**EXPERIMENT NO. 08**

**Title:** Laravel Framework

**Aim:** To Understand the Basic of Laravel Framework of PHP including view, controller and route.

**Objective:** To implement the program using Basics of Laravel Framework

**Contents:**

MCQ and Technical Questions on Laravel framework

Program to implement in Lab:

1. **Create the Registration page in Laravel**

Step 1: crate a Laravel application

Step 2: create the registration view with required fields

Step 3: create route to registration which call method in controller and return registration view

Once click to submit, call method in controller and return view which shows all details submitted

Step 4: create controller for above

1. **Create the Login Application.**

Step 1: crate a Laravel application

Step 2: create the login view with username and password

Step 3: create route to login which call method in controller and return login view

Also create the route for post login which will call method from controller to check username and password. If success and profile view et open another open login view again

Step 4: create controller for above

Extra:

To manually create login and registration functionality in Laravel, you'll need to perform the following steps:

1. **Create a Registration Controller**
2. **Create a Login Controller**
3. **Set up Routes**
4. **Create Views for Registration and Login Forms**
5. **Handle Authentication Logic**

### 1. ****Create the Registration Controller****

To create a registration controller, run the following artisan command:

php artisan make:controller Auth/RegisterController

Now, let's add the registration logic to the RegisterController.

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller

{

public function showRegistrationForm()

{

return view('auth.register');

}

public function register(Request $request)

{

// Validate the form input

$validator = Validator::make($request->all(), [

'name' => ['required', 'string', 'max:255'],

'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],

'password' => ['required', 'string', 'min:8', 'confirmed'],

]);

if ($validator->fails()) {

return redirect()->back()->withErrors($validator)->withInput();

}

// Create and store the user

User::create([

'name' => $request->name,

'email' => $request->email,

'password' => Hash::make($request->password),

]);

// Automatically log in the user after registration

auth()->attempt($request->only('email', 'password'));

return redirect()->route('home');

}

}

### 2. ****Create the Login Controller****

Similarly, create a login controller:

php artisan make:controller Auth/LoginController

Now add the login logic to the LoginController.

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class LoginController extends Controller

{

public function showLoginForm()

{

return view('auth.login');

}

public function login(Request $request)

{

// Validate the form input

$credentials = $request->validate([

'email' => ['required', 'email'],

'password' => ['required'],

]);

// Attempt to log the user in

if (Auth::attempt($credentials)) {

// Authentication passed

$request->session()->regenerate();

return redirect()->intended('home');

}

// Authentication failed

return back()->withErrors([

'email' => 'The provided credentials do not match our records.',

])->onlyInput('email');

}

public function logout(Request $request)

{

Auth::logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

return redirect('/login');

}

}

### 3. ****Set Up Routes****

Now let's define the routes for registration, login, and logout in the routes/web.php file:

use App\Http\Controllers\Auth\RegisterController;

use App\Http\Controllers\Auth\LoginController;

Route::get('/register', [RegisterController::class, 'showRegistrationForm'])->name('register');

Route::post('/register', [RegisterController::class, 'register']);

Route::get('/login', [LoginController::class, 'showLoginForm'])->name('login');

Route::post('/login', [LoginController::class, 'login']);

Route::post('/logout', [LoginController::class, 'logout'])->name('logout');

### 4. ****Create Views for Registration and Login Forms****

Create the registration and login views in the resources/views/auth directory.

#### Registration Form (resources/views/auth/register.blade.php):

<form method="POST" action="{{ route('register') }}">

@csrf

<div>

<label for="name">Name</label>

<input id="name" type="text" name="name" value="{{ old('name') }}" required autofocus>

@error('name')

<span>{{ $message }}</span>

@enderror

</div>

<div>

<label for="email">Email</label>

<input id="email" type="email" name="email" value="{{ old('email') }}" required>

@error('email')

<span>{{ $message }}</span>

@enderror

</div>

<div>

<label for="password">Password</label>

<input id="password" type="password" name="password" required>

@error('password')

<span>{{ $message }}</span>

@enderror

</div>

<div>

<label for="password\_confirmation">Confirm Password</label>

<input id="password\_confirmation" type="password" name="password\_confirmation" required>

</div>

<button type="submit">Register</button>

</form>

#### Login Form (resources/views/auth/login.blade.php):

html

Copy code

<form method="POST" action="{{ route('login') }}">

@csrf

<div>

<label for="email">Email</label>

<input id="email" type="email" name="email" value="{{ old('email') }}" required autofocus>

@error('email')

<span>{{ $message }}</span>

@enderror

</div>

<div>

<label for="password">Password</label>

<input id="password" type="password" name="password" required>

@error('password')

<span>{{ $message }}</span>

@enderror

</div>

<button type="submit">Login</button>

</form>

### 5. ****Handle Authentication Logic****

By default, Laravel provides the User model in the App\Models\User.php file. If you need to modify user authentication, you can adjust this model or the Auth facade.

### 6. ****Run Migrations****

Ensure the users table is created by running the migrations:

php artisan migrate

### 7. ****Set the Home Route****

After a successful login or registration, you'll want to redirect users to a homepage. Set a simple route in your web.php:

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

return 'Welcome to the home page!';

})->name('home')->middleware('auth');

### Conclusion

This approach allows you to manually create login and registration functionality in Laravel. You can extend this by adding password resets, email verification, and other authentication-related features as needed! Let me know if you'd like to add those.

**Conclusion:**

Write the crux of the experiment. Here crux means very important point, complicated and again it should be understandable by you. Do not write conclusion like thus we have studied the ……… in PHP programming.