Laravel Framework for Web Applications

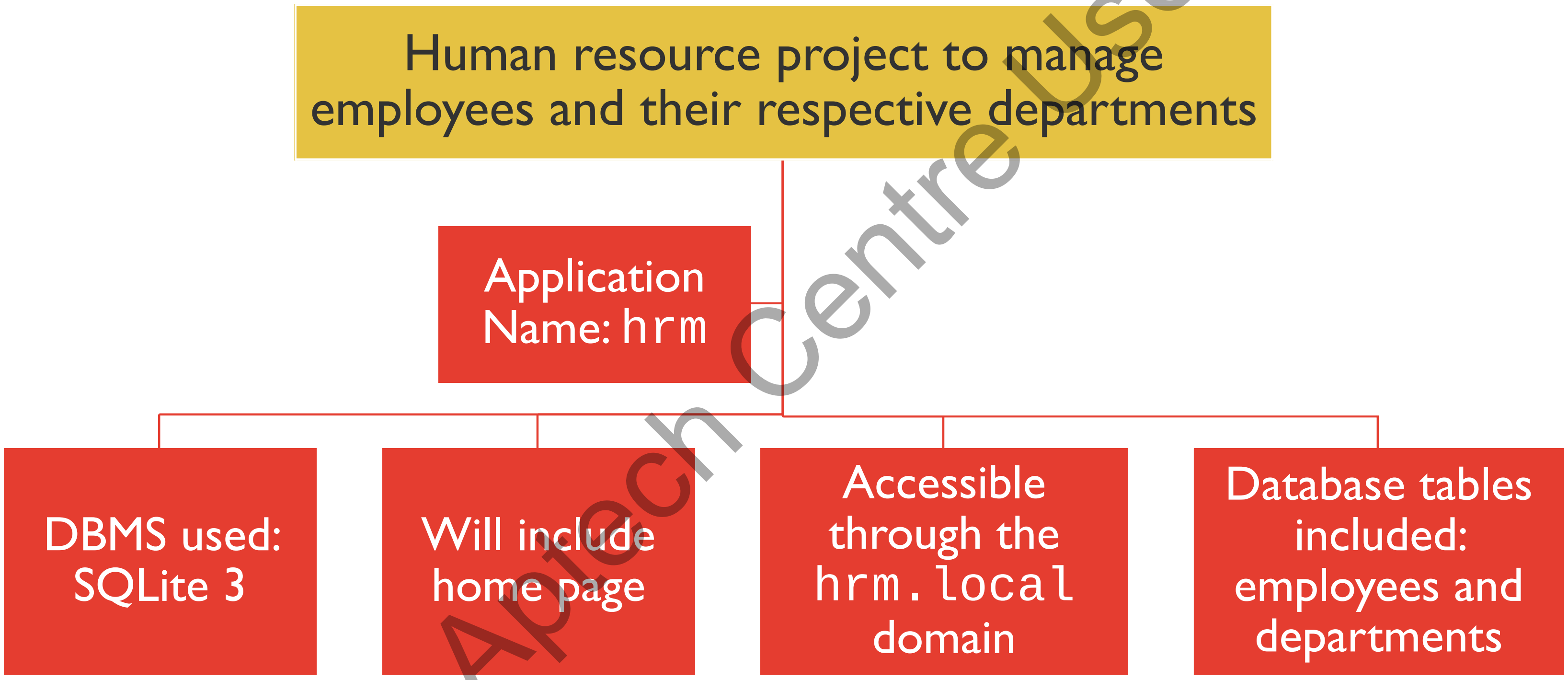
SESSION 04

# CREATING A WEB APPLICATION

# OBJECTIVES

- Learn to create a new Laravel application
- Understand integration of Laravel application with database
- Learn to create table schema
- Explain process migration
- Understand Web page creation using Master Blade Layout, Forms, and Routes

# PLANNING THE APPLICATION



Human resource project to manage employees and their respective departments

Application Name: `hrm`

DBMS used: SQLite 3

Will include home page

Accessible through the `hrm.local` domain

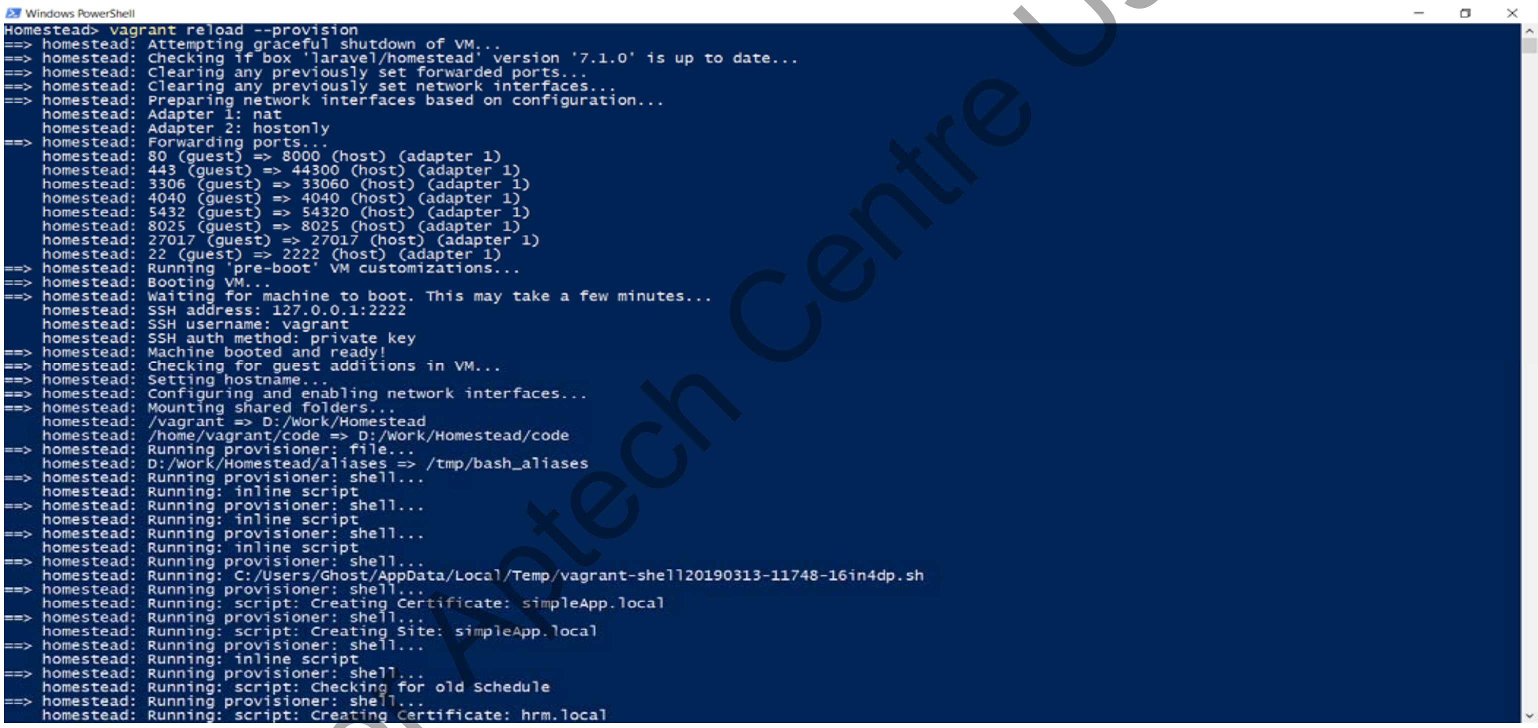
Database tables included: employees and departments

# BUILDING A LARAVEL APPLICATION (1-4)

```
ip: 192.168.10.10
memory: 2048
cpus: 1
provider: virtualbox
authorize: C:\Users\Ghost\.ssh\id_rsa.pub
keys:
    - C:\Users\Ghost\.ssh\id_rsa
folders:
    -
        map: 'D:\Work\Homestead\code'
        to: /home/vagrant/code
sites:
    -
        map: simpleApp.local
        to: /home/vagrant/code/simpleApp/public
    -
        map: hrm.local
        to: /home/vagrant/code/hrm/public
databases:
    - homestead
name: homestead
hostname: homestead
```

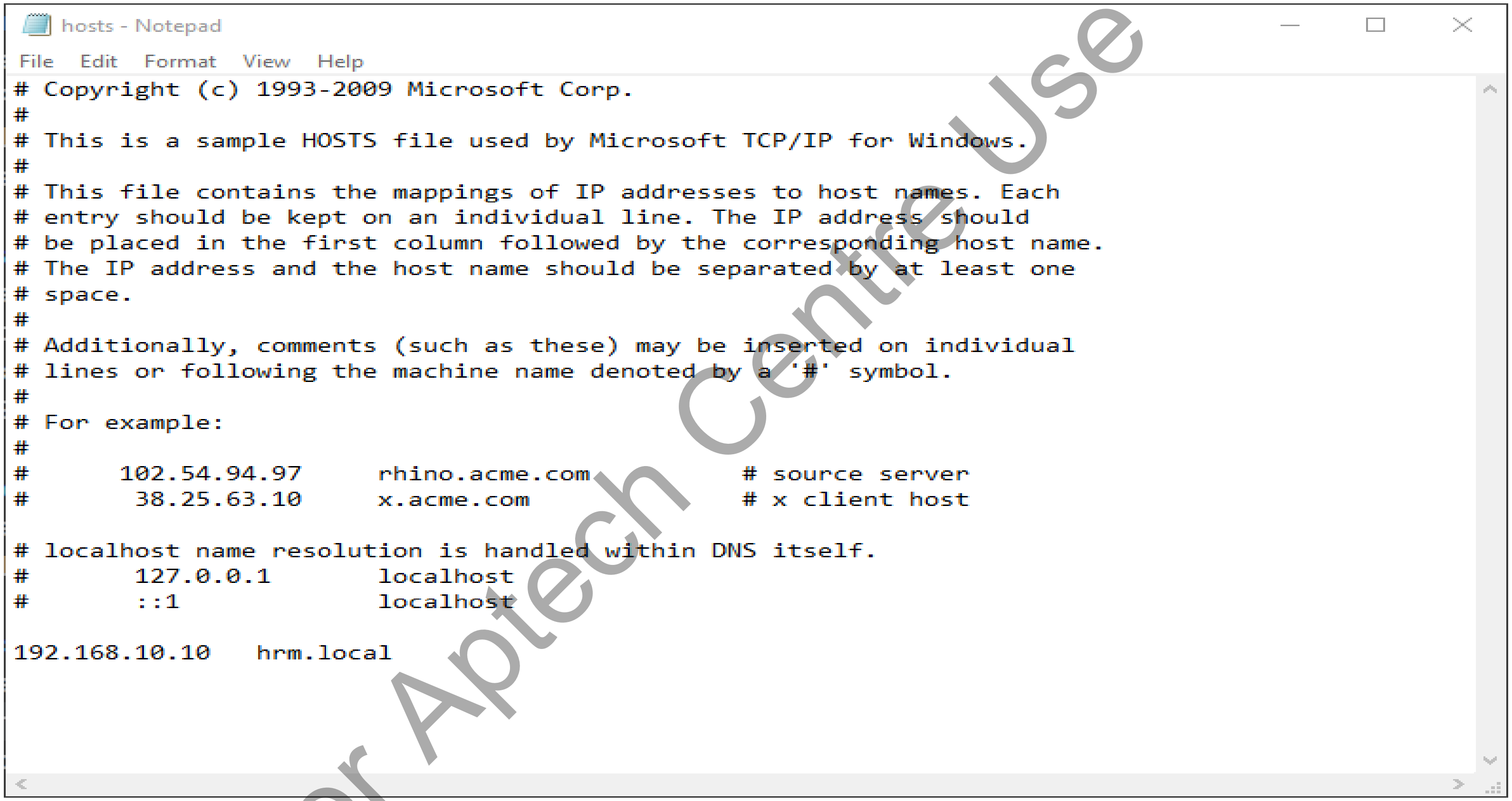
# BUILDING A LARAVEL APPLICATION (2-4)

**vagrant reload –-provision**



```
Homestead> vagrant reload --provision
==> homestead: Attempting graceful shutdown of VM...
==> homestead: Checking if box 'laravel/homestead' version '7.1.0' is up to date...
==> homestead: Clearing any previously set forwarded ports...
==> homestead: Clearing any previously set network interfaces...
==> homestead: Preparing network interfaces based on configuration...
    homestead: Adapter 1: nat
    homestead: Adapter 2: hostonly
==> homestead: Forwarding ports...
    homestead: 80 (guest) => 8000 (host) (adapter 1)
    homestead: 443 (guest) => 44300 (host) (adapter 1)
    homestead: 3306 (guest) => 33060 (host) (adapter 1)
    homestead: 4040 (guest) => 4040 (host) (adapter 1)
    homestead: 5432 (guest) => 54320 (host) (adapter 1)
    homestead: 8025 (guest) => 8025 (host) (adapter 1)
    homestead: 27017 (guest) => 27017 (host) (adapter 1)
    homestead: 22 (guest) => 2222 (host) (adapter 1)
==> homestead: Running 'pre-boot' VM customizations...
==> homestead: Booting VM...
==> homestead: Waiting for machine to boot. This may take a few minutes...
    homestead: SSH address: 127.0.0.1:2222
    homestead: SSH username: vagrant
    homestead: SSH auth method: private key
==> homestead: Machine booted and ready!
==> homestead: Checking for guest additions in VM...
==> homestead: Setting hostname...
==> homestead: Configuring and enabling network interfaces...
==> homestead: Mounting shared folders...
    homestead: /vagrant => D:/Work/Homestead
    homestead: /home/vagrant/code => D:/Work/Homestead/code
==> homestead: Running provisioner: file...
    homestead: D:/Work/Homestead/aliases => /tmp/bash_aliases
==> homestead: Running provisioner: shell...
    homestead: Running: inline script
==> homestead: Running provisioner: shell...
    homestead: Running: inline script
==> homestead: Running provisioner: shell...
    homestead: Running: inline script
==> homestead: Running provisioner: shell...
    homestead: Running: C:/Users/Ghost/AppData/Local/Temp/vagrant-shell20190313-11748-16in4dp.sh
==> homestead: Running provisioner: shell...
    homestead: Running: script: Creating Certificate: simpleApp.local
==> homestead: Running provisioner: shell...
    homestead: Running: script: Creating Site: simpleApp.local
==> homestead: Running provisioner: shell...
    homestead: Running: inline script
==> homestead: Running provisioner: shell...
    homestead: Running: script: Checking for old Schedule
==> homestead: Running provisioner: shell...
    homestead: Running: script: Creating Certificate: hrm.local
```

# BUILDING A LARAVEL APPLICATION (3-4)



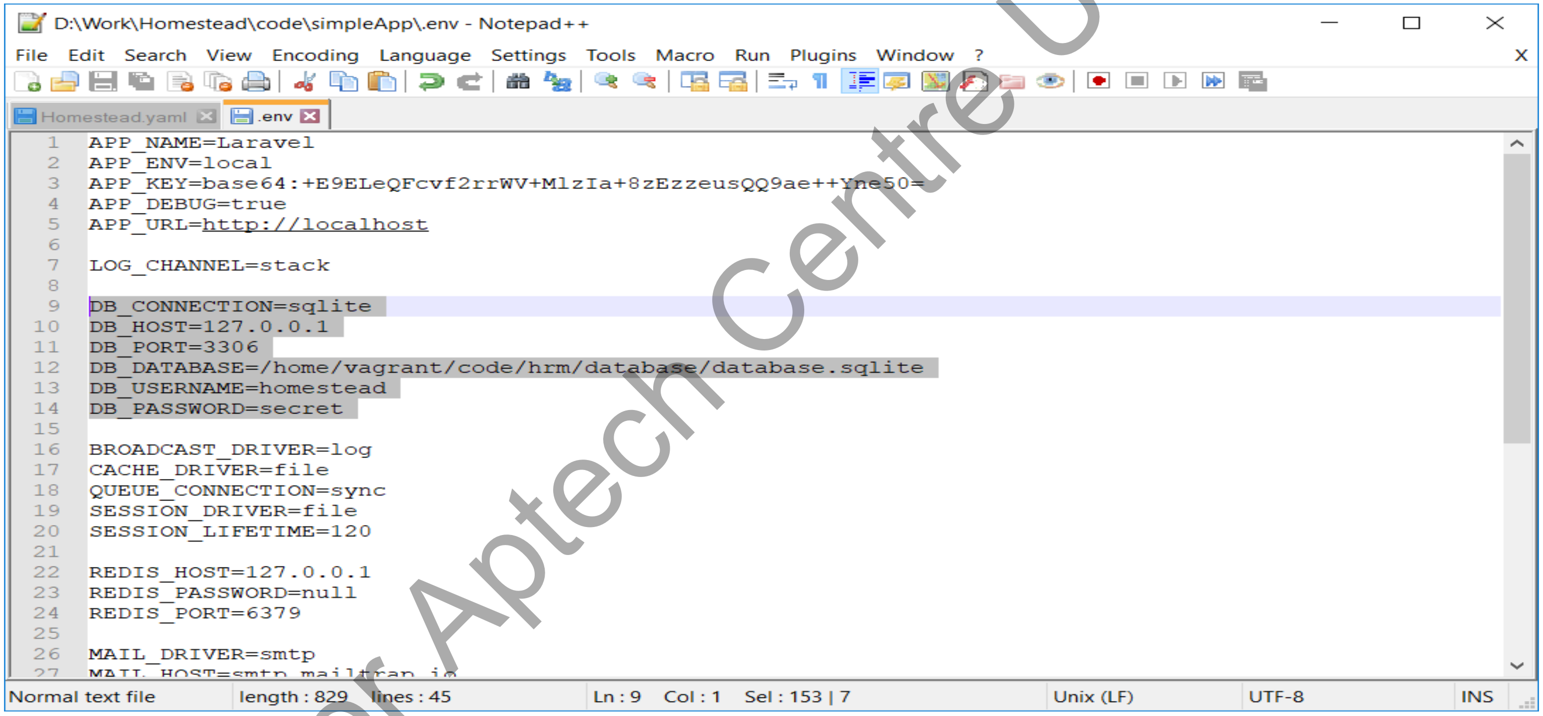
```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

192.168.10.10   hrm.local
```

# BUILDING A LARAVEL APPLICATION (4-4)

```
cd code
laravel new hrm
cd hrm
php artisan app:name HRM
```

# PREPARING THE DATABASE (1-5)

```
touch database/database.sqlite
```



D:\Work\Homestead\code\simpleApp\.env - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Homestead.yaml
.env

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:+E9ELeQFcvf2rrWV+MlzIa+8zEzzeusQQ9ae++Yne50=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=sqlite
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=/home/vagrant/code/hrm/database/database.sqlite
DB_USERNAME=homestead
DB_PASSWORD=secret

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
```

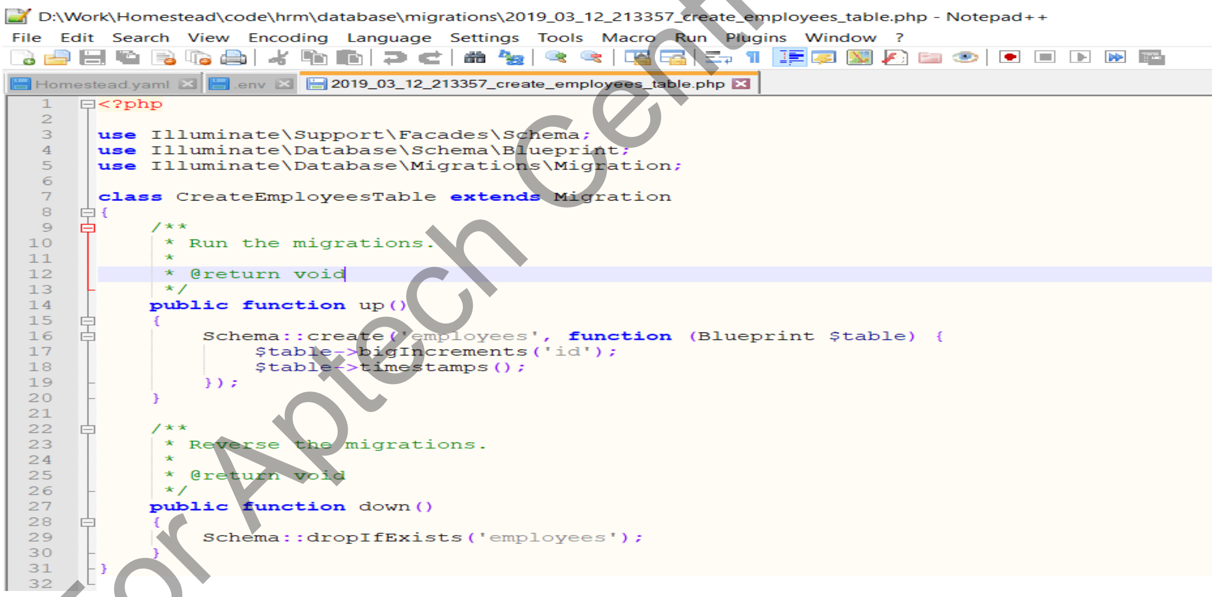
# PREPARING THE DATABASE (2-5)

```
Schema::create('employees', function (Blueprint
$table) {
    $table->bigIncrements('id');
    $table->string('name');
    $table->string('designation');
    $table->integer('salary');
});
```

# PREPARING THE DATABASE (3-5)

```
rm database/migrations/*
```

```
php artisan make:migration create_employees_table
--create=employees
```

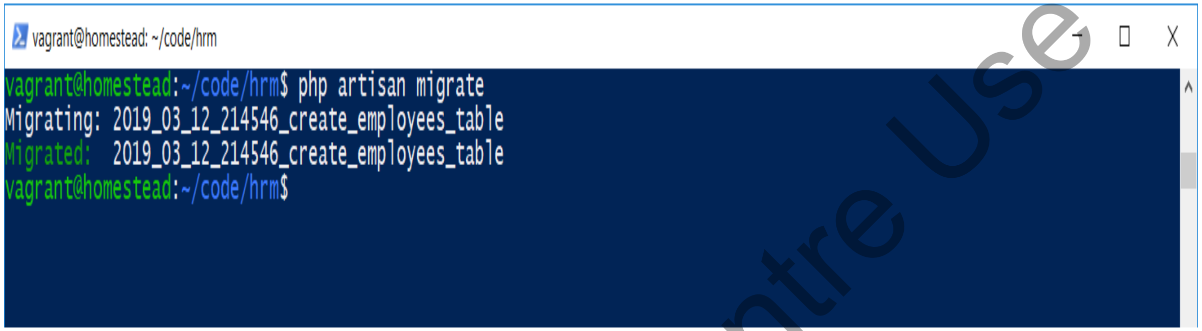

D:\Work\Homestead\code\hrm\database\migrations\2019_03_12_213357_create_employees_table.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Homestead.yaml
.env
2019_03_12_213357_create_employees_table.php

```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateEmployeesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('employees', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('employees');
    }
}
```

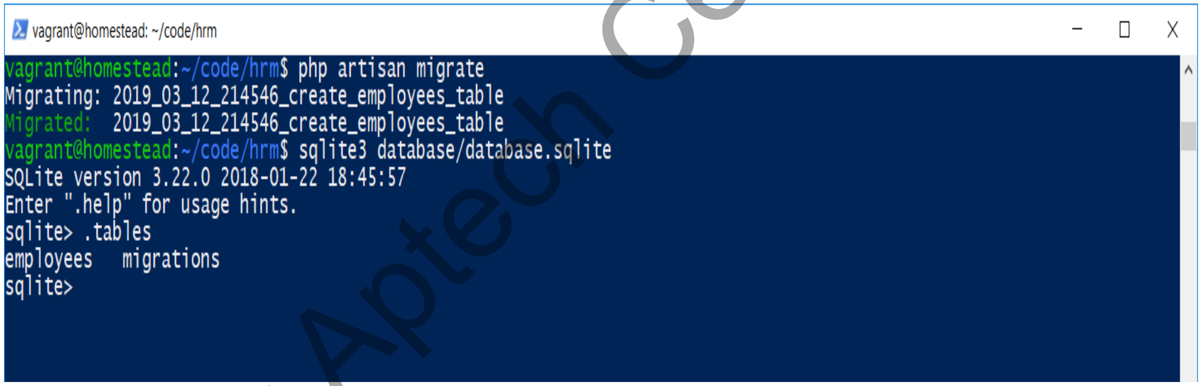
# PREPARING THE DATABASE (4-5)

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateEmployeesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('employees', function (Blueprint $table) {
            $table->bigIncrements('id’); $table->string('name');
            $table->string('designation');$table->integer('salary');
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('employees');
    }
};
```

# PREPARING THE DATABASE (5-5)



```
vagrant@homestead:~/code/hrm$ php artisan migrate
Migrating: 2019_03_12_214546_create_employees_table
Migrated:  2019_03_12_214546_create_employees_table
vagrant@homestead:~/code/hrm$
```



```
vagrant@homestead:~/code/hrm$ php artisan migrate
Migrating: 2019_03_12_214546_create_employees_table
Migrated:  2019_03_12_214546_create_employees_table
vagrant@homestead:~/code/hrm$ sqlite3 database/database.sqlite
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> .tables
employees   migrations
sqlite>
```

# CREATING WEB PAGES (1-9)

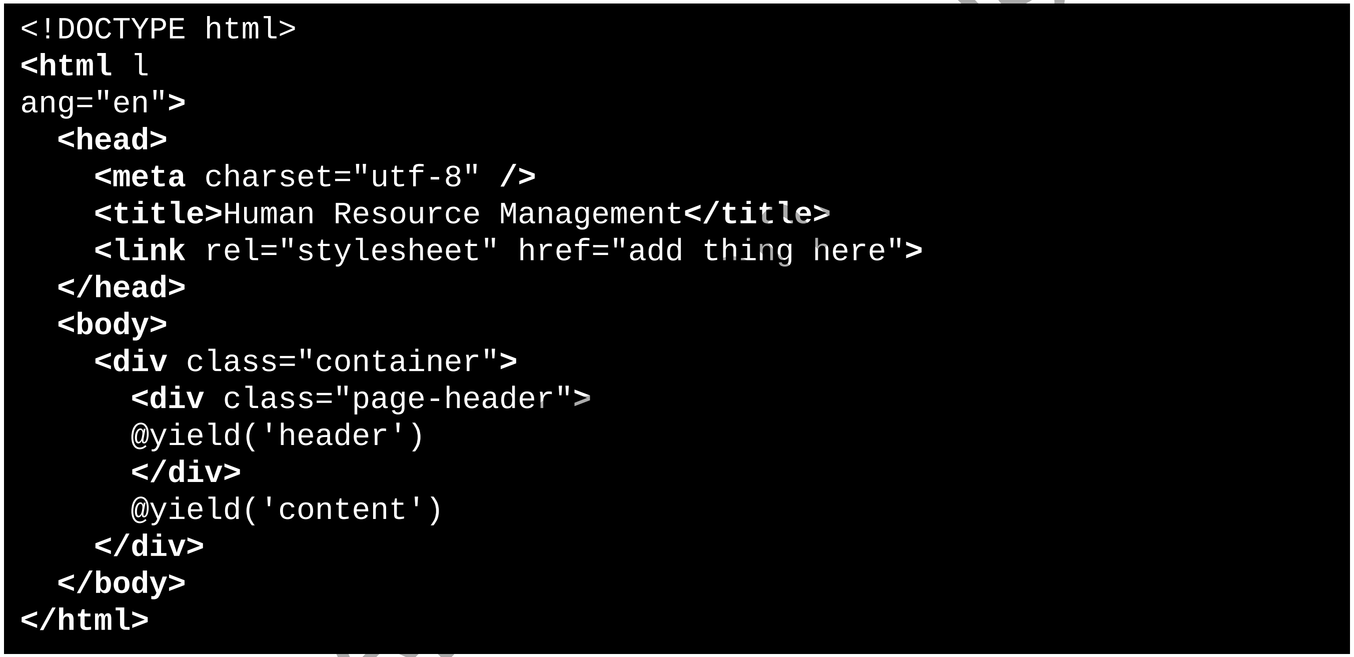
Views

Blade Templating Engine

Master Template

Individual Views and Blade Directives

# CREATING WEB PAGES (2-9)



```
<!DOCTYPE html>
<html l
ang="en">
  <head>
    <meta charset="utf-8" />
    <title>Human Resource Management</title>
    <link rel="stylesheet" href="add thing here">
  </head>
  <body>
    <div class="container">
      <div class="page-header">
      @yield('header')
      </div>
      @yield('content')
    </div>
  </body>
</html>
```

# CREATING WEB PAGES (3-9)

## Adding Forms

- Necessary to link other pages
- Receive input from users
- Created using the form builder class
- LaravelCollective

# CREATING WEB PAGES (4-9)

```
composer require laravelcollective/html
```

```
'providers' => [
  ...
    Collective\Html\HtmlServiceProvider::class,
  ...
  ],
```

```
'aliases' => [
   ...
        'Form' => Collective\Html\FormFacade::class,
   ...
  ],
```

# CREATING WEB PAGES (5-9)

```
<!-- Stored in resources/views/add.blade.php -->

@extends('layouts.master')
@section('header')
  <h1>Use the following form to add a record for new employee</h1>
@endsection
@section('content')
  {{ Form::open(['url'=>'/add', 'method'=>'post']) }}
    <pre>
        {{ Form::label('name','Name:') }} {{ Form::text('username') }}
        {{ Form::label('designation','Designation:') }}
{{Form::text('designation')}}
        {{ Form::label('salary','Salary:') }} {{ Form::number('name', 'value') }}
        {{ Form::submit('Submit') }}
    </pre>
  {{ Form::close() }}
@endsection
```

# CREATING WEB PAGES (6-9)

```
<!-- Stored in resources/views/display.blade.php -->

@extends('layouts.master')
@section('header')
  <h1>Use the following form to display a record for an employee with their
id</h1>
@endsection
@section('content')
  {{ Form::open(['url'=>'/display', 'method'=>'post']) }}
    <pre>
        {{ Form::label('id','ID:') }}   {{ Form::text('id') }}
        {{ Form::submit('Search') }}
    </pre>
  {{ Form::close() }}
@endsection
```

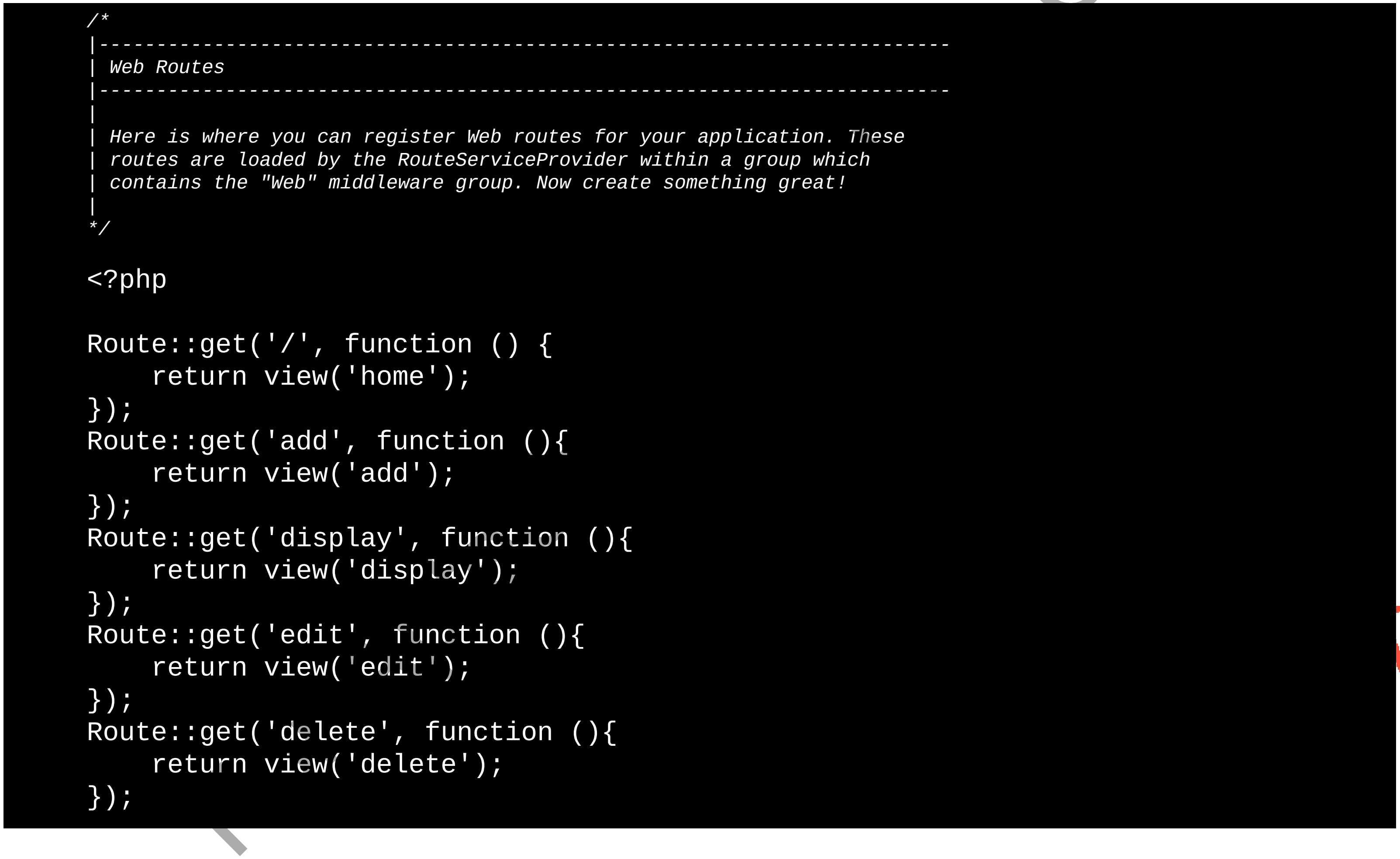
# CREATING WEB PAGES (7-9)

```
<!-- Stored in resources/views/edit.blade.php -->

@extends('layouts.master')
@section('header')
  <h1>Use the following form to fetch and edit an employee record with their id</h1>
@endsection
@section('content')
  {{ Form::open(['url'=>'/edit', 'method'=>'post']) }}
    <pre>
        {{ Form::label('id','ID:') }} {{ Form::text('id') }}
        {{ Form::submit('Submit') }}
    </pre>
  {{ Form::close() }}
@endsection
```

# CREATING WEB PAGES (8-9)

```
<!-- Stored in resources/views/delete.blade.php -->

@extends('layouts.master')
@section('header')
  <h1>Use the following form to delete an employee record</h1>
@endsection
@section('content')
  {{ Form::open(['url'=>'/delete', 'method'=>'delete']) }}
    <pre>
        {{ Form::label('id','ID:') }}   {{ Form::text('id') }}
        {{ Form::submit('Search') }}
    </pre>
  {{ Form::close() }}
@endsection
```

# CREATING WEB PAGES (9-9)



```
/*
|--------------------------------------------------------------------------
| 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!
|
*/

<?php

Route::get('/', function () {
    return view('home');
});
Route::get('add', function (){
    return view('add');
});
Route::get('display', function (){
    return view('display');
});
Route::get('edit', function (){
    return view('edit');
});
Route::get('delete', function (){
    return view('delete');
});
```

# SUMMARY

- A query builder class is used to Create, Read, Update, and Delete records (CRUD operations).
- SQLite is a fast and reliable database engine that is used as the DBMS software for Web applications.
- Schema is a builder class used to create tables.
- The Migrations feature acts as a version control system for a database that includes many tables.
- The Bootstrap CSS framework uses the asset() helper function to wrap the other folders and resources together.
- LaravelCollective repository is created by the open source community to help developers who use Form builder class.