MODULE:3

1. Create Following Routes.(for frontend sideapplication)

¬ Home page

```
->Route::get('/',[clientcontroller::class,'home'])-
>name('clienthome');
```

¬ About us

```
->Route::get('/aboutus',[clientcontroller::class,'aboutus'])-
>name('aboutus');
```

¬ Gallery

```
->Route::get('/galary',[clientcontroller::class,'galary'])-
>name('galary');
```

¬ Registration

```
>Route::get('/registration',[clientcontroller::class,'registrat
ion'])->name('registration');
```

¬ Login

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

2. How to pass Multiple Variables in route?

```
->Route::get('/cart/{id}',[clientcontroller::class,'cart'])-
>name('cart');
```

3. How to pass variables that can be null in Route?

```
-> Route::get('/',[clientcontroller::class,'home'])-
>nullable('');
```

4 Create custom auth using middleware.

```
->Route::middleware(['my'])->group(function (){
    Route::get('/myprofile',[clientcontroller::class,'myprofile
'])->name('myprofile');
Route::get('/cart/{id}',[clientcontroller::class,'cart'])-
>name('cart');
Route::post('/cart/{id}',[clientcontroller::class,'cart'])-
>name('cart');
Route::get('/order/{id}',[clientcontroller::class,'order'])-
>name('order');
Route::post('/order/{id}',[clientcontroller::class,'order'])-
>name('order');
Route::get('/mycart',[clientcontroller::class,'mycart'])-
>name('mycart');
Route::get('/myorder',[clientcontroller::class,'myorder'])-
>name('myorder');
});
```

5 Check IP address is valid or not for the home page using middleware.

->

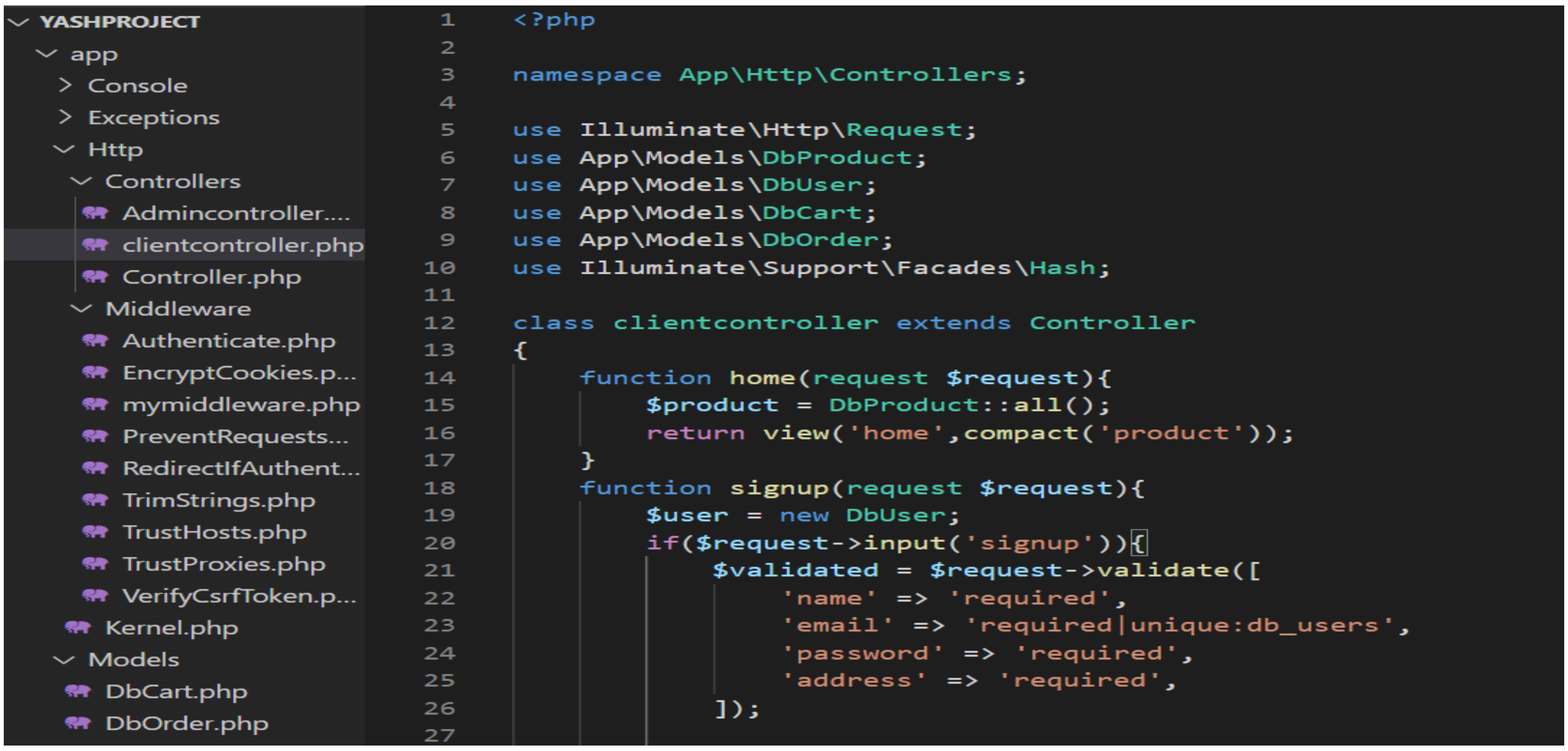
6 List Types Of middleware in Laravel with examples.

->Goble middleware

->group middleware

->route middleware

1. Generate Resource Controller For employees.



```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\DbProduct;
use App\Models\DbUser;
use App\Models\DbCart;
use App\Models\DbOrder;
use Illuminate\Support\Facades\Hash;

class clientcontroller extends Controller
{
    function home(request $request){
        $product = DbProduct::all();
        return view('home',compact('product'));
    }
    function signup(request $request){
        $user = new DbUser;
        if($request->input('signup')){
            $validated = $request->validate([
                'name' => 'required',
                'email' => 'required|unique:db_users',
                'password' => 'required',
                'address' => 'required',
            ]);

```

2. In Employee Controller’s action Call Middleware

```
->Route::middleware(['my'])->group(function (){
Route::get('/myprofile',[clientcontroller::class,'myprofile'])->name('myprofile');
Route::get('/cart/{id}',[clientcontroller::class,'cart'])->name('cart');
Route::post('/cart/{id}',[clientcontroller::class,'cart'])->name('cart');
Route::get('/deletecart/{id}',[clientcontroller::class,'deletecart'])->name('deletecart');
Route::get('/order/{id}',[clientcontroller::class,'order'])->name('order');
Route::get('/deleteorder/{id}',[clientcontroller::class,'deleteorder'])->name('deleteorder');
Route::post('/order/{id}',[clientcontroller::class,'order'])->name('order');
Route::get('/mycart',[clientcontroller::class,'mycart'])->name('mycart');
Route::get('/myorder',[clientcontroller::class,'myorder'])->name('myorder');
});
```

3. How to remove route caching?

-> PHP artisan configuration:clear

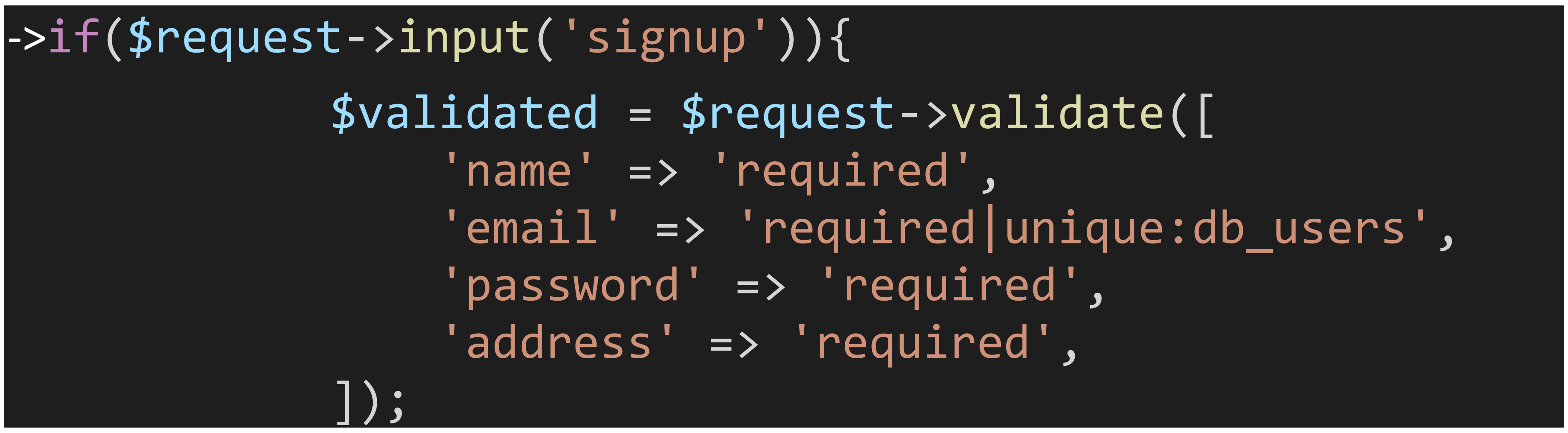
4. Create Custom Macro For search User

```
-> use Illuminate\Database\Eloquent\Builder;

// ...

Builder::macro('search', function(string $attribute, string $searchTerm) {
   return $this->where($attribute, 'LIKE', "%{$searchTerm}%");
});
```

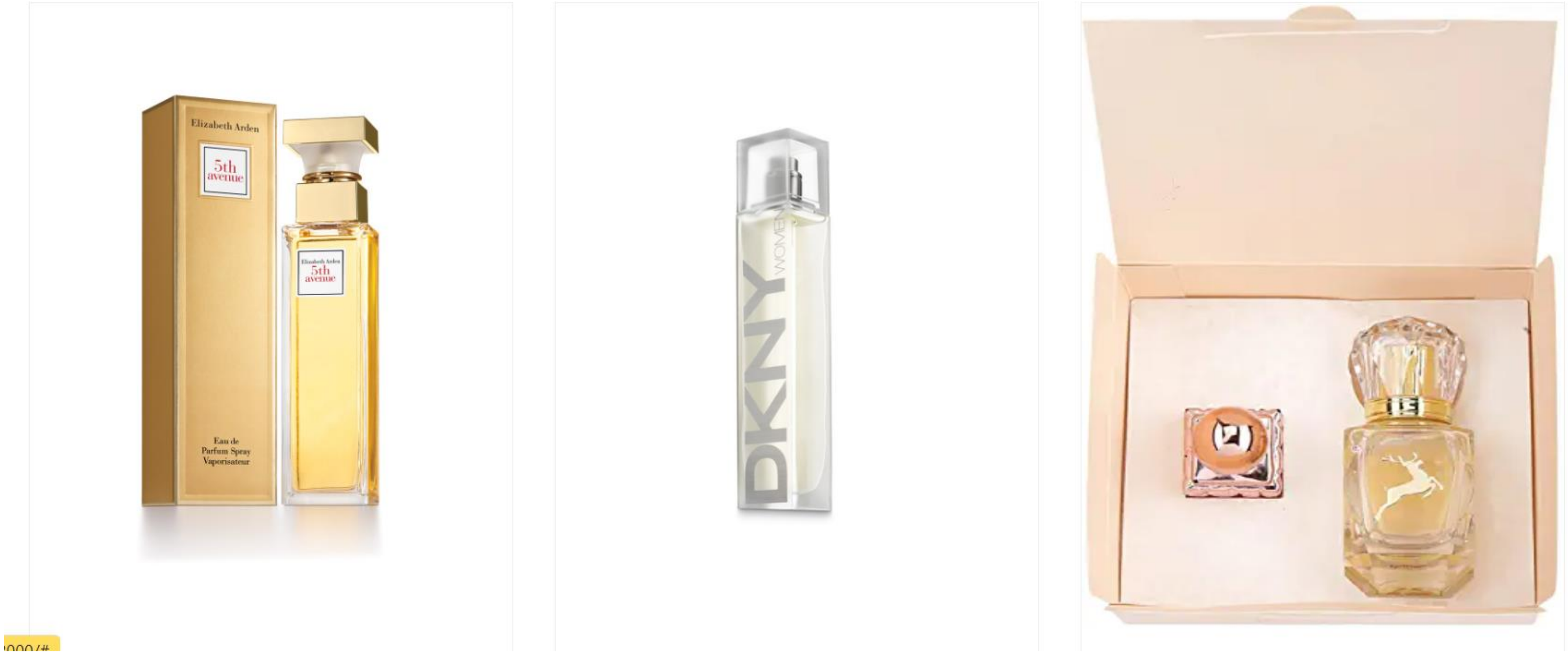
5. Add server-side Form validation



```
->if($request->input('signup')){
            $validated = $request->validate([
                'name' => 'required',
                'email' => 'required|unique:db_users',
                'password' => 'required',
                'address' => 'required',
            ]);
```

Practical 1. Create Photo Gallery for above route.

Note: upload files from the admin side. Show all the images on the user side



MODULE:4

1. How to do config database in laravel

-> Laravel makes connecting with databases and running queries extremely simple. The database configuration file is app/config/database.php. In this file you may define all of your database connections, as well as specify which connection should be used by default.

2. Call MySQLi Store Procedure from Laravel.

-> Step 1: DB::statement('call new_user(?, ?, ?, ?)',["myemail@test.com","mypassword","myname","mysurname"]);

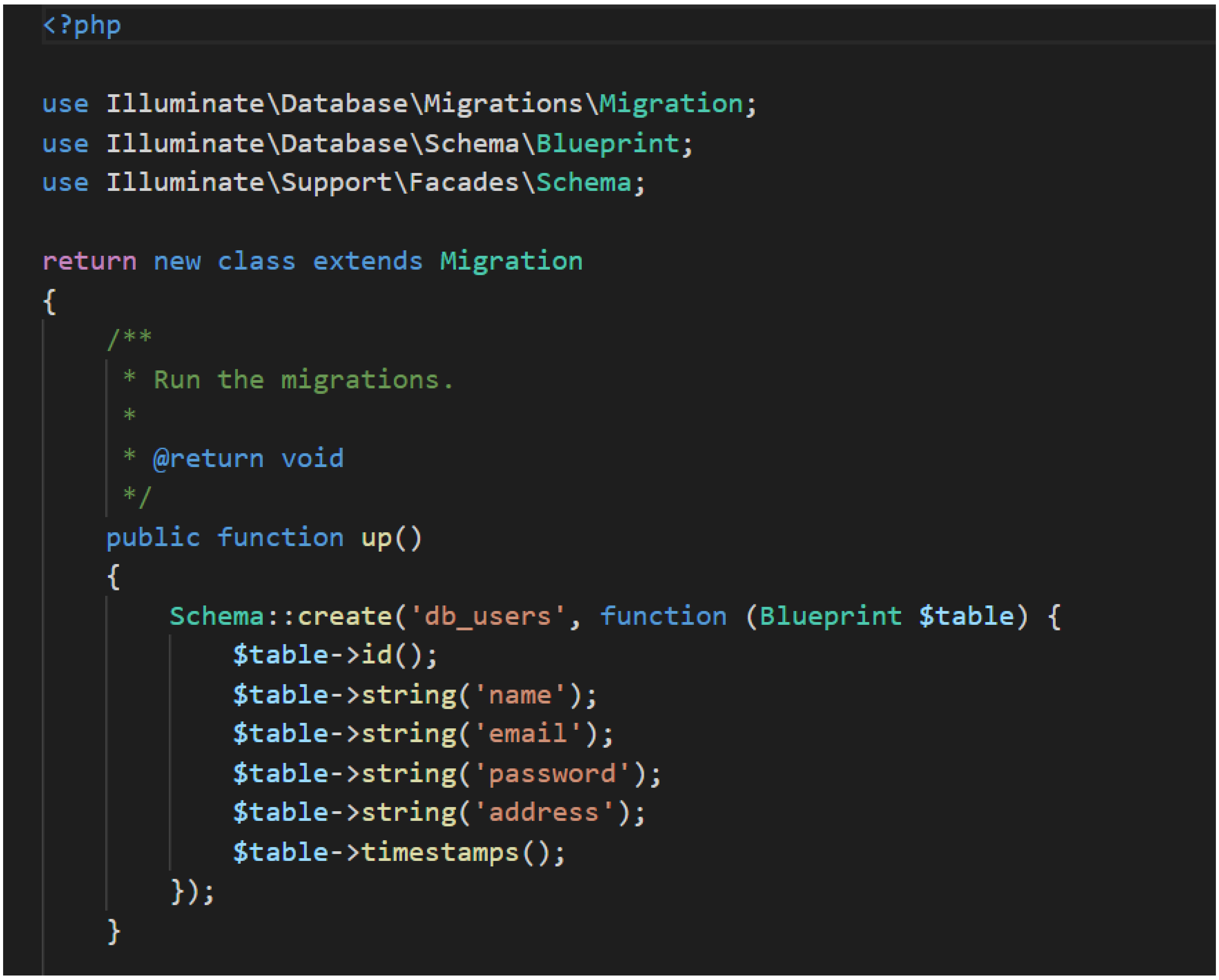
3. Apply Curd Operation through Query Builder For Employee Management

```
-> function signup(request $request){
        $user = new DbUser;
        if($request->input('signup')){
            $validated = $request->validate([
                'name' => 'required',
                'email' => 'required|unique:db_users',
                'password' => 'required',
                'address' => 'required',
            ]);

            $user->name=$request->input('name');
            $user->email=$request->input('email');
            $user->password= Hash::make($request-
>input('password'));
            $user->address=$request->input('address');
            $user->role_id=2;
            $user->save();
            return redirect()->route('login');
        }
        return view('signup');
    }
```

4. Create All Migration For Employee management

->



```php
<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('db_users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->string('password');
            $table->string('address');
            $table->timestamps();
        });
    }
```

MODULE:5

1. Explain ORM

->Eloquent is an object relational mapper (ORM) that is included by default within the Laravel framework. An ORM is software that facilitates handling database records by representing data as objects, working as a layer of abstraction on top of the database engine used to store an application’s data.

2. Do Curd using Eloquent Query.

```
->function myorder(request $request){
        if($request->session()->has('user')){
            $s = $request->session()->get('user');
        }
        $order = 
DbOrder::select("db_orders.id","db_products.p_name","db_p
roducts.price","db_orders.quentity","db_products.image","
db_orders.status")
        -
>join('db_products','db_products.id','db_orders.pid')
        ->where('db_orders.uid',$s->id)
        ->get();
        return view('myorder',compact('order'));
    }
```

3. Explain - Eloquent Relationships

->Eloquent relationships are defined as methods in your eloquent model classes. Since relationships also serve as powerful query builders, defining relationships as method provide powerful method chaining and querying capabilities.

4. What is Eager Loading and lazy loading?

->Dynamic properties are “lazy loading”, meaning they will only load their relationship data when you actually access them. Because of this, developers often use eager loading to pre-load relationships they know will be accessed after loading the model. Eager loading provides a significant reduction in SQL queries that must be executed to load a model’s relations.

5. Do Session for Employee Management System.

```
->function login(request $request){
        $user = DbUser::all();
        if($request->input('login')){
            $email = $request->input('email');
            $password = $request->input('password');
            $user = DbUser::where('email',$email)-
>first();
            if(Hash::check($password,$user->password)){
                if($user->role_id == 2){
                    $request->session()-
>put('user',$user);
                    return redirect()-
>route('clienthome');
                }
                else{
                    $request->session()-
>put('admin',$user);
                    return redirect()->route('home');
                }
            }
        }
        return view('login');
    }
```