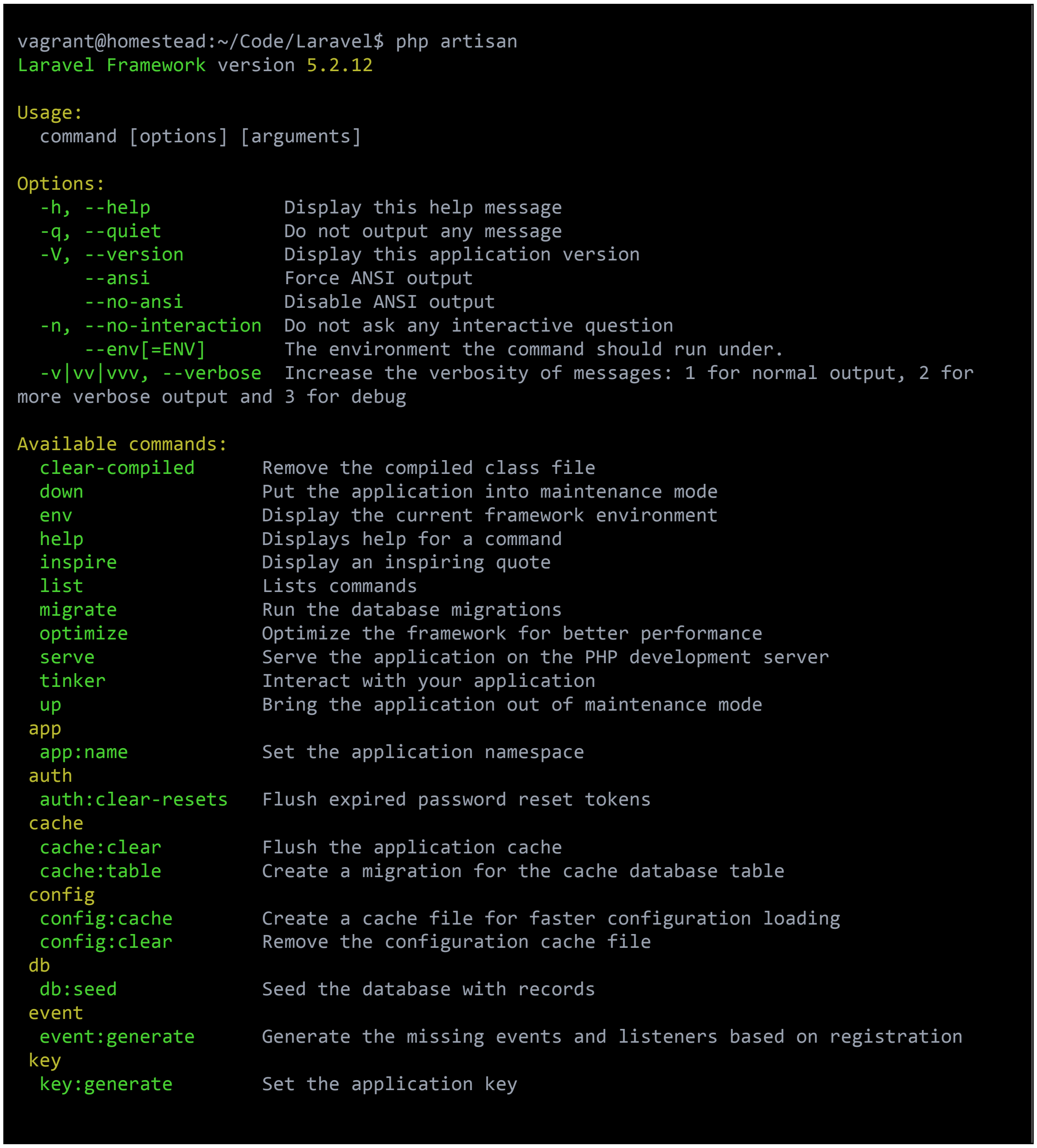
1. Using Laravel artisan tool, create 2 controllers: **PageController** and **WeatherController**.

   First, you will need to enter your project directory. By default we use ~/Code/Laravel. In case you chose to change that location, please make sure to keep in mind the paths you have set.

```
cd ~/Code/Laravel
php artisan
```

   php artisan command will give you a list of all the commands Laravel artisan tool is capable of.

   Sample output:

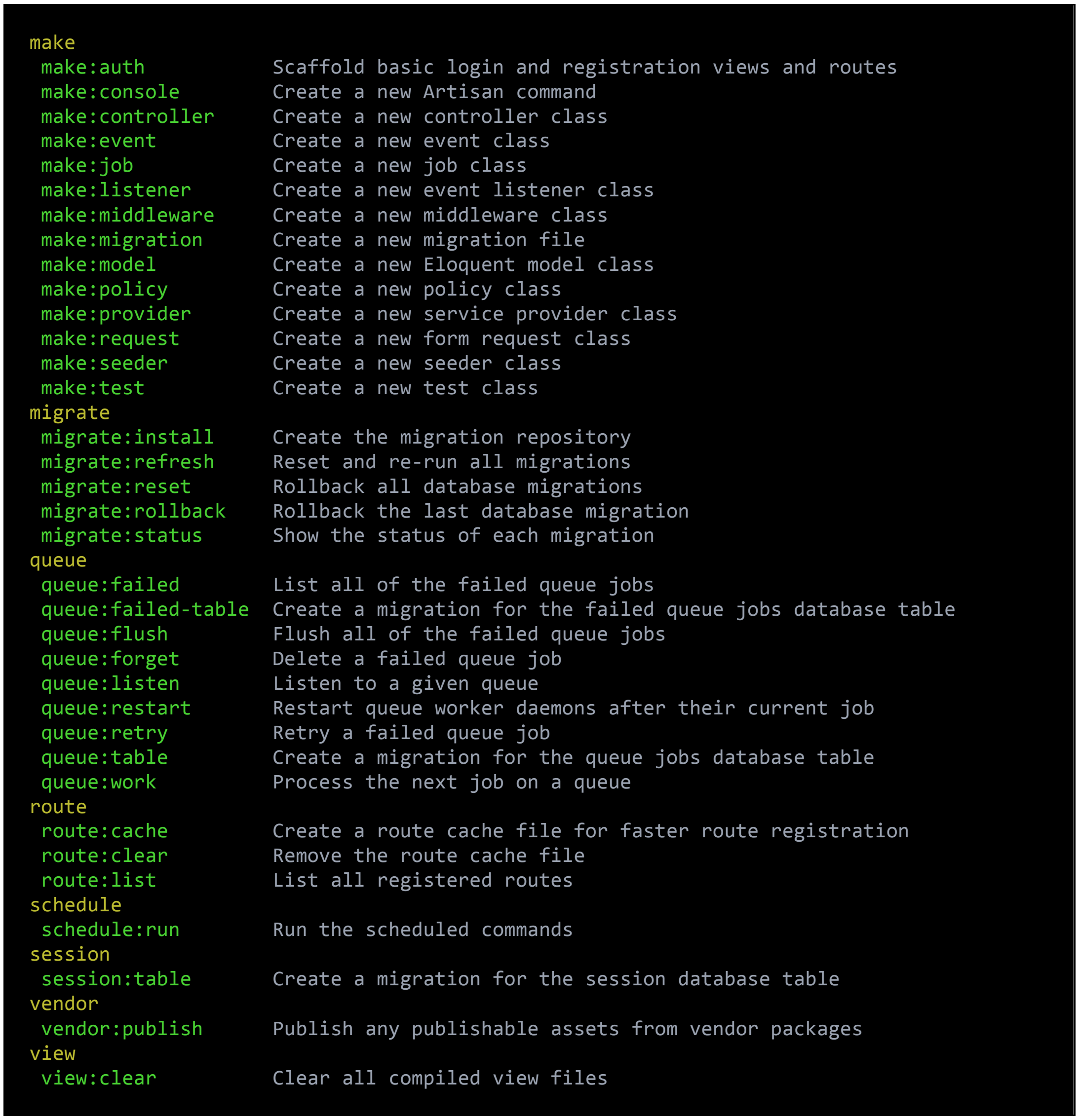


```
vagrant@homestead:~/Code/Laravel$ php artisan
Laravel Framework version 5.2.12

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
      --ansi            Force ANSI output
      --no-ansi         Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
      --env[=ENV]       The environment the command should run under.
  -v|vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output, 2 for
more verbose output and 3 for debug

Available commands:
  clear-compiled      Remove the compiled class file
  down                Put the application into maintenance mode
  env                 Display the current framework environment
  help                Displays help for a command
  inspire             Display an inspiring quote
  list                Lists commands
  migrate             Run the database migrations
  optimize            Optimize the framework for better performance
  serve               Serve the application on the PHP development server
  tinker              Interact with your application
  up                  Bring the application out of maintenance mode
 app
  app:name            Set the application namespace
 auth
  auth:clear-resets   Flush expired password reset tokens
 cache
  cache:clear         Flush the application cache
  cache:table         Create a migration for the cache database table
 config
  config:cache        Create a cache file for faster configuration loading
  config:clear        Remove the configuration cache file
 db
  db:seed             Seed the database with records
 event
  event:generate      Generate the missing events and listeners based on registration
 key
  key:generate        Set the application key
```

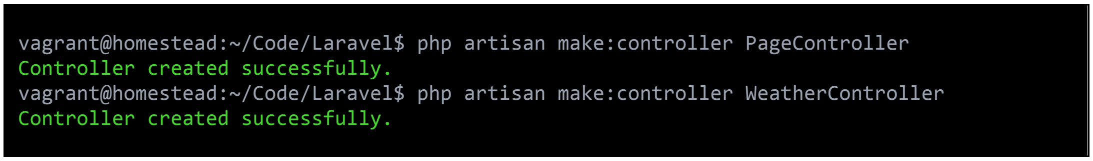


```
make
 make:auth            Scaffold basic login and registration views and routes
 make:console         Create a new Artisan command
 make:controller      Create a new controller class
 make:event           Create a new event class
 make:job             Create a new job class
 make:listener        Create a new event listener class
 make:middleware      Create a new middleware class
 make:migration       Create a new migration file
 make:model           Create a new Eloquent model class
 make:policy          Create a new policy class
 make:provider        Create a new service provider class
 make:request         Create a new form request class
 make:seeder          Create a new seeder class
 make:test            Create a new test class
migrate
 migrate:install      Create the migration repository
 migrate:refresh      Reset and re-run all migrations
 migrate:reset        Rollback all database migrations
 migrate:rollback     Rollback the last database migration
 migrate:status       Show the status of each migration
queue
 queue:failed         List all of the failed queue jobs
 queue:failed-table   Create a migration for the failed queue jobs database table
 queue:flush          Flush all of the failed queue jobs
 queue:forget         Delete a failed queue job
 queue:listen         Listen to a given queue
 queue:restart        Restart queue worker daemons after their current job
 queue:retry          Retry a failed queue job
 queue:table          Create a migration for the queue jobs database table
 queue:work           Process the next job on a queue
route
 route:cache          Create a route cache file for faster route registration
 route:clear          Remove the route cache file
 route:list           List all registered routes
schedule
 schedule:run         Run the scheduled commands
session
 session:table        Create a migration for the session database table
vendor
 vendor:publish       Publish any publishable assets from vendor packages
view
 view:clear           Clear all compiled view files
```

For this assignment, we will use **make** command to create new controllers.

```
php artisan make:controller PageController
php artisan make:controller WeatherController
```

Sample output:



```
vagrant@homestead:~/Code/Laravel$ php artisan make:controller PageController
Controller created successfully.
vagrant@homestead:~/Code/Laravel$ php artisan make:controller WeatherController
Controller created successfully.
```

From now on for the sake of simplicity, we will assume that our current directory is ~/Code/Laravel, and all subsequent paths will be contained within that directory.

Controllers you have just created will be located at app/Http/Controllers.

First, open ***PageController***, and create a new method, called **index**. I will provide first example of the method:

Sample code:

```
public function index()
{
    return "Hello from index page";
}
```

Next, you will create another method called **greet**. This method will take a parameter with variable **name** and return “Hello from **name**”, where name will be the value of that variable.

Sample code:

```
public function greet($name)
{
    return "Hello from ". $name;
}
```

Now, we will work with the ***WeatherController***. Create a new method called **getWeather**. It will take parameter with variable **location**. Please output the location the same way as in the greet method.

For more information about Controllers, please refer to https://laravel.com/docs/master/controllers

2. In the previous task we created our Controllers and methods that contain basic logic for this exercise. Now we will set up the routes in order to access that information.

   Let’s assign our GET / route to a **PageController index** method. You have already edited GET / route before, in the Programming Assignment 1. This will be a little bit different.

   routes.php file is located in app/Http/routes.php.

   Erase the route closure for GET /. Instead place the following:

   Sample code:

```
Route::get(‘/’, ‘PageController@index’);
```

   This way, we tell our Web Application to handle GET / HTTP request with ***PageController’s* index** method.

The next step will be to create a route that will use our **greet** method. Please note that the **greet** method will need an input parameter, name. Laravel provides an elegant way to handle variables within URI.

Sample code:

```
Route::get(‘/greet/{name}’, ‘PageController@greet);
```

Please create another route: GET /weather that will have an input parameter location. Use ***WeatherController’s*** method **getWeather** to handle this request.

For more information about Routes, please refer to
https://laravel.com/docs/master/routing

Please read the Routing and Controller sections of the manual. They are very important for further Web Application development.

What to submit:

Please update your fork of ***NYU-CS6015/Programming-Assignments***. Place your routes.php, PageController.php and WeatherController.php files in **Programming Assignment 2** directory. **Do not forget to commit** your programming assignment! No need to create pull requests.