Programming Assignment 3 – Architecture Foundations

Models
Using Laravel artisan tool, create ***Location***, ***Story*** and ***Tag*** models. These models will represent blueprint for the ***locations*** *(example: New York City, Los Angeles, Brooklyn, Paris…)*, news ***stories*** and ***tags*** relevant to particular stories.

Let's think how we will structure these models. First, we will need to figure out the fields and data types that will be used in these blueprints.

Location will need the value where we will store the name for that particular location. Location name is almost always will be a string, so let's use that.

Location structure:

Location	
Field	**Data type**
value	string

Please note that our fields will ***always be lowercase***, and if we need to combine two words for a field name, we will use ***camelCase***.

Next, we will design a structure for the **Story** model. Since we mentioned word blueprint, think of each model as an object that has common structure. Each story will need a title, and the story itself. Let's make it a step more advanced, and add a switch for this story to be published or not.

Story structure:

Story	
Field	**Data type**
title	
story	
published	boolean

1. Fill in the data types for the story structure above.

Finally, we will work on our **Tag** model. Each tag will have its value.

Tag structure:

Location	
Field	**Data type**
value	string

Database

1. **Configure .env**
   In Laravel root directory, ~/Code/Laravel there is a file **.env**. This file is used to store configuration for your web application.

   Database configuration block:

```
DB_HOST=localhost
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret
```

   More information: https://laravel.com/docs/master/homestead#connecting-to-databases

   These credentials can be used to connect to the database created in vagrant box. You can always go to MySQL console and change password, create new databases or users.

2. **Create migrations**
   Migrations are like version control for your database, allowing a team to easily modify and share the application's database schema. Migrations are typically paired with Laravel's schema builder to easily build your application's database schema. [1]

   Please review official migrations documentation:
   https://laravel.com/docs/master/migrations

   Using Laravel artisan tool, create the following migrations:

   a. **create_locations_table**
   b. **create_stories_table**
   c. **create_tags_table**

   Using the fields and data types we designed earlier, fill in table structures for each of the migrations.

   For example, **create_locations_table** will be located at ~/Code/Laravel/database/migrations/<*timestamp*>_create_locations_table.php. Please refer to the migration structure: https://laravel.com/docs/master/migrations#migration-structure to create *up* and *down* methods.

Make sure that $table->increments('id'); and $table->timestamps(); are present in your up methods in every migration.

3. **Create seeds**
   Seeds are used to pre-populate your database with sample data for testing purposes.

   Before you begin, please review model factories:
   https://laravel.com/docs/master/testing#model-factories

   You might need to install faker package using composer.

Using Laravel artisan tool, create few seeds:

a. LocationsTableSeeder
b. StoriesTableSeeder
c. TagsTableSeeder

Example seed for Locations:

```
<?php

use Illuminate\Database\Seeder;

class LocationsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $faker = Faker\Factory::create();

        $limit = 20;

        for ($i = 0; $i < $limit; $i++) {
            DB::table('locations')->insert([ //,
                'value' => $faker->city.', '. $faker->stateAbbr,
            ]);
        }
    }
}
```

Let's discuss the code above. First, we create an instance of a faker model factory. Then we set our limit to 20 locations that we will pre-populate. Inside for loop, we insert randomly generated locations into the database.

More Faker information: https://github.com/fzaninotto/Faker

Views
Views contain the HTML served by your application and separate your controller / application logic from your presentation logic. Views are stored in the **resources/views** directory. [3]

Create the following views:

1. **locations.blade.php**
2. **stories.blade.php**
3. **tags.blade.php**

For now, let's display the sample data we put into database using seeds.

Example code for locations.blade.php:

```
<html>
    <body>
        @foreach ($locations as $location)
        <p>{{ $location->value }}</p>
```

```
        @endforeach
    </body>
</html>
```

Controllers
Using Laravel artisan tool, please create the following controllers:

1. **LocationController**
2. **StoryController**

These controllers will be used to interact with models and render views.

Example code for LocationController:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;
use App\Http\Controllers\Controller;

use App\Location;

class LocationController extends Controller
{
    public function index()
    {
        $locations = Location::all();
        return view('locations', ['locations' => $locations]);
    }
}
```

1. Create **index** method in ***StoryController*** that will retrieve all stories, and display them in the stories.blade.php view.
2. Create **tags** method in ***StoryController*** that will retrieve all tags, and display them in the tags.blade.php view.

Putting it all together
Please create the following routes:

HTTP Request Method	Route name	Parameters	Controller	Method
GET	locations	*None*	LocationController	index
GET	stories	*None*	StoryController	index
GET	tags	*None*	StoryController	tags

What to submit:

Please fork ***NYU-CS6015/Programming-Assignments***. Place your:

1. routes.php
2. StoryController.php
3. LocationController.php
4. create_locations_table
5. create_stories_table
6. create_tags_table
7. LocationsTableSeeder.php
8. StoriesTableSeeder.php
9. TagsTableSeeder.php
10. locations.blade.php
11. stories.blade.php
12. tags.blade.php

inside **Programming Assignment 3** directory. **Do not forget to commit** your programming assignment! No need to create pull requests.

Sources:

1. https://laravel.com/docs/master/migrations
2. https://laravel.com/docs/master/eloquent-relationships
3. https://laravel.com/docs/master/views#basic-usage