếp các công việc sau:

- Tạo thư mục frontend vào thư mục public
- Sao chép thư mục assets trong thư mục DoAn vào thư mục frontend.
- Sao chép thư mục image trong thư mục DoAn vào thư mục frontend.

Bước 02: Trong thư mục **views** của Laravel tạo thư mục mới tên là **layouts** và trong thư mục Layouts tạo 1 view tên là **master.blade.php** và mở file **index.html** trong thư mục source rồi sao chép toàn bộ code của nó sang **master.blade.php**.

Bước 04: Hiệu chỉnh đường dẫn trong file master.

- Thêm thẻ `<base href="{{asset('')}}"/>` ngay phía dưới thẻ `Title`.
- Thêm `"frontend/"` vào trước các đường dẫn đến file `css`, `js` và `hình ảnh` trong trang.

Bước 05: Tạo 1 controller tên là PageCongtroller bằng lệnh:

php artisan make:controller PageController

```

MINGW64/g/ThucHanh/Laravel/shop
HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$ php artisan make:controller Pagecontroller
Controller created successfully.

HP@DESKTOP-A2DVCCB MINGW64 /g/ThucHanh/Laravel/shop
$

```

Bước 06: Tạo 1 folder mới trong views tên là **page**, trong folder page tạo một view tên là **trangchu.blade.php**.

Bước 07: Mở trang `master.blade.php`, chọn từ thẻ `<div class="rev-slider"> ... </div>` đến `<div class="container"> ... </div>` và paste sang `trangchu.blade.php` và sửa code lại như sau:

```

@extends("master")
@section("content")
<div class="rev-slider"> ...
</div>
<div class="container"> ...
</div> <!-- .container -->
@endsection

```

Sửa lại thành `@extends ("layout.master")`

Bước 08: Mở trang `master.blade.php`, chọn cắt phần header vào trang `header.blade.php` và cắt phần footer vào trang `footer.blade.php`.

- `header.blade.php`

```

<div id="header"> ...
</div> <!-- #header -->

```

- `footer.blade.php`

```

<div id="footer" class="color-div"> ...
</div> <!-- #footer -->
<div class="copyright"> ...
</div> <!-- .copyright -->

```

Bước 09: hiệu chỉnh code file `master.blade.php` như sau:

```

<body>
    @include("header")
    @yield("content")
    @include("footer")

```

Bước 10: Mở file **Pagecontroller.php**, tạo function truy cập trang chủ:

```
class Pagecontroller extends Controller
{
    //test
    public function getIndex()
    {
        return view("page.trangchu");
    }
}
```

Và vào routes/web.php:

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Pagecontroller;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', function () {
    return view('welcome');
});

Route::get("/index",[Pagecontroller::class, "getIndex"]);
```

1.8./ Cắt các trang còn lại

1) Cắt trang loại sản phẩm (view loi_sanpham.blade.php)

Bước 01: Mở tập tin **type_product.html**, cho các phần theo mẫu sau:

```
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
```

Bước 02: trong thư mục page, tạo file **loi_sanpham.blade.php** và paste đoạn code vào rồi code lại theo mẫu:

```
@extends("master")
@section("content")
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
@endsection
```

Bước 03: Mở Pagecontroller.php thêm function:

```
public function getLoaisanpham()
{
    return view("page.loai_sanpham");
}
```

Bước 04: Mở tập tin **Web.php** thêm route sau:

```
Route::get("/loai-san-pham", [Pagecontroller::class, "getLoaisanpham"]);
```

2) Cài trang chi tiết sản phẩm (chitiet_sanpham.blade.php)

Bước 01: Mở tập tin **product.html**, cho các phần theo mẫu sau:

```
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
```

Bước 02: Tạo view **chitiet_sanpham.blade.php** và paste đoạn code vào rồi code lại theo mẫu:

```
@extends("master")
@section("content")
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
@endsection
```

Bước 03: Mở Pagecontroller.php thêm function:

```
public function getChitietsanpham()
{
    return view("page.chitiet_sanpham");
}
```

Bước 04: Mở tập tin **Web.php** thêm route sau:

```
Route::get("/chi-tiet-san-pham", [Pagecontroller::class, "getChitietsanpham"]);
```

3) Cài trang liên hệ (lienhe.blade.php)

Bước 01: Mở tập tin **contacts.html**, cho các phần theo mẫu sau:

```
<div class="inner-header">...
</div>
<div class="beta-map">...
</div>
<div class="container">...
</div> <!-- .container -->
```

Bước 02: Tạo view **lienhe.blade.php** và paste đoạn code vào rồi code lại theo mẫu:

```
@extends("master")
@section("content")
<div class="inner-header">...
</div>
<div class="beta-map">...
</div>
<div class="container">...
</div> <!-- .container -->
@endsection
```

Bước 03: Mở Pagecontroller.php thêm function:

```
public function getLienhe()
{
    return view("page.lienhe");
}
```

Bước 04: Mở tập tin **Web.php** thêm route sau:

```
Route::get("/lien-he", [Pagecontroller::class, "getLienhe"]);
```

4) Cài trang giới thiệu (gioithieu.blade.php)

Bước 01: Mở tập tin **about.html**, cho các phần theo mẫu sau:

```
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
```

Bước 02: Tạo View **gioithieu.blade.php** và paste đoạn code vào rồi code lại theo mẫu:

```
@extends("master")
@section("content")
<div class="inner-header">...
</div>
<div class="container">...
</div> <!-- .container -->
@endsection
```

Bước 03: Mở Pagecontroller.php thêm function:

```
public function getGioithieu()
{
    return view("page.gioithieu");
}
```

Bước 04: Mở tập tin **Web.php** thêm route sau:

```
Route::get("/gioi-thieu", [Pagecontroller::class, "getGioithieu"]);
```