

[www.mladensblogger.rs](http://www.mladensblogger.rs)

---

Dokumentacija

---

Mladen Karić 32/15

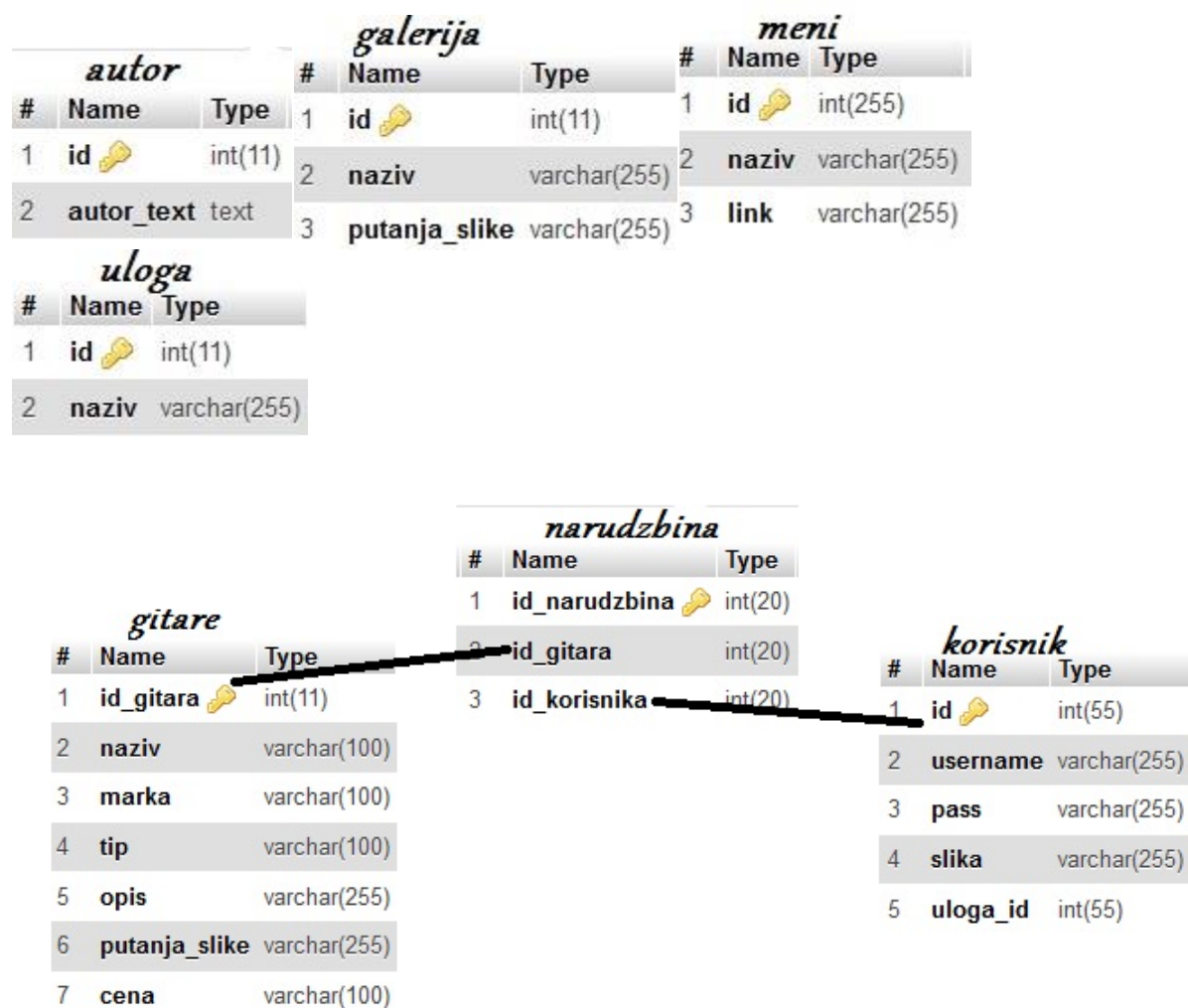
---

## Contents

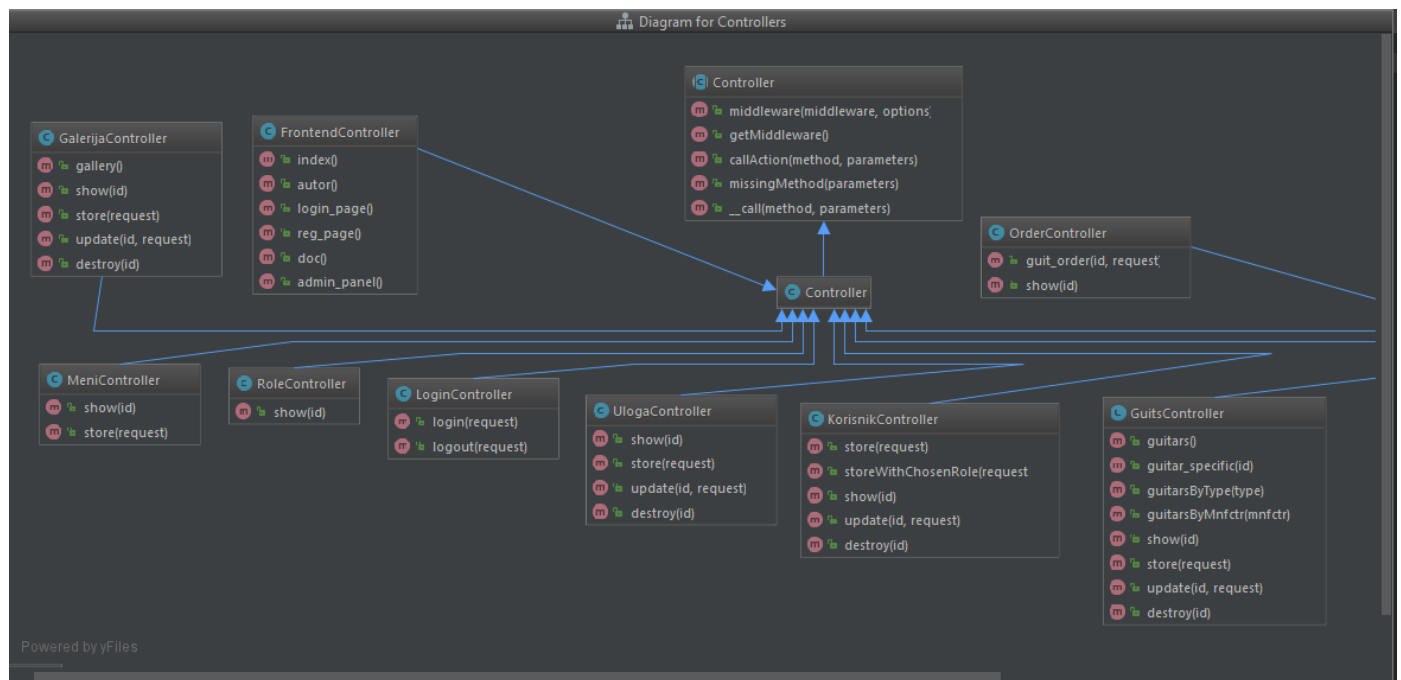
Dizajn baze .....	4
Kontroleri .....	4
FrontendController .....	5
GalerijaController.....	6
GuitsController.....	8
KorisnikController .....	11
LoginController .....	14
MeniController.....	15
OrderController.....	16
RoleController .....	16
UlogaController.....	17
Midlver .....	18
CheckIfLogged .....	19
CheckUserRole .....	19
Modeli .....	20
Autors.....	20
Galerija .....	21
Guit.....	22
Korisnik.....	24
Meni .....	25
Order .....	27
Role .....	28
Uloga .....	28
Views.....	29
Components.....	30
Footer.blade.....	30
Nav.blade .....	31
Sidebar.blade .....	32
Layouts.....	33
Front.blade.....	33
Pages .....	35

Admin_panel.blade .....	35
AdminGalerija.blade .....	35
AdminGuit.blade .....	36
AdminKorisnik.blade .....	38
AdminMenu.blade .....	40
AdminOrder.blade .....	40
AdminRole.blade.....	41
Gallery.blade .....	42
Guits.blade .....	43
Guits_show.blade .....	43
Home.blade.....	44
Login.blade.....	44
Mine.blade .....	45
Reg.blade .....	46
Rute.....	46
Web.....	46

## Dizajn baze



## Kontroleri



## FrontendController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Meni;
use App\Models\Guit;
use App\Models\Autor;
use App\Http\Controllers\Response;

class FrontendController extends Controller
{
    private $data = [];

    public function __construct() {
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();
    }

    public function index() {
        return view('pages.home', $this->data);
    }

    public function autor() {
        $autor = new Autor();
        $this->data['mine'] = $autor->getAll();
        return view('pages.mine', $this->data);
    }

    public function login_page() {
        return view('pages.login', $this->data);
    }
}
  
```

```

    public function reg_page(){
        return view('pages.reg', $this->data);
    }

    public function doc(){
        $file = base_path() . '/' . 'public/pics/doc.pdf';
        return response()->download($file);
    }

    public function admin_panel(){
        return view('pages.admin_panel', $this->data);
    }
}

```

## GalerijaController

```

<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Meni;
use App\Models\Guit;
use App\Models\Autor;
use App\Models\Galerija;
use Illuminate\Support\Facades\File;

class GalerijaController extends Controller
{
    private $data = [];

    public function __construct(){
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();
    }

    public function gallery(){
        $gall = new Galerija();
        $this->data['galls'] = $gall->getAll();
        return view('pages.gallery', $this->data);
    }

    public function show($id = null){
        $galerija = new Galerija();
        $this->data['galerije'] = $galerija->getAllnoPaginate();

        if(!empty($id)){
            $galerija->id = $id;
            $this->data['galerija'] = $galerija->get($galerija);
        }

        return view('pages.adminGalerija', $this->data);
    }

    public function store(Request $request){
        $this->validate($request, [
            'tbNaziv' => 'unique:galerija,naziv',
            'tbPic' => 'required|mimes:jpg,jpeg,png,gif'
        ]);
    }
}

```

```

        $naziv = $request->get('tbNaziv');

        $pic = $request->file('tbPic');

        $temp_path = $pic->getPathname();
        $ext = $pic->getClientOriginalExtension();
        $picName = time() . '.' . $ext;
        $path = 'pics/gallery/' . $picName;
        $server_path = public_path($path);

        try{
            File::move($temp_path, $server_path);
            $galerija = new Galerija();
            $galerija->naziv = $naziv;
            $galerija->putanja_slike = $path;

            $result = $galerija->save();

            if($result == 1){
                return redirect()->route('home')->with('success', 'Uspesan unos');}
            else{
                return redirect()->route('home')->with('error', 'Neuspesan unos');}
        }
        catch (\Exception $exception){
            \Log::error('Greska: ' . $exception->getMessage());
        }
    }

    public function update($id, Request $request) {

        $naziv = $request->get('tbNaziv');

        $pic = $request->file('tbPic');

        $galerija = new Galerija();
        $galerija->id = $id;
        $galerija->naziv = $naziv;

        if(!empty($pic)){

            $gall_to_update = $galerija->get();
            File::delete($gall_to_update->putanja_slike);

            $temp_path = $pic->getPathname();
            $ext = $pic->getClientOriginalExtension();
            $picName = time() . '.' . $ext;
            $path = 'pics/gallery/' . $picName;
            $server_path = public_path($path);

            File::move($temp_path, $server_path);

            $galerija->putanja_slike = $path;
        }

        $rez = $galerija->update();

        if($rez == 1){ // ako je uspeo update
            return redirect('/galls')->with('message','Uspesan update!');
        }
        else {
            return redirect('/galls')->with('message','Greska pri update-u!');
        }
    }

```

```

    }

    public function destroy($id){
        $galerija = new Galerija();
        $galerija->id = $id;

        // brisanje stare slike sa servera
        $gall_to_update = $galerija->get($id);
        File::delete($gall_to_update->putanja_slike);

        $rez = $galerija->delete();
        if($rez == 1){
            return redirect('/galls')->with('message','Uspesan delete!');
        }
        else {
            return redirect('/gallss')->with('message','Greska pri delete-u!');
        }
    }
}

```

## GuitsController

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/8/2018
 * Time: 1:57 AM
 */

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Korisnik;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\File;
use App\Models\Meni;
use App\Models\Uloga;
use App\Models\Guit;

class GuitsController extends Controller
{
    private $data = [];

    public function __construct(){
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();
    }

    public function guitars(){
        $guits = new Guit();
        $this->data['guits'] = $guits->getAll();
        return view('pages.guits', $this->data);
    }

    public function guitar_specific($id){
        $guits = new Guit();
    }
}

```



```

        $this->data['guit'] = $guits->get($id);
        return view('pages.guits_show', $this->data);
    }

    public function guitarsByType($type){
        $guits = new Guit();
        $this->data['guits'] = $guits->getByType($type);
        return view('pages.guits', $this->data);
    }

    public function guitarsByMnfctr($mnfctr){
        $guits = new Guit();
        $this->data['guits'] = $guits->getByMnfctr($mnfctr);
        return view('pages.guits', $this->data);
    }

    public function show($id = null){
        $guit = new Guit();
        $this->data['guits'] = $guit->getAll();

        if(!empty($id)){
            $guit->id = $id;
            $this->data['guit'] = $guit->gets();
        }

        return view('pages.adminGuit', $this->data);
    }

    public function store(Request $request){
        $this->validate($request, [
            'tbNaziv' => 'required|regex:/^[A-Za-z0-9]+(?:[ _-][A-Za-z0-9]+)*$/ ',
            'tbMarka' => 'required|alpha',
            'tbTip' => 'required|alpha',
            'tbOpis' => 'required|max:400',
            'tbCena' => 'required|alpha_num',
            'tbSlika' => 'required|mimes:jpg,jpeg,png,gif'
        ]);

        $naziv = $request->get('tbNaziv');
        $marka = $request->get('tbMarka');
        $tip = $request->get('tbTip');
        $opis = $request->get('tbOpis');
        $cena = $request->get('tbCena');
        $pic = $request->file('tbSlika');

        $temp_path = $pic->getPathname();
        $ext = $pic->getClientOriginalExtension();
        $picName = time() . '.' . $ext;
        $path = 'pics/guits/' . $picName;
        $server_path = public_path($path);

        try{
            File::move($temp_path, $server_path);
            $guit = new Guit();
            $guit->naziv = $naziv;
            $guit->marka = $marka;
            $guit->tip = $tip;
            $guit->opis = $opis;
            $guit->cena = $cena;
            $guit->putanja_slike = $path;
        }
    }

```

```

        $result = $guit->save();

        if($result == 1){
            return redirect()->route('home')->with('success', 'Uspesan unos');}
        else{
            return redirect()->route('home')->with('error', 'Neuspesan unos');}
    }
    catch (\Exception $exception){
        \Log::error('Greska: ' . $exception->getMessage());
    }
}

public function update($id, Request $request) {
    $naziv = $request->get('tbNaziv');
    $marka = $request->get('tbMarka');
    $tip = $request->get('tbTip');
    $opis = $request->get('tbOpis');
    $cena = $request->get('tbCena');

    $slika = $request->file('tbSlika');

    $guit = new Guit();
    $guit->id = $id;
    $guit->naziv = $naziv;
    $guit->marka = $marka;
    $guit->tip = $tip;
    $guit->opis = $opis;
    $guit->cena = $cena;

    if(!empty($slika)){

        $guit_to_update = $guit->gets();
        File::delete($guit_to_update->putanja_slike);

        $tmp_putanja = $slika->getFileName();
        $ime_fajla = time().'.'.$slika->getClientOriginalExtension();
        $putanja = 'pics/guits/' . $ime_fajla;
        $putanja_server = public_path($putanja);

        File::move($tmp_putanja, $putanja_server);

        $guit->putanja_slike = $putanja;
    }

    $rez = $guit->update();

    if($rez == 1){ // ako je uspeo update
        return redirect('/guits')->with('message','Uspesan update!');
    }
    else {
        return redirect('/guits')->with('message','Greska pri update-u!');
    }
}

public function destroy($id){
    $guit = new Guit();
    $guit->id = $id;

    // brisanje stare slike sa servera
    $guit_to_update = $guit->gets();
    File::delete($guit_to_update->putanja_slike);

    $rez = $guit->delete();
}

```

```

        if($rez == 1){
            return redirect('/guits')->with('message','Uspesan delete!');
        }
        else {
            return redirect('/guits')->with('message','Greska pri delete-u!');
        }
    }
}

```

## KorisnikController

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/3/2018
 * Time: 9:05 PM
 */

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Korisnik;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\File;
use App\Models\Meni;
use App\Models\Uloga;

class KorisnikController extends Controller
{
    private $data = [];

    public function __construct() {
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();
        $uloga = new Uloga();
        $this->data['uloge'] = $uloga->getAll();
    }

    public function store(Request $request) {
        /* $this->validate($request, [
            'tbUsername' => 'unique:korisnik,username',
            'tbPass' => 'required|alpha',
            'tbPic' => 'required|mimes:jpg,jpeg,png,gif'
        ]); */

        $this->validate($request, [
            [
                'tbUsername' => 'regex:/^[a-z]+(\s[\w\d\-\
])+$|/unique:korisnik,username',
                'tbPass' => 'required|alpha_num',
                'tbPic' => 'required|mimes:jpg,jpeg,png,gif',
                'ddlUloga' => 'required|not_in:0'
            ],
            [
                'required' => 'Polje :attribute je obavezno!',
                'tbUsername.regex' => 'Polje username nije u ispravnom formatu!',

```

```

        'mimes' => 'Dozvoljeni formati su: :values.'
    ]
});

$userName = $request->get('tbUsername');
$pass = $request->get('tbPass');
$pic = $request->file('tbPic');

$temp_path = $pic->getPathname();
$ext = $pic->getClientOriginalExtension();
$picName = time() . '.' . $ext;
$path = 'pics/' . $picName;
$server_path = public_path($path);

try{
    File::move($temp_path, $server_path);
    $korisnik = new Korisnik();
    $korisnik->username = $userName;
    $korisnik->pass = $pass;
    $korisnik->slika = $path;

    $result = $korisnik->saveWithRegularUserRole();

    if($result == 1){
        return redirect()->route('home')->with('success', 'Uspesan unos');}
    else{
        return redirect()->route('home')->with('error', 'Neuspesan unos');}
    }
catch (\Exception $exception){
    \Log::error('Greska: ' . $exception->getMessage());
}
}

public function storeWithChosenRole(Request $request){
    $this->validate($request, [
        'tbUsername' => 'unique:korisnik,username',
        'tbPass' => 'required|alpha',
        'tbPic' => 'required|mimes:jpg,jpeg,png,gif',
        'ddlUloga' => 'required|not_in:0'
    ]);

    $userName = $request->get('tbUsername');
    $pass = $request->get('tbPass');
    $pic = $request->file('tbPic');
    $role = $request->get('ddlUloga');

    $temp_path = $pic->getPathname();
    $ext = $pic->getClientOriginalExtension();
    $picName = time() . '.' . $ext;
    $path = 'pics/' . $picName;
    $server_path = public_path($path);

    try{
        File::move($temp_path, $server_path);
        $korisnik = new Korisnik();
        $korisnik->username = $userName;
        $korisnik->pass = $pass;
        $korisnik->slika = $path;
        $korisnik->uloga_id = $role;
        $result = $korisnik->save();

        if($result == 1){
            return redirect()->route('home')->with('success', 'Uspesan unos');}
    }
}

```

```

        else{
            return redirect()->route('home')->with('error', 'Neuspesan unos');}
    }
    catch (\Exception $exception){
        \Log::error('Greska: ' . $exception->getMessage());
    }
}

public function show($id = null){
    $korisnik = new Korisnik();
    $this->data['korisnici'] = $korisnik->getAll();

    if(!empty($id)){
        $korisnik->id = $id;
        $this->data['korisnik'] = $korisnik->get();
    }

    return view('pages.adminKorisnik', $this->data);
}

public function update($id, Request $request) {
    $korisnicko_ime = $request->get('tbUsername');
    $lozinka = $request->get('tbPass');
    $uloga_id = $request->get('ddlUloga');

    $slika = $request->file('tbPic');

    $korisnik = new Korisnik();
    $korisnik->id = $id;
    $korisnik->username = $korisnicko_ime;
    $korisnik->pass = $lozinka;
    $korisnik->uloga_id = $uloga_id;

    if(!empty($slika)){

        $korisnik_to_update = $korisnik->get();
        File::delete($korisnik_to_update->slika);

        $tmp_putanja = $slika->getPathName();
        $ime_fajla = time().'.'.$slika->getClientOriginalExtension();
        $putanja = 'pics/' . $ime_fajla;
        $putanja_server = public_path($putanja);

        File::move($tmp_putanja, $putanja_server);

        $korisnik->slika = $putanja;
    }

    $rez = $korisnik->update();

    if($rez == 1){ // ako je uspeo update
        return redirect('/users')->with('message','Uspesan update!');
    }
    else {
        return redirect('/users')->with('message','Greska pri update-u!');
    }
}

public function destroy($id){
    $korisnik = new Korisnik();
    $korisnik->id = $id;

    $korisnik_to_update = $korisnik->get();

```

```

        File::delete($korisnik_to_update->slika);

        $rez = $korisnik->delete();
        if($rez == 1){
            return redirect('/users')->with('message','Uspesan delete!');
        }
        else {
            return redirect('/users')->with('message','Greska pri delete-u!');
        }
    }
}

```

## LoginController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Korisnik;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
use App\Http\Controllers\Controller;

class LoginController extends Controller
{
    public function login(Request $request){
        $this->validate($request, [
            'tbUsername' => 'required|alpha',
        ]);

        $username = $request->get('tbUsername');
        $pass = $request->get('tbPass');

        $korisnik = new Korisnik();
        $korisnik->username = $username;
        $korisnik->pass = $pass;

        $loginKorisnik = $korisnik->getByUsernameAndPass();

        if(!empty($loginKorisnik)){
            $request->session()->push('user', $loginKorisnik);
            if($loginKorisnik->username == 'adminko'){
                $request->session()->push('isAdmin', true);
            }
            return redirect()->route('home')->with('success', 'Uspesno ste se
ulogovali!');
        }
        return redirect()->back()->with('error', 'Niste reg');
    }

    public function logout(Request $request){
        $request->session()->forget('user');
        $request->session()->forget('isAdmin');
        $request->session()->flush();
        return redirect('/');
    }
}

```

## MeniController

```
<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/9/2018
 * Time: 5:49 PM
 */

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Korisnik;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\File;
use App\Models\Meni;
use App\Models\Uloga;

class MeniController extends Controller
{
    private $data = [];

    public function show($id = null) {
        $menu = new Meni();
        $this->data['menus'] = $menu->getAll();

        if(!empty($id)) {
            $menu->id = $id;
            $this->data['menu'] = $menu->get();
        }

        return view('pages.adminMenu', $this->data);
    }

    public function store(Request $request) {
        $this->validate($request, [
            'tbNaziv' => 'unique:meni,naziv',
            'tbLink' => 'required|alpha'
        ]);

        $naziv = $request->get('tbNaziv');
        $link = $request->get('tbLink');

        /* try{ */
            $menu = new Meni();
            $menu->naziv = $naziv;
            $menu->link = $link;
            $result = $menu->save();

            if($result == 1) {
                return redirect()->route('home')->with('success', 'Uspesan unos');
            } else {
                return redirect()->route('home')->with('error', 'Neuspesan unos');
            }
        /* }
        catch (\Exception $exception) {
            \Log::error('Greska: ' . $exception->getMessage());
        } */
    }
}
```

## OrderController

```
<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/7/2018
 * Time: 10:11 PM
 */

namespace App\Http\Controllers;
use App\Models\Order;
use Illuminate\Http\Request;
use App\Models\Meni;

class OrderController extends Controller
{
    public $data = [];

    public function __construct() {
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();
    }

    public function quit_order($id, Request $request) {
        if(empty(session('user'))){
            return redirect()->route('home')->with('error', 'Morate biti ulogovani da biste narucili gitaru');
        } else {
            $order = new Order();
            $user_session = $request->session()->get('user')[0];
            $order->id_korisnika = $user_session->id;
            $order->id_gitara = $id;
            $order->save();
            return redirect()->route('home')->with('success', 'Uspesno ste narucili gitaru');
        }
    }

    public function show($id = null) {
        $order = new Order();
        $this->data['orders'] = $order->getAll();

        if(!empty($id)) {
            $order->id_narudzbina = $id;
            $this->data['order'] = $order->get();
        }

        return view('pages.adminOrder', $this->data);
    }
}
```

## RoleController

```
<?php
/**
```



```

* Created by PhpStorm.
* User: Mladen
* Date: 5/9/2018
* Time: 7:22 PM
*/

namespace App\Http\Controllers;
use App\Models\Order;
use App\Models\Uloga;
use Illuminate\Http\Request;
use App\Models\Role;
use App\Models\Meni;

class RoleController extends Controller
{
    private $data = [];

    public function show($id = null){
        $role = new Uloga();
        $this->data['roles'] = $role->getAll();
        $meni = new Meni();
        $this->data['menus'] = $meni->getAll();

        if(!empty($id)){
            $role->id = $id;
            $this->data['role'] = $role->get();
        }

        return view('pages.adminRole', $this->data);
    }
}

```

## UlogaController

```

<?php

namespace App\Http\Controllers;
use App\Models\Uloga;
use Illuminate\Http\Request;

class UlogaController extends Controller
{
    private $data = [];
    private $model = null;

    public function show($id = null){
        $uloga = new Uloga();
        $this->data['roles'] = $uloga->getAll();

        if(!empty($id)){
            $uloga->id = $id;
            $this->data['updateRole'] = $uloga->get();
        }
        return view('pages.adminUloga', $this->data);
    }

    public function store(Request $request){
        $request->validate([
            'naziv' => [

```

```

        'required', 'unique:uloga'
    ]
  ], [
    'required' => 'Polje :attribute je obavezno',
    'unique' => 'Polje :attribute mora biti jedinstveno u bazi.'
  ]);

  $naziv = $request->get('naziv');

  $uloga = new Uloga();
  $uloga->naziv = $naziv;

  $rez = $uloga->save();

  if($rez == 1){
    return redirect()->back()->with('success', 'Dodata uloga!');
  }
  else {
    return redirect()->back()->with('error','Greska pri dodavanju uloge!');
  }
}

public function update($id, Request $request){
  $naziv = $request->get('naziv');

  $uloga = new Uloga();
  $uloga->id = $id;
  $uloga->naziv = $naziv;

  $rez = $uloga->update();

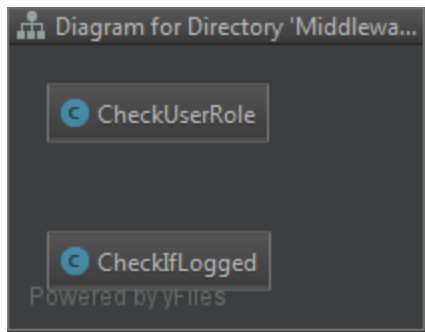
  if($rez == 1){
    return redirect()->back()->with('success', 'Uspesno azurirana uloga!');
  }
  else {
    return redirect()->back()->with('error','Greska pri azuriranju uloge!');
  }
}

public function destroy($id){
  $uloga = new Uloga();
  $uloga->id = $id;
  $rez = $uloga->delete();

  if($rez == 1){
    return redirect()->back()->with('success', 'Obrisana uloga!');
  }
  else {
    return redirect()->back()->with('error','Greska pri brisanju uloge!');
  }
}
}

```

## Midlver



## CheckIfLogged

```
<?php

namespace App\Http\Middleware;

use Closure;

class CheckIfLogged
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has('user'))
            return redirect()->back()->with('error', 'Nemate pravo pristupa ovoj stranici');
        else return $next($request);
    }
}
```

## CheckUserRole

```
<?php

namespace App\Http\Middleware;

use Closure;

class CheckUserRole
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has('user')){
```

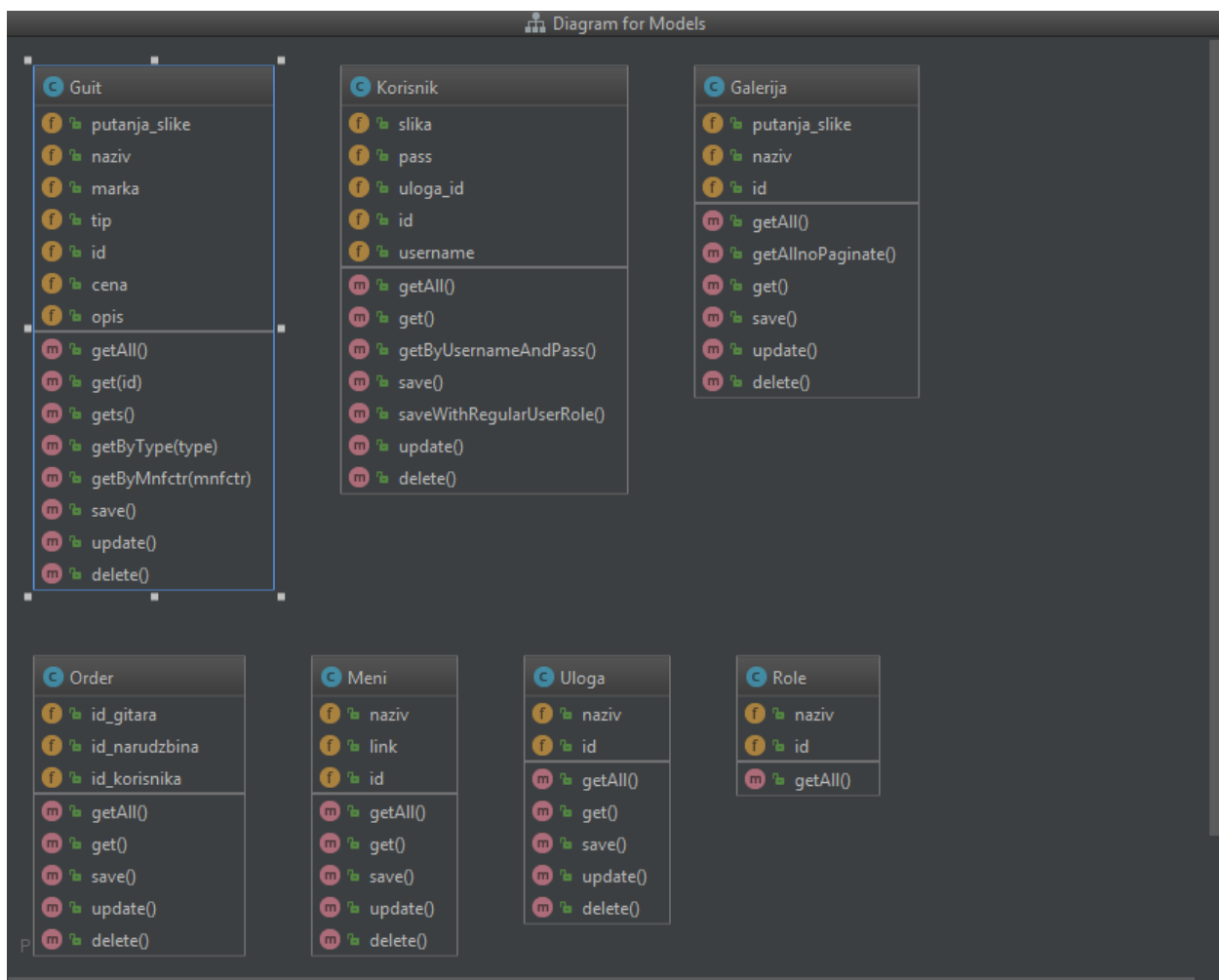
```

        $korisnik = $request->session()->get('user')[0];
        if($korisnik->username == 'adminko'){
            return $next($request);
        } else {
            return redirect()->back()->with('error','Nemate pravo pristupa ovoj
stranici!');
        }
    }

    return redirect()->back()->with('error','Nemate pravo pristupa ovoj
stranici!');
}
}

```

## Modeli



## Autors

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/8/2018
 * Time: 2:51 AM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Autor
{
    public function getAll() {
        $rezultat =
            DB::table('autor')
                ->get();
        return $rezultat;
    }
}

```

## Galerija

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/8/2018
 * Time: 5:09 AM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Galerija
{
    public $id;
    public $naziv;
    public $putanja_slike;

    public function getAll() {
        $rezultat = DB::table('galerija')->paginate(5);
        return $rezultat;
    }

    public function getAllnoPaginate() {
        $rezultat = DB::table('galerija')->get();
        return $rezultat;
    }

    public function get() {
        $rezultat =
            DB::table('galerija')
                ->select('*')
                ->where('id', $this->id)
                ->first();
        return $rezultat;
    }
}

```

```

    public function save() {
        $rezultat = DB::table('galerija')->insert([
            'naziv' => $this->naziv,
            'putanja_slike' => $this->putanja_slike
        ]);
        return $rezultat;
    }

    public function update(){
        $data = [
            'naziv' => $this->naziv
        ];

        if(!empty($this->putanja_slike)){ // ako je upload-ovana slika
            $data['putanja_slike'] = $this->putanja_slike;
        }

        $rez = DB::table('galerija')
            ->where('id',$this->id)
            ->update($data)
        ;
        return $rez;
    }

    public function delete(){
        $rezultat = DB::table('galerija')
            ->where('id', $this->id)
            ->delete();
        return $rezultat;
    }
}

```

## Guit

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/7/2018
 * Time: 6:19 PM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Guit
{
    public $id;
    public $naziv;
    public $marka;
    public $tip;
    public $opis;
    public $putanja_slike;
    public $scena;

    public function getAll() {
        $rezultat = DB::table('gitare')->get();
        return $rezultat;
    }

    public function get($id) {

```

```

        $rezultat =
            DB::table('gitare')
                ->select('*')
                ->where('id_gitara', $id)
                ->first();
        return $rezultat;
    }

    public function gets(){
        $rezultat =
            DB::table('gitare')
                ->select('*')
                ->where('id_gitara', $this->id)
                ->first();
        return $rezultat;
    }

    public function getByType($type){
        $rezultat =
            DB::table('gitare')
                ->select('*')
                ->where('tip', $type)
                ->get();
        return $rezultat;
    }

    public function getByMnfctr($mnfctr){
        $rezultat =
            DB::table('gitare')
                ->select('*')
                ->where('marka', $mnfctr)
                ->get();
        return $rezultat;
    }

    public function save() {
        $rezultat = DB::table('gitare')->insert([
            'naziv' => $this->naziv,
            'marka' => $this->marka,
            'tip' => $this->tip,
            'opis' => $this->opis,
            'putanja_slike' => $this->putanja_slike,
            'cena' => $this->cena
        ]);
        return $rezultat;
    }

    public function update(){
        $data = [
            'naziv' => $this->naziv,
            'marka' => $this->marka,
            'tip' => $this->tip,
            'opis' => $this->opis,
            'cena' => $this->cena
        ];

        if(!empty($this->putanja_slike)){ // ako je upload-ovana slika
            $data['putanja_slike'] = $this->putanja_slike;
        }

        $rez = DB::table('gitare')
            ->where('id_gitara', $this->id)
            ->update($data)
    }

```

```

        ;
        return $rez;
    }

    public function delete(){
        $rezultat = DB::table('gitare')
            ->where('id_gitara', $this->id)
            ->delete();
        return $rezultat;
    }
}

```

## Korisnik

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 4/30/2018
 * Time: 6:47 PM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Korisnik
{
    public $id;
    public $username;
    public $pass;
    public $slika;
    public $uloga_id;

    public function getAll(){
        $rezultat =
            DB::table('korisnik')
                ->select('*',
                    'korisnik.id AS korisnikId')
                ->join('uloga', 'uloga.id', '=', 'korisnik.uloga_id')
                ->get();
        return $rezultat;
    }

    public function get(){
        $rezultat =
            DB::table('korisnik')
                ->select('*')
                ->where('id', $this->id)
                ->first();
        return $rezultat;
    }

    //Za Lozinku md5

    public function getByUsernameAndPass(){
        $rezultat =

```



```

        DB::table('korisnik')
        ->select('korisnik.*', 'uloga.naziv')
        ->join('uloga', 'korisnik.uloga_id', '=', 'uloga.id')
        ->where(['username' => $this->username, 'pass' => md5($this->pass)])
        ->first();
        return $rezultat;
    }

    // SAVE UPDATE I DELETE

    // SAVE i SAVEUSER

    public function save() {
        $rezultat = DB::table('korisnik')->insert([
            'username' => $this->username,
            'pass' => md5($this->pass),
            'slika' => $this->slika,
            'uloga_id' => $this->uloga_id
        ]);
        return $rezultat;
    }

    public function saveWithRegularUserRole() {
        $rezultat = DB::table('korisnik')->insert([
            'username' => $this->username,
            'pass' => md5($this->pass),
            'slika' => $this->slika,
            'uloga_id' => '2'
        ]);
        return $rezultat;
    }

    public function update() {
        $data = [
            'username' => $this->username,
            'pass' => md5($this->pass),
            'uloga_id' => $this->uloga_id
        ];

        if(!empty($this->slika)){ // ako je upload-ovana slika
            $data['slika'] = $this->slika;
        }

        $rez = DB::table('korisnik')
            ->where('id', $this->id)
            ->update($data)
        ;
        return $rez;
    }

    public function delete() {
        $rezultat = DB::table('korisnik')
            ->where('id', $this->id)
            ->delete();
        return $rezultat;
    }
}

```

**Meni**

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/6/2018
 * Time: 6:14 PM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Meni
{
    public $id;
    public $naziv;
    public $link;

    public function getAll(){
        $rezultat = DB::table('meni')->get();
        return $rezultat;
    }

    public function get(){
        $rezultat =
            DB::table('meni')
                ->select('*')
                ->where('id', $this->id)
                ->first();
        return $rezultat;
    }

    public function save() {
        $rezultat = DB::table('meni')->insert([
            'id' => $this->id,
            'naziv' => $this->naziv,
            'link' => $this->link
        ]);
        return $rezultat;
    }

    public function update(){
        $data = [
            'id' => $this->id,
            'naziv' => $this->naziv,
            'link' => $this->link
        ];

        $rez = DB::table('meni')
            ->where('id',$this->id)
            ->update($data)
        ;
        return $rez;
    }

    public function delete(){
        $rezultat = DB::table('meni')
            ->where('id', $this->id)
            ->delete();
        return $rezultat;
    }
}

```

## Order

```
<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/7/2018
 * Time: 10:03 PM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Order
{
    public $id_narudzbina;
    public $id_gitara;
    public $id_korisnika;

    public function getAll() {
        $rezultar = DB::table('narudzbina')->
            join('gitare', 'gitare.id_gitara', '=', 'narudzbina.id_gitara')->
            join('korisnik', 'id', '=', 'narudzbina.id_korisnika')->
            get();
        return $rezultar;
    }

    public function get() {
        $rezultat = DB::table('narudzbina')
            ->select('*')
            ->where('id_narudzbina', $this->id_narudzbina)
            ->first();
        return $rezultat;
    }

    public function save() {
        $rezultat = DB::table('narudzbina')->insert([
            'id_gitara' => $this->id_gitara,
            'id_korisnika' => $this->id_korisnika
        ]);
        return $rezultat;
    }

    public function update() {
        $data = [
            'id_gitara' => $this->id_gitara,
            'id_korisnika' => $this->id_korisnika
        ];

        $rez = DB::table('narudzbina')
            ->where('id', $this->id_narudzbina)
            ->update($data)
        ;
        return $rez;
    }

    public function delete() {
        $rezultat = DB::table('narudzbina')
```

```

        ->where('id', $this->id_narudzbina)
        ->delete();
        return $rezultat;
    }
}

```

## Role

```

<?php
/**
 * Created by PhpStorm.
 * User: Mladen
 * Date: 5/9/2018
 * Time: 7:23 PM
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

// THIS CLASS IS MY ROLE MODEL (PUN INTENDED)

class Role
{
    public $id;
    public $naziv;

    public function getAll(){
        $rezultar = DB::table('uloga')->get();
        return $rezultar;
    }
}

```

## Uloga

```

<?php

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Uloga {

    public $id;
    public $naziv;

    public function getAll(){
        $rezultat = DB::table('uloga')->get();
        return $rezultat;
    }
}

```

```

public function get(){
    return DB::table('uloga')
        ->select('*')
        ->where('id',$this->id)
        ->first();
}

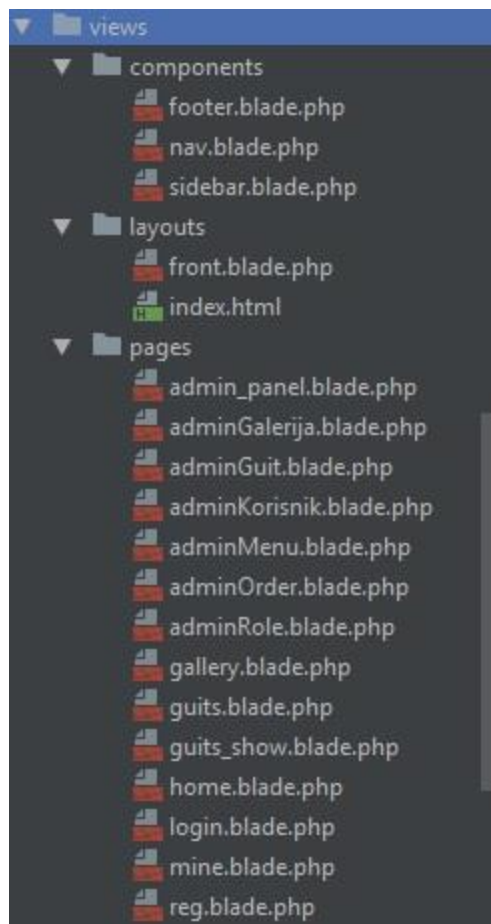
public function save(){
    return DB::table('uloga')
        ->insert([
            'naziv' => $this->naziv
        ]);
}

public function update(){
    return DB::table('uloga')
        ->where('id', $this->id)
        ->update([
            'naziv' => $this->naziv
        ]);
}

public function delete(){
    return DB::table('uloga')
        ->where('id', $this->id)
        ->delete();
}
}

```

## Views



## Components

### Footer.blade

```
<div class="col-md-12 footer-box">

    <div class="row">
        <div class="col-md-4">

        </div>

        <div class="col-md-4"><br><br><Br>
            <strong>Beograd</strong>
            <hr>
            <p>

                Created by:
                <br>Mladen Karic
                Email: mysteryx96@yahoo.com<br>
            </p>

            2018 www.mladensblogger.rs | All Right Reserved
```

```

        </div>
        <div class="col-md-4 social-box">

        </div>
    </div>
    <hr>
</div>
<!-- /.col -->
<div class="col-md-12 end-box ">
    &copy; 2018 | &nbsp; All Rights Reserved | &nbsp; www.mladensblogger.com |
&nbsp; &nbsp; Email me: mysteryx96@yahoo.com
</div>
<!-- /.col -->

```

## Nav.blade

```

<nav class="navbar navbar-default" role="navigation">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="{{ route('home') }}">Mladen's Guitar
Shop</a>
        </div>

        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

            <ul class="nav navbar-nav navbar-right">
                @isset($menus)
                    @foreach($menus as $menu)
                        <li><a href="{{ route($menu->link) }}">{{ $menu->naziv
}}</a></li>
                    @endforeach
                @endisset
                @empty(session('user'))
                    <li><a href="{{ route('login_page') }}">Logovanje</a></li>
                @endempty
                @if(session()->has('user'))
                    <li><a href="{{ route('logout') }}">Logout</a></li>
                @endif
                @empty(session('user'))
                    <li><a href="{{ route('reg_page') }}">Registracija</a></li>
                @endempty
                @if(session('isAdmin'))
                    <li><a href="{{ route('admin_panel') }}">Admin Panel</a></li>
                @endif

            </ul>

        </div>
    <!-- /.navbar-collapse -->

```

```

    </div>
    <!-- /.container-fluid -->
</nav>

```

## Sidebar.blade

```

<div class="col-md-3">

    <div>
        <a href="{{ asset('/guitarsType/Acoustic') }}" class="list-group-item
active">Akustične gitare
        </a>
        <ul class="list-group">

            <li class="list-group-item"><a href="{{
asset('/guitarsMnfctr/Yamaha') }}">Yamaha</a>
<span class="label label-primary pull-right"></span>
            </li>
            <li class="list-group-item"><a href="{{
asset('/guitarsMnfctr/Epiphone') }}">Epiphone</a>
            <span class="label label-success pull-right"></span>
            </li>
        </ul>
    </div>
    <!-- /.div -->
    <div>
        <a href="{{ asset('/guitarsType/Electric') }}" class="list-group-item
active list-group-item-success">Električne gitare
        </a>
        <ul class="list-group">
            <li class="list-group-item"><a href="{{ asset('/guitarsMnfctr/LTD')
}}">LTD</a>
            <span class="label label-warning pull-right"></span>
            </li>
            <li class="list-group-item"><a href="{{
asset('/guitarsMnfctr/Fender') }}">Fender</a>
            <span class="label label-success pull-right"></span>
            </li>
        </ul>

    </div>
    <!-- /.div -->
    <div>
        <a href="{{ asset('/guitarsType/Bass') }}" class="list-group-item
active">Bas Gitare
        </a>
        <ul class="list-group">
            <li class="list-group-item"><a href="{{ asset('/guitarsMnfctr/LTD')
}}">LTD</a>
            <span class="label label-warning pull-right"></span>
            </li>
            <li class="list-group-item"><a href="{{
asset('/guitarsMnfctr/Fender') }}">Fender</a>
            <span class="label label-success pull-right"></span>
            </li>
        </ul>
    </div>

```



```
<!-- /.div -->

</div>
```

## Layouts

### Front.blade

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Mladen's Guitar Shop</title>

    <!-- LINKS -->

    <!-- Bootstrap core CSS -->
    <link href="{{ asset('/') }}vendor/bootstrap/css/bootstrap.css" rel="stylesheet">
    <!-- Fontawesome core CSS -->
    <link href="{{ asset('/') }}vendor/bootstrap/css/font-awesome.min.css"
rel='stylesheet'>
    <!--GOOGLE FONT -->
    <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet'
type='text/css'>
    <!--Slide Show Css -->
    <link href="{{ asset('/') }}css/main-style.css" rel='stylesheet'>
    <!-- custom CSS here -->
    <link href="{{ asset('/') }}css/style.css" rel='stylesheet'>

    <!-- ENDLINKS -->

</head>

<body>

    @include('components.nav')

    <!-- GLAVNI_KONTEJNER -->

    <div class="container">

        <div class="col-lg-12">
            @empty(!session('error'))
                <div class="alert alert-danger">{{ session('error') }}</div>
            @endempty

            @empty(!session('success'))
                <div class="alert alert-success">{{ session('success') }} </div>
            @endempty

            @empty(!session('message'))
```

```

        <div class="alert alert-success">{{ session('success') }} </div>
    @endempty
</div>
    @yield('container_1')

    <div class="row">

<!-- SIDEBAR OVDE -->

        @include('components.sidebar')

<!-- CONTAINER OVDE -->
        <div class="col-md-9">
            <div class="row">
                @yield('container_2')
            </div>
        </div>

    </div>

</div>

<!-- ENDGLAVNI_KONTEJNER -->

<!-- PRE_FOOTER -->

<div class="col-md-12 download-app-box text-center">

</div>

<!-- ENDPRE_FOOTER -->

<!-- FOOTER -->

    @include('components.footer')

<!-- ENFOOTER-->

<!-- JSCRIPTS -->

<!--Core JavaScript file -->
<script src="{{ asset('/') }}vendor/jquery/jquery-1.10.2.js" >
</script>
<!--bootstrap JavaScript file -->
<script src="{{ asset('/') }}vendor/bootstrap/js/bootstrap.js" ></script>
<!--Slider JavaScript file -->
<script src="{{ asset('/') }}js/modernizr.custom.63321.js"></script>
<script src="assets/ItemSlider/js/jquery.catslider.js"></script>
<script src="{{ asset('/') }}js/jquery.catslider.js"></script>
<script>
    $(function () {

        $('#mi-slider').catslider();

    });
</script>

<!-- ENDJSCRIPTS -->

```

```
</body>
</html>
```

## Pages

### Admin\_panel.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <a href="{{ asset('/users') }}" >Korisnici</a><br><br>
    <a href="{{ asset('/galls') }}" >Galerija</a><br><br>
    <a href="{{ asset('/guits') }}" >Gitare</a><br><br>
    <a href="{{ asset('/menus') }}" >Meni</a><br><br>
    <a href="{{ asset('/orders') }}" >Narudzbina</a><br><br>
    <a href="{{ asset('/roles') }}" >Uloga</a><br>

@endsection
```

### AdminGalerija.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @empty(!session('message'))
        <div class="alert alert-danger">
            {{ session('message') }}
        </div>
    @endempty

    <form action="{{ (isset($galerija)) ? asset('/galls/update/' . $galerija->id) :
asset('/galls/store') }}" method="POST" enctype="multipart/form-data">

        {{ csrf_field() }}

    </form>
```

```

        <div class="form-group">
            <label>Naziv</label>
            <input type="text" name="tbNaziv" class="form-control" value="{{
(isset($galerija))? $galerija->naziv : old('tbNaziv') }}" />
        </div>
        <label>Slika:</label>

        @isset($galerija)
            
        @endisset

        <input type="file" name="tbPic" class="form-control" />

    </div>
    <div class="form-group">
        <input type="submit" name="addGall" value="{{ (isset($galerija))? 'Change
gall' : 'Add gall' }}" class="btn btn-default" />
    </div>
</form>

<table class="table">
    <tr>
        <td>Id</td>
        <td>Naziv</td>
        <td>Slika</td>
    </tr>
    <!-- Prikaz korisnika-->
    @isset($galerije)
        @foreach($galerije as $galerija)
            <tr>
                <td> {{ $galerija->id }} </td>
                <td> {{ $galerija->naziv }} </td>
                <td>  </td>
                <td> <a href="{{ asset('/galls/' . $galerija->id) }}">Izmeni</a>
</td>
                <td> <a href="{{ asset('/galls/destroy/' . $galerija->id)
}}">Obrisi</a> </td>
            </tr>
        @endforeach
    @endisset
</table>

@endsection

```

## AdminGuit.blade

```

<@extends('layouts.front')

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

```

```

@empty(!session('message'))
    <div class="alert alert-danger">
        {{ session('message') }}
    </div>
@endempty

<form action="{{ (isset($guit)) ? asset('/guits/update/' . $guit->id_gitara) :
asset('/guits/store') }}" method="POST" enctype="multipart/form-data">

    {{ csrf_field() }}

    <div class="form-group">
        <label>Naziv:</label>
        <input type="text" name="tbNaziv" class="form-control" value="{{
(isset($guit)) ? $guit->naziv : old('tbNaziv') }}" />
    </div>
    <div class="form-group">
        <label>Marka:</label>
        <input type="text" name="tbMarka" class="form-control" value="{{
(isset($guit)) ? $guit->marka : old('tbMarka') }}" />
    </div>
    <div class="form-group">
        <label>Tip:</label>
        <input type="text" name="tbTip" class="form-control" value="{{
(isset($guit)) ? $guit->tip : old('tbTip') }}" />
    </div>
    <div class="form-group">
        <label>Opis:</label>
        <input type="text" name="tbOpis" class="form-control" value="{{
(isset($guit)) ? $guit->opis : old('tbOpis') }}" />
    </div>
    <div class="form-group">
        <label>Cena:</label>
        <input type="text" name="tbCena" class="form-control" value="{{
(isset($guit)) ? $guit->cena : old('tbCena') }}" />
    </div>
    <div class="form-group">
        <label>Slika:</label>

        @isset($guit)
            
        @endisset

        <input type="file" name="tbSlika" class="form-control" />

    </div>

    <div class="form-group">
        <input type="submit" name="addGuit" value="{{ (isset($guit)) ? 'Change
guit' : 'Add guit' }}" class="btn btn-default" />
    </div>
</form>

<table class="table">
    <tr>
        <td>Id</td>
        <td>Naziv</td>
        <td>Marka</td>
        <td>Tip</td>
        <td>Opis</td>
    </tr>

```

```

        <td>Cena</td>
        <td></td>
        <td>Izmeni</td>
        <td>Obrisi</td>
    </tr>
    <!-- Prikaz korisnika-->
    @isset($guits)
        @foreach($guits as $guit)
            <tr>
                <td> {{ $guit->id_gitara }} </td>
                <td> {{ $guit->naziv }} </td>
                <td> {{ $guit->marka }} </td>
                <td> {{ $guit->tip }} </td>
                <td> {{ $guit->opis }} </td>
                <td> {{ $guit->cena }} </td>
                <td> 
            </td>
                <td> <a href="{{ asset('/guits/' . $guit->id_gitara) }}">Izmeni</a>
            </td>
                <td> <a href="{{ asset('/guits/destroy/' . $guit->id_gitara)
            }}">Obrisi</a> </td>
            </tr>
        @endforeach
    @endisset
</table>
@endsection

```

## AdminKorisnik.blade

```

@extends('layouts.front')

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @empty(!session('message'))
        <div class="alert alert-danger">
            {{ session('message') }}
        </div>
    @endempty

    <form action="{{ (isset($korisnik)) ? asset('/users/update/' . $korisnik->id) :
asset('/users/store') }}" method="POST" enctype="multipart/form-data">

        {{ csrf_field() }}

        <div class="form-group">
            <label>Korisnicko ime:</label>
            <input type="text" name="tbUsername" class="form-control" value="{{
(isset($korisnik)) ? $korisnik->username : old('tbUsername') }}" />
        </div>
        <div class="form-group">

```

```

        <label>Lozinka:</label>
        <input type="password" name="tbPass" class="form-control" value="{{
(isset($korisnik))? $korisnik->pass : old('tbPass') }}" />
    </div>
    <div class="form-group">
        <label>Slika:</label>

        @isset($korisnik)
            
        @endisset

        <input type="file" name="tbPic" class="form-control" />

    </div>
    <div class="form-group">
        <label>Uloga:</label>
        <select name="ddlUloga">
            <option value="0">Izaberite</option>

            @foreach($uloge as $uloga)
                <option value="{{ $uloga->id }}" {{ (isset($korisnik) &&
$korisnik->uloga_id == $uloga->id)? 'selected' : (old('ddlUloga')== $uloga->id)?
'selected' : ' ' }} > {{ $uloga->naziv }} </option>
            @endforeach

        </select>
    </div>
    <div class="form-group">
        <input type="submit" name="addKorisnik" value="{{ (isset($korisnik))?
'Change korisnik' : 'Add korisnik' }}" class="btn btn-default" />
    </div>
</form>

<table class="table">
    <tr>
        <td>Id</td>
        <td>Username</td>
        <td>Slika</td>
        <td>Uloga</td>
        <td>Izmeni</td>
        <td>Obriši</td>
    </tr>
    <!-- Prikaz korisnika-->
    @isset($korisnici)
        @foreach($korisnici as $korisnik)
            <tr>
                <td> {{ $korisnik->korisnikId }} </td>
                <td> {{ $korisnik->username }} </td>
                <td>  </td>
                <td> {{ $korisnik->naziv }} </td>
                <td> <a href="{{ asset('/users/' . $korisnik->korisnikId)
}}">Izmeni</a> </td>
                <td> <a href="{{ asset('/users/destroy/' . $korisnik->korisnikId)
}}">Obrisi</a> </td>
            </tr>
        @endforeach
    @endisset
</table>

@endsection

```

## AdminMenu.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @empty(!session('message'))
        <div class="alert alert-danger">
            {{ session('message') }}
        </div>
    @endempty

    <table class="table">
        <tr>
            <td>Id</td>
            <td>Naziv</td>
            <td>Link</td>
        </tr>
        <!-- Prikaz korisnika-->
        @isset($menus)
            @foreach($menus as $menu)
                <tr>
                    <td> {{ $menu->id }} </td>
                    <td> {{ $menu->naziv }} </td>
                    <td> {{ $menu->link }} </td><!--
                    <td> <a href="{{ asset('/menus/'.$menu->id) }}">Izmeni</a> </td>
                    <td> <a href="{{ asset('/menus/destroy/'.$menu->id) }}">Obrisi</a>
                </tr>
            @endforeach
        @endisset
    </table>

@endsection
```

## AdminOrder.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
```



```

        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

@empty(!session('message'))
    <div class="alert alert-danger">
        {{ session('message') }}
    </div>
@endempty

<table class="table">
    <tr>
        <td>Id</td>
        <td>Gitara</td>
        <td>Narucio</td>
    </tr>
    @isset($orders)
        @foreach($orders as $order)
            <tr>
                <td> {{ $order->id_narudzbina }} </td>
                <td> {{ $order->naziv }} </td>
                <td> {{ $order->username }} </td>
            </tr>
        @endforeach
    @endisset
</table>

@endsection

```

## AdminRole.blade

```

@extends('layouts.front')

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @empty(!session('message'))
        <div class="alert alert-danger">
            {{ session('message') }}
        </div>
    @endempty

```

```

<table class="table">
  <tr>
    <td>Id</td>
    <td>Naziv</td>
  </tr>
  @isset($roles)
    @foreach($roles as $role)
      <tr>
        <td> {{ $role->id }} </td>
        <td> {{ $role->naziv }} </td>
      </tr>
    @endforeach
  @endisset
</table>

@endsection

```

## Gallery.blade

```

@extends('layouts.front')

@section('container_2')
  @if ($errors->any())
    <div class="alert alert-danger">
      <ul>
        @foreach ($errors->all() as $error)
          <li>{{ $error }}</li>
        @endforeach
      </ul>
    </div>
  @endif

  @isset($galls)

    @foreach($galls as $gall)
      <div class="col-md-4 text-center col-sm-6 col-xs-6">
        <div class="thumbnail product-box">
          <a href="{{ asset($gall->putanja_slike) }}"></a>
          <div class="caption">
            <p>Naziv: <strong>{{ $gall->naziv }}</strong> </p>
          </div>
        </div>
      </div>
    @endforeach

  @endisset

  {{ $galls->links() }}

@endsection

```

## Guits.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @isset($guits)
        @foreach($guits as $guit)
            <div class="col-md-4 text-center col-sm-6 col-xs-6">
                <div class="thumbnail product-box">
                    <a href="{{ asset('/guits_show/' . $guit->id_gitara) }}"></a>
                    <div class="caption">
                        <h3><a href="{{ asset('/guits_show/' . $guit->id_gitara)
}}">{{ $guit->naziv }}</a></h3>
                        <p>Cena: <strong>{{ $guit->cena }}</strong> </p>
                        <p><a href="{{ asset('/guits_show/' . $guit->id_gitara)
}}">Detaljnije</a>
                    </div>
                </div>
            </div>
        @endforeach
    @endisset
@endsection
```

## Guits\_show.blade

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

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
```

```

@isset($guit)

    <div class="col-md-4 text-center col-sm-6 col-xs-6">
        <div class="thumbnail product-box">
            
            <div class="caption">
                <h3>{{ $guit->naziv }}</h3>
                <p>Cena: <strong>{{ $guit->cena }}</strong> </p>
                <p>{{ $guit->opis }}</p>
                <p><a href="{{ asset('/guit_order/' . $guit->id_gitara) }}"
class="btn btn-success" role="button">Naruci</a> </p>
            </div>
        </div>
    </div>

@endisset

@endsection

```

## Home.blade

```

@extends('layouts.front')

@section('container_1')

    <div class="row">
        <div class="col-md-12">

            <div>

            </div>
            <br>
        </div>
        <!-- /.col -->

    </div>

@endsection

@section('container_2')

@endsection

```

## Login.blade

```

@extends('layouts.front')

@section('container_1')

    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <!-- LOGIN FORMA SREDI-->
    <form action="{{ route('login') }}" method="POST">
        {{ csrf_field() }}
        <div class="form-group">
            <label>Korisničko ime:</label><br>
            <input type="text" name="tbUsername" class="form-control-mine" />
        </div>

        <div class="form-group">
            <label>Lozinka:</label><br>
            <input type="password" name="tbPass" class="form-control-mine"/>
        </div>

        <input type="submit" name="btnLogin" value="Login" class="btn btn-primary"/>
    </form>
    <!--// LOGIN FORMA -->

@endsection

```

## Mine.blade

```

@extends('layouts.front')

@section('container_2')
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    
    <br><br>

    @isset($mine)
        @foreach($mine as $min)
            <p>{{ $min->autor_text }}</p>
        @endforeach
    @endisset

```

```
@endsection
```

## Reg.blade

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

@section('container_2')

    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <!-- LOGIN FORMA SREDI-->
    <form action="{{ route('reg') }}" method="POST" enctype="multipart/form-data">
        {{ csrf_field() }}
        <div class="form-group">
            <label>Korisničko ime:</label><br>
            <input type="text" name="tbUsername" class="form-control-mine" />
        </div>

        <div class="form-group">
            <label>Lozinka:</label><br>
            <input type="password" name="tbPass" class="form-control-mine"/>
        </div>

        <div class="form-group">
            <label>Slika:</label>
            <input type="file" name="tbPic" class="form-control-mine" />
        </div>

        <input type="submit" name="btnLogin" value="Registruj se" class="btn btn-
primary"/>
    </form>
    <!--// LOGIN FORMA -->

@endsection
```

## Rute

## Web

```

<?php

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

// INDEX

Route::get('/', 'FrontendController@index')->name('home');

// MIDDLEWARE

Route::group(['middleware' => 'admin'], function () {

    Route::get('/admin_panel', 'FrontendController@admin_panel')->name('admin_panel');

    // KORISNICI

    Route::get('/users/{id?}', 'KorisnikController@show');
    Route::post('/users/store', 'KorisnikController@storeWithChosenRole');
    Route::post('/users/update/{id}', 'KorisnikController@update');
    Route::get('/users/destroy/{id}', 'KorisnikController@destroy');

    // GALERIJA

    Route::get('/galls/{id?}', 'GalerijaController@show');
    Route::post('/galls/store', 'GalerijaController@store');
    Route::post('/galls/update/{id}', 'GalerijaController@update');
    Route::get('/galls/destroy/{id}', 'GalerijaController@destroy');

    // GITARE

    Route::get('/guits/{id?}', 'GuitsController@show');
    Route::post('/guits/store', 'GuitsController@store');
    Route::post('/guits/update/{id}', 'GuitsController@update');
    Route::get('/guits/destroy/{id}', 'GuitsController@destroy');

    // MENI

    Route::get('/menus/{id?}', 'MeniController@show');
    Route::post('/menus/store', 'MeniController@store');
    Route::post('/menus/update/{id}', 'MeniController@update');
    Route::get('/menus/destroy/{id}', 'MeniController@destroy');

    // NARUDZBINA

    Route::get('/orders/{id?}', 'OrderController@show');
    Route::post('/orders/store', 'OrderController@store');
    Route::post('/orders/update/{id}', 'OrderController@update');
    Route::get('/orders/destroy/{id}', 'OrderController@destroy');

    /*
    */

    // ULOGE

    Route::get('/roles', 'RoleController@show');

```

```
});

// LOGIN I LOGOUT I REGISTRACIJA SVI POD MIDLVEROM
Route::get('/logout', 'LoginController@logout')->name('logout');

Route::group(['middleware' => 'login'], function(){
    Route::get('/login_page', 'FrontendController@login_page')->name('login_page');
    Route::post('/login', 'LoginController@login')->name('login');
    Route::get('/reg_page', 'FrontEndController@reg_page')->name('reg_page');
    Route::post('/reg', 'KorisnikController@store')->name('reg');
});

// GUIITS
Route::get('/guitars', 'GuitsController@guitars')->name('guitars');
Route::get('/guitarsType/{type}', 'GuitsController@guitarsByType');
Route::get('/guitarsMnfctr/{mnfctr}', 'GuitsController@guitarsByMnfctr');
Route::get('/guits_show/{id}', 'GuitsController@guitar_specific');
Route::get('/guit_order/{id}', 'OrderController@guit_order');

// OAUTORU
Route::get('/autor', 'FrontendController@autor')->name('autor');

// GALERIJA
Route::get('/gallery', 'GalerijaController@gallery')->name('gallery');

// DOKUMENTACIJA
Route::get('/doc', 'FrontendController@doc')->name('doc');
```