

```

GenreController.php x
1 |<?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\View;
7 use Illuminate\Support\Facades\Redirect;
8 use Illuminate\Support\Facades\Validator;
9 use App\Genre;
10
11 class GenreController extends Controller
12 {
13     /**
14      * Display a listing of the resource.
15      *
16      * @return \Illuminate\Http\Response
17      */
18     public function index()
19     {
20         $genres = Genre::all();
21         return View::make('genre.index',compact('genres'));
22     }
23
24     /**
25      * Show the form for creating a new resource.
26      *
27      * @return \Illuminate\Http\Response
28      */
29     public function create()
30     {
31         return View::make('genre.create');
32     }
33
34     /**
35      * Store a newly created resource in storage.
36      *
37      * @param \Illuminate\Http\Request $request
38      * @return \Illuminate\Http\Response
39      */
40     public function store(Request $request)
41     {
42         $rules = [
43             'Genre' => 'required|alpha|min:3|max:32',
44         ];
45         $messages = [
46             'required' => 'This :attribute field is required',
47             'min' => 'masyadong maliit ang :attribute',
48             'alpha' => 'pawang mga letra lamang',
49             'name.required' => 'Anong pangalan mo?'
50         ];
51
52         $formData = $request->all();
53         $validator = Validator::make($formData, $rules, $messages);
54         if ($validator->passes()){
55             Genre::create($request->all());
56             return Redirect::to('genre')->with('success','New Genre added!');
57         }
58         //return Redirect::route('customer.create')->withInput();
59         return redirect()->back()->withInput()->withErrors($validator);
60     }
61
62     /**
63      * Display the specified resource.
64      *
65      * @param int $id
66      * @return \Illuminate\Http\Response
67      */
68     public function show($id)
69     {
70         //
71     }
72
73     /**

```

ProducerController.php x

!>php

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\View;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Validator;
use App\Producer;

class ProducerController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $producers = Producer::all();
        return View::make('producer.index', compact('producers'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return View::make('producer.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $rules = [
            'name' => 'required|alpha|min:6|max:32',
            'email' => 'required',
            'website' => 'required',
        ];
        $messages = [
            'required' => 'This :attribute field is required',
            'min' => 'masyadong maliit ang :attribute',
            'alpha' => 'pawang mga letra lamang',
            'name.required' => 'Anong pangalan mo?'
        ];

        $formData = $request->all();
        $validator = Validator::make($formData, $rules, $messages);
        if ($validator->passes()){
            Producer::create($request->all());
            return Redirect::to('producer')->with('success', 'New Producer added!');
        }
        //return Redirect::route('customer.create')->withInput();
        return redirect()->back()->withInput()->withErrors($validator);
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }
}
```

```

RatingController.php x
50 {
51     //
52 }
53
54 /**
55  * Show the form for editing the specified resource.
56  *
57  * @param int $id
58  * @return \Illuminate\Http\Response
59  */
60 public function edit($id)
61 {
62     //
63 }
64
65 /**
66  * Update the specified resource in storage.
67  *
68  * @param \Illuminate\Http\Request $request
69  * @param int $id
70  * @return \Illuminate\Http\Response
71  */
72 public function update(Request $request, $id)
73 {
74     Rating::where('film_id','=', $id)->where('user_id','=',Auth::user()->id)->delete();
75     echo ('rate');
76     $rating=Rating::create(['film_id' => $id,
77         'user_id' => Auth::user()->id,
78         'rate' => $request->radio,
79         'comment' => $request->comment]);
80
81     return Redirect::to('film')->with('success','Film updated!');
82 }
83
84 /**
85  * Remove the specified resource from storage.
86  *
87  * @param int $id
88  * @return \Illuminate\Http\Response
89  */
90 public function destroy($id)
91 {
92     //
93 }
94 }
95

```

```

39  */
40  public function store(Request $request)
41  {
42      $rules = [
43          'role' => 'required|alpha|min:3|max:32',
44      ];
45      $messages = [
46          'required' => 'This :attribute field is required',
47          'min' => 'masyadong maliit ang :attribute',
48          'alpha' => 'pawang mga letra lamang',
49          'name.required' => 'Anong pangalan mo?'
50      ];
51
52      $formData = $request->all();
53      $validator = Validator::make($formData, $rules, $messages);
54      if ($validator->passes()){
55          role::create($request->all());
56          return Redirect::to('role')->with('success','New role added!');
57      }
58      //return Redirect::route('customer.create')->withInput();
59      return redirect()->back()->withInput()->withErrors($validator);
60  }
61
62  /**
63   * Display the specified resource.
64   *
65   * @param int $id
66   * @return \Illuminate\Http\Response
67   */
68  public function show($id)
69  {
70      //
71  }
72
73  /**
74   * Show the form for editing the specified resource.
75   *
76   * @param int $id
77   * @return \Illuminate\Http\Response
78   */
79  public function edit($id)
80  {
81      $role = role::find($id);
82      return view('role.edit',compact('role'));
83  }
84
85  /**
86   * Update the specified resource in storage.
87   *
88   * @param \Illuminate\Http\Request $request
89   * @param int $id
90   * @return \Illuminate\Http\Response
91   */
92  public function update(Request $request, $id)
93  {
94      $role = role::find($id);
95      // dd($customer);
96      $role->update($request->all());
97      return Redirect::to('role')->with('success','role updated!');
98  }
99
100  /**
101   * Remove the specified resource from storage.
102   *
103   * @param int $id
104   * @return \Illuminate\Http\Response
105   */
106  public function destroy($id)
107  {
108      $role = role::findOrFail($id);
109      $role->delete();
110      return Redirect::to('role')->with('success','role deleted!');

```

```
Genre.php x
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Genre extends Model
{
    protected $fillable = ['genre'];

    public function films() {
        return $this->belongsToMany('App\Film', 'film_genre');
    }
}
```

```
ActorController.php x Producer.php x
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Producer extends Model
{
    protected $fillable = ['name', 'email', 'website'];

    public function films() {
        return $this->belongsToMany('App\Producer', 'film_producer');
    }
}
```

```
Rating.php x
1 |<?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Relations\Pivot;
6
7 class Rating extends Pivot
8 {
9     protected $fillable=['user_id','film_id','rate','comment'];
10    protected $table='ratings';
11 }
12
```

```
Role.php x
1 |<?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Role extends Model
8 {
9     protected $fillable = ['role'];
10
11     public function films() {
12         return $this->belongsToMany('App\Film','film_actor_role');
13     }
14
15     public function actors() {
16         return $this->belongsToMany('App\Role','film_actor_role');
17     }
18 }
19
```

