

---

## 4. Laboratorinis darbas. Duomenų įvedimo formų kūrimas

---

Laboratorinis darbas skirtas sukurti duomenų įvedimo formos vaizdą, valdiklio bei modelio funkcijas, skirtas patikrinti bei įrašyti vartotojo įvestus duomenis į duomenų bazę. Darbas atliekamas iki semestro pabaigos.

### I. Darbo užduotis

Sukurti formas visų duomenų įvedimui, redagavimui. Duomenų įvedimo formos turėtų būti validuojamos pasinaudojant karkaso funkcionalumu. Sukurti galimybę ištrinti duomenis.

### II. Formos atvaizdavimas HTML

Vartotojo įvestų duomenų siuntimui į serverį naudojamas *HTML form* elementas. Esminiai šio elemento atributai yra *action*, kuris skirtas nurodyti puslapio, kuriam siunčiami duomenys adresą bei *method*, kuriame nustatomas duomenų siuntimo metodas (*get* ar *post*).

*HTML* įvedimo formose naudojami įvairių tipų laukai. Pagrindinis *input* elementas gali būti įvairių tipų:

- *text* – tekstinis laukas;
- *select* – pasirinko laukas;
- *password* – slaptažodžio įvedimo laukas;
- *radio* – vieno elemento iš grupės pasirinkimo laukas;
- *checkbox* – keleto elementų iš grupės pasirinkimo laukas;
- *submit* – formos pateikimo mygtukas;
- *hidden* – paslėptas laukas, skirtas perduoti vartotojui nematomus parametrus.

Formose taip pat naudojami *select* (elemento pasirinkimas iš sąrašo), *textarea* (tekstinis daugelio eilučių laukas) ir kiti elementai.

Aprašant formas svarbu nepamiršti nurodyti *name* atributą – tik tada laukų reikšmės būna prieinamos *PHP*. Daugiau apie *HTML* įvedimo formas galite paskaityti šiame puslapyje:

[http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)

### III. Laravel įvedamų duomenų tikrinimas

Įvedamų duomenų tikrinimo taisyklės *Laravel* karkase gali būti aprašomos valdiklyje, bei kitais sudėtingesniais būdais kurių neaptarsime šio laboratorinio darbo metu. Duomenų teisingumo tikrinimą vykdo *Laravel Validator* biblioteka. Validatorius – tai atskira klasė su savo funkcijomis, kurioje aprašymas atitinkamas duomenų tikrinimas. Jų yra įvairių, pvz: el. pašto adreso tikrinimui, *IP* adreso tikrinimui ir kiti.

Daugiau apie Laravel validatorius ir duomenų užklausas galima rasti čia:

<https://laravel.com/docs/master/requests>

<https://laravel.com/docs/master/validation>

### IV. Darbo eiga

#### 1. Duomenų vaizdavimas svetainėje

##### 1.1. Šablono papildymas klaidų atvaizdavimu

Kadangi svetainėje gali pasitaikyti daug formų ir norime turėti standartinį klaidų atvaizdavimo būdą. Papildome pagrindinę svetainės šabloną klaidų ir sėkmės žinučių pranešimo blokais.

```
<div class="col-12 col-md-9">
    @if (\Session::has('success'))
        <div class="alert alert-success">
            <p>{!! \Session::get('success') !!}</p>
        </div>
    @endif
    @if (count($errors) > 0)
        <div class="alert alert-danger">
            <p>jūsų įvedamuose duomenyse yra klaidų:</p>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    @yield('content')
</div>
```

### 1.1. Formos sukūrimas

Sukuriame HTML tipo formą su tiek įvesties vietų, kiek turime lentelėje stulpelių. Pavyzdinės formos HTML kodas pateikiamas apačioje. Atkreipkite dėmesį, kad pavyzdyje naudojami trys įvedimo būdai, input, text, bei select. Jeigu kurio nors iš šių tipų nenaudojate pakoreguokite savo šabloną atitinkamai.

```
@extends('layouts.app')
@section('content')
    <h1 class="align">Adding User </h1>
    <div class="row">
        <div class="col-md-12 col-md-offset-3">
            <div class="panel panel-default">
                <div class="panel-body">
                    <form class="form-horizontal" role="form" method="POST"
action="{{ url('newsManagement/insertNews/') }}">
                        <input type="hidden" name="_token" value="{{ csrf_token()
}}">

                        <div class="form-group">
                            <label class="col-md-4 control-label">Pavadinimas</label>
                            <div class="col-md-12">
                                <input type="text" class="form-control" name="name"
value="">

                                </div>
                            </div>
                        <div class="form-group">
                            <label class="col-md-4 control-label">Aprašymas</label>
                            <div class="col-md-12">
                                <input type="text" class="form-control"
name="description" value="">

                                </div>
                            </div>
                        <div class="form-group">
                            <label class="col-md-4 control-label">Turinys</label>
                            <div class="col-md-12">
                                <textarea type="text" class="form-control"
name="content"></textarea>

                                </div>
                            </div>
                        <div class="form-group">
                            <label class="control-label">Tipas</label>
                            <div class="col-md-12">
                                <select class="form-control" name="newsType">
                                    @foreach($newsType as $nt)
                                        <option value="{{ $nt->getKey() }}">{{ $nt-
>pavadinimas }}</option>
```

```

                                @endforeach
                                </select>
                                </div>
                            </div>
                            <div class="form-group">
                                <div class="col-md-12 col-md-offset-4">
                                    <button type="submit" class="btn btn-primary">
                                        <span class="glyphicon glyphicon-refresh"></span>
                                        Add
                                    </button>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    @endsection

```

Duomenų redagavimo forma nuo duomenų įvedimo formos skiriasi tik tuo, kad įvedimo laukai užpildomi jau įvestomis reikšmėmis, bei nurodoma kitas formos veiksmas. Duomenų atvaizdavimo įvedimo laukuose pavyzdys pateikiamas apačioje.

```

<div class="form-group">
<label class="col-md-4 control-label">Pavadinimas</label>
    <div class="col-md-12">
        <input type="text" name="pavadinimas" value="{ { $selectedNews->pavadinimas } }">
    </div>
</div>

```

### 1.1. Visų naujienų atvaizdavimo vaizdas

Visų naujienų atvaizdavimo vaizdą papildome trimis naujais mygtukais duomenų pridėjimo, redagavimo ir šalinimo. Papildytas naujienų atvaizdavimo failas galėtų atrodyti taip:

```
@extends('layouts.app')

@section('content')
    <div class="row">
        <h3 class="font-italic">
            Naujienos
        </h3>
    </div>
    <div class="row">
        @foreach($allNews as $newsData )
            <div class="blog-post">
                <h2 class="blog-post-title"> {{ $newsData->pavadinimas }} </h2>
                <p class="blog-post-meta">{{ $newsData->sukurimo_data }} by <a href="#"> Jacob</a></p>
                <h4 class="font-italic">
                    Santrauka
                </h4>
                <div><p>{{ $newsData->santrauka }} </p></div>
                <h4 class="font-italic">
                    Turinys
                </h4>
                <p>{{ $newsData->turinys }} </p>
                {{ $allNews->links() }}

                <a class="btn btn-primary" href="{{route('editUserData',$user->id)}}">
                    <span>Pridėti</span>
                </a>

                <a class="btn btn-primary" href="{{ route('editNews', $newsData->getKey()) }}">
                    <span>Redaguoti</span>
                </a>

                <a class="btn btn-danger" onclick="javascript:return confirm('Do you really want to delete this?')" href="{{route('deleteNews',$newsData->getKey())}} ">
                    <span>Pašalinti</span>
                </a>
            </div><!-- /.blog-post -->
        @endforeach
    </div><!--/row-->
@endsection
```

## 1.2. Web.php kelių failo papildymas

Web.php faile papildome kelius sukuriant GET, bei POST HTTP užklausas. GET tipo užklausa naudosime duomenų atvaizdavimui ir duomenų šalinimui. POST tipo užklausa naudosime didesnio kiekio duomenų perdavimui saugojimui / atnaujinimui. Pavyzdinis Web.php failas pateikiamas apačioje. Šiame pavyzdyje naudosime du skirtingus vaizdinių atvaizduoti naujienoms, kurios turi verslo ir mokslo tipus. Bei naujienų valdymo keliai, skirti pridėti ( *addNews* , *insertNews* ), atnaujinti ( *edit* , *confirmEdit* ) bei trinti ( *delete* )

```
<?php

/*News routes*/

Route::get('/allNews', 'news\PublicNewsController@index')->name('PublicNews');

Route::get('/business', 'news\BusinessController@index')->name('business');

Route::get('/science', 'news\ScienceController@index')->name('science');

/*News management routes*/

Route::get('/newsManagement/addNews', 'news\NewsController@index')->name('addNews');

Route::post('/newsManagement/insertNews', 'news\NewsController@insertNews')->name('insertNews');

Route::get('/newsManagement/delete/{id}', 'news\NewsController@deleteNews')->name('deleteNews');

Route::get('/newsManagement/edit/{id}', 'news\NewsController@editNews')->name('editNews');

Route::post('/newsManagement/confirmEdit/{id}', 'news\NewsController@confirmEditedNews')->name('confirmEditedNews');

/* End of news management routes*/
/* End of news routes*/
```

### 1.3. Valdiklio kūrimas/papildymas

Valdiklyje sukuriame funkcijas duomenų įterpimo formos atvaizdavimui ( *index* ). Šioje funkcijoje kategorijos naudojamos kai atvaizduojama naujienos įvedimo forma, kurioje yra kategorijos pasirinkimo forma.

Duomenų įvedimui (*insertNews*) į šią funkciją duomenys iš HTML formos perduodamos pasinaudojant Laravel Request. Švalidator kintamasis užtikrina, kad įvedami duomenys atitiktų mūsų nurodytas taisykles. Jei bent viena taisyklė netenkinama gražiname naudotoją atgal į puslapį su klaidos pranešimu. Jei tikrinimo taisyklės tenkinamos sukuriamas naujas naujienos įrašas duomenų bazės lentelėje ir grįžtame į naujienos pridėjimo puslapį su pranešimu apie sėkmingai pridėtą naujieną.

*app/controllers/NewsController.php*

```
<?php

namespace App\Http\Controllers\News;

use App\News;
use App\NewsCategory;
use Carbon\Carbon;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Validator;

class NewsController extends Controller
{
    public function Index()
    {
        $newsType = NewsCategory::all();
        return view('insertNews', compact('newsType'));
    }

    public function insertNews(Request $request)
    {
        $validator = Validator::make(
            [
                'name' => $request->input('name'),
                'description' => $request->input('description'),
                'content' => $request->input('content'),
                'newsType' => $request->input('newsType')
            ],
            [
                'name' => 'required|alpha_num',
                'description' => 'required|alpha_num',
                'content' => 'required|min:3|max:150',
                'newsType' => 'required'
            ]
        );

        if ($validator->fails())
        {
            return Redirect::back()->withErrors($validator);
        }
        else
        {
            $news = new News();
            $news->pavadinimas = $request->input('name');
            $news->santrauka = $request->input('description');
            $news->turinys = $request->input('content');
            $news->sukurimo_data = Carbon::now();
        }
    }
}
```

```
$news->naujienos_kategorija_id= $request->input('newsType');

$news->save();
}
return Redirect::to('/newsManagement/addNews')->with('success', 'Naujiena
pridėta');
}

public function confirmEditedNews(Request $request, $id)
{
    $validator = Validator::make(
        [
            'name' =>$request->input('pavadinimas'),
            'description' =>$request->input('santrauka'),
            'content' =>$request->input('turinys'),
            'newsType' => $request->input('naujienos_kategorija_id')
        ],
        [
            'name' => 'required|alpha_num',
            'description' => 'required|alpha_num',
            'content' =>'required|min:3|max:150',
            'newsType' => 'required'
        ]
    );

    if ($validator->fails())
    {
        return Redirect::back()->withErrors($validator);
    }
    else
    {
        $appendableData = $request->all();

        //pašaliname iš masyvo saugumui naudojamą _token kintamąjį
        unset($appendableData['_token']);

        $news = News::where('id','=',$id)->update($appendableData);
    }
    return redirect()->back()->with('success', 'Naujiena atnaujinta');
}

public function deleteNews($id)
{
    News::where('id','=',$id)->delete();
    $allNews= News::paginate(1);
    return Redirect::to('/allNews ')->with('success', 'Naujiena pašalinta');
}

public function editNews($id)
{
    $newsType = NewsCategory::all();
    $selectedNews = News::where('id','=',$id)->first();
    return view('editNews', compact('selectedNews', 'newsType'));
}
}
```

Pavyzdyje realizuotas duomenų įvedimas pirmojo laboratorinio darbo duomenų bazės lentelei „naujiena“. Visi laukai privalomi, pavadinimas ir santrauka gali būti sudaryta tik iš skaičių ir skaitmenų, turinys turi būti ilgesnis nei 3 simboliai bet mažesnis nei 150. Naujienos sėkmingo ištrynimo metu nukreipiama į visą naujienų sąrašą, sėkmės pranešimu. Atkreipkite dėmesį kad pavyzdyje pateikiami du būdai kaip galima išsaugoti duomenis iš užklauso (*Request*). Vienas būdas pavaizduotas *InsertNews* funkcijoje, kitas *confirmEditedNews*. savod arba rekomenduojama išsirinkti vieną ir naudoti vienodą duomenų išsaugojimo stilių.