

Laravel

Ths. Ngô Lê Quân
ngolequanit@gmail.com
www.Vku.udn.vn

Bài 6. Ứng dụng Migration trong Laravel

6.1. Thiết lập kết nối cơ sở dữ liệu mysql

- Mở PhpMyadmin tạo cơ sở dữ liệu có tên: mylaravel
- Mở file có phần mở rộng .env và thiết lập kết nối như sau

APP_NAME=Laravel

APP_ENV=local

APP_KEY=base64:VcKbAe9m5kyhkTGac9qDznAJ06bmJ1evDuLcBgM6H4w=

APP_DEBUG=true

APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=mylaravel // tên CSDL đã tạo

DB_USERNAME=root // user đăng nhập mysql

DB_PASSWORD= // mật khẩu để trống

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.2. Migration là gì?

- Migration là một công cụ rất hay của Laravel, cho phép tạo các bảng, các cột cơ sở dữ liệu (Database) bằng các dòng lệnh PHP, ngoài ra có thể dễ dàng cập nhật nội dung các bảng, các cột đã tồn tại.
- Migration giúp cho việc đồng bộ Database giữa các môi trường khá thuận lợi, thay vì muốn thêm một bảng hay một cột trong các môi trường, thì ta phải làm thủ công từng môi trường, thì với Migration bạn chỉ cần thực hiện bằng dòng lệnh đơn giản.
- Điểm tốt của Migration là làm việc được với nhiều cơ sở dữ liệu như: MySQL, SQL Server, PostgreSQL, ...

Bài 6. Ứng dụng Migration trong Laravel

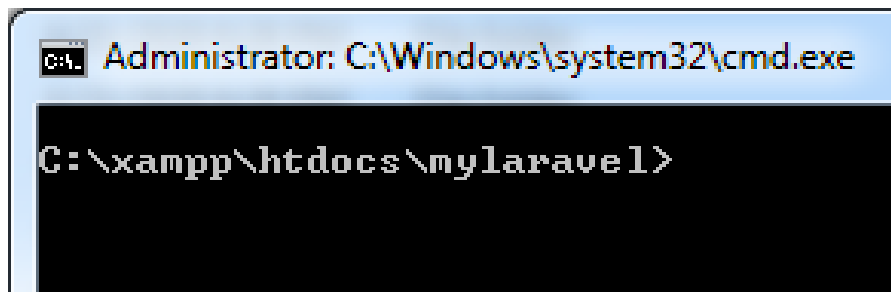
❑ 6.3. Tạo dữ liệu với Migration

- Migration được đặt bên trong thư mục */database/migrations/*.
- Để Migration hoạt động, chúng ta cần kết nối với cơ sở dữ liệu, nếu chưa biết cách kết nối thì bạn xem phần [Kết nối database](#).

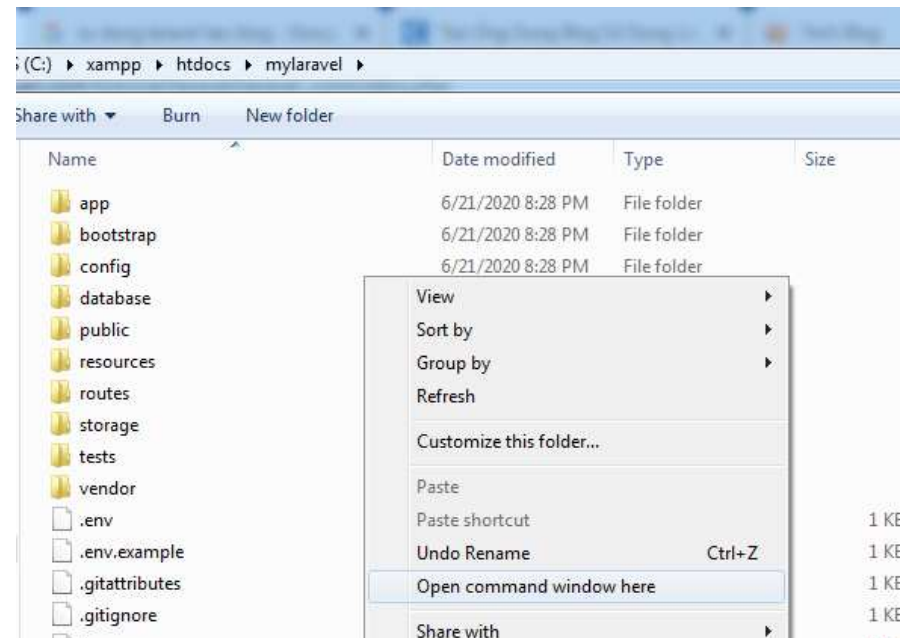
Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

- Thao tác với Migration, trước tiên, mở cửa sổ lệnh cmd, di chuyển tới thư mục myproject bằng lệnh:



```
Administrator: C:\Windows\system32\cmd.exe  
C:\xampp\htdocs\mylaravel>
```



Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

➤ Tạo Migration bằng lệnh Artisan:

```
php artisan make:migration create_news_table
```

make:migration - Lệnh tạo Migration.

create_news_table - Tên Migration do mình tự đặt.

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

dòng lệnh trên thực thi sẽ cho ta kết quả sau:



```

C:\> Select C:\WINDOWS\system32\cmd.exe

C:\xampp\htdocs\myproject>php artisan make:migration create_news_table
Created Migration: 2019_03_01_031800_create_news_table

C:\xampp\htdocs\myproject>_

```

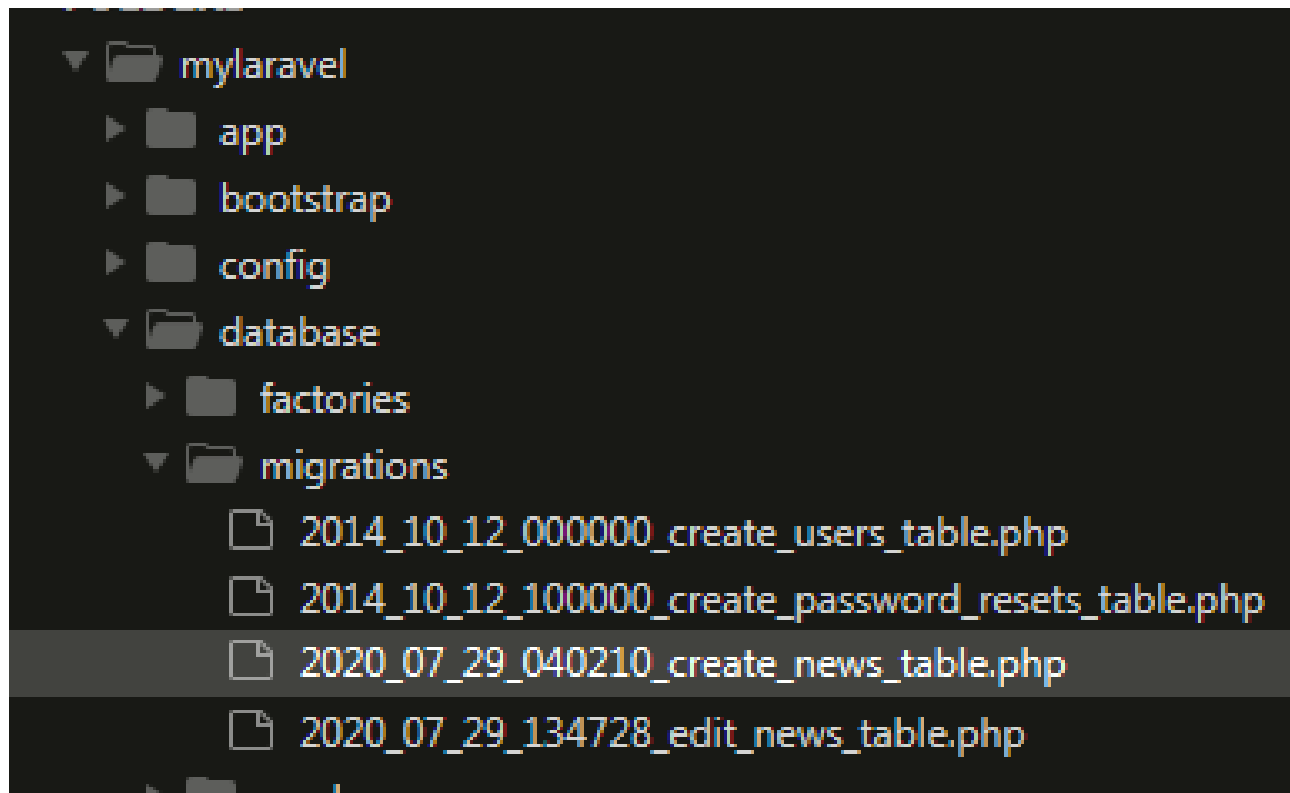
Sau khi thực thi lệnh, chúng ta sẽ nhận được thông báo file Migration đã được tạo "*Created Migration: 2019_03_01_031800_create_news_table*".

.

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

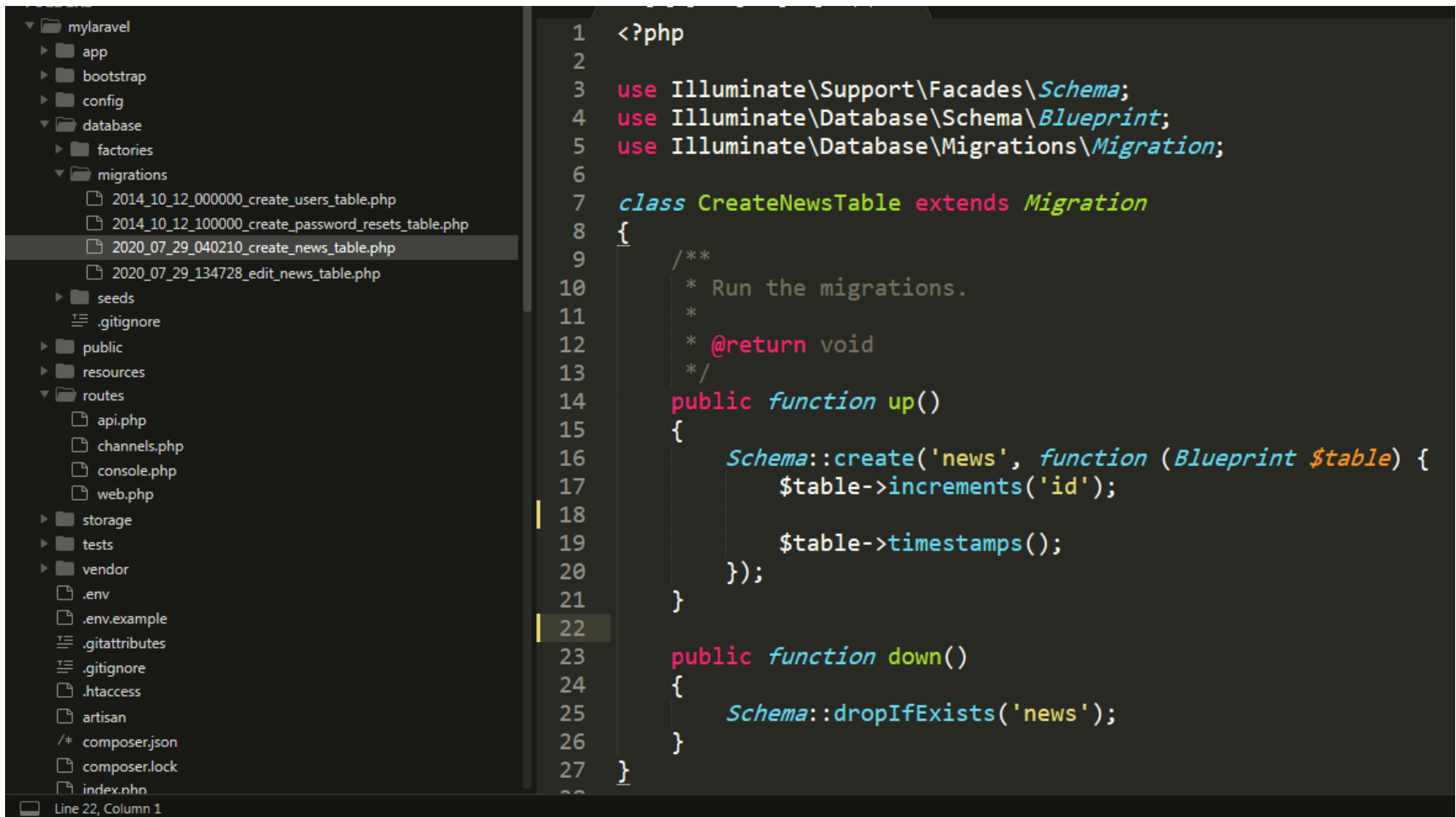
File Migration vừa tạo sẽ được lưu tại thư mục */database/migrations/*:



Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

Nội dung file vừa tạo như sau



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure of a Laravel application, including folders like 'app', 'bootstrap', 'config', 'database', 'factories', 'migrations', 'seeds', 'public', 'resources', 'routes', 'storage', 'tests', and 'vendor'. The 'migrations' folder is expanded, showing several migration files. The code editor on the right displays the content of a migration file, which is a PHP class named 'CreateNewsTable' extending 'Migration'. The class has two methods: 'up()' and 'down()'. The 'up()' method uses 'Schema::create()' to create a new table named 'news' with an 'id' column and a timestamp column. The 'down()' method uses 'Schema::dropIfExists()' to drop the 'news' table if it exists.

```
1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateNewsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('news', function (Blueprint $table) {
17             $table->increments('id');
18
19             $table->timestamps();
20         });
21     }
22
23     public function down()
24     {
25         Schema::dropIfExists('news');
26     }
27 }
```

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

- use Illuminate\Support\Facades\Schema; use Illuminate\Database\Schema\Blueprint; use Illuminate\Database\Migrations\Migration; là các khai báo sử dụng cần cho thao tác Migration.
- Chúng ta cần quan tâm là 2 function up() và down():
 - **public function up()** dùng để thêm, bớt, thay đổi, ... nội dung bảng cơ sở dữ liệu - để thực thi ta cần sử dụng lệnh php artisan migrate
 - **public function down()** dùng phục hồi hay xóa bảng, ... - để thực thi ta cần sử dụng lệnh php artisan migrate:rollback

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

- Với nội dung trên, ta đã có thể tạo bảng *news* với 1 cột là *id*, để có thể nhiều cột hơn ta thêm trực tiếp vào bên trong function `up()`:

```
public function up()
{
    Schema::create('news', function (Blueprint $table) {
        $table->increments('id');
        $table->string('headline');
        $table->string('email');

        $table->timestamps();
    });
}
```

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

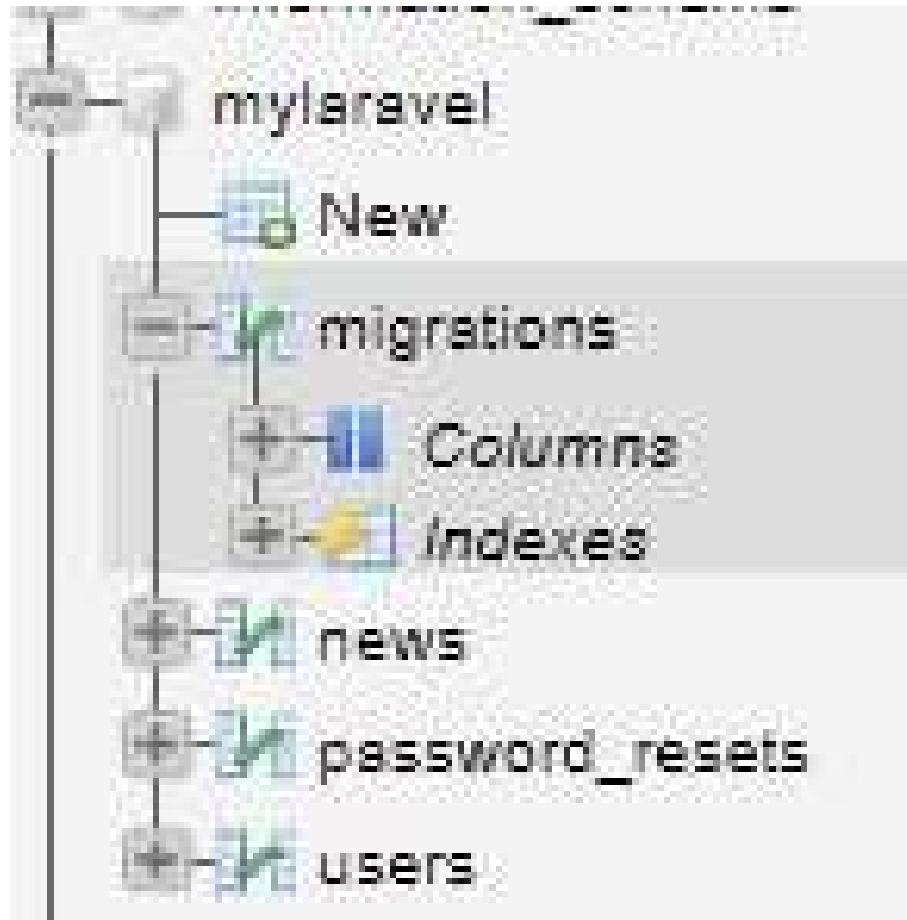
Khi chạy lệnh trên, nếu kết quả xuất hiện thông báo sau là thành công:

```
Migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table  
Migrating: 2019_03_01_031800_create_news_table  
Migrated: 2019_03_01_031800_create_news_table
```

Bài 6. Ứng dụng Migration trong Laravel

❑ 6.3. Tạo dữ liệu với Migration

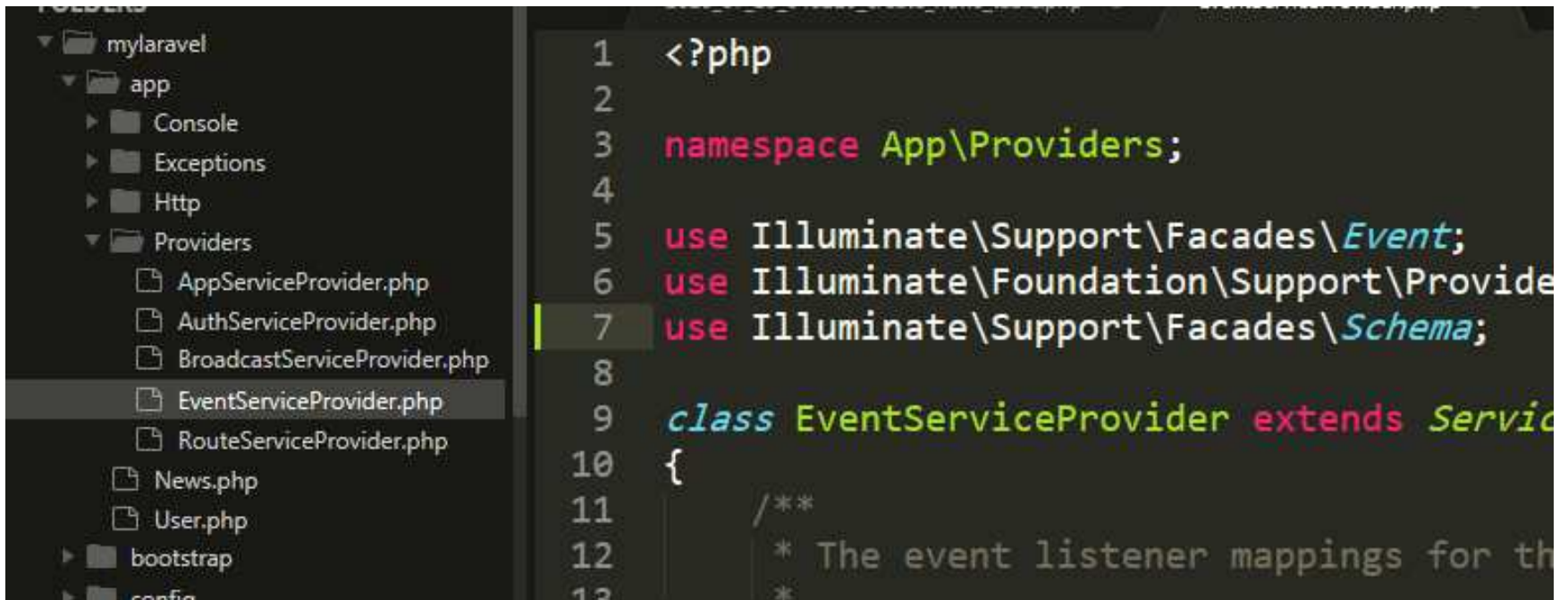
Lúc này đã có 4 bảng được tạo trong Database myproject:



Bài 6. Ứng dụng Migration trong Laravel

❑ 6.4. Xử lý lỗi khi chạy lệnh Migration

Nếu gặp lỗi sau khi chạy lệnh `php artisan migrate` thì nguyên nhân có thể là phiên bản MySQL thấp hơn yêu cầu, để chỉnh lỗi, thực hiện mở file `/app/Providers/AppServiceProvider.php` thêm các lệnh như bên dưới:



```
1 <?php
2
3 namespace App\Providers;
4
5 use Illuminate\Support\Facades\Event;
6 use Illuminate\Foundation\Support\Providers\RouteServiceProvider;
7 use Illuminate\Support\Facades\Schema;
8
9 class EventServiceProvider extends ServiceProvider
10 {
11     /**
12      * The event listener mappings for the application.
13      *
```

Bài 6. Ứng dụng Migration trong Laravel

6.5. Thay đổi Migration

Để sửa một table trong cơ sở dữ liệu, ta tạo một Migration mới dành riêng cho việc thay đổi như sau: tạo file *Migration edit_news_table.php*

`php artisan make:migration edit_news_table`

A screenshot of a Windows command prompt window. The title bar reads "Administrator: C:\Windows\system32\cmd.exe". The command prompt shows the following text:
C:\xampp\htdocs\mylaravel>php artisan make:migration edit_news_table
Created Migration: 2020_08_17_163838_edit_news_table
C:\xampp\htdocs\mylaravel>cfnhgh
The text is displayed in a monospaced font on a black background.

Bài 6. Ứng dụng Migration trong Laravel

6.5.1. Thêm cột dữ liệu

Mở file *create_news_table.php* trong thư mục */database/migrations*, nhưng 2 function *up()* và *down()* đang là rỗng

Để thêm cột mô tả *Description* ta thực hiện như sau

```
public function up()  
{ Schema::table('news', function (Blueprint $table)  
{ $table->string('description'); });  
}
```

Bài 6. Ứng dụng Migration trong Laravel

6.5.1. Thêm cột dữ liệu

Chạy lại lệnh php artisan migrate

php artisan migrate

Kiểm tra lại bảng *news* trên phpMyAdmin để kiểm tra kết quả

Bài 6. Ứng dụng Migration trong Laravel

6.5.1. Thêm cột dữ liệu

Chạy lại lệnh php artisan migrate

php artisan migrate

Kiểm tra lại bảng *news* trên phpMyAdmin để kiểm tra kết quả

Bài 6. Ứng dụng Migration trong Laravel

6.5.2. Sửa tên cột dữ liệu

Giả sử ta cần thay đổi tên cột headline thành title, và cần thêm một cột mới là slug, ta tiến hành thay đổi file *edit_news_table.php* như sau:

```
public function up()
{
    Schema::table('news', function (Blueprint $table) {
        // $table->string('descriptions');
        $table->renameColumn('headline', 'title');
        $table->string('slug');
    });
}
```

Bài 6. Ứng dụng Migration trong Laravel

6.5.2. Sửa tên cột dữ liệu

khi chạy lệnh `php artisan migration` thì sẽ xuất hiện thông báo "Nothing to migrate.": do bên trong table *migrations* đã tồn tại tên migration *yyyy_mm_dd_xxxxxx_edit_news_table*:
Mở table migrations trong phpmyadmin và xóa dòng này đi



The screenshot shows the phpMyAdmin interface for a database. At the top, there's a '+ Options' dropdown. Below it is a table with columns 'id', 'migration', and 'batch'. The table contains four rows of migration records. The fourth row, with id 7 and migration name '2020_08_17_163838_edit_news_table', is highlighted with a red rectangular box. Each row has a checkbox on the left and action icons (Edit, Copy, Delete) to its right. At the bottom of the table, there are additional action buttons: 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'.

	id	migration	batch
<input type="checkbox"/>	1	2014_10_12_000000_create_users_table	1
<input type="checkbox"/>	2	2014_10_12_100000_create_password_resets_table	1
<input type="checkbox"/>	3	2020_07_29_040210_create_news_table	1
<input type="checkbox"/>	7	2020_08_17_163838_edit_news_table	2

Bài 6. Ứng dụng Migration trong Laravel

6.5.2. Sửa tên cột dữ liệu

Chạy lại lệnh php artisan migration

Và mở cơ sở dữ liệu kiểm tra kết quả

```
C:\xampp\htdocs\mylaravel>php artisan migrate  
Nothing to migrate.
```

```
C:\xampp\htdocs\mylaravel>php artisan migrate  
Migrating: 2020_08_17_163838_edit_news_table  
Migrated: 2020_08_17_163838_edit_news_table
```

```
C:\xampp\htdocs\mylaravel>_
```

Bài 6. Ứng dụng Migration trong Laravel

6.5.3. Xóa cột dữ liệu

Giả sử ta cần xóa cột slug, ta tiến hành thay đổi file *edit_news_table.php* như sau:

```
*/  
public function up()  
{  
  
    Schema::table('news', function (Blueprint $table) {  
        // $table->string('descriptions');  
        // $table->renameColumn('headline', 'title');  
        // $table->string('slug');  
        $table->dropColumn('slug');  
    });  
  
}
```