

Dokumentace

Webové aplikace

Instalace

1. Stáhnout projekt z GitHub do složky *xampp/htdocs*
2. Zapnout si server v terminálu VS Code (`php artisan serve`)
3. Do prohlížeče napsat url *localhost:8000/skoly*
4. Možná budete potřebovat nainstalovat composer (`composer install`)
5. Také bude možná potřebovat vygenerovat APP_KEY (`php artisan key:generate`)
6. Projekt by měl fungovat

Úvod

Aplikace je naprogramována v jazyce PHP ve frameworku **Laravel**. Použité byl použit template **Blade**, **Bootstrap**, **FontAwesome** a Admin panel **Voyager**.

Controller

V aplikaci se nachází celkem 5 controllerů pro každou položku v navbaru.

dokumentaceController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class dokumentaceController extends Controller
8  {
9      public function index()
10     {
11         return view('download.download');
12     }
13 }
```

V této metodě index jsem načetl view download ze složky download.

lokaceController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Skola;
7
8  class lokaceController extends Controller
9  {
10     public function index()
11     {
12         $lokace = Skola::join('mesto', 'skola.mesto', '=', 'mesto.id_mesto')
13             ->join('pocet_prijatych', 'pocet_prijatych.skola', '=', 'skola.id_skola')
14             ->select('skola.*', 'mesto.nazev_mesta', 'pocet_prijatych.pocet')
15             ->get();
16
17         return view('lokace.lokace', compact('lokace'));
18     }
19
20 }
```

Zde jsem načetl model *Skola* (use App\Skola). V metodě index jsem načetl cizí klíče pomocí join a následně načetl všechny položky z tabulky *skola*, nazev

města z tabulky *mesto*, počet přijatých studentů z tabulky *pocet*. Dal jsem proměnnou *lokace* do view *lokace* ve složce *lokace*.

mestaController

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Mesto;
7
8  class mestaController extends Controller
9  {
10     public function index()
11     {
12         $mesta = Mesto::paginate(15);
13         return view('mesta.mestaVypis', compact('mesta'));
14     }
15 }
```

Načtení modelu *Mesto* (use App\Mesto). V metodě *index* jsem načel všechny z tabulky *mesto* a udělal paginaci po 15-ti položkách na stránku. Proměnnou *mesta* jsem dal do view *mestaVypis* ve složce *mesta*.

prijatiStudenti

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\PocetPrijatych;
7  use App\Obor;
8  use App\Skola;
9
```

V úvodu jsem si načel modely *PocetPrijatych*, *Obor*, *Skola*.

```

10 class prijatiStudenti extends Controller
11 {
12     public function index()
13     {
14         $pocet = PocetPrijatych::join('obor', 'pocet_prijatych.obor', '=', 'obor.id_obor')
15             ->join('skola', 'pocet_prijatych.skola', '=', 'skola.id_skola')
16             ->select('pocet_prijatych.*', 'obor.nazev_oboru', 'skola.nazev_skoly')
17             ->orderBy('nazev_skoly', 'asc')
18             ->paginate(15);
19
20         $obor = Obor::get();
21
22         $skoly = Skola::orderBy('nazev_skoly', 'asc')
23             ->get();
24
25         return view('pocetPrijatych.pocetPrijatych', compact('pocet', 'obor', 'skoly'));
26     }
27 }
28

```

V proměnné *pocet* jsem načetl potřebné cizí klíče. Vytvořil paginaci po 15-ti položkách na stránku a zároveň všechny data načetl. V proměnné *obor* jsem načetl všechny data. V proměnné *skoly* načet všechny školy a seřadil podle názvu školy. Všechny proměnné jsem dal do view *pocetPrijatych* ve složce *pocetPrijatych*.

skolyController

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Skola;
7 use App\Mesto;
8
9 class skolyController extends Controller
10 {
11     public function index()
12     {
13         $skolaPag = Skola::join('mesto', 'skola.mesto', '=', 'mesto.id_mesto')
14             ->select('skola.*', 'mesto.nazev_mesta')
15             ->orderBy('skola.nazev_skoly', 'asc')
16             ->paginate(15);
17
18         $skola = Skola::orderBy('nazev_skoly', 'asc')
19             ->get();
20
21         $mesto = Mesto::orderBy('nazev_mesta', 'asc')
22             ->get();
23
24
25         return view('skoly.skolyVypis', compact('skolaPag', 'skola', 'mesto'));
26     }
27 }
28
29

```

V tomto controlleru je vše stejné jako v předchozím.

Model

Všechny modely jsou stejné a slouží pouze pro načtení dat z databáze pro konkrétní tabulku a pro nastavení jiného id, které jsem potřeboval přejmenovat z důvodu administrace.

```
7 class Mesto extends Model
8 {
9     /**
10      * The table associated with the model.
11      *
12      * @var string
13      */
14     protected $table = 'mesto';
15     protected $primaryKey = 'id_mesto';
16
17     /**
18      * Indicates if the model should be timestamped.
19      *
20      * @var bool
21      */
22     public $timestamps = false;
23 }
24
```

View

V aplikaci se nachází 7 souborů ve složce view.

app.blade.php

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6
7     <title>Laravel</title>
8
9     <!-- Fonts -->
10    <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">
11    <!-- Bootstrap -->
12    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0h"
13    <!-- Icons -->
14    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.14.0/css/all.css">
15    <!-- CSS -->
16    <link rel="stylesheet" type="text/css" href="{{ asset('css/style.css') }}" />
17
18  </head>
19  <body>
20    @include('templates.header')
21    <div class="container">
22      @yield('content')
23    </div>
24  </body>
25  <footer>
26    <script src="https://cdnjs.cloudflare.com/ajax/libs/gmaps.js/0.4.24/gmaps.js"></script>
27    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DFdz2htPH01sSSs5nCTpuj/zy4C+0GpamoFVy38MVBnE+IbbVYUew+0rCXaRkfj" crossorigin="anonymou
28    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKAiZap3H661Z81PoY1FhbGU+6BZp6G7niu735Sk71N" c
29    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-B4gt1jrGC7Jh4AgTPSdUt0Bvf08shuf57BaghqFfPlYxofvL8/KUEfY1JOMMV+rV"
30  </footer>
31 </html>
32
```

V tomto view jsem si nalinkoval všechny potřebné věci (bootstrap, style.css, fontawesome). Dále jsem zde vytvořil @include pro přidání navbaru do všech částí aplikace které budou začínat @extends('layout.app') a @section('content'). Končit musí @endsection.

download.blade.php

```
1 @extends('layout.app')
2
3 @section('content')
4
5 <a type="button" class="download btn btn-primary" href="/dokumentace/download">Stáhnout dokumentaci</a>
6
7 @endsection
```

Zde je pouze tlačítko na stažení dokumentace.

lokace.blade.php

```
9 <script>
10 function initMap() {
11
12     var options = {
13         zoom: 10,
14         center: {lat:49.005500, lng:17.173950}
15     }
16
17     var map = new
18     google.maps.Map(document.getElementById('map'), options);
19
20     @foreach($lokace as $skola)
21
22     addMarker({
23         coords: {
24             lat: {{ $skola->geo_lat}},
25             lng: {{ $skola->geo_long}}
26         },
27         content: "<div><h4>{{ trim($skola->nazev_skoly) }}</h4>" +
28                 "<strong>Město: </strong>{{ $skola->nazev_mesta }}</div>" +
29                 "<strong>Počet přijatých: </strong>{{ $skola->pocet }}</div>"
30     });
31
32     @endforeach
33
34     function addMarker(props)
35     {
36         var marker = new google.maps.Marker({
37             position: props.coords,
38             map: map
39         });
40
```

```
41         if(props.content)
42         {
43             var infoWindow = new google.maps.InfoWindow({
44                 content: props.content
45             });
46
47             marker.addListener('click', function(){
48                 infoWindow.close();
49                 infoWindow.open(map, marker);
50             });
51         }
52     }
53 }
54
55 </script>
56
57 <script async defer
58     src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAwBQy3UJEG-GeCe0aVkj90zb5joSGR2o&callback=initMap">
59 </script>
60 @endsection
```

Zde se nachází script na vytvoření mapy, přidání markerů s obsahem načteným z databáze.

mestaVypis.blade.php

```
5 <table class="table">
6   <thead>
7     <tr>
8       <th>Název města</th>
9     </tr>
10  </thead>
11  @foreach ($mesta as $m)
12    <tbody>
13      <tr>
14        <td>{{ $m->nazev_mesta }}</td>
15      </tr>
16    </tbody>
17  @endforeach
18 </table>
19
20
21 {{ $mesta->links() }}
```

V tomto view je pouze tabulka s daty načtenými z databáze. Jako poslední je zde metoda, která přidá tlačítka na přepínání stran (paginace).

pocetPrijetych.blade.php

```
5 <table class="table">
6   <thead>
7     <tr>
8       <th>Obor</th>
9       <th>Škola</th>
10      <th>Počet</th>
11      <th>Rok</th>
12    </tr>
13  </thead>
14  @foreach ($pocet as $p)
15    <tbody>
16      <tr>
17        <td>{{ $p->nazev_oboru }}</td>
18        <td>{{ $p->nazev_skoly }}</td>
19        <td>{{ $p->pocet }}</td>
20        <td>{{ $p->rok }}</td>
21      </tr>
22    </tbody>
23  @endforeach
24 </table>
25
26 {{ $pocet->links() }}
```

V tomto view jsem udělal to stejné jako v předchozím view.

skolyVypis.blade.php

```
5 <table class="table">
6     <thead>
7         <tr>
8             <th>Název školy</th>
9             <th>Město</th>
10        </tr>
11    </thead>
12    @foreach ($skolaPag as $s)
13    <tbody>
14        <tr>
15            <td>{{ $s->nazev_skoly }}</td>
16            <td>{{ $s->nazev_mesta }}</td>
17        </tr>
18    </tbody>
19    @endforeach
20 </table>
21
22 {{ $skolaPag->links() }}
```

header.blade.php

```
1 <nav class="navbar navbar-expand-lg navbar-light bg-light">
2     <div class="container">
3         <a class="navbar-brand" href="/">Lokace škol</a>
4         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">
5             <span class="navbar-toggler-icon"></span>
6         </button>
7         <div class="collapse navbar-collapse" id="navbarNavDropdown">
8
9             <ul class="navbar-nav">
10                <li class="nav-item">
11                    <a class="nav-link" href="/skoly">Výpis škol</a>
12                </li>
13                <li class="nav-item">
14                    <a class="nav-link" href="/mesta">Výpis měst</a>
15                </li>
16                <li class="nav-item">
17                    <a class="nav-link" href="/pocetPrijetych">Počet přijatých</a>
18                </li>
19                <li class="nav-item">
20                    <a class="nav-link" href="/dokumentace">Stahování dokumentace</a>
21                </li>
22            </ul>
23
24            <ul class="navbar-nav ml-auto">
25                <li class="nav-item">
26                    <a class="nav-link" href="/admin">Přihlášení do administrace</a>
27                </li>
28            </ul>
29        </div>
30    </nav>
```

Zde je view, který obsahuje navbar celé aplikace.

Routes

Routes jsou v frameworku Laravel v souboru web.php.

```
16 Route::get('/skoly', 'skolyController@index');
17
18 Route::get('/mesta', 'mestaController@index');
19
20 Route::get('/pocetPrijatych', 'prijatiStudenti@index');
21
22 Route::get('/', 'lokaceController@index');
23
24 Route::get('/dokumentace', 'dokumentaceController@index');
25
```

První část souboru obsahu je jednoduché routy na načtení metody z controlleru a to co uživatel napíše za base url do prohlížeče.

```
26 Route::get('/dokumentace/download', function(){
27     $file = public_path()."/dokumentace.pdf";
28
29     $headers = array(
30         'Content-type: application/pdf',
31     );
32
33     return Response::download($file, "Dokumentace.pdf", $headers);
34 });
35
36
37 Route::group(['prefix' => 'admin'], function () {
38     Voyager::routes();
39 });
```

V druhé části je routa, která slouží na stažení souboru dokumentace ze složky *public*. Poté je zde routa sloužící Admin panelu Voyager.