



Neumann János Egyetem

GAMF Műszaki és Informatikai Kar

GSZABAN-RAPIDWAD-1 –

Gyakorlatvezető: Dr. Subecz Zoltán

Fehér Donát – ECFE35

mérnökinformatika szak – levelező tagozat

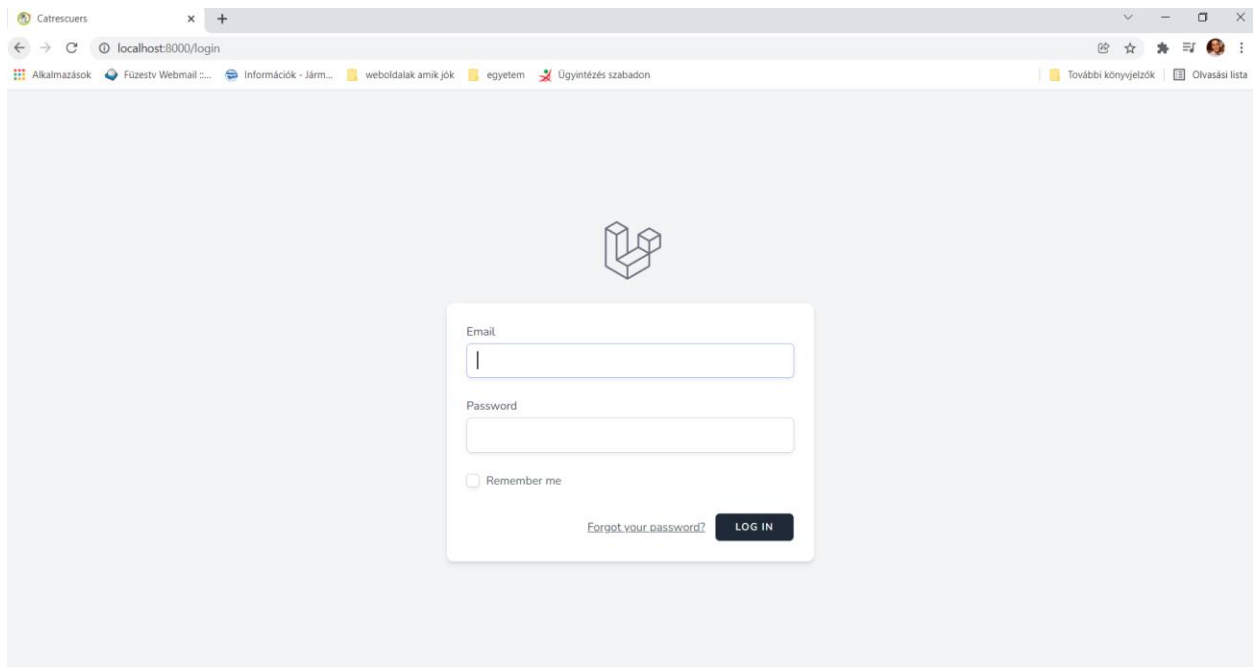
Rapid web-application development with Laravel
framework - online course

I have chosen the <https://taxdesignation.org/> nonprofit foundation's webpage to recreate.

The following paths have been defined in web.php. /Dashboard is only available after login. The authentication is provided by Laravel Breeze, which I used for the project with the recommendation of an acquaintance. It also handles login and registration. It also receives get and post http requests on the /contact path.

```
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/', function () {
6     return view('welcome');
7 });
8
9 //Auth:routes();
10
11 Route::get('/contact', [\App\Http\Controllers\ContactController::class, 'show'])->middleware('guest');
12 Route::post('/contact', [\App\Http\Controllers\ContactController::class, 'store'])->middleware('guest');
13 Route::get('/galery', [\App\Http\Controllers\GaleryController::class, 'show'])->middleware('guest');
14 Route::get('/regional', [\App\Http\Controllers\RegionalController::class, 'show'])->middleware('guest');
15
16
17 Route::get('/dashboard', function () {
18     return view('dashboard');
19 })->middleware(['auth'])->name('dashboard');
20
21 require __DIR__.'/auth.php';
22
23
```

Login screen:



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/login'. The browser's tab is titled 'Catrescuers'. The page features a light gray background with a central white login form. Above the form is a logo consisting of three stacked cubes. The form contains the following elements: an 'Email' label above a text input field, a 'Password' label above a text input field, a 'Remember me' checkbox, a 'Forgot your password?' link, and a dark blue 'LOG IN' button.

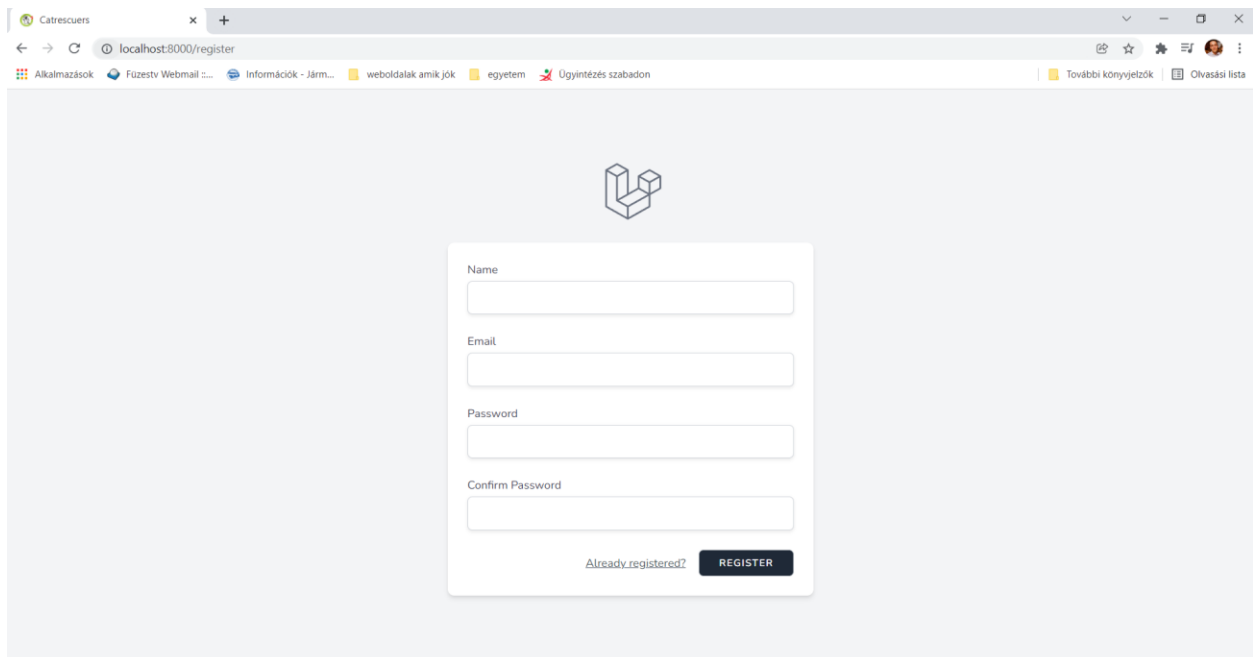
Email

Password

☐ Remember me

[Forgot your password?](#) **LOG IN**

Register screen:



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/register'. The browser's tab is titled 'Catrescuers'. The page features a light gray background with a central white registration form. Above the form is a logo consisting of three stacked cubes. The form contains the following elements: a 'Name' label above a text input field, an 'Email' label above a text input field, a 'Password' label above a text input field, a 'Confirm Password' label above a text input field, a link for 'Already registered?', and a dark blue 'REGISTER' button.

Name

Email

Password

Confirm Password

[Already registered?](#) **REGISTER**

A multi-level menu has also been implemented, which also includes the address of the current website of the selected non-profit organization.

[Taxdesignation](#) [research project](#) [team](#) [regional synthesis report](#) [Contact](#)

Other ▾

taxdesignation.org/

Register

Login

tax des

signation

research project

Research commissioned by ERSTE Foundation to assess the impact of the percentage tax designation on civil society based on experience from Hungary, Lithuania, Poland, Slovakia and Romania.

Research project
Assessment of the impact of the percentage tax designation concept and systems with recommendations for future policies

The percentage tax designation concept (sometimes referred to as tax assignment or percentage tax giving) emerged in the policy debate in the late 1990s and materialised first in Hungary (1996), later in Slovakia, Romania, Lithuania and Poland, as an innovative mechanism that channelled public funds in a decentralized way to public benefit purposes. Today, in each country it has a slightly different form, but the core principle is the same: it grants a right to a taxpayer to designate 1% or 2% of paid income tax to a non-profit, non-governmental organization or other type of public benefit entity or to a church.

Contact menu:

[Taxdesignation](#) [research project](#) [team](#) [regional synthesis report](#) [Contact](#)

Other ▾

Register

Login

Contact

Name:

Name

Phone:

Phone number

E-mail:

E-mail

Message:

Message

Send

If the data processing was successful and the provided data meet the required criteria, we will be notified in the following form:

Contact

Thanks for contacting us, we will reply soon!

Name:

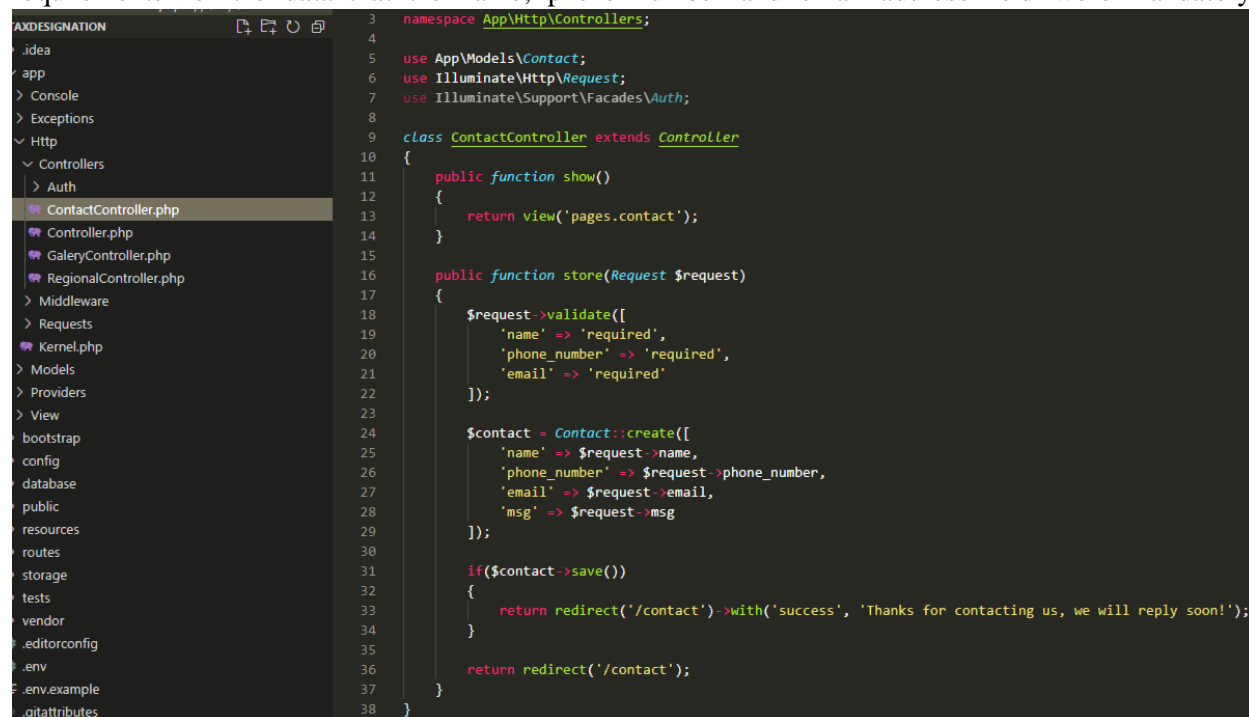
Phone:

E-mail:

Message:

Send

Validation was implemented on the server side in the ContactController. There were only so many requirements for the data that the name, phone number and email address field were mandatory.



```

1 namespace App\Http\Controllers;
2
3 use App\Models\Contact;
4 use Illuminate\Http\Request;
5 use Illuminate\Support\Facades\Auth;
6
7 class ContactController extends Controller
8 {
9     public function show()
10     {
11         return view('pages.contact');
12     }
13
14     public function store(Request $request)
15     {
16         $request->validate([
17             'name' => 'required',
18             'phone_number' => 'required',
19             'email' => 'required'
20         ]);
21
22         $contact = Contact::create([
23             'name' => $request->name,
24             'phone_number' => $request->phone_number,
25             'email' => $request->email,
26             'msg' => $request->msg
27         ]);
28
29         if($contact->save())
30         {
31             return redirect('/contact')->with('success', 'Thanks for contacting us, we will reply soon!');
32         }
33
34         return redirect('/contact');
35     }
36 }

```

If the data does not meet the specified conditions, you will receive an error message.

The phone number field is required.

The Contact and Gallery and User models have also been created. A separate migration was also created for each database table. Example of one migration (contact):

```

C:\xampp\htdocs\          \taxdesignation>php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (170.59ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (287.05ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (198.76ms)
Migrating: 2021_11_08_093609_galery_migration
Migrated: 2021_11_08_093609_galery_migration (61.93ms)
Migrating: 2021_11_08_094914_contact_migration
Migrated: 2021_11_08_094914_contact_migration (55.65ms)
Migrating: 2021_11_08_143217_create_permission_tables
Migrated: 2021_11_08_143217_create_permission_tables (2,064.46ms)

```

```
<?php
```

```

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

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

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

```

Seeders have been created for several tables, an example of one seeder (UserSeeder):

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use DB;

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert([
            'id' => 1,
            'name' => 'Donát Fehér',
            'email' => 'donat.feher@admin.com',
            'password' => bcrypt('admin123')
        ]);

        DB::table('users')->insert([
            'id' => 2,
            'name' => 'Fehér Edvárd',
            'email' => 'feheredvard@gmail.com',
            'password' => bcrypt('user123')
        ]);
    }
}
```

```
C:\xampp\htdocs>php artisan db:seed
Seeding: Database\Seeders\RoleSeeder
Seeded: Database\Seeders\RoleSeeder (194.73ms)
Seeding: Database\Seeders\UserSeeder
Seeded: Database\Seeders\UserSeeder (331.12ms)
Database seeding completed successfully.
```

Due to the success of the migration and seeding, the tables are created and the data was loaded into it:

Tabla	Művelet	Sorok	Tipus	Illesztés	Méret	Felülírás
<input type="checkbox"/> contact	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> failed_jobs	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> galery	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> migrations	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	6	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
<input type="checkbox"/> model_has_permissions	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> model_has_roles	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	2	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> password_resets	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> permissions	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> roles	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	2	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> role_has_permissions	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
<input type="checkbox"/> users	★ Tartalom Szerkesztés Keresés Beszúrás Kiürítés Eldobás	2	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
11 tábla	Összesen	12	InnoDB	utf8mb4_general_ci	384.0 KB	0 B

Seeded Data:

✓ Sorok megjelenítése 0- 1 (összesen 2, A lekérdezés 0,0016 másodpercig tartott.)

```
SELECT * FROM `users`
```

☐ Adatgyűjtés [Szerkesztés helyben] [Módosítás] [Az SQL magyarázata] [PHP-kód létrehozása] [Frissítés]

☐ Összes megjelenítése | Sorok száma: 25 | Sorok szűrése: Keresés a táblában | Sort by key: Nincs

Beállítások

	id	name	email	email_verified_at	password	remember_token
<input type="checkbox"/> Módosítás Másolás Törés	1	Donát Fehér	donat.fehér@admin.com	NULL	\$2y\$10\$VJitUATgNrdYnmEA0115OEhSJ7EMTS/PEZweNgb2...	NULL
<input type="checkbox"/> Módosítás Másolás Törés	2	Fehér Edvárd	feheredvard@gmail.com	NULL	\$2y\$10\$wGRSAL5q6AII0aXNbYyo4epHVvl/qQBfCuKuWCH0KiY...	NULL

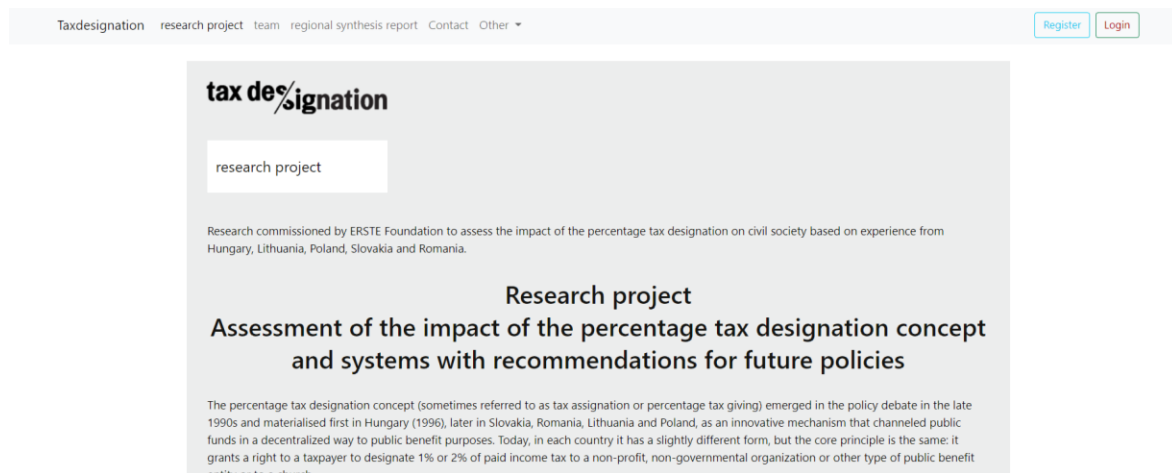
↑ ☐ Összes bejelölése A kijelöltekkel végzendő művelet: ☐ Módosítás ☐ Másolás ☐ Törés ☐ Exportálás

I used the latest bootstrap 5 currently available everywhere to implement the frontend. The website is fully responsive.

Phone:



PC:



Which, unfortunately, have not been implemented:

Roles, photo gallery and the saved messages (under contact menu) will not be displayed in the admin interface. Although these were prepared, but due to lack of time and knowledge, I was unable to complete these tasks.

Github link: <https://github.com/Dodni/Laravel-online-class-assignment>