**Web development with php & laravel**

**Assignment of Module - 15**

**Assignment Title: Laravel Concepts Assessment**

**Name : MD. RAQIBUL ISLAM HOWLADER**

**Batch No : 01**

**Ostad login No : 01681533509**

**WhatsApp No : 01782333171**

**Email : raqibulislam907@gmail.com**

***Submit date:29 -05-2023***

**Assignment:** Laravel Concepts Assessment

**Task 1: Request Validation**

Implement request validation for a registration form that contains the following fields: name, email, and password. Validate the following rules:

name: required, string, minimum length 2.

email: required, valid email format.

password: required, string, minimum length 8.

**Answer:**

php artisan make:request RegistrationRequest

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegistrationRequest extends FormRequest

{

public function authorize()

{

return true;

}

public function rules()

{

return [

'name' => 'required|string|min:2',

'email' => 'required|email',

'password' => 'required|string|min:8',

];

}

}

use App\Http\Requests\RegistrationRequest;

public function register(RegistrationRequest $request)

{

// Registration logic goes here

}

**Task 2: Request Redirect**

Create a route /home that redirects to /dashboard using a 302 redirect.

**Answer:**

Route::get('/home', function () {

return redirect('/dashboard');

});

**Task 3: Global Middleware**

Create a global middleware that logs the request method and URL for every incoming request. Log the information to the Laravel log file.

**Answer**:

php artisan make:middleware LogRequest

protected $middleware = [

// Other middleware entries...

\App\Http\Middleware\LogRequest::class,

];

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Support\Facades\Log;

class LogRequest

{

public function handle($request, Closure $next)

{

$method = $request->method();

$url = $request->fullUrl();

Log::info("Request Method: $method, URL: $url");

return $next($request);

}

}

**Task 4: Route Middleware**

Create a route group for authenticated users only. This group should include routes for /profile and /settings. Apply a middleware called AuthMiddleware to the route group to ensure only authenticated users can access these routes.

**Answer:**

Route::middleware(['auth'])->group(function () {

Route::get('/profile', 'ProfileController@index');

Route::get('/settings', 'SettingsController@index');

});

**Task 5: Controller**

Create a controller called ProductController that handles CRUD operations for a resource called Product. Implement the following methods:

index(): Display a list of all products.

create(): Display the form to create a new product.

store(): Store a newly created product.

edit($id): Display the form to edit an existing product.

update($id): Update the specified product.

destroy($id): Delete the specified product.

**Answer:**

php artisan make:controller ProductController –resource

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;

class ProductController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$products = Product::all();

return view('products.index', compact('products'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

return view('products.create');

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

// Validate the request data

$validatedData = $request->validate([

'name' => 'required',

'price' => 'required|numeric',

// Add other validation rules for product fields

]);

// Create a new product

$product = Product::create($validatedData);

return redirect('/products')->with('success', 'Product created successfully.');

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$product = Product::findOrFail($id);

return view('products.edit', compact('product'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

// Validate the request data

$validatedData = $request->validate([

'name' => 'required',

'price' => 'required|numeric',

// Add other validation rules for product fields

]);

// Find the product by ID and update its attributes

$product = Product::findOrFail($id);

$product->update($validatedData);

return redirect('/products')->with('success', 'Product updated successfully.');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

$product = Product::findOrFail($id);

$product->delete();

return redirect('/products')->with('success', 'Product deleted successfully.');

}

}

**Task 6: Single Action Controller**

Create a single action controller called ContactController that handles a contact form submission. Implement the \_\_invoke() method to process the form submission and send an email to a predefined address with the submitted data.

**Answer:**

php artisan make:controller ContactController –invokable

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Mail;

use App\Mail\ContactFormMail;

class ContactController extends Controller

{

/\*\*

\* Process the contact form submission and send an email.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function \_\_invoke(Request $request)

{

$request->validate([

'name' => 'required',

'email' => 'required|email',

'message' => 'required',

]);

$data = [

'name' => $request->name,

'email' => $request->email,

'message' => $request->message,

];

// Send email to the predefined address

Mail::to('your-email@example.com')->send(new ContactFormMail($data));

return redirect()->back()->with('success', 'Thank you for your message. We will be in touch soon.');

}

}

php artisan make:mail ContactFormMail

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

class ContactFormMail extends Mailable

{

use Queueable, SerializesModels;

public $data;

/\*\*

\* Create a new message instance.

\*

\* @param array $data

\*/

public function \_\_construct($data)

{

$this->data = $data;

}

/\*\*

\* Build the message.

\*

\* @return $this

\*/

public function build()

{

return $this->view('emails.contact')->subject('New Contact Form Submission');

}

}

**Task 7: Resource Controller**

Create a resource controller called PostController that handles CRUD operations for a resource called Post. Ensure that the controller provides the necessary methods for the resourceful routing conventions in Laravel.

**Answer:**

php artisan make:controller PostController –resource

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Post;

class PostController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

$posts = Post::all();

return view('posts.index', compact('posts'));

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

return view('posts.create');

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

// Validate the request data

$validatedData = $request->validate([

'title' => 'required',

'content' => 'required',

// Add other validation rules for post fields

]);

// Create a new post

$post = Post::create($validatedData);

return redirect('/posts')->with('success', 'Post created successfully.');

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

$post = Post::findOrFail($id);

return view('posts.show', compact('post'));

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

$post = Post::findOrFail($id);

return view('posts.edit', compact('post'));

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

// Validate the request data

$validatedData = $request->validate([

'title' => 'required',

'content' => 'required',

// Add other validation rules for post fields

]);

// Find the post by ID and update its attributes

$post = Post::findOrFail($id);

$post->update($validatedData);

return redirect('/posts')->with('success', 'Post updated successfully.');

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

$post = Post::findOrFail($id);

$post->delete();

return redirect('/posts')->with('success', 'Post deleted successfully.');

}

}

**Task 8: Blade Template Engine**

Create a Blade view called welcome.blade.php that includes a navigation bar and a section displaying the text "Welcome to Laravel!".

**Answer:**

//app.blade.php

<!DOCTYPE html>  
<html>  
<head>  
 <title>Welcome to Laravel</title>  
</head>  
<body>  
<nav>  
 <!-- Navigation bar code goes here -->  
</nav>  
  
<section>  
 <h1>Welcome to Laravel!</h1>  
 <!-- Additional content goes here -->  
</section>  
</body>  
</html>

//welcome.blade.php

@extends(‘layout.app’)

php artisan make:controller HomeController

<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
  
class HomeController extends Controller  
{  
 function page(Request $request){  
 return view( 'Home');  
 }  
}

//Route:

<?php  
  
use App\Http\Controllers\HomeController;  
use Illuminate\Support\Facades\Route;  
  
Route::get('/',[HomeController::class,'page']);