**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 6

la cursul ***„Tehnologii web”***

***Tema: Utilizarea unui CMS sau Framework pentru dezvoltarea unei aplicații Web.***

**A efectuat :**   **St. gr. CR-221FR Serba Cristina**

**A verificat: Rusu Viorel**

**Chișinău 2024**

# Sarcina lucrării:

# a) Alegeti un CMS sau Famework (JS sau pe oricare script server) pentru dezvoltarea unei aplicatii Web. Motivati alegerea.

# b) Dezvoltati o aplicatie functionala in baza alegerii facute.

# c) Puneti in raport PrintScreen-uri la paginile create. In cazul CMS-ului ilustrati moduld e interactiune cu partea administrativa unde ati intervenit pentru adaptari, iar in cazul Framework-ului - citeva secvente de cod creat de voi pe care-l considerati important sau interesant .

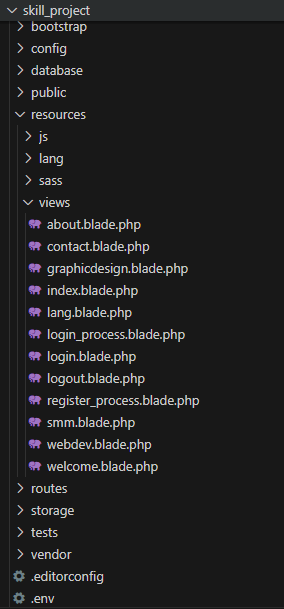
# Mersul lucrării:

Pentru a dezvolta aplicația, am ales framework-ul Laravel în primul rând deoarece este bazat pe limbajul PHP, așa cum aplicația din lucrările trecute era în mare parte scrisă în PHP. De asemenea Laravel vine cu o gamă largă de caracteristici și funcționalități încorporate care simplifică sarcinile comune de dezvoltare web. Acestea includ autentificare, rutare, stocare în cache, sesiuni, coadă, validare și multe altele. Ecosistemul Laravel oferă, de asemenea, pachete suplimentare prin Composer, facilitând extinderea capabilităților aplicației.

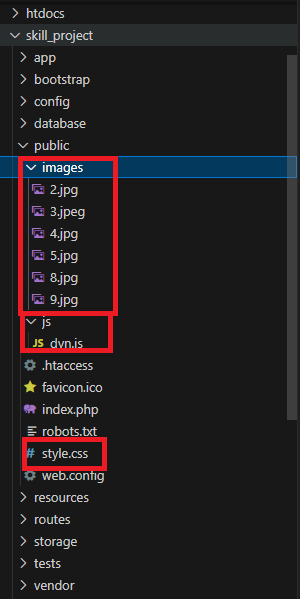
Astfel, pentru a crea proiectul, am introdus comanda

composer create-project --prefer-dist laravel/laravel skill\_project

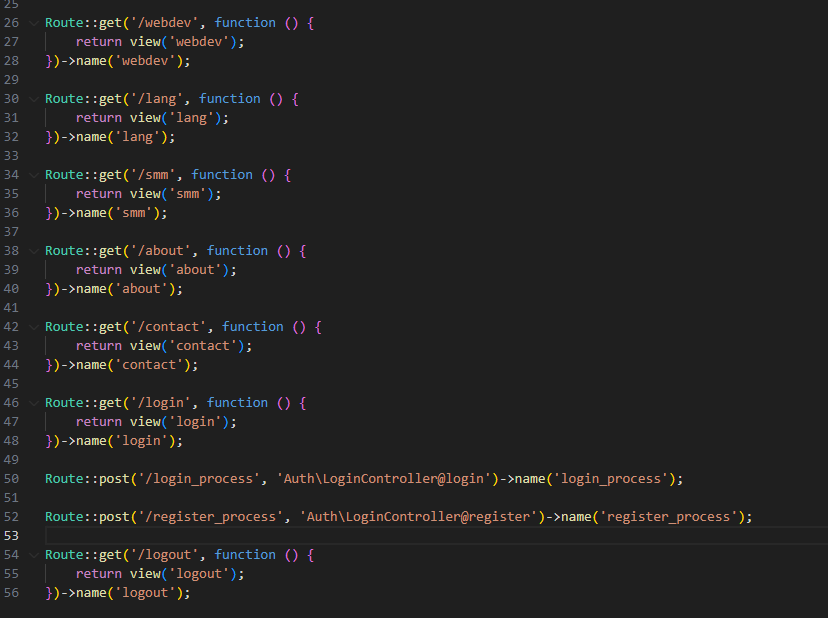
apoi am mutat toate fișierele sursă existente în folderul dat:



Imaginile, stilul css și codul javascript au fost adăugate în



Apoi am definit rutele spre fișierele adăugate recent



Iar calea spre ele în codul sursă a fost definită în felul următor:

  <link rel="stylesheet" href="{{ asset('style.css') }}">

<div class="dropdown-menu" aria-labelledby="navbarDropdown" id="dropdownMenuContent">

              <a class="dropdown-item" href="{{ route('graphicdesign') }}">Design grafic</a>

              <a class="dropdown-item" href="{{ route('webdev') }}">Web Development</a>

              <a class="dropdown-item" href="{{ route('lang') }}">Limbi</a>

              <a class="dropdown-item" href="{{ route('smm') }}">SMM</a>

            </div>

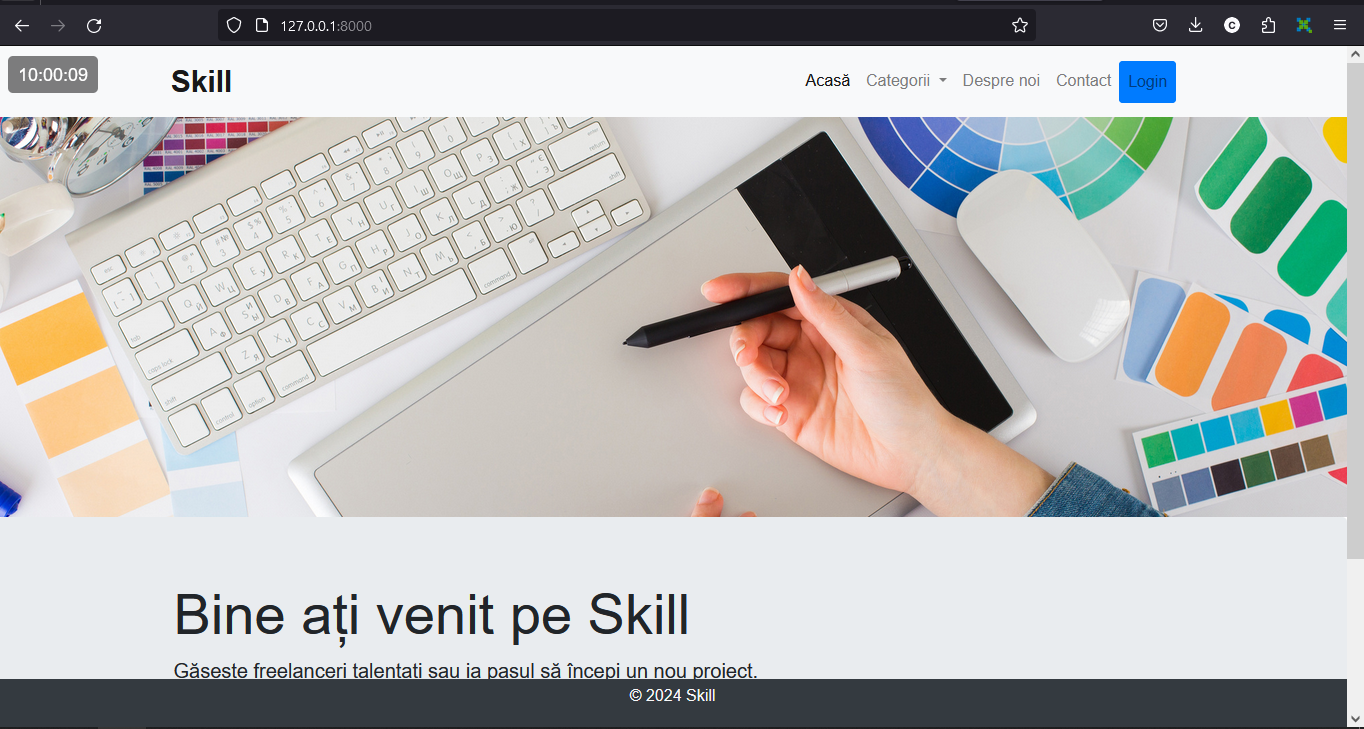
  <script src="{{ asset('js/dyn.js') }}"></script>

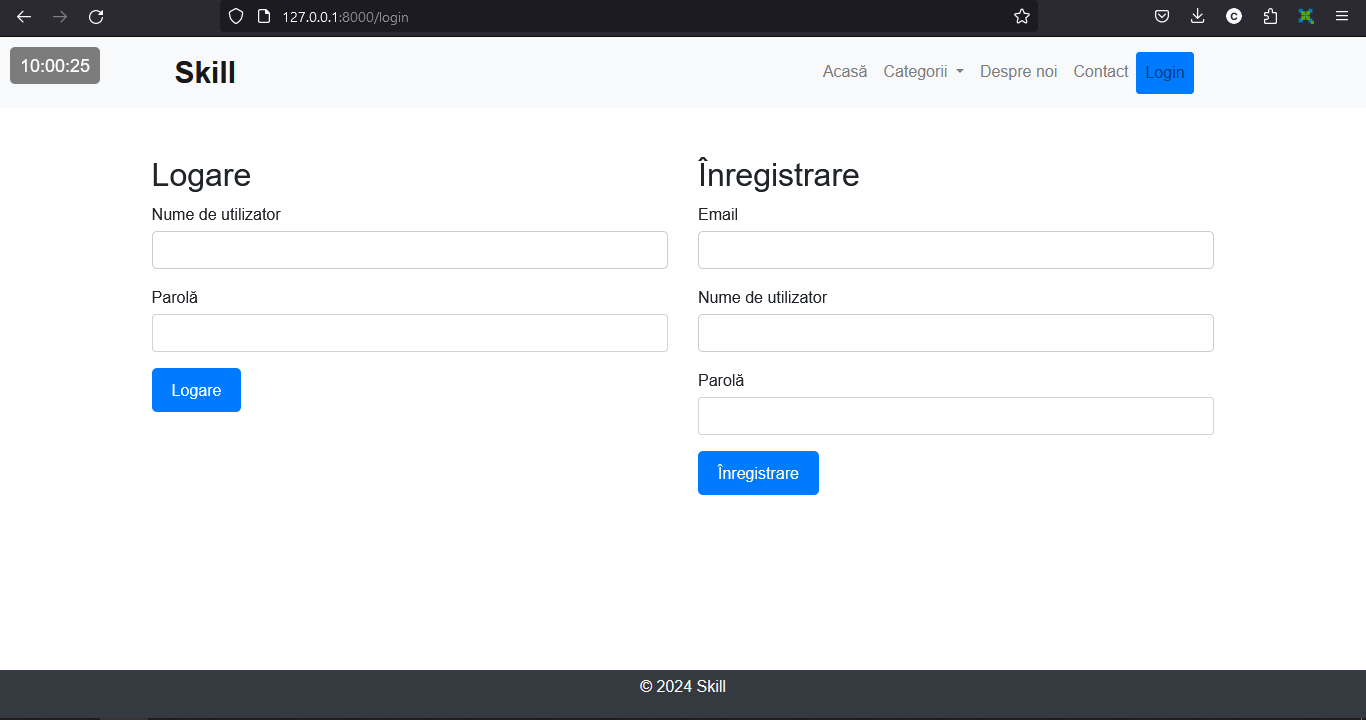
Definirea metodelor pentru logare/înregistrare a fost definită în LoginController și, respectiv, RegisterController

Exemplu:

|  |
| --- |
| <?php  namespace App\Http\Controllers\Auth;  use App\Http\Controllers\Controller;  use App\Providers\RouteServiceProvider;  use App\User;  use Illuminate\Foundation\Auth\RegistersUsers;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Hash;  use Illuminate\Support\Facades\Validator;  class RegisterController extends Controller  {      use RegistersUsers;      protected $redirectTo = RouteServiceProvider::HOME;      public function \_\_construct()      {          $this->middleware('guest');      }      protected function validator(array $data)      {          return Validator::make($data, [              'name' => ['required', 'string', 'max:255'],              'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],              'password' => ['required', 'string', 'min:8', 'confirmed'],          ]);      }      protected function create(array $data)      {          return User::create([              'name' => $data['name'],              'email' => $data['email'],              'password' => Hash::make($data['password']),          ]);      }      public function registerProcess(Request $request)      {          $this->validator($request->all())->validate();          $user = $this->create($request->all());          $this->guard()->login($user);          return $this->registered($request, $user)                          ?: redirect($this->redirectPath());      }  } |

***Exemplu de pagini***:





# Concluzii:

În concluzie, în această lucrare am optat să-mi extind aplicația existentă folosind Laravel din mai multe motive: cu acest framework, am putut avea o abordare mai organizată și mai modulară a dezvoltării. Separând componente precum controlerele, modelele și vizualizările, am reușit să-mi păstrez baza de cod curată și întreținabilă, făcând mai ușoară adăugarea de noi funcții și scalarea aplicației în timp. Sistemul de rutare și middleware mi-au permis să gestionez eficient cererile primite și să aplic autentificarea, înregistrarea în database și modificarea cererilor. În concluzie, folosirea Laravel în proiectul meu mi-a permis să-i îmbunătățesc funcționalitatea, să-i îmbunătățesc mentenabilitatea și să-i eficientizez procesul de dezvoltare. Valorificând caracteristicile puternice și sintaxa expresivă a lui Laravel, am reușit să construiesc o aplicație robustă și bogată în funcții, care să răspundă eficient nevoilor utilizatorilor.