Visoka škola strukovnih studija za informacione i komunikacione tehnologije Zdravka Čelara 16, Beograd



**Dokumentacija**

**za sajt TopNews**

Smer: **Internet tehnologije**

Modul: **Web programiranje**

Predmet: **Web programiranje PHP2**

**Aleksandra Fridl 311/14**

**Beograd, Mart 2021. Godine**

SADRŽAJ

1.UVOD

1.1. Korišćeni programski jezici

1.2. Opis funkcionalnosti

1.3. Template

1.4 Korišćeni CSS

2. ORGANIZACIJA

2.1. Organizaciona šema

2.2. Mapa sajta

2.3. Slike stranica I opis funkcionalnosti

2.3.1. php2Sajt2\_8.03/public/ - Početna

2.3.2. – php2Sajt2\_8.03/public/registracija -Registracija

2.3.3. – php2Sajt2\_8.03/public/logovanje – Logovanje

2.3.4. - php2Sajt2\_8.03/public/kontakt – Kontakt

2.3.5. – php2Sajt2\_8.03/public/autor - Autor

2.3.6. – php2Sajt2\_8.03/public/galerija – Galerija

2.3.7. – php2Sajt2\_8.03/public/korisnik/2 – Korisnik

2.3.8. – php2Sajt2\_8.03/public/adminCategory – Kategorije

2.3.9. – php2Sajt2\_8.03/public/adminCategory\_insert – Dodaj kategoriju

2.3.10 - php2Sajt2\_8.03/public/adminCategory\_update/19 – Izmeni kategoriju

2.3.11- php2Sajt2\_8.03/public/adminCategory\_delete/12 -Delete kategoriju

2.3.12 - php2Sajt2\_8.03/public/adminUsers – Korisnici

2.3.13 – php2Sajt2\_8.03/public/adminUsers\_insert – Dodaj korisnika

2.3.14 – php2Sajt2\_8.03/public/adminUsers\_update/10 – Izmeni korisnika

2.3.15 – php2Sajt2\_8.03/public/adminUser\_delete - Delete korisnika

2.3.16 – php2Sajt2\_8.03/public/adminRoles – Uloge

2.3.17 – php2Sajt2\_8.03/public/adminRoles\_update/2 – Izmeni uloge

2.3.18– php2Sajt2\_8.03/public/adminRoles\_update/2 – Dodaj ulogu

2.3.19 – php2Sajt2\_8.03/public/adminRoles\_delete -Delete uloge

2.3.20 - php2Sajt2\_8.03/public/adminNews – Vesti

2.3.21 - php2Sajt2\_8.03/public/adminNews\_update/22 – Izmeni vest

2.3.22 - php2Sajt2\_8.03/public/adminNews\_insert – Dodaj vest

2.3.23 – php2Sajt2\_8.03/public/adminNews\_delete -Delete vesti

2.3.24 - php2Sajt2\_8.03/public/adminComments – Komentari

2.3.25 – php2Sajt2\_8.03/public/adminComments\_delete - Delete komentar

2.3.26 - php2Sajt2\_8.03/public/admin\_contactForm – Poruke administrator

2.3.27 – php2Sajt2\_8.03/public/adminContactForm\_delete /28- Delete poruka

2.3.28 - php2Sajt2\_8.03/public/activity – Aktivnosti

2.3.29 - php2Sajt2\_8.03/public/logout – Logout

2.3.30 - php2Sajt2\_8.03/public/news/39

3. KODOVI

**3.1. RUTE**

**Routes->web.php**

**HTML**

3.2. **HTML (resources->views->layout)**

3.2.1. index.blade.php

3.3. **HTML (resources->views->components)**

3.3.1. footer.blade.php

3.3.2. head.blade.php

3.3.3. nav.blade.php

3.3.4. sidebar.blade.php

3.4. **HTML (resources->views->pages)**

3.4.1. admin\_contactForm.blade.php

3.4.2. adminActivity.blade.php

3.4.3. adminCategory.blade.php

3.4.4. adminCategory\_insert.blade.php

3.4.5. adminCategory\_update.blade.php

3.4.6. adminComments.blade.php

3.4.7. adminNews.blade.php

3.4.8. adminNews\_insert.blade.php

3.4.9. adminNews\_update.blade.php

3.4.10. adminRoles.blade.php

3.4.11. adminRoles\_insert.blade.php

3.4.12. adminRoles\_update.blade.php

3.4.13. adminUsers.blade.php

3.4.14. adminUsers\_insert.blade.php

3.4.15. adminUsers\_update.blade.php

3.4.16. author.blade.php

3.4.17. galery.blade.php

3.4.18. [home.blade.php](http://home.blade.php)

3.4.19. kontakt.blade.php

3.4.20. logovanje.blade.php

3.4.21. oneNews.blade.php

3.4.22. registracija.blade.php

3.4.23. showKorisnik.blade.php

3.5. **CSS (public/js/)**

3.5.1. css/style.css

3.5.2. public/js/script.js

3.5.3. public/js/jquery.js

3.6. **PHP (app/http/Controlers)**

3.6.1 – AuthController.php

3.6.2 – BeckendController

3.6.3 – CategoryController.php

3.6.4 – CommentController

3.6.5 - FrontendController.php

3.6.6 - KorisnikController.php

3.6.7 – NewsController.php

3.6.8 – RoleController.php

3.6.9 – UserController.php

3.7. **PHP - Model (app/http/Models)**

3.7.1 Activity

3.7.2 Author

3.7.3. Comment

3.7.4. Contact

3.7.5. Kategorija

3.7.6. Korisnik

3.7.7. News

3.7.8. Uloga

3.8. **Middelware (app/http/Middleware)**

3.8.1. AdminMiddleware.php

3.8.2. KorisnikMiddleware.php

3.9. **Kernel.php (app/http/Kernel.php)**

4.1 **Requests (app/http/Requests)**

4.1. CategoryUpdateRequest.php

4.2. KorisnikRequest.php

4.3. NewsRequest.php

4.4. RegistracijaRequest.php

4.5. RoleRequest.php

4.6. UserRequest.php

4. BAZA PODATAKA

4.1 Tabele

4.1.1 Tabela activities

4.1.2 Tabela author

4.1.3 Tabela categories

4.1.4 Tabela comments

4.1.5 Tabela contacts

4.1.6 Tabela menus

4.1.7 Tabela news

4.1.8 Tabela roles

4.1.9 Tabela users

**UVOD**

1.1. Korišćeni programski jezici

Za izradu ovog sajta korisćeni su sledeći jezici, biblioteke i tehnologije:

* **HTML**
* **CSS**
* **PHP**
* **JavaScript**
* **AJAX**
* **JQuery**
* **PHPMailer**
* **jquery-2.1.3.min**
* **Bootstrap**

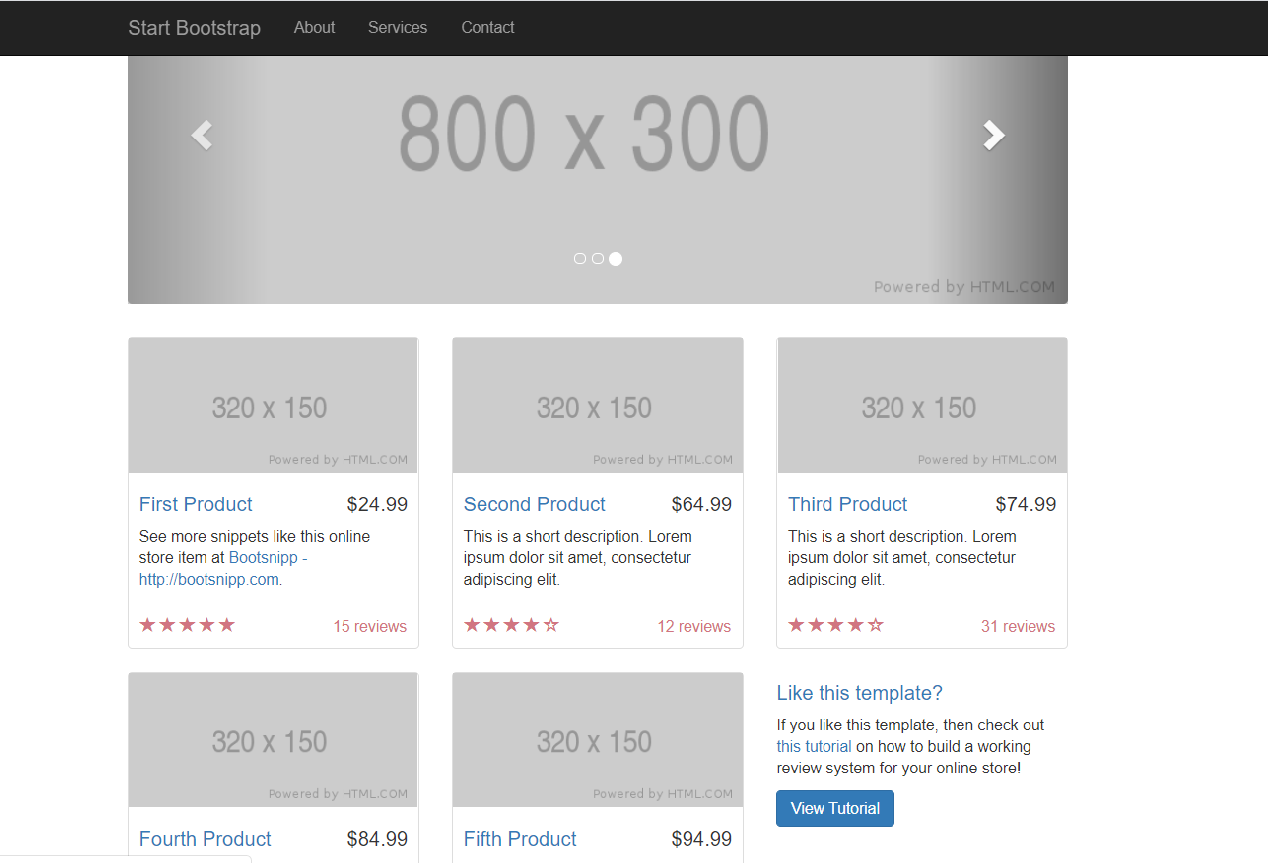
Ceo kod je pisan u **PhpStorm** text editoru.

1.2. Opis funkcionalnosti

Sajt ima sledeće funkcionalnosti:

* Sajt je realizovan u MVC arhitekturi i izrađen korišćenjem OOPHP u Laravel framework-u.
* Korišćen je Bootrap templejt sa dodatim komponentama I neophodnim bibliotekama.
* orišćena je Mysql baza podataka.
* Sadržaj sajta je realizovan dinamičkim ispisom iz baze
* Registracija i logovanje
* Postoje tri uloge: korisnik, administrator i neautorizovan
* Ulogovani korisnik – može da ostavlja komentare na vesti I pregleda galeriju, radi pretragu I pregled vesti po kategorijama
* Ulogovani administrator – ima pregled svih podataka iz baze u vidu tabela raspoređenih po linkovima koji odgovaraju tabelama u bazi. Na njima može da doda, izmeni ili obriše podatak. Kod pregleda komentara i poruka administratoru u tabelama može da izvrši samo brisanje. Na stranici za pregled aktivnosti može da uradi filtriranje po datumu i tada rezultat biva prikazan u vidu tabele na linku Aktivnosti. Ovo je realizovano putem AJAXa. Takođe ovde je unos kategorije realizovan putem AJAXa kao i pretraga vesti na linku Vesti.
* Neautorizovani korisnik može samo da radi pregled vesti, pretragu vesti na početnoj strani, registraciju I logovanje, kao i slanje mejla administratoru sajta koje je realizovano pomoću biblioteke PHPMailer.
* Pomoći **JavaScript-a i Ajaxa** realizovana je pretraga na strainci za korisnika, kao I na strainici za administratora, upis komentara, upis kategorije na admin panelu, filtriranje po datumu iz tabele radi prikaza u tabeli
* Paginacija vesti, admin penelu I galerija slika.
* Provera podataka preko reguralnih izraza u formama

1.3 Template



<body>

<nav></nav>

<div class=”container”>

<footer>

<div class=”row”></div>

</footer>

</div>

<div class=”container”>

<div class=”row”>

<div class=” row carusel-holder”></div>

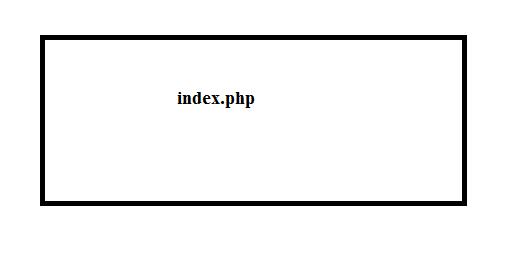
<div class=”row”></div>

</div>

</div>

**2.** **ORGANIZACIJA**

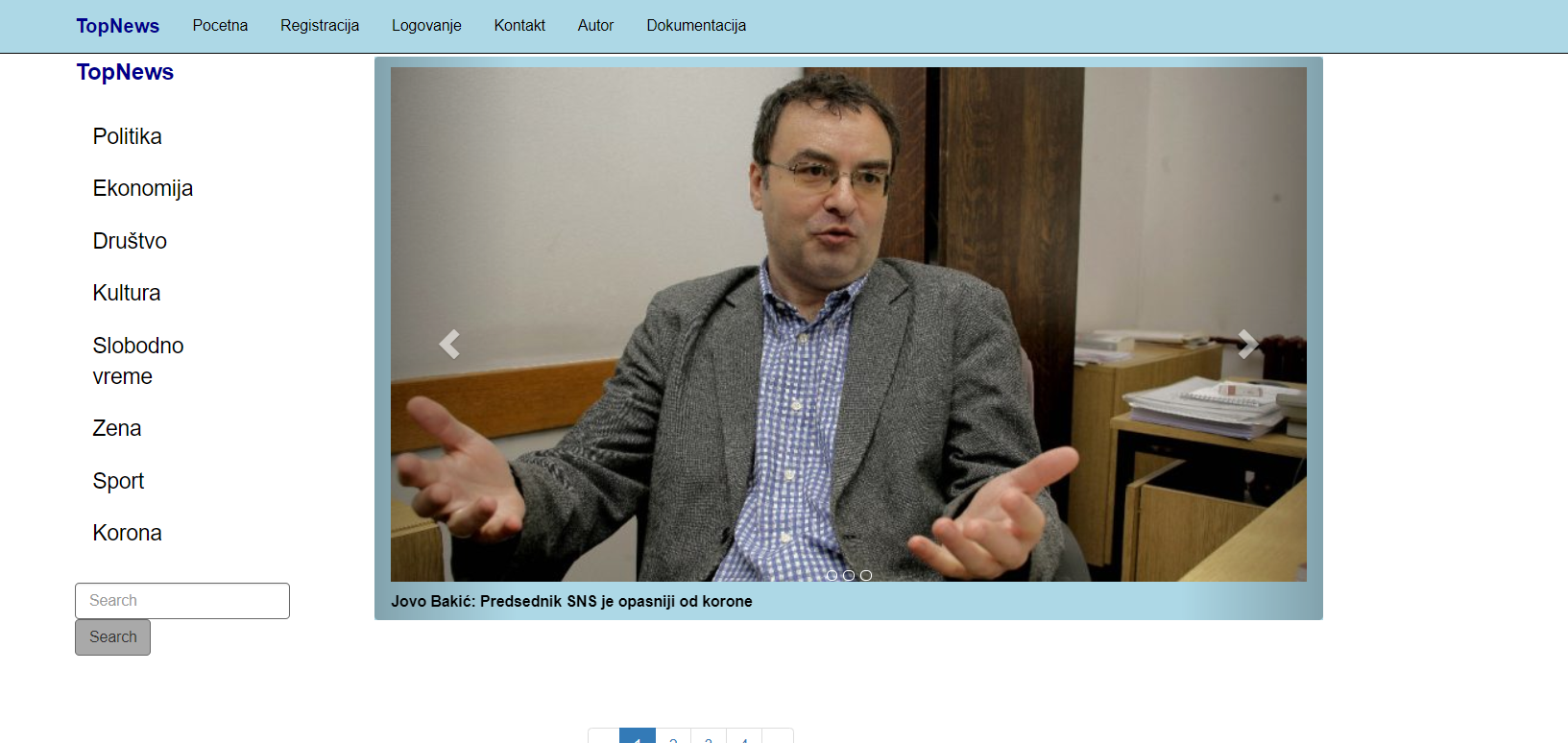
2.1. Organizaciona šema



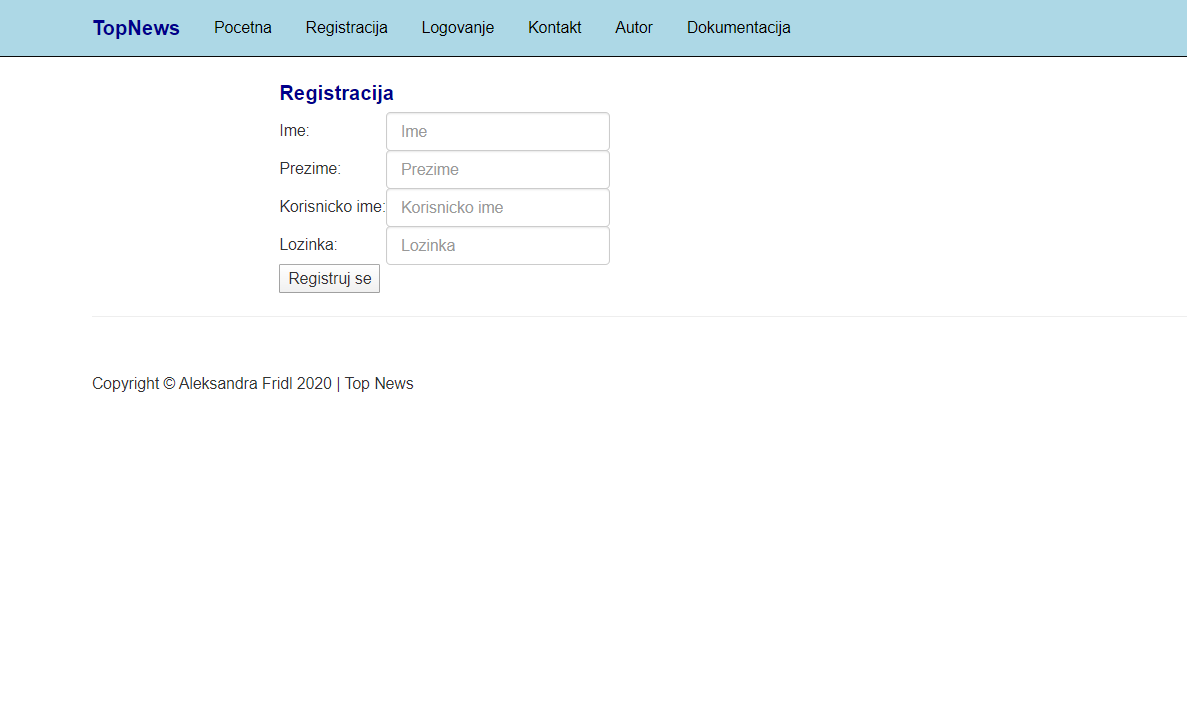
2.2. Mapa sajta

2.3. Slike stranica i opis funkcionalnosti

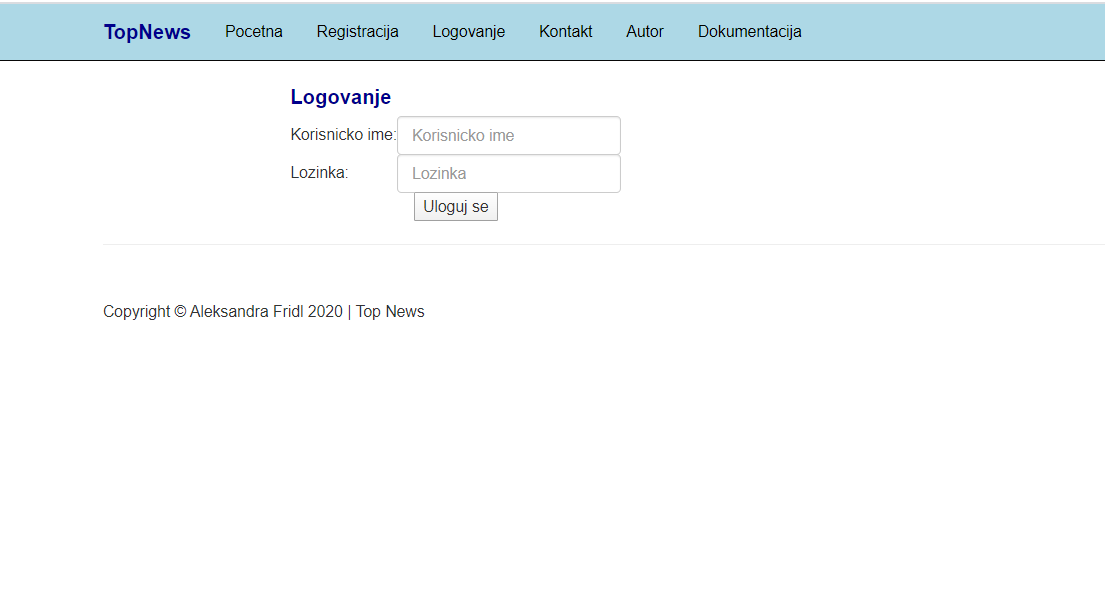
2.3.1. php2Sajt2\_8.03/public/ - Početna – Na ovoj stranici postoji dinamičko ispisivanje kategorija iz baze. Postoji I slajder koji prikazuje slike vesti iz tabele u bazi. Vesti su spakovane u manje divove koji se ponavljaju. Klikom na naslov ili na link Više moguće je dobiti pregled te odabrane vesti. Napravljena je pretraga koja se dešava nad tabelom News iz baze. Moguće je uraditi pretragu tako da se pojavljuju vesti sa zadatim pojmom. Ukoliko ima više od šest vesti po pretrazi uvodi se paginacija radi lepšeg prikaza.



2.3.2. – php2Sajt2\_8.03/public/registracija –Registracija - Na ovoj stranici nalazi se forma za registraciju.

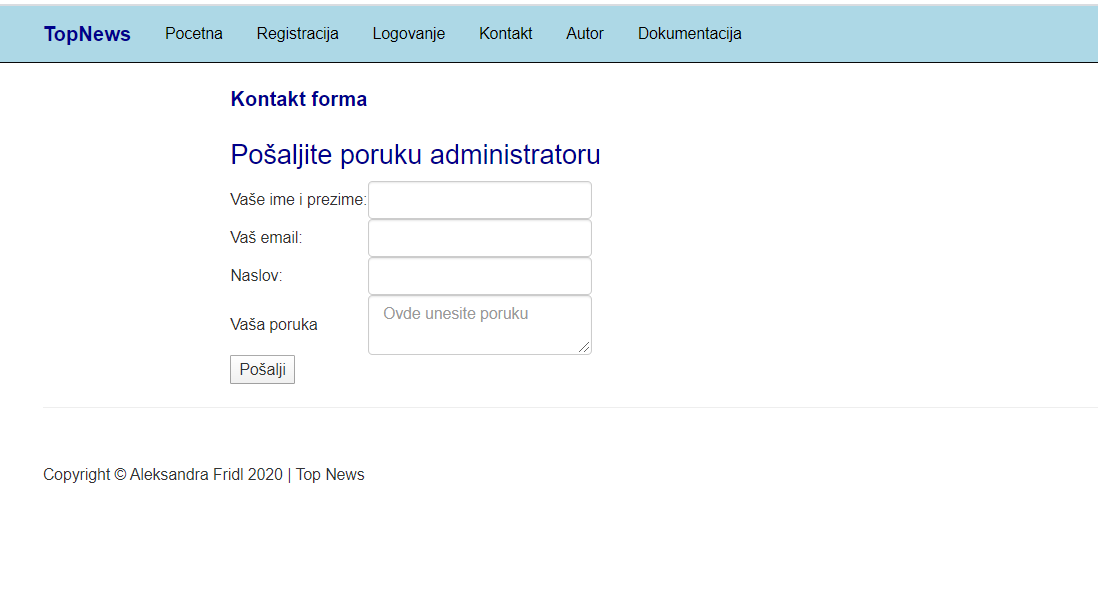


2.3.3. – php2Sajt2\_8.03/public/logovanje – Logovanje – Na ovoj stranici nalazi se forma za logovanje



2.3.4. - php2Sajt2\_8.03/public/kontakt - Kontakt

- Na ovoj stranici je realizovana forma za slanje mejla administratoru. Slanje mejla je realizovano putem PHPMailera.

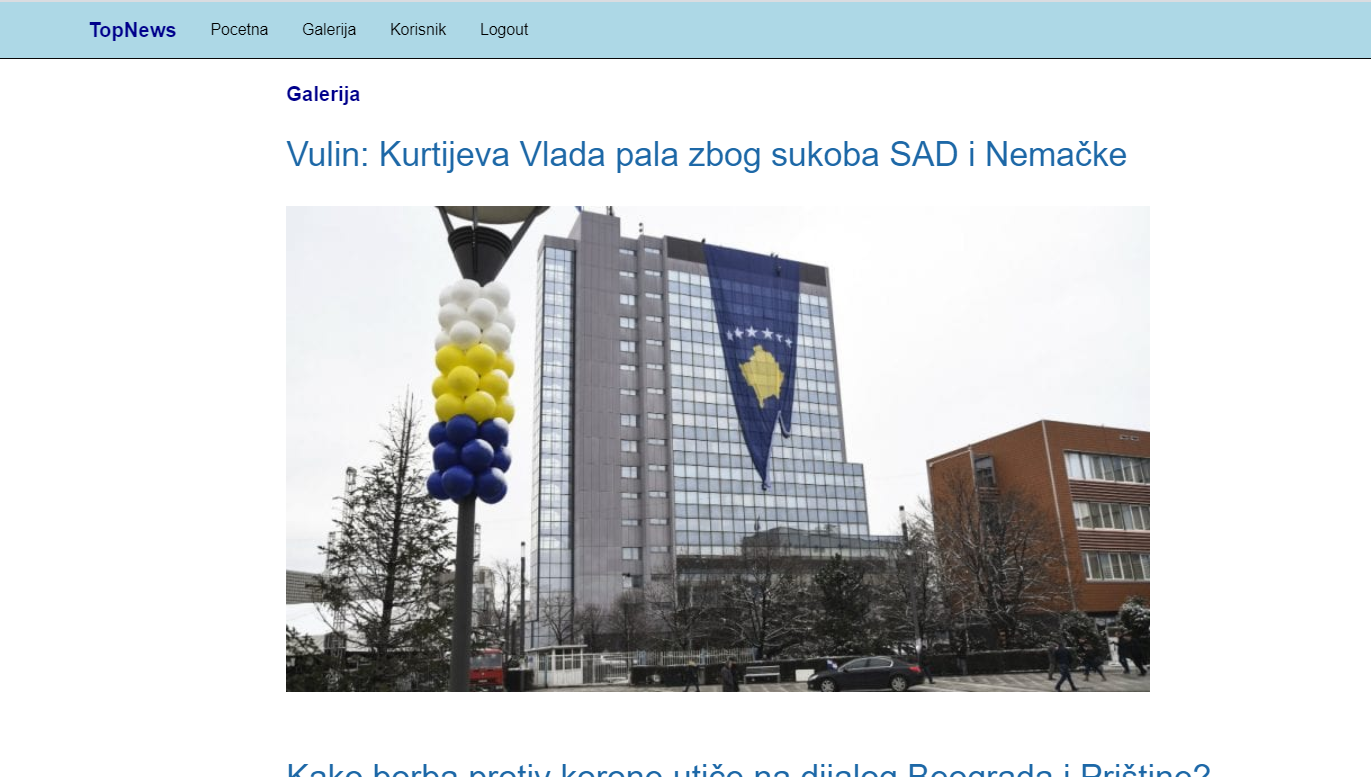


2.3.5. – php2Sajt2\_8.03/public/autor - Autor

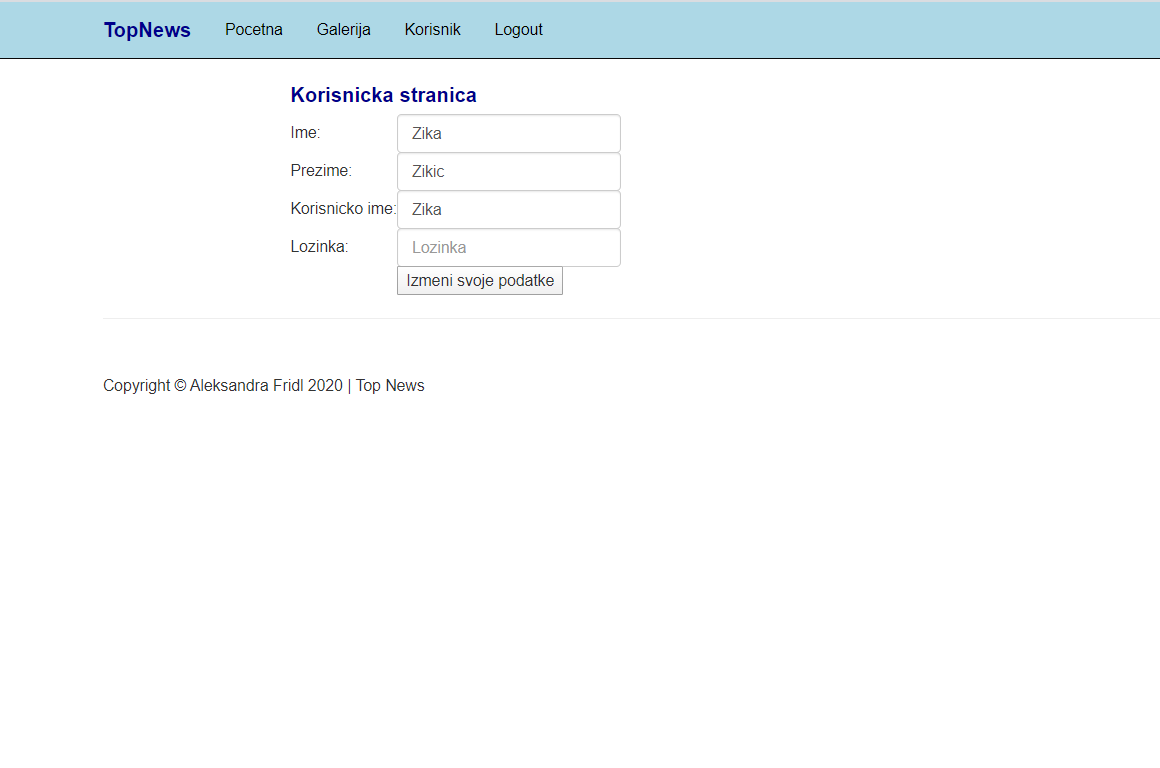
Na ovoj stranici su dinamički iz baze prikazani osnovni podaci o autoru sajta.



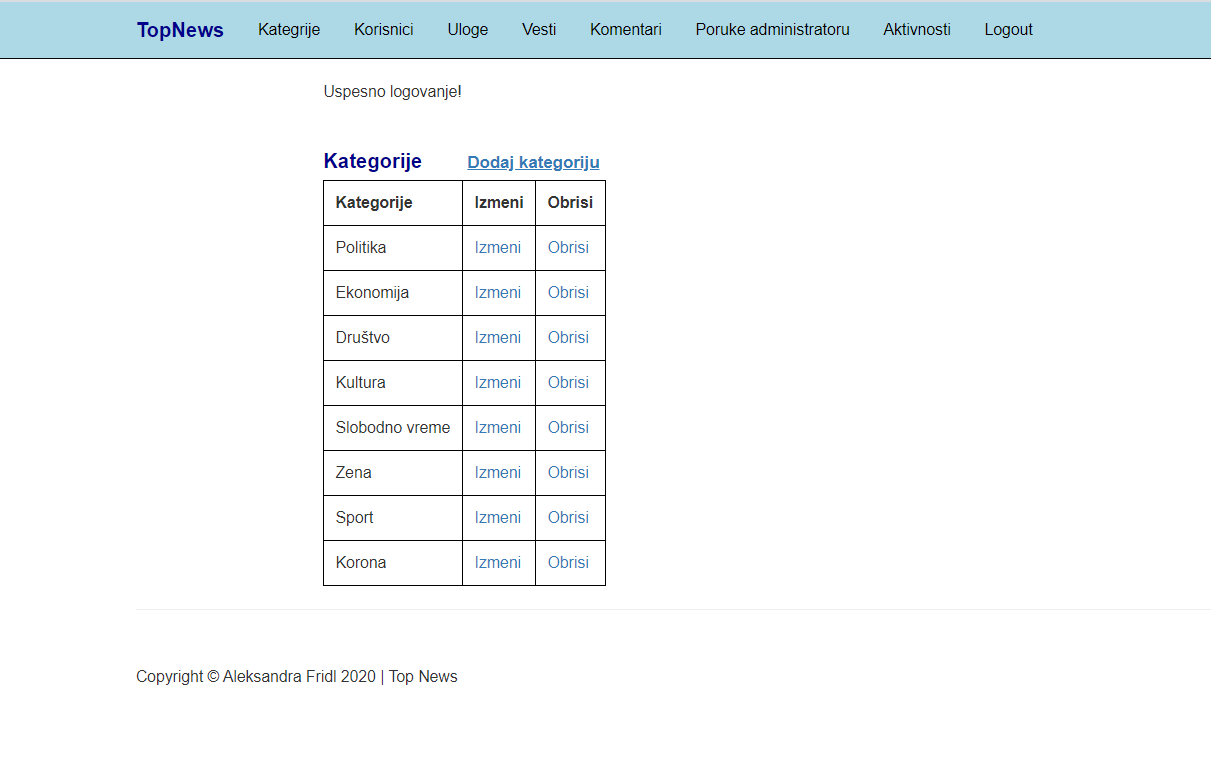
2.3.6. – php2Sajt2\_8.03/public/galerija – Galerija – Na ovoj stranici je prikazana galerija I uradjena je paginacija radi lakšeg pregleda slika.



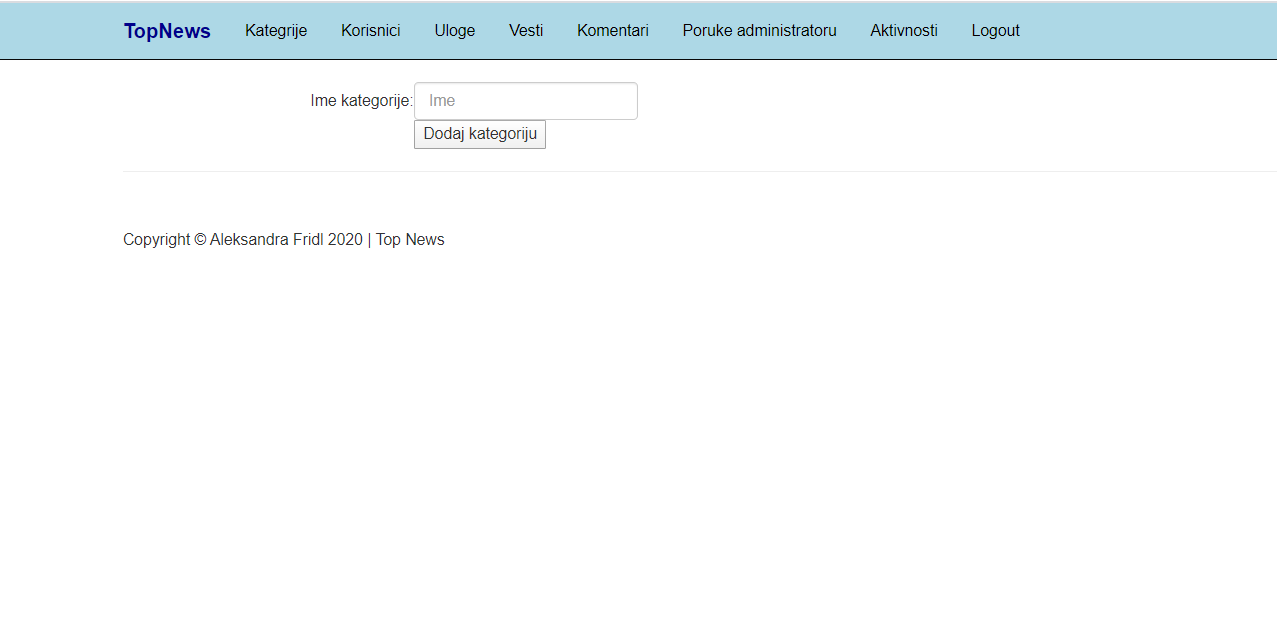
2.3.7. – php2Sajt2\_8.03/public/korisnik/2 – Korisnik – Na ovoj stranici je prikazana forma u kojoj korisnik može da izmeni svoje podatke za logovanje.



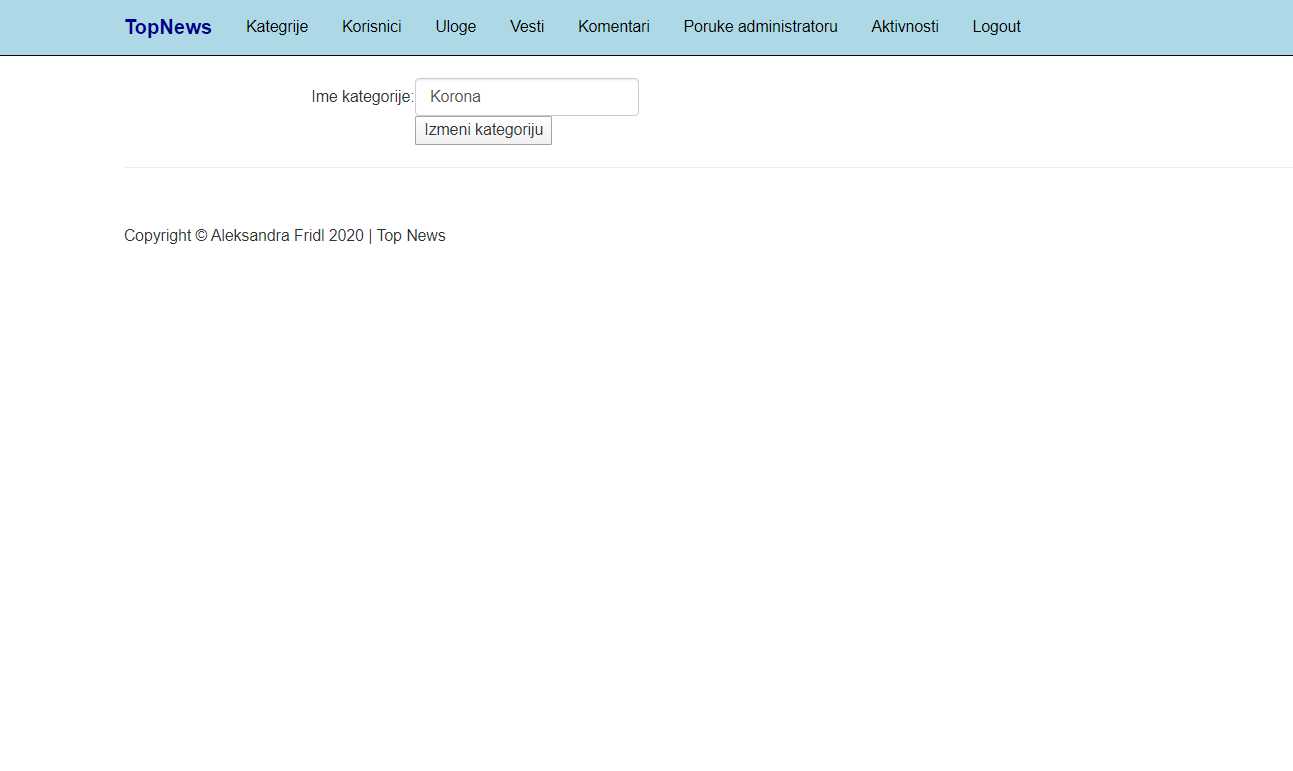
2.3.8. – php2Sajt2\_8.03/public/adminCategory – Kategorije – Na ovoj stranici je urađen prikaz svih kategorija iz tabele categories iz baze. Na njoj je moguće uraditi dodavanje nove kategorije, izmena postojeće i brisanje postojeće kategorije.



2.3.9. – php2Sajt2\_8.03/public/adminCategory\_insert – Dodaj kategoriju – Na ovoj stranici je moguće uneti novu kategoriju. Ovo je realizovano putem AJAXa.

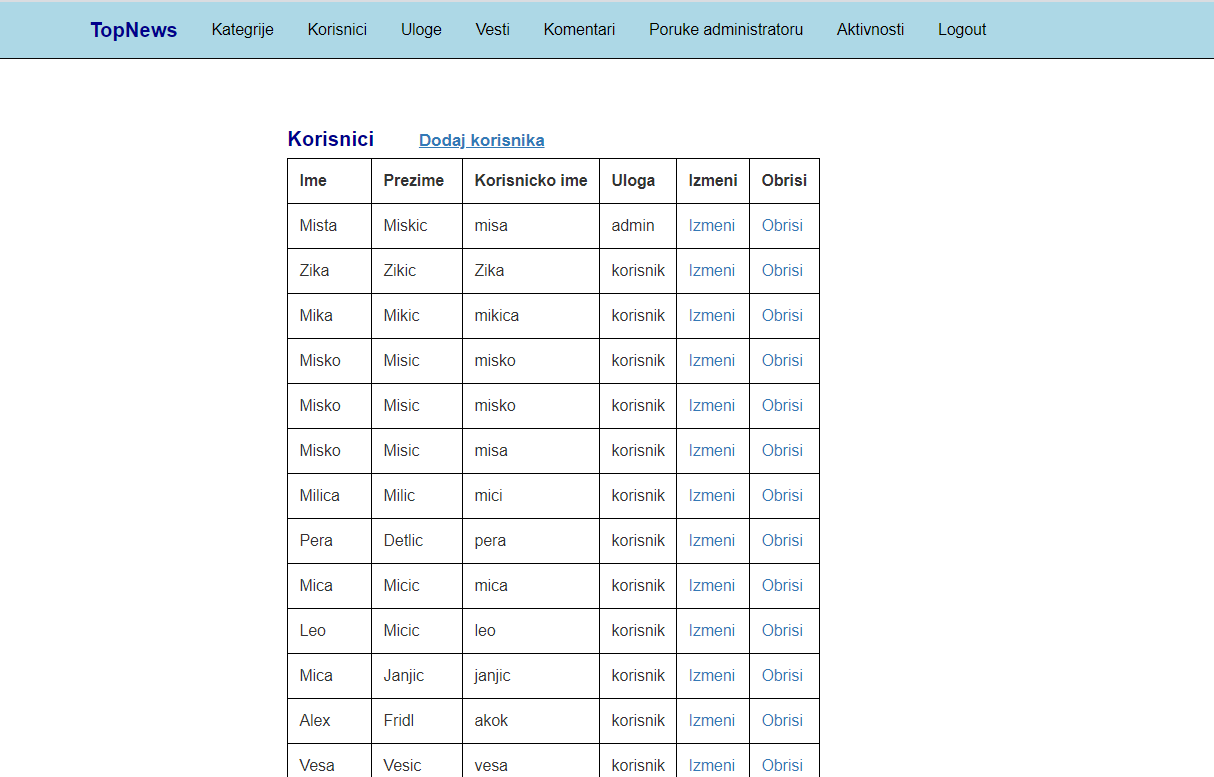


2.3.10 - php2Sajt2\_8.03/public/adminCategory\_update/19 – Izmeni kategoriju – Na ovoj stranici je moguće uraditi izmenu postojeće kategorije.

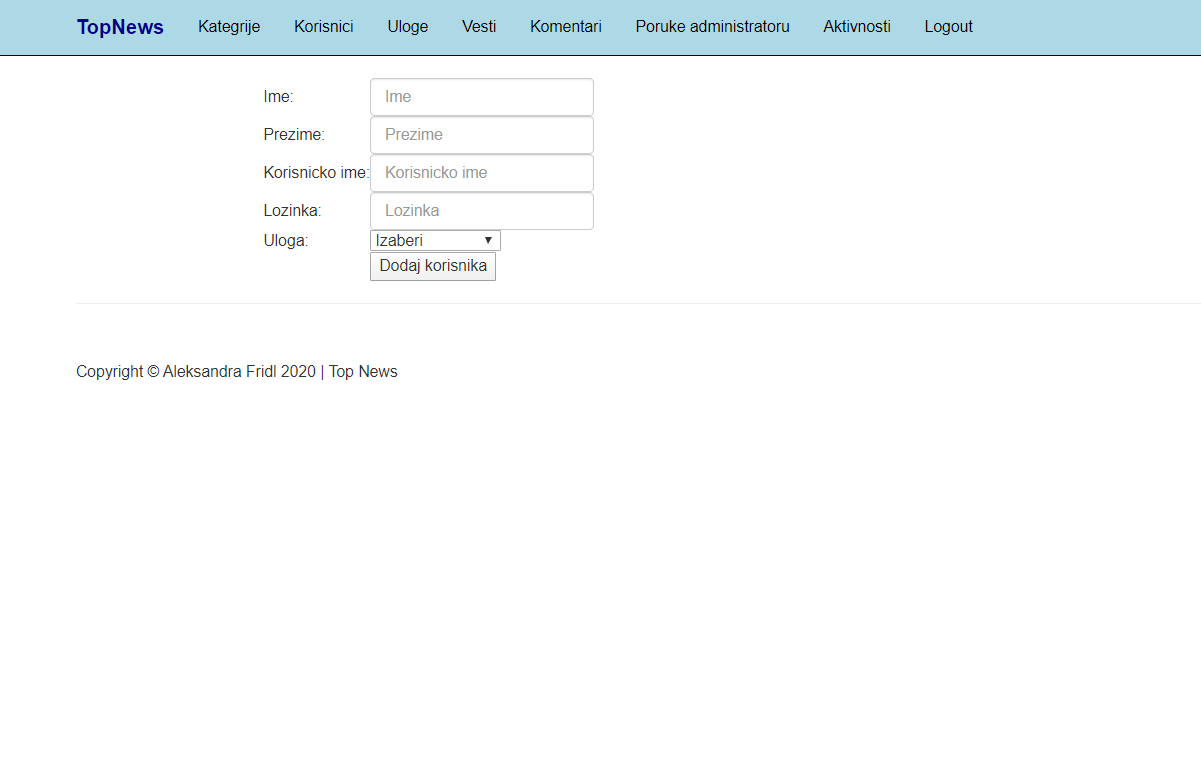


2.3.11- php2Sajt2\_8.03/public/adminCategory\_delete/12 -Delete kategoriju – Na ovom linku se dešava brisanje odabrane kategorije.

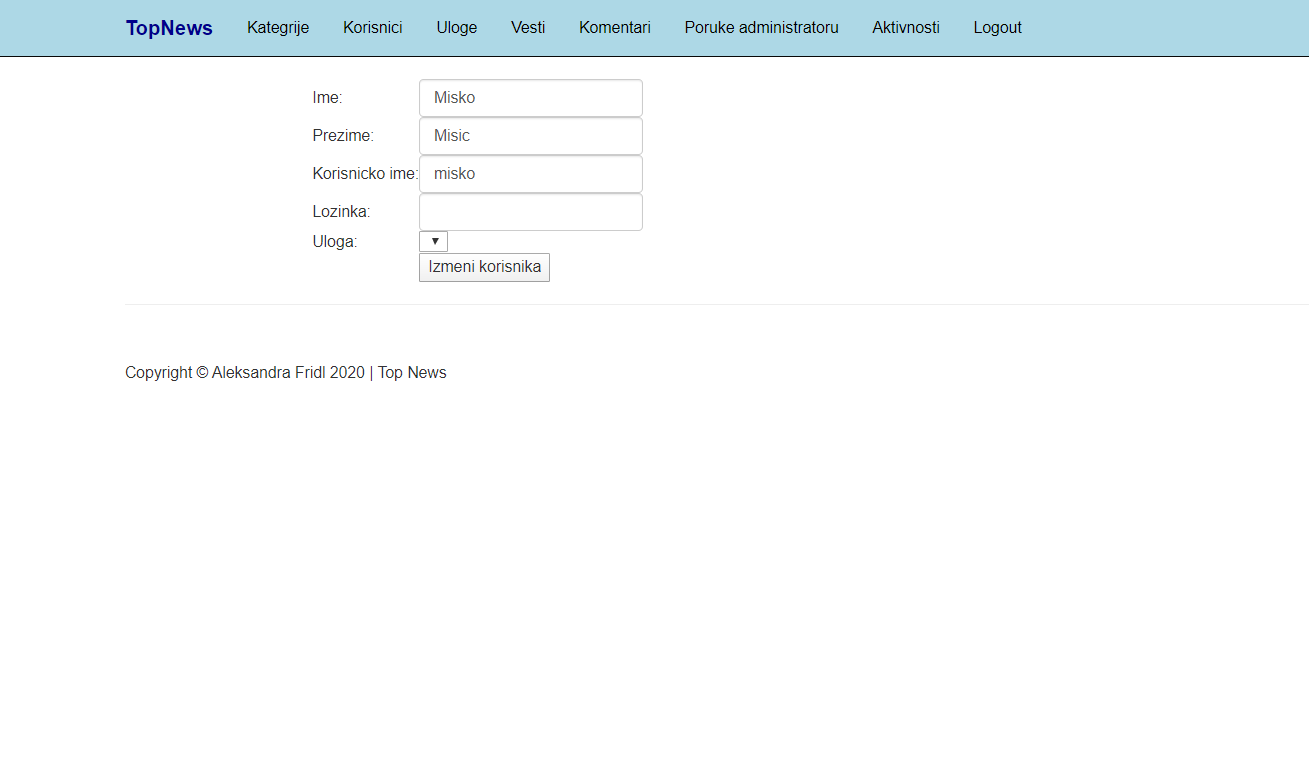
2.3.12 - php2Sajt2\_8.03/public/adminUsers – Korisnici – Ovde je tabelarno dat prikaz svoh korisnika koji su registrovani I koji mogu obavljati logovanje na sajt.



2.3.13 – php2Sajt2\_8.03/public/adminUsers\_insert – Dodaj korisnika – Ovde je moguće dodati novog korisnika.

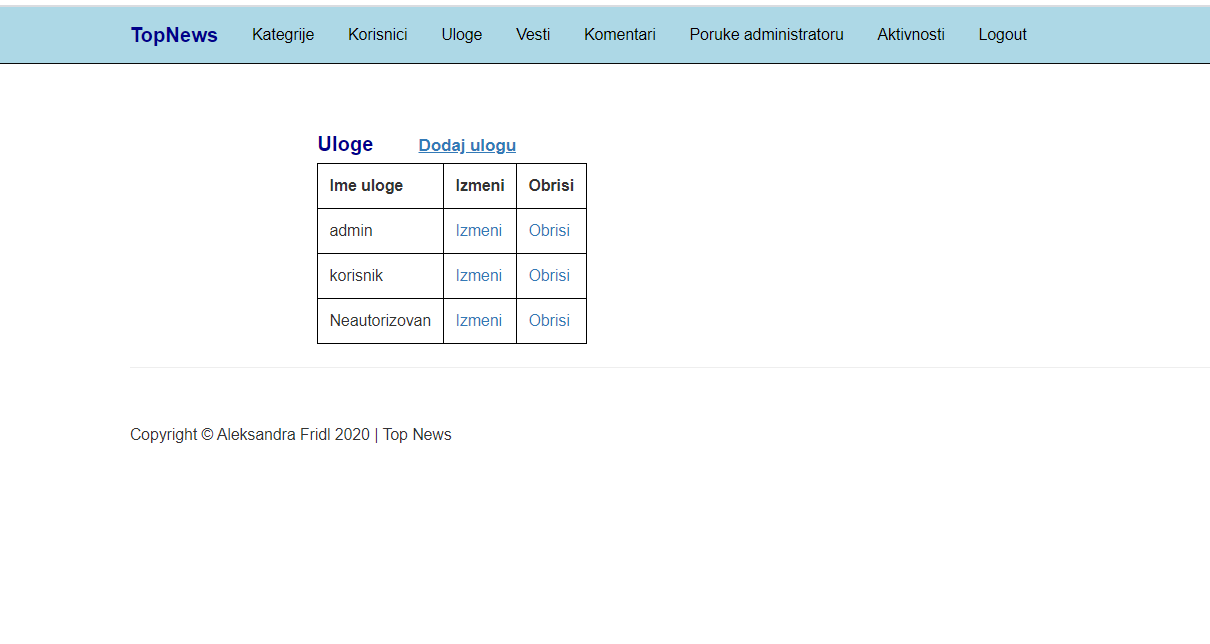


2.3.14 – php2Sajt2\_8.03/public/adminUsers\_update/10 – Izmena korisnika – Na ovoj stranici data je forma za izmenu odabranog korisnika.

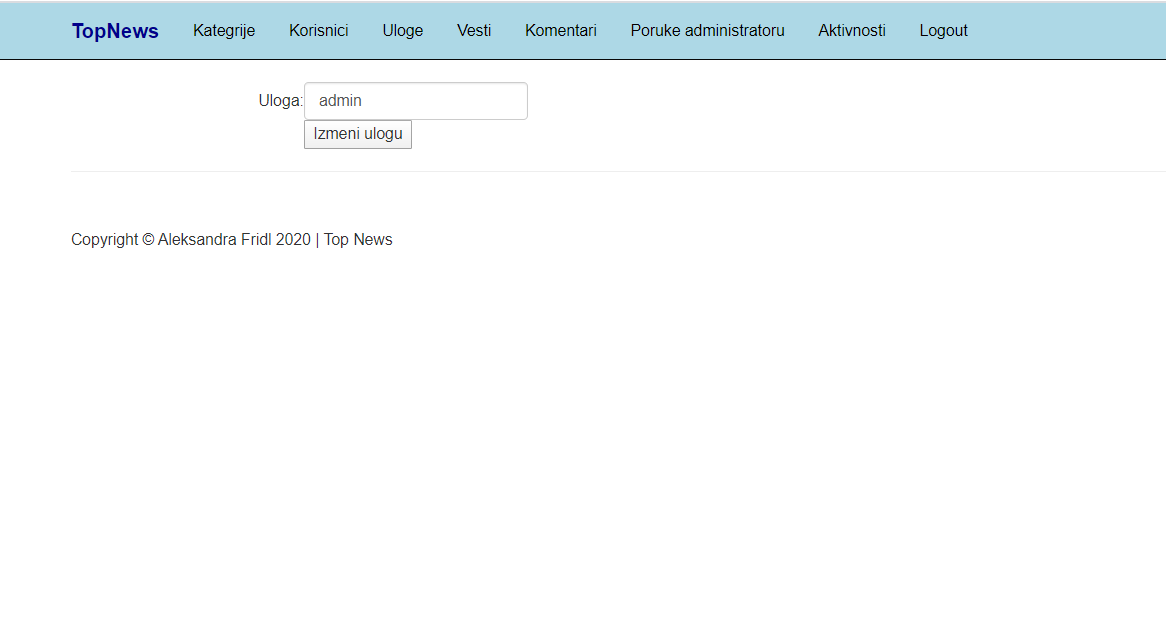


2.3.15 – php2Sajt2\_8.03/public/adminUser\_delete - Delete korisnika – Na ovom linku moguće je izbrisati korisnika.

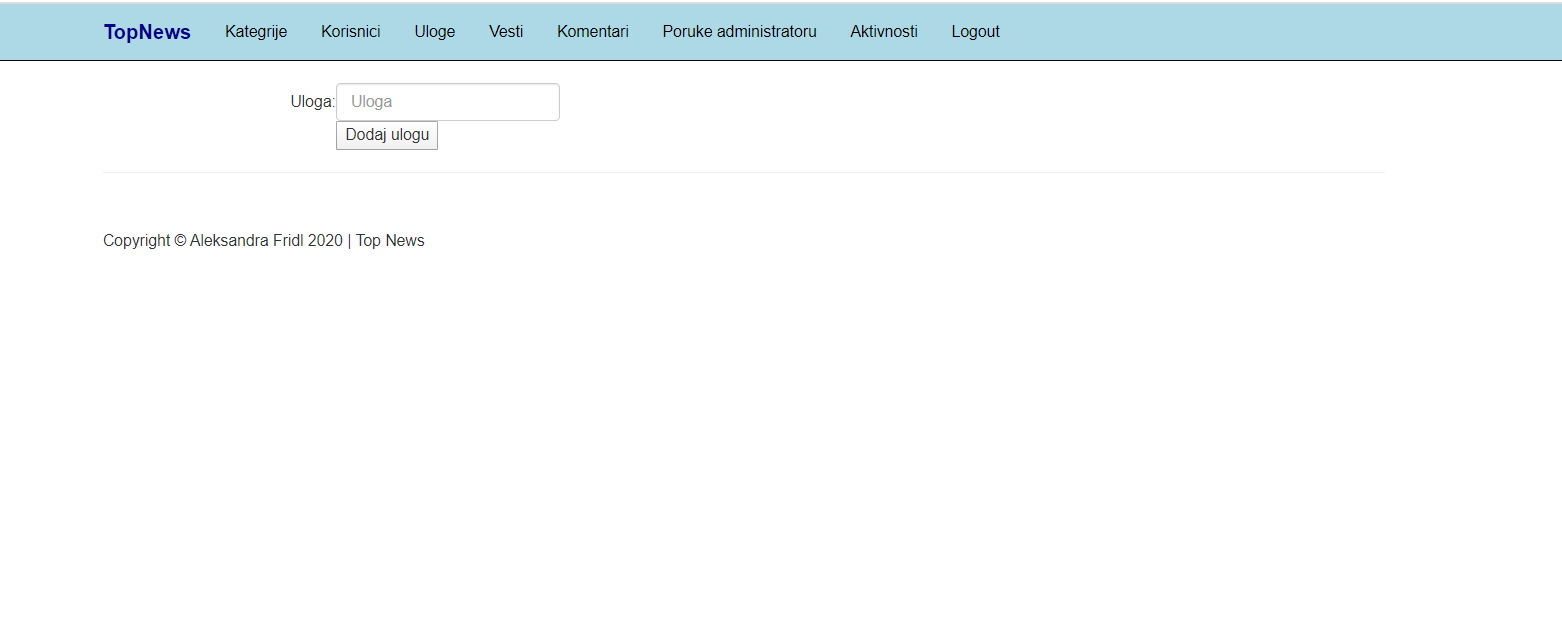
2.3.16 – php2Sajt2\_8.03/public/adminRoles – Uloge – Ovde je dat tabelarni prikaz svih uloga koje postoje u bazi u tabeli Roles. Ovde su i linkovi na kojima je moguće uneti izmenu imena postojeće kategorije kao i brisanje kategorije.



2.3.17 – php2Sajt2\_8.03/public/adminRoles\_update/2 – Izmeni uloge – Na ovoj stranici moguće je izvršiti izmenu odabrane uloge.

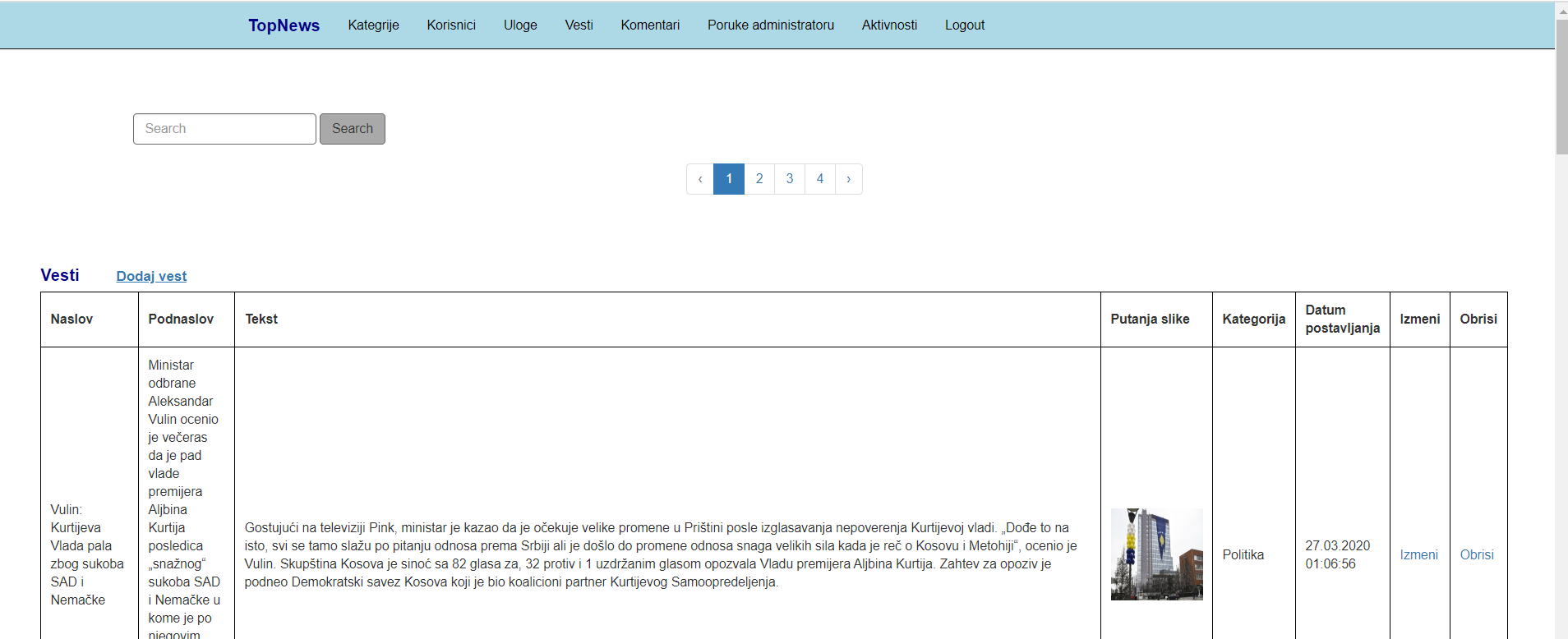


2.3.18– php2Sajt2\_8.03/public/adminRoles\_update/2 – Dodaj ulogu – Na ovoj stranici moguće je dodati novu ulogu.

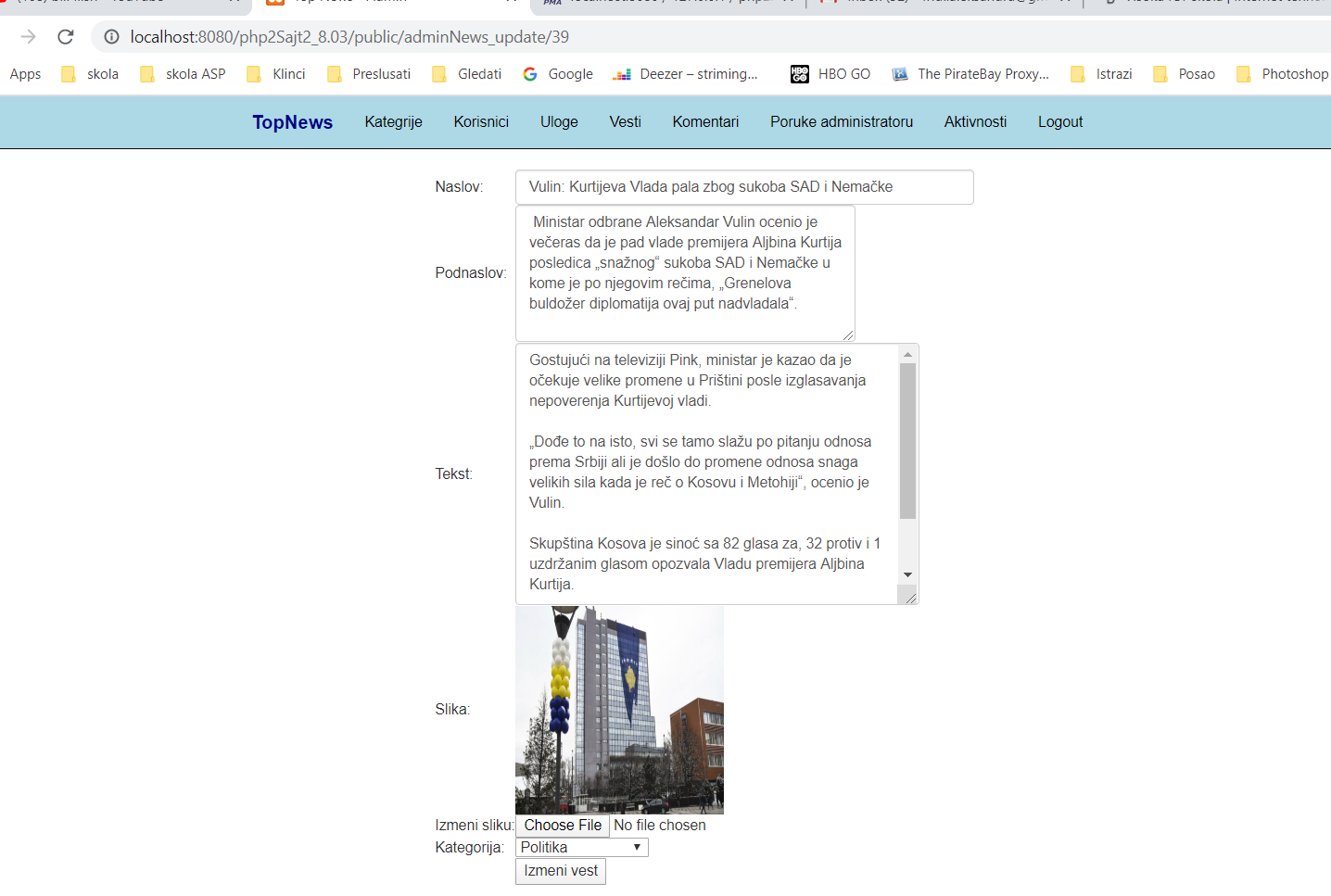


2.3.19 – php2Sajt2\_8.03/public/adminRoles\_delete -Delete uloge – Na ovom linku moguće je izrbrisati odabranu ulogu.

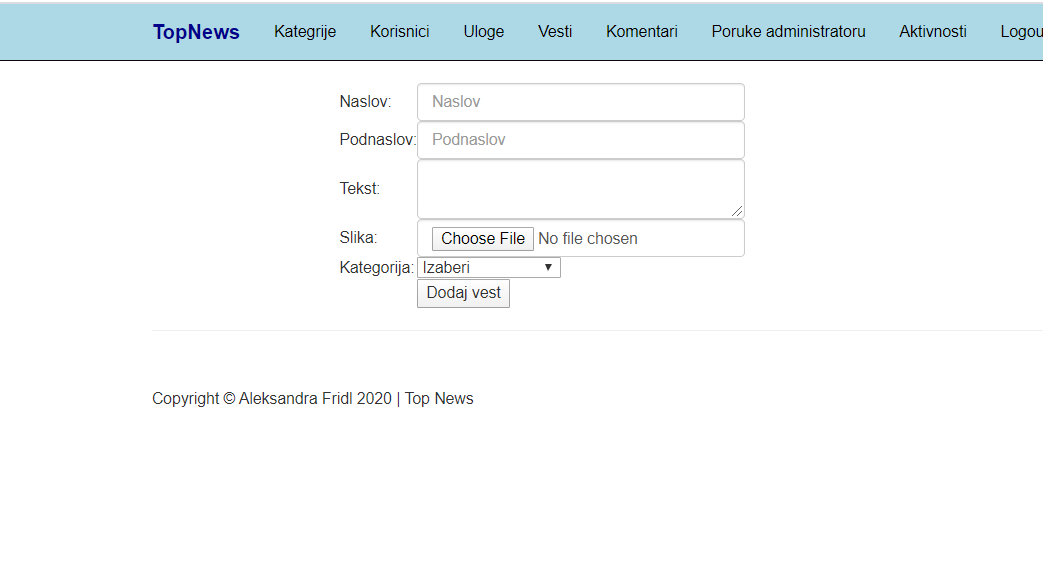
2.3.20 - php2Sajt2\_8.03/public/adminNews – Vesti – Na ovom linku moguće je u tabelarnom prikazu pregledati sve vesti koji se nalaze u tabeli News u bazi podataka. Takođe urađena je I paginacija tabele kako bi se daobio lepši I funkcionalniji prikaz. Na stranici se nalazi I search box tj pretraga u kojoj je moguće uraditit pretragu po zadatom pojmu. Na stranici se nalazi I link za dodavanje nove vesti, kao I za izmenu I brisanje odabrane vesti.



2.3.21 - php2Sajt2\_8.03/public/adminNews\_update/22 – Izmeni vest – Na ovoj stranici je moguće obaviti izmenu odabrane vesti.

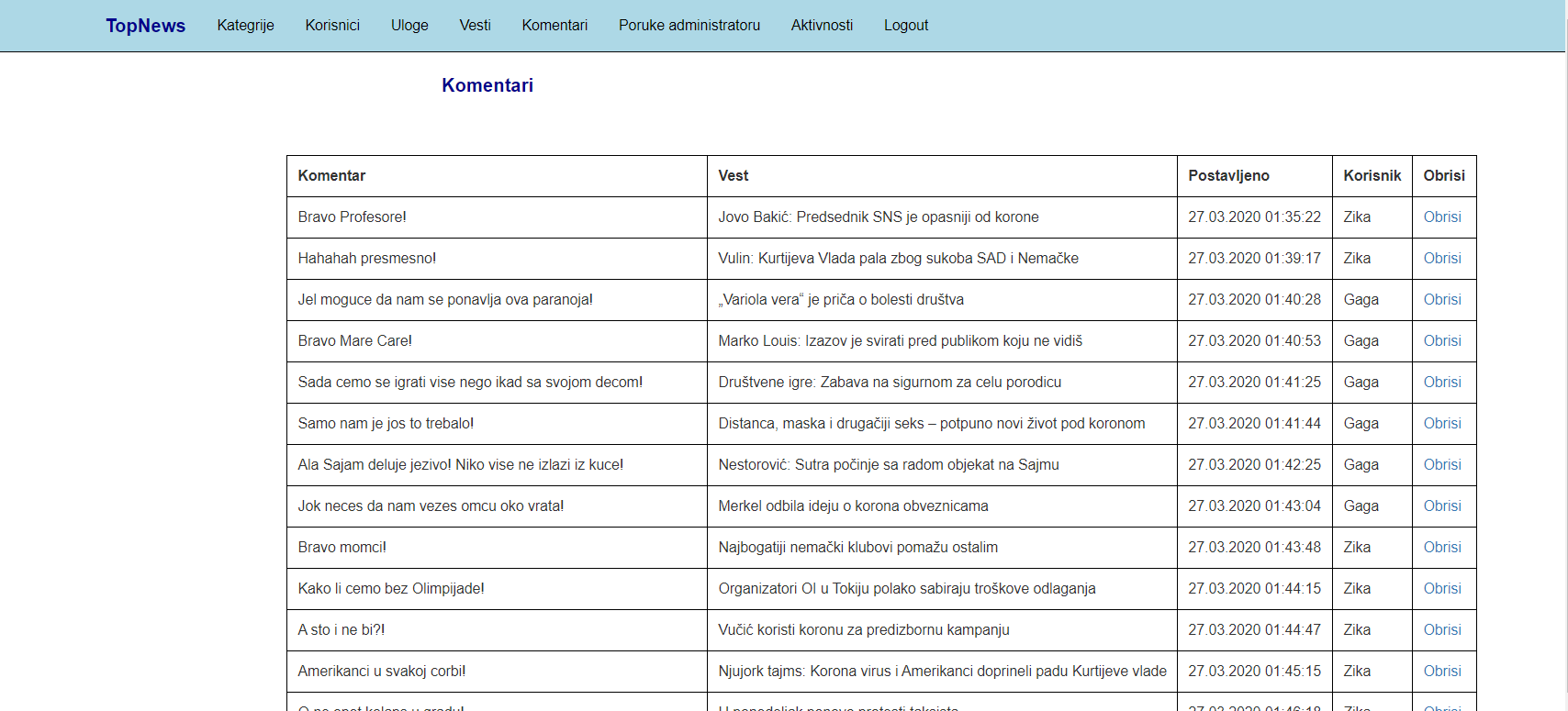


2.3.22 - php2Sajt2\_8.03/public/adminNews\_insert – Dodaj vest – Na ovom linku moguće je uraditi dodavanje nove vesti.



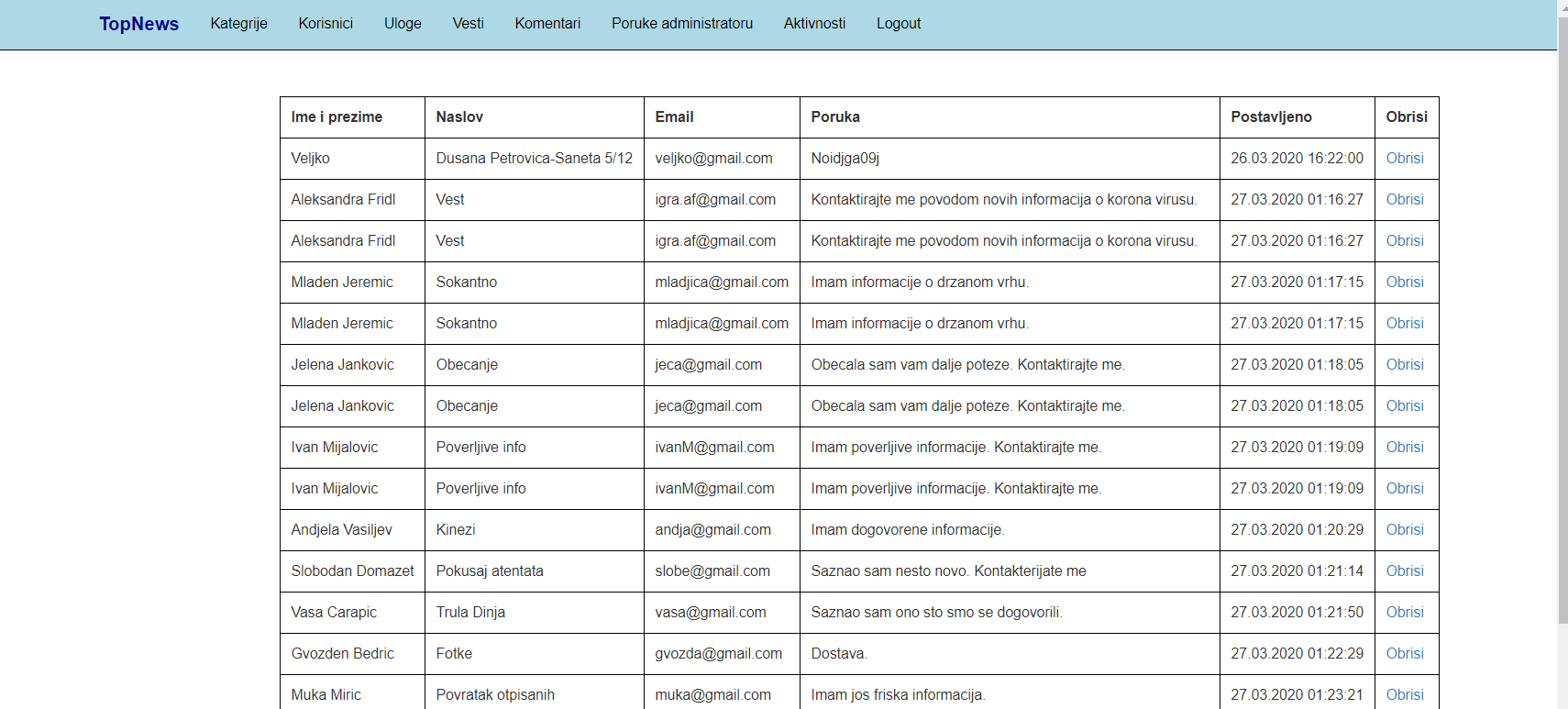
2.3.23 – php2Sajt2\_8.03/public/adminNews\_delete -Delete vesti – Na ovom linku moguće je uraditi brisanje odabrane vesti.

2.3.24 - php2Sajt2\_8.03/public/adminComments – Komentari – Na ovoj stranici nalaze se u tabelarnom prikazu svi komentari koje je korisnik uneo ispod neke vesti kada je bio ulogovan.Ovde se nalazi I mogućnost za brisanje komentara iz tabele.



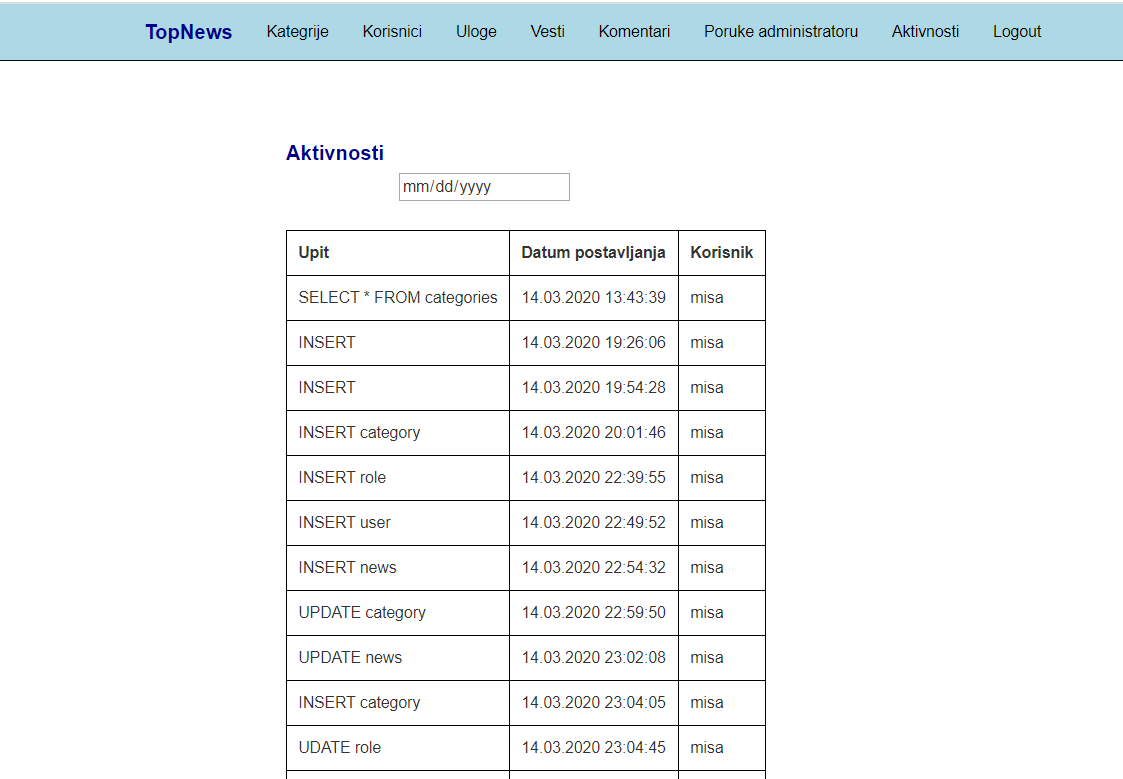
2.3.25 – php2Sajt2\_8.03/public/adminComments\_delete - Delete komentar – Na ovom linku moguće je izvršiti brisanje odabranog komentara.

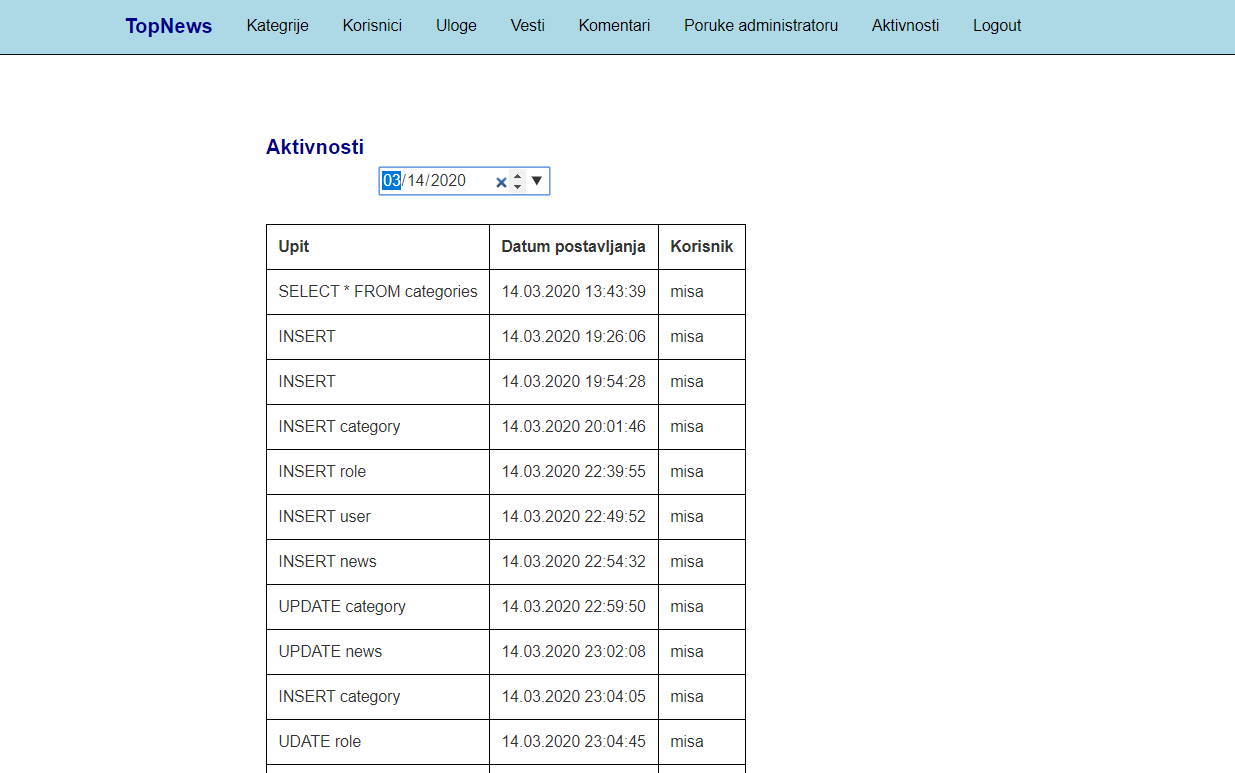
2.3.26 - php2Sajt2\_8.03/public/admin\_contactForm – Poruke administrator – U tabelarnom prikazu date su sve poruke koje je administrator primio od korisnik koje je poslao putem Kontakt forme. Ovde se nalazi I mogućnost brisanja odabrane poruke.



2.3.27 – php2Sajt2\_8.03/public/adminContactForm\_delete /28- Delete poruka – Na ovom linku moguće je brisanje poruke koja je poslata administratoru

2.3.28 - php2Sajt2\_8.03/public/activity – Aktivnosti – Ovde se nalazi tabelarni prikaz aktivnosti koje se upisuju u tabelu prilikom aktivnosti kao što su registracija, logovanje, insert, update, delete. Takođe moguće je uraditi filtriranje po datumu I na osnovu filtriranja dobiti opet tabelarni prikaz za odabrani dan.

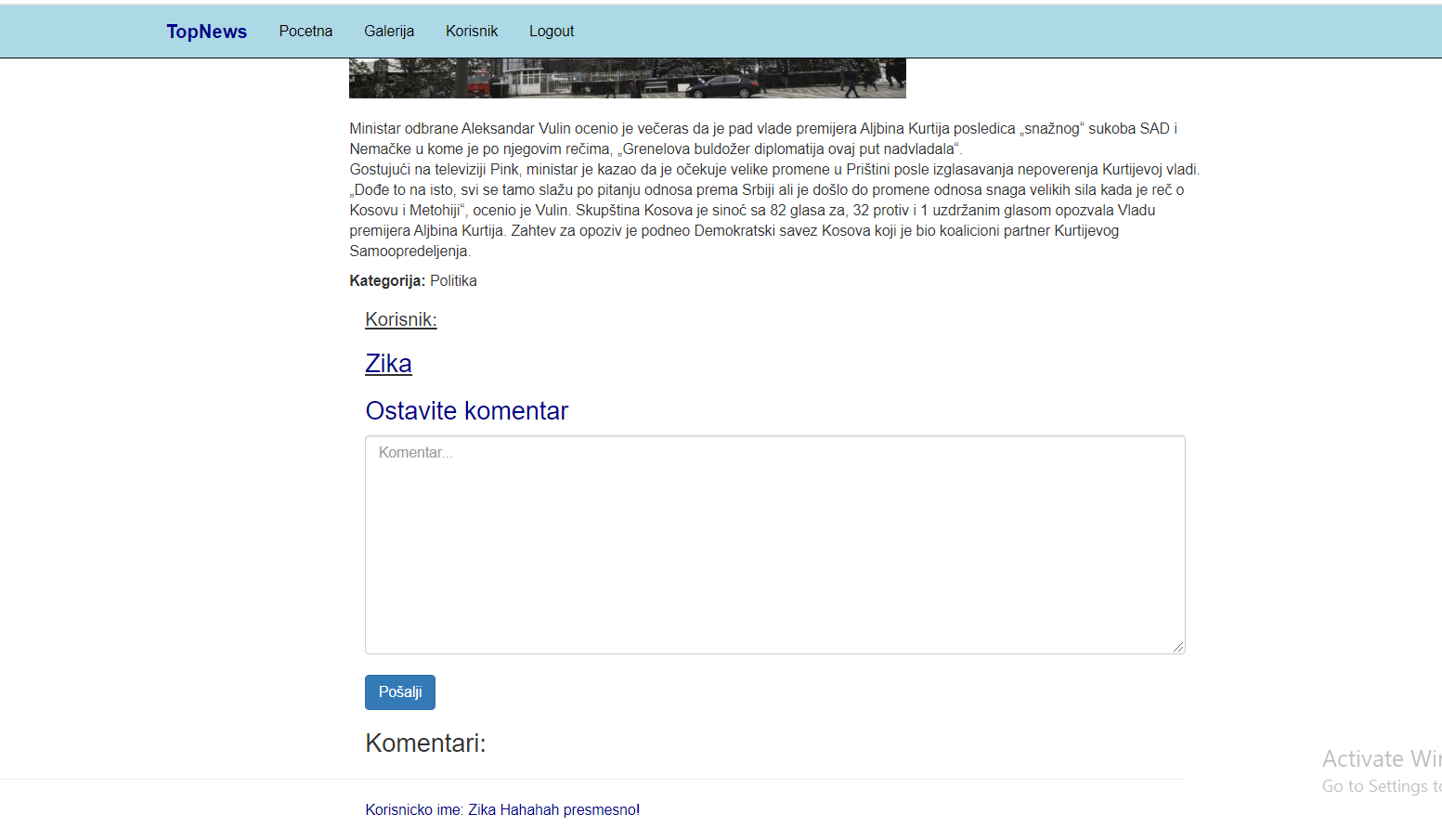




2.3.29 - php2Sajt2\_8.03/public/logout – Logout – Na ovom linku moguće je izlogovati se ukoliko ste ulogovani kao korisnik ili administrator.

2.3.30 - php2Sajt2\_8.03/public/news/39 – Na ovom linku moguće je pregledati jednu vest koja je odabrana na početnoj strani. Na njoj je, ako je korisnik ulogovan, moguće ostaviti komentar. Ukoliko nije korisnik ulogovan ne može da ostavi komentar ali ispod vesti su prikazane poruke koje su postavljene na tu vest.





1. **KODOVI**

**3.1. RUTE**

**Routes->web.php**

<?php  
  
use \App\Http\Controllers\FrontendController;  
use \App\Http\Controllers\AuthController;  
use \App\Http\Controllers\KorisnikController;  
use \App\Http\Controllers\BackendController;  
use \App\Http\Controllers\CategoryController;  
use \App\Http\Controllers\NewsController;  
use \App\Http\Controllers\UserController;  
use \App\Http\Controllers\RoleController;  
use \App\Http\Controllers\CommentController;  
  
  
 Route::*get*('/', [FrontendController::class, 'index'])->name('index');  
  
 Route::*get*('/category/{id}', [FrontendController::class, 'category'])->name('category');  
  
 Route::*get*('/news/{id?}', [FrontendController::class, 'showOne'])->name('showOne');  
  
 Route::*get*('/galerija', [FrontendController::class, 'galerija'])->name('galerija');  
  
 Route::*get*('/kontakt', [FrontendController::class, 'kontakt'])->name('kontakt');  
 Route::*post*('/kontakt', [FrontendController::class, 'doKontakt'])->name('doKontakt');  
  
 Route::*get*('/registracija', [AuthController::class, 'registracija'])->name('regForma');  
 Route::*post*('/registracija', [AuthController::class, 'doReg'])->name('doReg');  
  
 Route::*get*('/logovanje', [AuthController::class, 'logovanje'])->name('loginForma');  
 Route::*post*('/doLogovanje', [AuthController::class, 'doLog'])->name('doLogin');  
 Route::*get*('/logout', [AuthController::class, 'logout'])->name('logout');  
  
 Route::*get*('/author', [FrontendController::class, 'showAuthor'])->name('author');  
  
  
//KorisnikMiddleware  
Route::*group*(['middleware'=>'korisnik'],function(){  
 Route::*get*('/galerija',[FrontendController::class,'galerija'])->name('galerija');  
 Route::*get*('/korisnik/{id\_user}',[KorisnikController::class,'showKorisnik'])->name('showKorisnik');  
 Route::*post*('/korisnik/{id\_user}',[KorisnikController::class,'changeUserData'])->name('changeUserData');  
  
 Route::*post*('/ajax/insert\_comment',[KorisnikController::class,'insert\_comment'])->name('insert\_comment');  
});  
  
//AdminMiddleware  
Route :: *group*(['middleware'=>'admin'],function (){  
  
 Route::*get*('/adminCategory',[CategoryController::class,'adminCategory'])->name('adminCategory'); //CATEGORY  
 Route::*get*('/adminNews',[NewsController::class,'adminNews'])->name('adminNews'); //NEWS  
 Route::*get*('/adminUsers',[UserController::class,'adminUsers'])->name('adminUsers'); //USER  
 Route::*get*('/adminRoles',[RoleController::class,'adminRoles'])->name('adminRoles'); //ROLE  
  
 Route::*get*('/adminComments',[CommentController::class,'adminComments'])->name('adminComments'); //COMMENT  
 Route::*get*('/admin\_contactForm',[BackendController::class,'admin\_contactForm'])->name('admin\_contactForm'); //BACKEND  
 Route::*get*('/activity',[BackendController::class,'activity'])->name('activity'); //BACKEND  
 Route::*get*('/admin\_contactForm',[BackendController::class,'admin\_contactForm'])->name('admin\_contactForm'); //BACKEND  
  
 //insert  
 Route::*get*('/adminCategory\_insert',[CategoryController::class,'adminCategory\_insert'])->name('adminCategory\_insert'); //CATEGORY  
  
 Route::*get*('/adminNews\_insert',[NewsController::class,'adminNews\_insert'])->name('adminNews\_insert'); //NEWS  
 Route::*post*('/adminNews\_insert',[NewsController::class,'doAdminNews\_insert'])->name('doAdminNews\_insert'); //NEWS  
  
 Route::*get*('/adminUsers\_insert',[UserController::class,'adminUsers\_insert'])->name('adminUsers\_insert'); //USER  
 Route::*post*('/adminUsers\_insert',[UserController::class,'doAdminUsers\_insert'])->name('doAdminUsers\_insert'); //USER  
  
 Route::*get*('/adminRoles\_insert',[RoleController::class,'adminRoles\_insert'])->name('adminRoles\_insert'); //ROLE  
 Route::*post*('/adminRoles\_insert',[RoleController::class,'doAdminRoles\_insert'])->name('doAdminRoles\_insert'); //ROLE  
  
 //update  
 Route::*get*('/adminCategory\_update/{id\_category}',[CategoryController::class,'adminCategory\_update'])->name('adminCategory\_update'); //CATEGORY  
 Route::*post*('/adminCategory\_update/{id\_category}',[CategoryController::class,'doAdminCategory\_update'])->name('doAdminCategory\_update'); //CATEGORY  
  
 Route::*get*('/adminUsers\_update/{id\_user}',[UserController::class,'adminUsers\_update'])->name('adminUsers\_update'); //USER  
 Route::*post*('/adminUsers\_update/{id\_user}',[UserController::class,'doAdminUsers\_update'])->name('doAdminUsers\_update'); //USER  
  
 Route::*get*('/adminRoles\_update/{id\_role}',[RoleController::class,'adminRoles\_update'])->name('adminRoles\_update'); //ROLE  
 Route::*post*('/adminRoles\_update/{id\_role}',[RoleController::class,'doAdminRoles\_update'])->name('doAdminRoles\_update'); //ROLE  
  
 Route::*get*('/adminNews\_update/{id\_news}',[NewsController::class,'adminNews\_update'])->name('adminNews\_update'); //NEWS  
 Route::*post*('/adminNews\_update/{id\_news}',[NewsController::class,'doAdminNews\_update'])->name('doAdminNews\_update'); //NEWS  
  
 //delete  
 Route::*get*('/adminCategory\_delete/{id\_category}',[CategoryController::class,'adminCategory\_delete'])->name('adminCategory\_delete'); //CATEGORY  
 Route::*get*('/adminRoles\_delete/{id\_role}',[RoleController::class,'adminRoles\_delete'])->name('adminRoles\_delete'); //ROLE  
 Route::*get*('/adminUsers\_delete/{id\_user}',[UserController::class,'adminUsers\_delete'])->name('adminUsers\_delete'); //USER  
 Route::*get*('/adminNews\_delete/{id\_news}',[NewsController::class,'adminNews\_delete'])->name('adminNews\_delete'); //NEWS  
 Route::*get*('/adminComments\_delete/{id\_comment}',[CommentController::class,'adminComments\_delete'])->name('adminComments\_delete'); //COMMENT  
 Route::*get*('/adminContactForm\_delete/{id\_contact}',[BackendController::class,'adminContactForm\_delete'])->name('adminContactForm\_delete');  
  
 Route::*post*('/ajax/adminSearch',[BackendController::class,'adminSearch'])->name('adminSearch');  
 Route::*post*('/ajax/adminSortByDate',[BackendController::class,'sortByDate'])->name('sortByDate');  
});  
  
//AJAX  
  
Route::*group*(['prefix'=>'/ajax'],function(){  
 Route::*post*('/adminCategory\_insert',[CategoryController::class,'doAdminCategory\_insert'])->name('doAdminCategory\_insert'); //CATEGORY  
 Route::*post*('/search',[FrontendController::class,'search'])->name('search');  
  
});

**3.2. HTML (resources->views->components)**

**3.2.1. footer.blade.php**

<div class="container">  
  
 <hr>  
  
 <!-- Footer -->  
 <footer>  
 <div class="row">  
 <div class="col-lg-12">  
 <p>Copyright &copy; Aleksandra Fridl 2021 | Top News</p>  
 </div>  
 </div>  
 </footer>  
  
</div>  
<!-- /.container -->  
@section('JavaScript')  
<!-- jQuery -->  
<script src="{{asset('/')}}js/jquery.js"></script>  
  
<!-- Bootstrap Core JavaScript -->  
<script src="{{asset('/')}}js/bootstrap.min.js"></script>  
  
<script language="JavaScript" src="{{asset('/')}}js/script.js"></script>  
<script type="text/javascript">  
 const baseUrl = '{{ route("index") }}';  
 const token = '{{ csrf\_token() }}';  
</script>  
  
@show  
</body>  
  
</html>

**3.2.2. head.blade.php**

<!DOCTYPE html>  
<html>  
  
<head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <meta name="description" content="Top News prikazuje vesti koje su preuzete sa sajta danas.rs">  
 <meta name="author" content="Sajt je realizovan za potrebe predmeta php2 na Visokoj školi strukovnih studija za informacione i komunikacione tehnologije. ">  
 <meta name="keywords" content="vesti, Top News, korona, politika, sport, žena, ekonomija, društvo, slobodno vreme">  
  
 <title>Top News - @yield('title')</title>  
  
 <!-- Bootstrap Core CSS -->  
 <link href="{{asset('/')}}css/bootstrap.min.css" rel="stylesheet">  
  
 <!-- Custom CSS -->  
 <link href="{{asset('/')}}css/shop-homepage.css" rel="stylesheet">  
 <link rel="stylesheet" href="{{asset('/')}}css/style.css"/>  
  
 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>  
</head>

**3.2.3. nav.blade.php**

<body>  
  
<!-- Navigation -->  
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation" style="background:lightblue">  
 <div class="container">  
 <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('index')}}" style="font-weight:bold;color: darkblue;">TopNews</a>  
 </div>  
 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  
 @if(session()->has('user') && session()->get('user')->name\_role == 'admin')  
  
 <ul class="nav navbar-nav" >  
 <!--ZA ADMINA-->  
 <li>  
 <a href="{{route('adminCategory')}}" style="color: black">Kategorije</a>  
 </li>  
 <li>  
 <a href="{{route('adminUsers')}}" style="color: black">Korisnici</a>  
 </li>  
 <li>  
 <a href="{{route('adminRoles')}}" style="color: black">Uloge</a>  
 </li>  
 <li>  
 <a href="{{route('adminNews')}}" style="color: black">Vesti</a>  
 </li>  
 <li>  
 <a href="{{route('adminComments')}}" style="color: black">Komentari</a>  
 </li>  
 <li>  
 <a href="{{route('admin\_contactForm')}}" style="color: black">Poruke administratoru</a>  
 </li>  
 <li>  
 <a href="{{route('activity')}}" style="color: black">Aktivnosti</a>  
 </li>  
 <li>  
 <a href="{{route('logout')}}" style="color: black">Logout</a>  
 </li>  
 </ul>  
 @elseif (session()->has('user') && session()->get('user')->name\_role == 'korisnik')  
  
 <ul class="nav navbar-nav" >  
 <li>  
 <a href="{{route('index')}}" style="color: black">Pocetna</a>  
 </li>  
 <li>  
 <a href="{{route('galerija')}}" style="color: black">Galerija</a>  
 </li>  
 <li>  
 <a href="{{route('showKorisnik',['id\_user'=>session()->get('user')->id\_user])}}" style="color: black">Korisnik</a>  
 </li>  
 <li>  
 <a href="{{route('logout')}}" style="color: black">Logout</a>  
 </li>  
 </ul>  
 @else  
 <ul class="nav navbar-nav" >  
 <li>  
 <a href="{{route('index')}}" style="color: black">Pocetna</a>  
 </li>  
 <li>  
 <a href="{{route('regForma')}}" style="color: black">Registracija</a>  
 </li>  
 <li>  
 <a href="{{route('loginForma')}}" style="color: black">Logovanje</a>  
 </li>  
  
 <li>  
 <a href="{{route('kontakt')}}" style="color: black">Kontakt</a>  
 </li>  
 <li>  
 <a href="{{route('author')}}" style="color: black">Autor</a>  
 </li>  
 <li>  
 <a href="{{asset('/')}}js/DokumentacijaTopNews.pdf" style="color: black">Dokumentacija</a>  
 </li>  
 </ul>  
 @endif  
 </div>  
 </div>  
</nav>

**3.2.4. sidebar.blade.php**

@section('sidebar')  
  
 <div class="container">  
 <div class="row">  
  
 <div class="col-md-3" style="">  
 <p class="lead" style="color:darkblue;font-weight: bolder;">TopNews</p>  
 <div class="list-group" style="width:100px;height: 100%;">  
  
  
 <nav class="navbar navbar-light bg-light" >  
 <ul class="nav" style="list-style:none;font-size:20px;">  
  
 @isset($kategorije)  
 @foreach($kategorije as $kategorija)  
 <li ><a href="{{route('category',['id\_category'=>$kategorija->id\_category])}}" class="nav-brand" style="display: block;width:120px;color:#000;">{{$kategorija->name\_category}}</a></li>  
  
 @endforeach  
 @endif  
 </ul>  
 </nav>  
 </div>  
@endsection

**3.3. HTML (resources->views->layout)**

**3.3.1. index.blade.php**

@include('components.head')  
@include('components.nav')  
@yield('sidebar')  
@yield('search')  
@yield('content')  
@include('components.footer')

**3.4. HTML (resources->views->pages)**

**3.4.1. admin\_contactForm.blade.php**

@extends('layout.index')  
  
@section('title')  
Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right" >  
 <label class="naslov\_news\_activity">Poruke administratoru  
 </label>  
 <table style="border:1px solid #000;margin-top: 25px;/\*margin-left: 350px;\*/">  
 <tr class="admin">  
 <th class="admin">Ime i prezime</th>  
 <th class="admin">Naslov</th>  
 <th class="admin">Email</th>  
 <th class="admin">Poruka</th>  
 <th class="admin">Postavljeno</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($contact)  
 @foreach($contact as $c)  
 <tr class="admin">  
 <td class="admin">{{$c->first\_last\_name}}</td>  
 <td class="admin">{{$c->title}}</td>  
 <td class="admin">{{$c->email}}</td>  
 <td class="admin">{{$c->text}}</td>  
 <td class="admin">{{date("d.m.Y H:i:s",$c->posted)}}</td>  
 <td class="admin"><a href="{{route('adminContactForm\_delete',['id\_contact'=>$c->id\_contact])}}">Obrisi</a></td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
@endsection

**3.4.2. adminActivity.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 <div class="table">  
 <form method="POST" action="{{route('sortByDate')}}">  
 <table class="move\_right1">  
 <label class="naslov\_news\_activity1">Aktivnosti  
 </label>  
  
 <tr>  
 <td><input type="date" name="sortByDate" id="sortByDate"/></td>  
 </tr>  
 </table>  
 </form>  
 <div id="ispis">  
 <table style="border:1px solid #000;margin-top: 25px;margin-left: 350px;">  
 <tr class="admin">  
 <th class="admin">Upit</th>  
 <th class="admin">Datum postavljanja</th>  
 <th class="admin">Korisnik</th>  
 </tr>  
 @isset($activity)  
 @foreach($activity as $a)  
 <tr class="admin">  
 <td class="admin">{{$a->query}}</td>  
 <td class="admin">{{date("d.m.Y H:i:s",$a->time)}}</td>  
 <td class="admin">{{$a->user\_name}}</td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
 </div>  
@endsection

**3.4.3. adminCategory.blade.php**

@extends('layout.index')  
  
@section('title')  
Admin  
@endsection  
  
@section('content')  
 <div class="move\_right1">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
</br>  
</br>  
<div class="move\_right1">  
 <table class="admin">  
  
 <label class="naslov">Kategorije  
 <a class="dodaj" href="{{route('adminCategory\_insert')}}">Dodaj kategoriju</a>  
 </label>  
 <div id="ispis">  
 <div>  
 <tr class="admin">  
 <th class="admin">Kategorije</th>  
 <th class="admin">Izmeni</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($kategorija)  
 @foreach($kategorija as $k)  
 <tr class="admin">  
 <td class="admin">{{$k->name\_category}}</td>  
 <td class="admin"><a href="{{route('adminCategory\_update',['id\_category'=>$k->id\_category])}}">Izmeni</a></td>  
 <td class="admin">  
 <a href="{{route('adminCategory\_delete',['id\_category'=>$k->id\_category])}}" class="obrisi" data-id="{{$k->id\_category}}">Obrisi</a>  
 </td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
 </table>  
</div>  
@endsection

**3.4.4. adminCategory\_insert.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <div id="ispis"></div>  
 <label class="naslov">Dodati novu kategoriju</label>  
 <form method="POST" action="{{route('doAdminCategory\_insert')}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" id="query" value="INSERT category"/>  
 <input type="hidden" name="id\_user" id="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Ime kategorije: </td>  
 <td><input type="text" name="name" class="form-control" placeholder="Ime"/></td>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="button" name="btnAddCategory" id="btnAddCategory" value="Dodaj kategoriju"/></td>  
 </tr>  
 </table>  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 150px;">  
 <ul>  
 @foreach($errors->all() as $error)  
 <li>{{$error}}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
@endsection

**3.4.5. adminCategory\_update.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 @isset($kategorija)  
 @foreach($kategorija as $k)  
 <div class="move\_right">  
 <label class="naslov">Izmeni naziv kategorije</label>  
 <form method="POST" action="{{route('doAdminCategory\_update',['id\_category'=>$k->id\_category])}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="UPDATE category"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Ime kategorije: </td>  
 <td><input type="text" name="ime" class="form-control" placeholder="Ime kategorije" value="{{isset($k)?$k->name\_category:old("Ime kategorije")}}"/></td>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnUpdateCategory" value="Izmeni kategoriju"/></td>  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert">  
 <ul>  
 @foreach($errors->all() as $e)  
 <li>{{$e}}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
 @endforeach  
 @endisset  
@endsection

**3.4.6. adminComments.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right" >  
 <table style="border:1px solid #000;margin-top: 50px;/\*margin-left: 150px;\*/">  
 <label class="naslov\_news1">Komentari  
 </label>  
 <tr class="admin">  
 <th class="admin">Komentar</th>  
 <th class="admin">Vest</th>  
 <th class="admin">Postavljeno</th>  
 <th class="admin">Korisnik</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($comments)  
 @foreach($comments as $c)  
 <tr class="admin">  
 <td class="admin">{{$c->text\_comment}}</td>  
 <td class="admin">{{$c->title\_news}}</td>  
 <td class="admin">{{date("d.m.Y H:i:s",$c->posted\_at)}}</td>  
 <td class="admin">{{$c->user\_name}}</td>  
 <td class="admin"><a href="{{route('adminComments\_delete',['id\_comment'=>$c->id\_comment])}}" data-id="{{$c->id\_comment}}">Obrisi</a></td>  
 @endforeach  
 </tr>  
 @endisset  
 </table>  
 </div>  
@endsection

**3.4.7. adminNews.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('search')  
 <div class="admin\_search" >  
 <form class="form-inline" >  
 {{csrf\_field()}}  
 <input class="form-control mr-sm-2" type="search" name="search" id="search" placeholder="Search" aria-label="Search" style="border:1px solid dimgray;">  
 <button class="btn btn-outline-success msearchy-2 my-sm-0" type="button" id="btnAdminSearch" name="btnAdminSearch" style="background-color:darkgray;border:1px solid dimgray;">Search</button>  
 </form>  
 </div>  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
  
 @isset($news)  
 <div id="news" style="text-align: center;">  
 {{$news->links()}}  
 </div>  
 @endisset  
 <div class="table">  
 <div id="ispis">  
 <table style="border:1px solid #000;">  
 <label class="naslov">Vesti  
 <a class="dodaj" href="{{route('adminNews\_insert')}}">Dodaj vest</a>  
 </label>  
 <tr class="admin">  
 <th class="admin">Naslov</th>  
 <th class="admin">Podnaslov</th>  
 <th class="admin">Tekst</th>  
 <th class="admin">Putanja slike</th>  
 <th class="admin">Kategorija</th>  
 <th class="admin">Datum postavljanja</th>  
 <th class="admin">Izmeni</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($news)  
 @foreach($news as $n)  
 <tr class="admin">  
 <td class="admin">{{$n->title\_news}}</td>  
 <td class="admin">{{$n->subtitle}}</td>  
 <td class="admin">{{$n->text\_news}}</td>  
 <td class="admin"><img src="{{$n->picture\_path}}" width="100px" height="100px" alt="slika"/></td>  
 <td class="admin">{{$n->name\_category}}</td>  
 <td class="admin">{{date("d.m.Y H:i:s",$n->posted\_date)}}</td>  
 <td class="admin"><a href="{{route('adminNews\_update',['id\_news'=>$n->id\_news])}}">Izmeni</a></td>  
 <td class="admin"><a href="{{route('adminNews\_delete',['id\_news'=>$n->id\_news])}}">Obrisi</a></td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
 </div>  
@endsection

**3.4.8. adminNews\_insert.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <label class="naslov">Dodati novu vest</label>  
 <form method="POST" action="{{route('doAdminNews\_insert')}}" enctype="multipart/form-data">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="INSERT news"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Naslov: </td>  
 <td><input type="text" name="naslov" class="form-control" placeholder="Naslov"/></td>  
 </tr>  
 <tr>  
 <td>Podnaslov: </td>  
 <td><input type="text" name="podnaslov" class="form-control" placeholder="Podnaslov"/></td>  
 </tr>  
 <tr>  
 <td>Tekst: </td>  
 <td><textarea name="tekst" class="form-control" placeholder="Tekst"> </textarea></td>  
 </tr>  
 <tr>  
 <td>Slika: </td>  
 <td><input type="file" name="slika" class="form-control" placeholder="slika"/></td>  
 </tr>  
 <tr>  
 <td>Kategorija: </td>  
 <td>  
 <select id="ddlKategorija" name="ddlKategorija">  
 <option value="0">Izaberi</option>  
 @isset($kategorija)  
 @foreach($kategorija as $k)  
 <option value="{{$k->id\_category}}">{{$k->name\_category}}</option>  
 @endforeach  
 @endisset  
 </select>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnAddNews" value="Dodaj vest"/></td>  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 150px;">  
 <ul>  
 @foreach ($errors->all() as $error)  
 <li>{{ $error }}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
  
  
@endsection

**3.4.9. adminNews\_update.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 @isset($news)  
 @foreach($news as $n)  
 <div class="move\_right">  
 <label class="naslov">Izmeni vest</label>  
 <form method="POST" action="{{route('doAdminNews\_update',['id\_news'=>$n->id\_news])}}" enctype="multipart/form-data">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="UPDATE news"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Naslov: </td>  
 <td><input type="text" name="naslov" class="form-control" placeholder="Ime" value="{{$n->title\_news}}"/></td>  
 </tr>  
 <tr>  
 <td>Podnaslov: </td>  
 <td><textarea name="podnaslov" class="form-control" placeholder="Podnaslov"> {{$n->subtitle}}</textarea></td>  
 </tr>  
 <tr>  
 <td>Tekst: </td>  
 <td><textarea name="tekst" class="form-control" placeholder="Tekst" >{{$n->text\_news}}</textarea></td>  
 </tr>  
 <tr>  
 <td>Slika: </td>  
 <td>  
 <img src="{{asset('/'.$n->picture\_path)}}" width="200px" height="200px" alt="{{$n->title\_news}}"/>  
 <input type="hidden" name="imageHidden" value="{{$n->picture\_path}}"/>  
 </td>  
 </tr>  
 <tr></tr>  
 <tr>  
 <td>Izmeni sliku:</td>  
 <td><input type="file" name="slikaUpdate" id="slikaUpdate" /></td>  
 </tr>  
 <tr>  
 <td>Kategorija: </td>  
 <td>  
 <select id="ddlKategorija" name="ddlKategorija">--}}  
 @isset($kategorija)  
 @foreach($kategorija as $k)  
 @if($n->category\_id == $k->id\_category)  
 <option value="{{$k->id\_category}}" selected>{{$k->name\_category}}</option>  
 @else  
 <option value="{{$k->id\_category}}">{{$k->name\_category}}</option>  
 @endif  
 @endforeach  
 @endisset  
 </select>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnUpdateNews" value="Izmeni vest"/></td>  
  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 50px;">  
 <ul>  
 @foreach ($errors->all() as $error)  
 <li>{{ $error }}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
 @endforeach  
 @endisset  
  
@endsection

**3.4.10. adminRoles.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right1">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 </br>  
 </br>  
 <div class="move\_right1">  
 <table class="admin">  
 <label class="naslov">Uloge  
 <a class="dodaj" href="{{route('adminRoles\_insert')}}">Dodaj ulogu</a>  
 </label>  
 <tr class="admin">  
 <th class="admin">Ime uloge</th>  
 <th class="admin">Izmeni</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($uloga)  
 @foreach($uloga as $u)  
 <tr class="admin">  
 <td class="admin">{{$u->name\_role}}</td>  
 <td class="admin"><a href="{{route('adminRoles\_update',['id\_role'=>$u->id\_role])}}">Izmeni</a></td>  
 <td class="admin"><a href="{{route('adminRoles\_delete',['id\_role'=>$u->id\_role])}}">Obrisi</a></td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
@endsection

**3.4.11. adminRoles\_insert.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <label class="naslov">Dodati novu ulogu</label>  
 <form method="POST" action="{{route('doAdminRoles\_insert')}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="INSERT role"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Uloga: </td>  
 <td><input type="text" name="uloga" class="form-control" placeholder="Uloga"/></td>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnAddRole" value="Dodaj ulogu"/></td>  
 </tr>  
 </table>  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 150px;">  
 <ul>  
 @foreach ($errors->all() as $error)  
 <li>{{ $error }}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
@endsection

**3.4.12. adminRoles\_update.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 @isset($uloga)  
 <div class="move\_right">  
 <label class="naslov">Izmeni naziv uloge</label>  
 <form method="POST" action="{{route('doAdminRoles\_update',['id\_role'=>$uloga->id\_role])}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="UDATE role"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Uloga: </td>  
 <td><input type="text" name="uloga" class="form-control" placeholder="Uloga" value="{{isset($uloga)?$uloga->name\_role:old("Uloga")}}"/></td>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnUpdateRole" value="Izmeni ulogu"/></td>  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert alert-danger">  
 <ul>  
 @foreach($errors->all() as $e)  
 <li>{{$e}}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
 @endisset  
@endsection

**3.4.13. adminUsers.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right1">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
</br>  
 </br>  
 <div class="move\_right1">  
 <table class="admin">  
 <label class="naslov">Korisnici  
 <a class="dodaj" href="{{route('adminUsers\_insert')}}">Dodaj korisnika</a>  
 </label>  
 <tr class="admin">  
 <th class="admin">Ime</th>  
 <th class="admin">Prezime</th>  
 <th class="admin">Korisnicko ime</th>  
 <th class="admin">Uloga</th>  
 <th class="admin">Izmeni</th>  
 <th class="admin">Obrisi</th>  
 </tr>  
 @isset($korisnik)  
 @foreach($korisnik as $k)  
 <tr>  
 <td class="admin">{{$k->first\_name}}</td>  
 <td class="admin">{{$k->last\_name}}</td>  
 <td class="admin">{{$k->user\_name}}</td>  
 <td class="admin">{{$k->name\_role}}</td>  
 <td class="admin"><a href="{{route('doAdminUsers\_update',['id\_user'=>$k->id\_user])}}">Izmeni</a></td>  
 <td class="admin"><a href="{{route('adminUsers\_delete',['id\_user'=>$k->id\_user])}}">Obrisi</a></td>  
 </tr>  
 @endforeach  
 @endisset  
 </table>  
 </div>  
@endsection

**3.4.14. adminUsers\_insert.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <label class="naslov">Dodavanje novog korisnika</label>  
 <form method="POST" action="{{route('doAdminUsers\_insert')}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="INSERT user"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Ime: </td>  
 <td><input type="text" name="ime" class="form-control" placeholder="Ime"/></td>  
 </tr>  
 <tr>  
 <td>Prezime: </td>  
 <td><input type="text" name="prezime" class="form-control" placeholder="Prezime"/></td>  
 </tr>  
 <tr>  
 <td>Korisnicko ime: </td>  
 <td><input type="text" name="korisnicko\_ime" class="form-control" placeholder="Korisnicko ime"/></td>  
 </tr>  
 <tr>  
 <td>Lozinka: </td>  
 <td><input type="text" name="loznika" class="form-control" placeholder="Lozinka"/></td>  
 </tr>  
 <tr>  
 <td>Uloga: </td>  
 <td>  
 <select id="ddluloga" name="ddluloga">  
 <option value="0">Izaberi</option>  
  
 @isset($uloga)  
 @foreach($uloga as $u)  
 <option value="{{$u->id\_role}}">{{$u->name\_role}}</option>  
 @endforeach  
 @endisset  
 </select>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnAddUser" value="Dodaj korisnika"/></td>  
 </tr>  
 </table>  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 50px;">  
 <ul>  
 @foreach($errors->all() as $error)  
 <li>{{$error}}</li>  
 @endforeach  
 </ul>  
  
 </div>  
 @endif  
@endsection

**3.4.15. adminUsers\_update.blade.php**

@extends('layout.index')  
  
@section('title')  
 Admin  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 @isset($korisnik)  
 <div class="move\_right">  
 <label class="naslov">Izmeni podatke o korisniku</label>  
 <form method="POST" action="{{route('doAdminUsers\_update',['id\_user'=>$korisnik->id\_user])}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="UPDATE user"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <tr>  
 <td>Ime: </td>  
 <td><input type="text" name="ime" class="form-control" placeholder="Ime" value="{{isset($korisnik)?$korisnik->first\_name:old("Ime korisnika")}}"/></td>  
 </tr>  
 <tr>  
 <td>Prezime: </td>  
 <td><input type="text" name="prezime" class="form-control" placeholder="Prezime" value="{{isset($korisnik)?$korisnik->last\_name:old("Prezime korisnika")}}"/></td>  
 </tr>  
 <tr>  
 <td>Korisnicko ime: </td>  
 <td><input type="text" name="korisnicko\_ime" class="form-control" placeholder="Korisnicko ime" value="{{isset($korisnik)?$korisnik->user\_name:old('Korsinicko ime')}}"/></td>  
 </tr>  
 <tr>  
 <td>Lozinka: </td>  
 <td><input type="password" name="loznika" class="form-control" /></td>  
 </tr>  
 <tr>  
 <td>Uloga: </td>  
 <td>  
 <select id="ddluloga" name="ddluloga">  
 @isset($uloga)  
 @foreach($uloga as $u)  
 @if($korisnik->role\_id == $u->id\_role)  
 <option value="{{$korisnik->role\_id}}" selected>{{$korisnik->name\_role}}</option>  
 @else  
 <option value="{{$u->id\_role}}">{{$u->name\_role}}</option>  
 @endif  
 @endforeach  
 @endisset  
 </select>  
 </tr>  
 <tr>  
 <td align="center">  
 <td><input type="submit" name="btnUpdateUser" value="Izmeni korisnika"/></td>  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert alert-danger">  
 <ul>  
 @foreach($errors->all() as $e)  
 <li>{{$e}}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
 @endisset  
@endsection

**3.4.16. author.blade.php**

@extends('layout.index')  
@section('title')  
 Autor  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div><div class="author" style="width:800px;height:700px;margin:auto;">  
 <label class="naslov">Autor</label>  
 @isset($author)  
  
 <h2 style="color: darkblue;">{{$author->first\_last\_name}}</h2>  
 </br>  
 <img src="{{$author->image\_path}}" class="rounded float-left" alt="{{$author->first\_last\_name}}"/></br></br></br>  
 {{$author->text1}}</br>  
 {{$author->text2}}</br>  
 {{$author->text3}}</br>  
 {{$author->text4}}</br>  
 </br>  
 <b>Broj indeksa: </b>{{$author->no\_index}}  
 @endisset  
 </div>  
@endsection

**3.4.17. galery.blade.php**

@extends('layout.index')  
  
@section('title')  
 Galerija  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right" style="/\*width:800px;height:700px;\*/">  
 <label class="naslov">Galerija</label>  
 @isset($slike)  
 <div class="picture">  
 @foreach($slike as $s)  
 <h2 style="color: #1d68a7;">{{$s->title\_news}}</h2>  
 </br>  
 <img src="{{$s->picture\_path}}" class="rounded float-left" alt="{{$s->title\_news}}"/></br></br></br>  
 @endforeach  
  
 </br>  
 </div>  
 @endif  
 <div id="galery">  
 {{$slike->links()}}  
 </div>  
 </div>  
  
@endsection

**3.4.18.** [**home.blade.php**](http://home.blade.php)

@extends('layout.index')  
  
@section('title')  
 Home  
@endsection  
  
@section('sidebar')  
 <div class="container">  
 <div class="row">  
  
 <div class="col-md-3" style="">  
 <p class="lead" style="color:darkblue;font-weight: bolder;">TopNews</p>  
 <div class="list-group" style="width:100px;height: 100%;">  
  
  
 <nav class="navbar navbar-light bg-light" >  
 <ul class="nav" style="list-style:none;font-size:20px;">  
  
 @isset($kategorije)  
 @foreach($kategorije as $kategorija)  
 <li ><a href="{{route('category',['id\_category'=>$kategorija->id\_category])}}" class="nav-brand" style="display: block;width:120px;color:#000;">{{$kategorija->name\_category}}</a></li>  
 @endforeach  
 @endif  
 </ul>  
 </nav>  
 </div>  
@endsection  
  
@section('search')  
 <div class="search" >  
 <form class="form-inline" >  
 {{csrf\_field()}}  
 <input class="form-control mr-sm-2" type="search" name="search" id="search" placeholder="Search" aria-label="Search" style="border:1px solid dimgray;">  
 <button class="btn btn-outline-success msearchy-2 my-sm-0" type="button" id="btnSearch" name="btnSearch" style="background-color:darkgray;border:1px solid dimgray;`">Search</button>  
 </form>  
 </div>  
 </div>  
@endsection  
  
@section('content')  
 <div class="col-md-9" style="padding-bottom: 50px;">  
 <div class="row carousel-holder" style="background-color: lightblue;" >  
 <div class="col-md-12">  
 <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">  
 <ol class="carousel-indicators">  
 <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>  
 <li data-target="#carousel-example-generic" data-slide-to="1"></li>  
 <li data-target="#carousel-example-generic" data-slide-to="2"></li>  
 </ol>  
 <div class="carousel-inner">  
 @isset($news)  
 @foreach($news as $n)  
 @php $class = $loop->first ? 'item active': 'item' @endphp  
 <div class="{{$class}}" style="background-color:lightblue;">  
  
 <img class="slide-image" src="{{asset('/').$n->picture\_path}}" alt="{{$n->title\_news}}" style="padding-top:10px;"/>  
  
 <h5><b><a href="{{route('showOne').'/'.$n->id\_news}}" style="color:#000;a:hover:#f2a29f;/\*width:100%;margin:0px auto;text-align:center;\*/">{{$n->title\_news}}</a></b></h5>  
 </div>  
 @endforeach  
 @endif  
 </div>  
 </div>  
 <a class="left carousel-control" href="#carousel-example-generic" data-slide="prev" >  
 <span class="glyphicon glyphicon-chevron-left"></span>  
 </a>  
 <a class="right carousel-control" href="#carousel-example-generic" data-slide="next">  
 <span class="glyphicon glyphicon-chevron-right"></span>  
 </a>  
 </div>  
 </div>  
 </div>  
  
 <div class="row">  
 <div id="ispis">  
 @if(isset($news))  
 <div id="news" style="text-align: center;">  
 {{$news->links()}}  
 </div>  
 @foreach($news as $n)  
 <div class="col-sm-4 col-lg-4 col-md-4" style="height:450px;text-overflow: ellipsis;">  
 <div class="thumbnail" style="padding-top: 0px;">  
 <img src="{{asset('/').$n->picture\_path}}" class="img-fluid" style="weight:150px;height:150px;" alt="{{$n->title\_news}}"/></br></br>  
 <div class="caption" style="padding-top: 0px;">  
 <h5><b><a href="{{route('showOne',['id\_news'=>$n->id\_news])}}" style="color: darkblue;">{{$n->title\_news}}</a></b>  
 </h5>  
 <p>  
 {{$truncated = Str::*limit*($n->text\_news, 200, ' (...)')}}  
 </p>  
 </div>  
 <div class="ratings">  
 <p>  
 <a href="{{route('category',['id\_category'=>$n->category\_id])}}" style="color:darkblue;">{{$n->name\_category}}</a>  
 </p>  
 <p>  
 <b><a href="{{route('showOne',['id\_news'=>$n->id\_news])}}" style="color:#000;">Više</a></b>  
 </p>  
 <p>  
 <b>{{date("d.m.Y H:i:s",$n->posted\_date)}}</b>  
 </p>  
 </div>  
 </div>  
 </div>  
 @endforeach  
 @endif  
 </div>  
 </div>  
 </div>  
 </div>  
@endsection

**3.4.19. kontakt.blade.php**

@extends('layout.index')  
  
@section('title')  
 Kontakt  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <label class="naslov">Kontakt forma</label>  
 <h3 style="color:darkblue; ">Pošaljite poruku administratoru</h3>  
 <form method="POST" action="{{route('doKontakt')}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="INSERT message"/>  
 <input type="hidden" name="id\_user" value="11"/>  
 <table>  
 <?php //echo (extension\_loaded('openssl')?'SSL loaded':'SSL not loaded')."\n"; ?>  
 <tr>  
 <td>Vaše ime i prezime:</td>  
 <td><input type="text" class="form-control" id="ImePrezime" name="ImePrezime"/></td>  
 </tr>  
 <tr>  
 <td>Vaš email:</td>  
 <td><input type="text" class="form-control" id="Email" name="Email"/></td>  
 </tr>  
 <tr>  
 <td>Naslov:</td>  
 <td><input type="text" class="form-control" id="Naslov" name="Naslov"/></td>  
 </tr>  
 <tr>  
 <td>Vaša poruka</td>  
 <td><textarea type="text" class="form-control" id="Poruka" name="Poruka" placeholder="Ovde unesite poruku"></textarea></td>  
 </tr>  
 <tr>  
 <td>  
 <input type="submit" id="btnPosalji" name="btnPosalji" value="Pošalji" />  
 </td>  
 </tr>  
 </table>  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 150px;">  
 <ul>  
 @foreach($errors->all() as $error)  
 <li>{{$error}}</li>  
 @endforeach  
  
 </ul>  
 </div>  
 @endif  
 @endsection

**3.4.20. logovanje.blade.php**

@extends('layout.index')  
  
@section('title')  
 Logovanje  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <form method="POST" action="{{route('doLogin')}}" >  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="SELECT user/logovanje"/>  
 <table>  
 <label class="naslov">Logovanje</label>  
 <tr>  
 <td>Korisnicko ime:</td>  
 <td><input type="text" name="korisnicko\_ime" class="form-control" placeholder="Korisnicko ime"/></td>  
 </tr>  
 <tr>  
 <td>Lozinka:</td>  
 <td><input type="password" name="lozinka" class="form-control" placeholder="Lozinka"/></td>  
 </tr>  
 <tr>  
 <td colspan="2" align="center"><input type="submit" name="btnLogovanje" value="Uloguj se"/></td>  
 </tr>  
 </table>  
 </form>  
 </div>  
@endsection

**3.4.21. oneNews.blade.php**

@extends('layout.index')  
  
@section('title')  
  
 @if(isset($news))  
 @foreach($news as $n)  
 {{$n->name\_category}}  
 @endforeach  
 @endif  
@endsection  
  
@section('content')  
 @isset($news)  
 @foreach($news as $n)  
  
 <div class="col-md-9">  
 <div class="row">  
 <div class="move\_right">  
 <h2>{{$n->title\_news}}</h2>  
 <h6>{{date("d.m.Y H:i:s",$n->posted\_date)}}</h6>  
 <img class="img-fluid" src="{{ asset('/'.$n->picture\_path) }}" style="weight:800px;height:300px;" alt="{{$n->title\_news}}"/></br></br>  
 <p>{{$n->subtitle}}</br>  
 {{$n->text\_news}}  
 </p>  
 <h5><b>Kategorija:</b> {{$n->name\_category}}</h5>  
 </div>  
 </div>  
 <div>  
 @if(empty(!session()->has('poruka')))  
 <div class="move\_right">  
 Korisnik:  
 <u><h3 style="color: darkblue;">{{session()->get('user')->user\_name}}</h3></u>  
 {{session('poruka')}}  
 </br>  
 </div>  
 @elseif(session()->has('user') && session()->get('user')->name\_role == 'korisnik')  
 <div class="move\_right">  
 <u><h4>Korisnik:</h4></u>  
 <u><h3 style="color: darkblue;">{{session()->get('user')->user\_name}}</h3></u>  
 </div>  
 <div id="ispis" class="move\_right"></div>  
 <h3 class="move\_right" style="color: darkblue;">Ostavite komentar</h3>  
 @else  
 <h4 class="move\_right" style="color: darkblue;">Ulogujte se da ostavite komentar!</h4>  
 <hr>  
 @endif  
 @if(session()->has('user') && session()->get('user')->name\_role == 'korisnik')  
 <div class="comment-form-wrap pt-5 move\_right">  
 <form action="{{route('insert\_comment')}}" method="post" name="commentform" class="p-5 bg-light">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" id="query" value="INSERT comment"/>  
 <textarea name="comment" id="comment" cols='30' rows='10'class="form-control" placeholder="Komentar..."></textarea>  
 <input type="hidden" name="news\_id" id="news\_id" value="{{$n->id\_news}}">  
 <input type="hidden" name="user\_id" id="user\_id" value="{{session()->get('user')->id\_user}}">  
 <br>  
 <input type="button" value="Pošalji" name="btnOstaviKomentar" id="btnOstaviKomentar" class="btn btn-primary py-3 px-5">  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger move\_right" style="margin-top: 10px;">  
 <ul>  
 @foreach($errors->all() as $error)  
 <li>{{$error}}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
 @endif  
 <div class="comment-form-wrap pt-5">  
 <div id="ispisKomentar">  
 @isset($comment)  
 <h3 class="move\_right">Komentari:</h3>  
 @foreach($comment as $c)  
 <hr>  
 <div class="move\_right">  
 <h4 style="color: darkblue;">Korisnicko ime: {{$c->user\_name}}</h4>  
 {{$c->text\_comment}}  
 <hr>  
 </div>  
 @endforeach  
 @endif  
 </div>  
 </div>  
 </div>  
 </div>  
 @endforeach  
 @endif  
@endsection

**3.4.22. registracija.blade.php**

@extends('layout.index')  
  
@section('title')  
 Registracija  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 <div class="move\_right">  
 <form method="POST" action="{{route('doReg')}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="INSERT user/registracija"/>  
 <table>  
 <label class="naslov">Registracija</label>  
 <tr>  
 <td>Ime:</td>  
 <td><input type="text" name="ime" class="form-control" placeholder="Ime"/></td>  
 </tr>  
 <tr>  
 <td>Prezime:</td>  
 <td><input type="text" name="prezime" class="form-control" placeholder="Prezime"/></td>  
 </tr>  
 <tr>  
 <td>Korisnicko ime:</td>  
 <td><input type="text" name="korisnicko\_ime" class="form-control" placeholder="Korisnicko ime"/></td>  
 </tr>  
 <tr>  
 <td>Lozinka:</td>  
 <td><input type="password" name="lozinka" class="form-control" placeholder="Lozinka"/></td>  
 </tr>  
 <tr>  
 <td><input type="submit" name="btnRegistracija" value="Registruj se"/></td>  
 </tr>  
 </table>  
 </form>  
 </div>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 50px;">  
 <ul>  
 @foreach ($errors->all() as $error)  
 <li>{{ $error }}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
@endsection

**3.4.23. showKorisnik.blade.php**

@extends('layout.index')  
  
@section('title')  
 Korisnik  
@endsection  
  
@section('content')  
 <div class="move\_right">  
 @empty(!session('poruka'))  
 {{session('poruka')}}  
 @endempty  
 </div>  
 @isset($korisnik)  
 <div class="move\_right">  
 <form method="POST" action="{{route('changeUserData',['id\_user'=>$korisnik->id\_user])}}">  
 {{csrf\_field()}}  
 <input type="hidden" name="query" value="UPDATE user"/>  
 <input type="hidden" name="id\_user" value="{{session()->get('user')->id\_user}}"/>  
 <table>  
 <label class="naslov">Korisnicka stranica</label>  
 <tr>  
 <td>Ime:</td>  
 <td><input type="text" name="ime" class="form-control" placeholder="Ime" value="{{$korisnik->first\_name}}"/></td>  
 </tr>  
 <tr>  
 <td>Prezime:</td>  
 <td><input type="text" name="prezime" class="form-control" placeholder="Prezime" value="{{$korisnik->last\_name}}"/></td>  
 </tr>  
 <tr>  
 <td>Korisnicko ime:</td>  
 <td><input type="text" name="korisnicko\_ime" class="form-control" placeholder="Korisnicko ime" value="{{$korisnik->user\_name}}"/></td>  
 </tr>  
 <tr>  
 <td>Lozinka:</td>  
 <td><input type="password" name="lozinka" class="form-control" placeholder="Lozinka" value=""/></td>  
 </tr>  
 <tr>  
 <td></td>  
 <td ><input type="submit" name="btnIzmeniPodatke" id="btnIzmeniPodatke" value="Izmeni svoje podatke"/> </td>  
 </tr>  
 </table>  
 </form>  
 @if($errors->any())  
 <div class="alert alert-danger" style="margin-top: 50px;">  
 <ul>  
 @foreach ($errors->all() as $error)  
 <li>{{ $error }}</li>  
 @endforeach  
 </ul>  
 </div>  
 @endif  
 </div>  
@endisset  
  
@endsection

3.5. **CSS (public/css/)**

3.5.1. css/style.css

.admin {  
 border:1px solid #000;  
 padding: 10px;  
}  
.admin1 {  
 border:1px solid #000;  
}  
.naslov{  
 font-size: large;  
 color:darkblue;  
}  
.naslov\_news{  
 font-size: large;  
 color:darkblue;  
 padding-left: 150px;  
}  
.naslov\_news1{  
 font-size: large;  
 color:darkblue;  
 padding-left: 500px;  
}  
.naslov\_news\_activity{  
 font-size: large;  
 color:darkblue;  
 padding-left: 480px;  
}  
.naslov\_news\_activity1{  
 font-size: large;  
 color:darkblue;  
 padding-left: 500px;  
 padding-bottom: 10px;  
}  
.dodaj{  
 font-size:15px;  
 padding-left:35px;  
 text-decoration: underline;  
}  
.move\_right1{  
 margin-left: 480px;  
}  
.move\_right{  
 margin-left: 200px;  
}  
  
.table{  
 padding: 50px;  
}  
.admin\_search{  
 margin-top: 50px;  
 padding-left: 150px;  
}

**3.5.2 public/js/script.js**

$(document).ready(function () {  
  
//Search na pocetnoj strani  
 $('#btnSearch').click(function () {  
  
 $unos = $('input[name="search"]').val();  
 $ispis = $('#ispis');  
 //console.log($unos);  
 $.ajax({  
 url: baseUrl + '/ajax/search',  
 type: 'POST',  
 data: {unos: $unos, \_token: token},  
 dataType: 'json',  
 success: function (data, xhr) {  
 //console.log(data);  
 //console.log(xhr);  
 $ispis.html('');  
 $ispis.html(data);  
 //$ispis.html(showNews(data));  
 },  
 error: function (xhr, status, error) {  
 console.log("Greska");  
 //console.log(xhr);  
 //console.log(status);  
 //console.log(error);  
 }  
 });  
 });  
//Search na admin strani  
 $('#btnAdminSearch').click(function () {  
 $unos = $('input[name="search"]').val();  
 $ispis = $('#ispis');  
 $ispis2 = $('#news');  
 //console.log($unos);  
  
 $.ajax({  
 url: baseUrl + '/ajax/adminSearch',  
 type: 'POST',  
 data: {unos: $unos, \_token: token},  
 dataType: 'json',  
 success: function (data, xhr) {  
 // console.log(data);  
 // console.log(xhr);  
 $ispis.html("");  
 $ispis2.html("");  
 $ispis.html(data);  
 },  
 error: function (xhr, status, error) {  
 console.log("Greska");  
 console.log(xhr);  
 console.log(status);  
 console.log(error);  
 }  
 });  
  
 });  
//AJAX insert kategorije  
 $('#btnAddCategory').click(function () {  
 $unos = $('input[name="name"]').val();  
 $query = $('#query').val();  
 $id\_user = $('#id\_user').val();  
 $ispis = $("#ispis");  
 //console.log($unos);  
  
 $.ajax({  
 url: baseUrl + '/ajax/adminCategory\_insert',  
 type: 'POST',  
 data: {unos: $unos, query: $query, id\_user: $id\_user, \_token: token},  
 dataType: 'json',  
 success: function (data, xhr) {  
 // console.log(data);  
 // console.log(xhr);  
 $ispis.html("");  
 $ispis.html(data);  
 $('input[name="name"]').val("");  
 },  
 error: function (xhr, status, error) {  
 console.log("Greska");  
 console.log(xhr);  
 console.log(status);  
 console.log(error);  
 }  
 });  
 });  
//Filtriranje aktivnosti po datumu  
 $('#sortByDate').change(function () {  
 $unos = $('input[name="sortByDate"]').val();  
 $ispis = $("#ispis");  
 console.log($unos);  
  
 $.ajax({  
 url: baseUrl + '/ajax/adminSortByDate',  
 type: 'POST',  
 data: {unos: $unos, \_token: token},  
 dataType: 'json',  
 success: function (data, xhr) {  
 // console.log(data);  
 // console.log(xhr);  
 $ispis.html("");  
 $ispis.html(data);  
 },  
 error: function (xhr, status, error) {  
 console.log("Greska!");  
 console.log(xhr);  
 console.log(status);  
 console.log(error);  
 }  
 });  
 });  
//AJAX insert komentara  
  
 $('#btnOstaviKomentar').click(function () {  
 $unos = $("#comment").val();  
 $news\_id = $("#news\_id").val();  
 $user\_id = $("#user\_id").val();  
 $query = $("#query").val();  
 $ispis = $("#ispis");  
 $ispisKomentar = $('#ispisKomentar');  
 $unos2 = $("#comment");  
 //console.log($unos);  
  
 $.ajax({  
 url: baseUrl + '/ajax/insert\_comment',  
 type: 'POST',  
 data: {unos: $unos, news\_id: $news\_id, user\_id: $user\_id, \_token: token},  
 dataType: 'json',  
 success: function (data, xhr) {  
 //console.log(data);  
 //console.log(xhr);  
 //$ispis.html('');  
 //$ispis.html(data);  
 $unos2.val("");  
 $ispisKomentar.html('');  
 $ispisKomentar.html(data);  
 },  
 error: function (xhr, status, error) {  
 console.log("Greska!");  
 console.log(xhr);  
 console.log(status);  
 console.log(error);  
 $ispis.html('');  
 $data = xhr.responseText;  
 if (JSON.parse($data) != "") {  
 $jsonResponse = JSON.parse($data);  
 console.log($jsonResponse["errors"]['unos']);  
 //$r = xhr.parse(this.response)  
 //$ispis.html($r);  
 $ispis.html($jsonResponse["errors"]['unos']);  
 }  
 }  
 });  
  
 });  
});

3.6. **PHP (app/http/Controlers)**

**3.6.1 – AuthController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\Korisnik;  
//use App\Http\Requests\RegisterRequest;  
use Illuminate\Validator;  
use App\Models\Activity;  
use App\Http\Requests\RegistracijaRequest;  
class AuthController extends FrontendController  
{  
 public $data = [];  
  
 public function registracija()  
 {  
 return view('pages.registracija');  
 }  
 public function logovanje()  
 {  
 return view('pages.logovanje');  
 }  
 public function doLog(Request $request)  
 {  
 if ($request->has('btnLogovanje')) {  
 //dd($request);  
 $user\_name = $request->input('korisnicko\_ime');  
 $password = md5($request->input('lozinka'));  
 $query = $request->input('query');  
  
 $korisnik = new Korisnik();  
 $korisnik->user\_name = $user\_name;  
 $korisnik->password = $password;  
 //dd($korisnik);  
 try{  
 $user = $korisnik->checkUser($user\_name, $password);  
 //dd($user);  
 if ($user) {  
 $request->session()->put('user', $user);  
  
 $id\_user = session()->get('user')->id\_user;  
  
 $activity = new Activity();  
 $write = $activity->insertActivity($query,$id\_user);  
  
 return redirect()->route('adminCategory')->with('poruka', 'Uspesno logovanje!');  
 } else {  
 return redirect()->route('loginForma')->with('poruka', 'Niste ulogovani!');  
  
 }  
 }  
 catch  
 (\Exception $ex) {  
 \Log::*error*('MOJA GRESKA: ' . $ex->getMessage());  
 }  
 }  
 }  
 public function logout(Request $request)  
 {  
 $request->session()->forget('user');  
 $request->session()->flush('user');  
 return redirect('/');  
 }  
 public function doReg(RegistracijaRequest $request)  
 {  
 if ($request->has('btnRegistracija')) {  
  
 $first\_name = $request->input('ime');  
 $last\_name = $request->input('prezime');  
 $user\_name = $request->input('korisnicko\_ime');  
 $password = $request->input('lozinka');  
 $query = $request->input('query');  
 $user = NULL;  
  
 $korisnik = new Korisnik();  
  
 $korisnik->first\_name = $first\_name;  
 $korisnik->last\_name = $last\_name;  
 $korisnik->user\_name = $user\_name;  
 $korisnik->password = $password;  
 try {  
 $user = $korisnik->insertRegistracija();  
  
 if ($user) {  
 $activity = new Activity();  
 $write = $activity->insertActivityFront($query,$user);  
 return redirect()->route('loginForma')->with('poruka', 'Uspesno ste se registrovali! Ulogujte se!');  
 } else {  
 return redirect()->back()->with('poruka', 'Nije uspela registracija!');  
 }  
 } //  
 catch  
 (\Exception $ex) {  
 \Log::*error*('MOJA GRESKA: ' . $ex->getMessage());  
 }  
 }  
 }  
}

**3.6.2 – BeckendController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Comment;  
use App\Models\News;  
use App\Models\Slika;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
use Illuminate\Support\Facades\File;  
use App\Models\Kategorija;  
use App\Models\Korisnik;  
use App\Models\Uloga;  
use App\Models\Activity;  
use App\Models\Contact;  
use Illuminate\Support\Facades\Response;  
  
  
class BackendController extends Controller {  
  
 public $data = [];  
  
 public function activity() {  
 $activity = new Activity();  
 try {  
 $this->data['activity'] = $activity->getAll();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminActivity', $this->data);  
 }  
  
 public function adminSearch(Request $request) {  
 $unos = $request->post('unos');  
  
 $news = new News();  
 $rez = $news->getBySearch($unos);  
  
// $totalCount = $rez->count();  
//  
// $perPage = 3;  
// $pageCount = ceil($totalCount/$perPage);  
  
// $page = $request->has('page') ? $request->get('page') : 1;  
//  
// $offset = ((int)$page-1) \* $perPage;  
//  
// $rez1 = $rez->skip($offset)->take($perPage);  
  
  
 $podaci = null;  
 $podaci .= "<table style='border:1px solid #000;'>";  
 $podaci .= "<label class='naslov'>Vesti";  
 $podaci .= "<a class='dodaj' href='" . route('adminNews\_insert') . "'>Dodaj vest</a>";  
 $podaci .= "</label>";  
 $podaci .= "<tr class='admin'>";  
 $podaci .= "<th class='admin'>Naslov</th>";  
 $podaci .= "<th class='admin'>Podnaslov</th>";  
 $podaci .= " <th class='admin'>Tekst</th>";  
 $podaci .= "<th class='admin'>Putanja slike</th>";  
 $podaci .= "<th class='admin'>Kategorija</th>";  
 $podaci .= "<th class='admin'>Datum postavljanja</th>";  
 $podaci .= "<th class='admin'>Izmeni</th>";  
 $podaci .= "<th class='admin'>Obrisi</th>";  
 $podaci .= "</tr>";  
 foreach ($rez as $item) {  
 $podaci .= "<tr class='admin'>";  
 $podaci .= "<td class='admin'>" . $item->title\_news . "</td>";  
 $podaci .= "<td class='admin'>" . $item->subtitle . "</td>";  
 $podaci .= "<td class='admin'>" . $item->text\_news . "</td>";  
 $podaci .= "<td class='admin'><img src='" . $item->picture\_path . "' width='100px' height='100px' alt='slika'/></td>";  
 $podaci .= "<td class='admin'>" . $item->name\_category . "</td>";  
 $podaci .= "<td class='admin'>" . date('d.m.Y H:i:s', $item->posted\_date) . "</td>";  
 $podaci .= "<td class='admin'><a href='" . route('adminNews\_update', ['id\_new' => $item->id\_news]) . "'>Izmeni</a></td>";  
 $podaci .= "<td class='admin'><a href='" . route('adminNews\_delete', ['id\_news' => $item->id\_news]) . "'>Obrisi</a></td>";  
 $podaci .= "</tr>";  
 }  
 $podaci .= "</table>";  
 return json\_encode($podaci);  
 }  
  
 public function sortByDate(Request $request) {  
 $unos = $request->post('unos');  
 $podaci = null;  
 list($year\_sort, $month\_sort, $day\_sort) = explode('-', $unos);  
  
 $activity = new Activity();  
 $rez = $activity->getAll();  
  
 $podaci .= "<table style='border:1px solid #000;margin-top: 25px;margin-left: 350px;'>";  
 $podaci .= "<tr class='admin'>";  
 $podaci .= "<th class='admin'>Upit</th>";  
 $podaci .= "<th class='admin'>Datum postavljanja</th>";  
 $podaci .= "<th class='admin'>Korisnik</th>";  
 $podaci .= "</tr>";  
 foreach ($rez as $r) {  
 $date = date('Y-m-d', $r->time);  
 list($year, $month, $day) = explode('-', $date);  
  
 if ($day\_sort == $day && $month\_sort == $month && $year\_sort == $year) {  
 $podaci .= "<tr class='admin'>";  
 $podaci .= "<td class='admin'>$r->query</td>";  
 $podaci .= "<td class='admin'>" . date('d.m.Y H:i:s', $r->time) . "</td>";  
 $podaci .= "<td class='admin'>$r->user\_name</td>";  
 $podaci .= "</tr>";  
 }  
 }  
 $podaci .= "</table>";  
 return json\_encode($podaci);  
 }  
  
 public function admin\_contactForm() {  
 $contact = new Contact();  
 $this->data['contact'] = $contact->getAll();  
 return view('pages.admin\_contactForm', $this->data);  
 }  
  
 public function adminContactForm\_delete($id\_contact) {  
 $contact = new Contact();  
 $contact->id\_contact = $id\_contact;  
 $query = "DELETE contact";  
 $id\_user = 1;  
  
 $rez = $contact->delete\_contact($id\_contact);  
  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
  
 return redirect()->route('admin\_contactForm')->with('poruka', 'Uspesno ste obavili brisanje poruke!');  
 } else {  
 return redirect()->route('admin\_contactForm')->with('poruka', 'Niste obrisali poruku!');  
 }  
 }  
  
}

**3.6.3 – CategoryController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use App\Models\Kategorija;  
use Illuminate\Http\Request;  
use App\Http\Requests\CategoryUpdateRequest;  
class CategoryController extends Controller  
{  
 public function adminCategory() {  
 $kategorija = new Kategorija();  
 try {  
 $this->data['kategorija'] = $kategorija->getAll();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages/adminCategory', $this->data);  
 }  
 public function adminCategory\_insert() {  
 return view('pages.adminCategory\_insert');  
 }  
 public function doAdminCategory\_insert(Request $request) {  
  
 $podaci = null;  
 $unos = $request->post('unos');  
  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 $category = new Kategorija();  
 $category->name\_category = $unos;  
  
 $rez = $category->insertCategory();  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
  
 $podaci .= "Uspesno ste uneli kategoriju!";  
 return json\_encode($podaci);  
 } else {  
 $podaci .= "Niste ste uneli kategoriju!";  
 return json\_encode($podaci);  
 }  
 }  
  
 public function adminCategory\_update($id\_category) {  
 $kategorija = new Kategorija();  
 $kategorija->id\_category = $id\_category;  
  
 $this->data['kategorija'] = $kategorija->getOne($id\_category);  
  
 return view('pages.adminCategory\_update', $this->data);  
 }  
 public function doAdminCategory\_update(CategoryUpdateRequest $request, $id\_category) {  
 if ($request->has('btnUpdateCategory')) {  
 $ime\_kategorije = $request->input('ime');  
  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 $kategorija = new Kategorija();  
 $kategorija->id\_category = $id\_category;  
 $kategorija->name\_category = $ime\_kategorije;  
  
 try {  
 $this->data['kategorija'] = $kategorija->update\_category();  
 if ($ime\_kategorije == $this->data['kategorija']['name\_category']) {  
 if ($this->data['kategorija']) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminCategory')->with('poruka', "Uspesno ste uneli izmenu!");  
 } else {  
 return redirect()->route('adminCategory\_update', ['id\_category' => $id\_category])->with('poruka', "Izmena nije uspela!");  
 }  
 } else {  
 return redirect()->route('adminCategory')->with('poruka', "Uspesno ste uneli izmenu!");  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
  
 public function adminCategory\_delete($id\_category) {  
 $kategorija = new Kategorija();  
 $kategorija->id\_category = $id\_category;  
 $query = 'DELETE\_category';  
 $id\_user = 1;  
  
 try {  
 $rez = $kategorija->delete\_category($id\_category);  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminCategory')->with('poruka', 'Uspesno brisanje!');  
 } else {  
 return redirect()->route('adminCategory')->with('poruka', 'Brisanje nije uspelo!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 return response(null, 500);  
 }  
 }  
}

**3.6.4 – CommentController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use App\Models\Comment;  
use Illuminate\Http\Request;  
  
class CommentController extends Controller  
{  
 public function adminComments() {  
 $comments = new Comment();  
  
 try {  
 $this->data['comments'] = $comments->get\_comments();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminComments', $this->data);  
 }  
 public function adminComments\_delete($id\_comment) {  
 $comment = new Comment();  
 $comment->id\_comment = $id\_comment;  
 $query = 'DELETE\_comment';  
 $id\_user = 1;  
 try {  
 $rez = $comment->delete\_comment($id\_comment);  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminComments')->with('poruka', 'Uspesno brisanje!');  
 } else {  
 return redirect()->route('adminComments')->with('poruka', 'Nije uspelo brisanje!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
}

**3.6.5 – FrontendController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Kategorija;  
use http\Message;  
use Illuminate\Http\Request;  
use App\Models\Author;  
use App\Models\News;  
use App\Models\Contact;  
use App\Models\Activity;  
use Illuminate\Support\Str;  
use Illuminate\Pagination\Paginator;  
use Illuminate\Pagination\LengthAwarePaginator;  
use function MongoDB\BSON\toJSON;  
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\SMTP;  
use PHPMailer\PHPMailer\Exception;  
  
class FrontendController extends Controller  
{  
 private $data = [];  
  
 public function index()  
 {  
 $kategorija = new Kategorija();  
 try{  
 $this->data['kategorije'] = $kategorija->getAll();  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 $news = new News();  
 try{  
 $this->data['news'] = $news->getAllNews();  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.home', $this->data);  
 }  
  
 public function category($category\_id)  
 {  
 $kategorija = new Kategorija();  
 $this->data['kategorije'] = $kategorija->getAll();  
  
 $news = new News();  
 $this->data['news'] = $news->showAllNewsFromCategory($category\_id);  
 return view('pages.home', $this->data);  
 }  
  
 public function showOne($id){  
 $news = new News();  
 $this->data['id'] = $id;  
  
 if(!empty($id)){  
 $news->id\_news = $id;  
 $this->data['news'] = $news->oneNews($id);  
  
 $this->data['comment'] = $news->getComment($id);  
 return view('pages.oneNews',$this->data);  
 }  
 }  
 public function kontakt(){  
 return view('pages.kontakt');  
 }  
 public function doKontakt(Request $request){  
 if($request->has('btnPosalji')){  
 $first\_last\_name = $request->input('ImePrezime');  
 $title = $request->input('Naslov');  
 $email = $request->input('Email');  
 $text = $request->input('Poruka');  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 require base\_path().'/vendor/autoload.php';  
 $mail = new PHPMailer(true);  
  
 try{  
 //Server settings  
 $mail->SMTPDebug = 4;  
 $mail->isSMTP();  
 $mail->Host = "smtp.gmail.com";  
 $mail->SMTPAuth = true;  
 $mail->Username = 'topNews.af@gmail.com';  
 $mail->Password = 'david280215';  
 //$mail->Mailer = "smtp";  
 $mail->SMTPSecure = 'ssl';  
 $mail->Port = 587;  
  
 //Recipients  
 $mail->setFrom('topNews.af@gmail.com', 'TopNews');  
 $mail->addAddress('bebelak@gmail.com', 'Aleksandra Fridl');  
  
 // Content  
 $mail->isHTML(true); // Set email format to HTML  
 $mail->CharSet = 'UTF-8';  
 $mail->Subject = $title;  
 $mailText = "<table>";  
 $mailText .= "<tr><td>Ime i Prezime: </td><td>".$first\_last\_name."</td></tr>";  
 $mailText .= "<tr><td>Email:</td><td>".$email."</td></tr>";  
 $mailText .= "<tr><td>Poruka:</td><td>".$text."</td></tr>";  
 $mail->Body = $mailText; //$text; //'This is the HTML message body <b>in bold!</b>';  
 $mail->send();  
 if($mail->send()) {  
 $contact = new Contact();  
 $contact->title = $title;  
 $contact->text = $text;  
 $contact->email = $email;  
 $contact->first\_last\_name = $first\_last\_name;  
 $rez = $contact->insert\_message();  
  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 if ($write == true) {  
 return redirect()->route('kontakt')->with('poruka', 'Uspesno ste poslali poruku administratoru!');  
 }  
 }  
 else {  
 return redirect()->route('kontakt')->with('poruka', 'Poruka nije poslata!');  
 }  
 }  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
  
 public function showAuthor(){  
 $author = new Author();  
 try{  
 $this->data['author']= $author->getAll();  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.author',$this->data);  
 }  
 public function search(Request $request){  
 $unos = $request->post('unos');  
 $news = new News();  
 $rez = $news->getBySearch($unos);  
 $podaci = null;  
 foreach($rez as $n) {  
 //$podaci .= $n->picture\_path;  
 $podaci .= "<div class='col-sm-4 col-lg-4 col-md-4' style='height:450px;text-overflow: ellipsis;'>";  
 $podaci .= "<div class='thumbnail' style='padding-top: 0px;'>";  
 $podaci .= "<img src='".asset('/').$n->picture\_path."' class='img-fluid' style='weight:150px;height:150px;' alt='".$n->title\_news."'/></br></br>";  
 $podaci .= "<div class='caption' style='padding-top: 0px;'>";  
 $podaci .= "<h5><b><a href='".route('showOne',['id\_news'=>$n->id\_news])."' style='color: darkblue;'>".$n->title\_news."</a></b>";  
 $podaci .= "</h5>";  
 $podaci .= "<p>";  
 $podaci .= $truncated = Str::*limit*($n->text\_news, 200, ' (...)');  
 $podaci .= "</p>";  
 $podaci .= "</div>";  
 $podaci .= "<div class='ratings'>";  
 $podaci .= "<p>";  
 $podaci .= "<a href='".route('category',['id\_category'=>$n->category\_id])."' style='color:darkblue;'>".$n->name\_category."</a>";  
 $podaci .= "</p>";  
 $podaci .= "<p>";  
 $podaci .= "<b><a href='".route('showOne',['id\_news'=>$n->id\_news])."' style='color:#000;'>Više</a></b>";  
 $podaci .= "</p>";  
 $podaci .= "<p>";  
 $podaci .= "<b>".date('d.m.Y H:i:s',$n->posted\_date)."</b>";  
 $podaci .= "</p>";  
 $podaci .= "</div>";  
 $podaci .= "</div>";  
 $podaci .= "</div>";  
 }  
 return json\_encode($podaci);  
 }  
 public function galerija(){  
 $news = new News();  
 try{  
 $this->data['slike'] = $news->getImages();  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.galery',$this->data);  
 }  
}

**3.6.6 – KorisnikController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use Illuminate\Http\Request;  
use App\Models\Comment;  
use App\Models\Korisnik;  
use App\Http\Requests\KorisnikRequest;  
class KorisnikController extends Controller  
{  
 public $data = [];  
  
 public function insert\_comment(Request $request)  
 {  
 $unos = $request->post('unos');  
  
 $rules = [  
 'unos' => 'required',  
 ];  
 $custom\_messages = [  
 'required' => 'Neophodno je uneti komentar!',  
 ];  
 $rez\_validate = $request->validate($rules, $custom\_messages);  
  
 $komentar = $request->input('unos');  
 $news\_id = $request->input('news\_id');  
 $user\_id = $request->input('user\_id');  
 $query = $request->input('query');  
  
 $comment = new Comment();  
 $comment->text\_comment = $komentar;  
 $comment->posted\_at = time();  
 $comment->user\_id = $user\_id;  
 $comment->news\_id = $news\_id;  
 try{  
 if ($rez\_validate == true) {  
 $rez = $comment->insert\_comment();  
 if ($rez == true) {  
 $commentNews = new Comment();  
 $commentNews->news\_id = $news\_id;  
 $rezComm = $commentNews->get\_commentsNews($news\_id);  
  
 $podaci = null;  
 $podaci .= "<h3 class='move\_right'>Komentari:</h3>";  
 foreach($rezComm as $c){  
 $podaci .= "<hr>";  
 $podaci .= "<div class='move\_right'>";  
 $podaci .= "<h4 style='color: darkblue;'>Korisnicko ime: $c->user\_name</h4>";  
 $podaci .= $c->text\_comment;  
 $podaci .= "<hr>";  
 $podaci .= "</div>";  
 }  
 return json\_encode($podaci);  
 }  
 }  
 else{  
 return json\_encode($rez\_validate);  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 public function showKorisnik($id\_user){  
 $korisnik = new Korisnik();  
 $korisnik->id\_user = $id\_user;  
 try{  
 $this->data['korisnik'] = $korisnik->getOne($id\_user);  
 return view('pages.showKorisnik',$this->data);  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 public function changeUserData(KorisnikRequest $request,$id\_user)  
 {  
 if ($request->has('btnIzmeniPodatke')) {  
 $first\_name = $request->input('ime');  
 $last\_name = $request->input('prezime');  
 $user\_name = $request->input('korisnicko\_ime');  
 $password = $request->input('lozinka');  
  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
  
 $korisnik = new Korisnik();  
 $korisnik->id\_user = $id\_user;  
 $korisnik->first\_name = $first\_name;  
 $korisnik->last\_name = $last\_name;  
 $korisnik->user\_name = $user\_name;  
 $korisnik->role\_id = 2;  
 $korisnik->password = $password;  
  
// if ($password == NULL) {  
// return redirect()->route('showKorisnik', ['id\_user' => $id\_user])->with('poruka', 'Morate uneti lozinku!');  
// }  
// else {  
  
 //}  
 try{  
 $rez = $korisnik->update\_User($id\_user);  
 if ($rez) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query,$id\_user);  
 return redirect()->route('showKorisnik', ['id\_user' => $id\_user])->with('poruka', 'Uspesno ste izmenili svoje podatke!');  
 } else {  
 return redirect()->route('showKorisnik', ['id\_user' => $id\_user])->with('poruka', 'Izmena nije izvrsena!');  
 }  
 }  
 catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
}

**3.6.7 – NewsController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use App\Models\Kategorija;  
use App\Models\News;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\File;  
use App\Http\Requests\NewsRequest;  
class NewsController extends Controller  
{  
 public function adminNews() {  
 $news = new News();  
  
 try {  
 $this->data['news'] = $news->getAllNews();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminNews', $this->data);  
 }  
 public function adminNews\_insert() {  
 $kategorija = new Kategorija();  
 try {  
 $this->data['kategorija'] = $kategorija->getAll();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminNews\_insert', $this->data);  
 }  
 public function doAdminNews\_insert(NewsRequest $request) {  
 if ($request->has('btnAddNews')) {  
  
 $title\_news = $request->input('naslov');  
 $subtitle = $request->input('podnaslov');  
 $text\_news = $request->input('tekst');  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 $photo = $request->file('slika');  
  
 $tmp\_path = $photo->getPathname();  
 $extension = $photo->getClientOriginalExtension();  
 $photo\_name = time() . '.' . $extension;  
 $new\_path = 'images/' . $photo\_name;  
  
 $category = $request->input('ddlKategorija');  
  
 // Upload slike na server  
 File::move($tmp\_path, $new\_path);  
  
 $news = new News();  
 $news->title\_news = $title\_news;  
 $news->subtitle = $subtitle;  
 $news->text\_news = $text\_news;  
 $news->picture\_path = $new\_path;  
 $news->category\_id = $category;  
 $news->posted\_date = time();  
 //$news->status = $status;  
  
 try {  
 $rez = $news->insert\_News();  
 if ($rez) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminNews')->with('poruka', 'Uspesno ste uneli vest!');  
 } else {  
 return redirect()->route('adminNews\_insert')->with('poruka', 'Niste uneli vest!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
 public function adminNews\_update($id\_news) {  
 $news = new News();  
 $news->id\_news = $id\_news;  
 $this->data['news'] = $news->oneNews($id\_news);  
  
 $kategorija = new Kategorija();  
 $this->data['kategorija'] = $kategorija->getAll();  
  
 return view('pages.adminNews\_update', $this->data);  
 }  
 public function doAdminNews\_update(NewsRequest $request, $id\_news) {  
 if ($request->has('btnUpdateNews')) {  
  
 $title\_news = $request->input('naslov');  
 $subtitle = $request->input('podnaslov');  
 $text\_news = $request->input('tekst');  
 $category\_id = $request->input('ddlKategorija');  
  
 if($request->file('slikaUpdate')){  
 $photo = $request->file('slikaUpdate');  
 }  
 $photo = $request->input('imageHidden');  
  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 $news = new News();  
 $news->id\_news = $id\_news;  
 $news->title\_news = $title\_news;  
 $news->subtitle = $subtitle;  
 $news->text\_news = $text\_news;  
  
 $news->category\_id = $category\_id;  
 $news->posted\_date = time();  
  
 if ($request->file('slikaUpdate')) {  
 $photo = $request->file('slikaUpdate');  
  
 $news = new News();  
 $news->id\_news = $id\_news;  
 $news->title\_news = $title\_news;  
 $news->subtitle = $subtitle;  
 $news->text\_news = $text\_news;  
 $news->category\_id = $category\_id;  
 $news->posted\_date = time();  
  
 $tmp\_path = $photo->getPathName();  
 $extension = $photo->getClientOriginalExtension();  
 $photo\_name = time() . '.' . $extension;  
 $new\_path = 'images/' . $photo\_name;  
  
 // Upload slike na server  
 File::move($tmp\_path, $new\_path);  
 $news->picture\_path = $new\_path;  
 }  
 else {  
 $photo = $request->input('imageHidden');  
 $news = new News();  
 $news->id\_news = $id\_news;  
 $news->title\_news = $title\_news;  
 $news->subtitle = $subtitle;  
 $news->text\_news = $text\_news;  
 $news->category\_id = $category\_id;  
 $news->posted\_date = time();  
 $news->picture\_path = $photo;  
  
// $rezImage = $news->getPicturePath($id\_news);  
// $news->picture\_path = $rezImage[0]->picture\_path;  
 }  
 try {  
 $rez = $news->update\_news($id\_news);  
 //var\_dump($rez);  
 if ($rez) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminNews')->with('poruka', 'Uspeno ste izmenili vest!');  
 } else {  
 return redirect()->route('adminNews\_update', ['id\_news' => $news->id\_news])->with('poruka', 'Izmena vesti nije izvrsena!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
 public function adminNews\_delete($id\_news) {  
 $news = new News();  
 $news->id\_news = $id\_news;  
 $query = 'DELETE\_news';  
 $id\_user = 1;  
 try {  
 $rez = $news->delete\_news($id\_news);  
  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminNews')->with('poruka', 'Uspesno ste izbrisali vest!');  
 } else {  
 return redirect()->route('adminNews')->with('poruka', 'Brisanje vesti nije izvrseno!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
  
 public static function adminNewsSearch($search) {  
 $news = new News();  
  
 try {  
  
 $news = $news->getBySearch($search);  
  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return $news;  
 }  
  
}

**3.6.8 – RoleController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use App\Models\Uloga;  
use Illuminate\Http\Request;  
use App\Http\Requests\RoleRequest;  
  
class RoleController extends Controller  
{  
 public function adminRoles() {  
 $uloga = new Uloga();  
  
 try {  
 $this->data['uloga'] = $uloga->getRoles();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminRoles', $this->data);  
 }  
 public function adminRoles\_insert() {  
 return view('pages.adminRoles\_insert');  
 }  
 public function doAdminRoles\_insert(RoleRequest $request) {  
 if ($request->has('btnAddRole')) {  
 $name\_role = $request->input('uloga');  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
// $rules = [  
// 'uloga' => 'required|regex:/^[A-Z][a-z]{4,}$/'  
// ];  
// $custom\_messages = [  
// 'required' => 'Polje :attribute je obavezno!',  
// 'uloga.regex' => 'Naziv uloge nije dobrog formata, mora sadrzati najmanje cetiri slova!',  
// ];  
// $request->validate($rules, $custom\_messages);  
  
 $uloga = new Uloga();  
  
 $uloga->name\_role = $name\_role;  
 try {  
 $rez = $uloga->insert\_role();  
 if ($rez) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminRoles')->with('poruka', 'Uspesno ste uneli ulogu!');  
 } else {  
 return redirect()->route('adminRoles\_insert')->with('poruka', 'Unos nije uspeo!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
 public function adminRoles\_update($id\_role) {  
 $uloga = new Uloga();  
 $uloga->id\_role = $id\_role;  
  
 $this->data['uloga'] = $uloga->getOne($id\_role);  
 return view('pages.adminRoles\_update', $this->data);  
 }  
 public function doAdminRoles\_update(RoleRequest $request, $id\_role) {  
 if ($request->has('btnUpdateRole')) {  
 $name\_role = $request->input('uloga');  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
// $rules = [  
// 'uloga' => 'required|regex:/^[A-Z][a-z]{4,}$/'  
// ];  
// $custom\_messages = [  
// 'uloga.required' => 'Polje :attribute je obavezno!',  
// 'uloga.regex' => 'Naziv uloge nije dobrog formata, mora sadrzati najmanje cetiri slova!'  
// ];  
//  
// $request->validate($rules, $custom\_messages);  
  
 $uloga = new Uloga();  
 $uloga->name\_role = $name\_role;  
 $uloga->id\_role = $id\_role;  
 try {  
 $this->data['uloga'] = $uloga->update\_Role($id\_role);  
  
  
// if ($name\_role == $this->data['uloga']['name\_role']) {  
 if ($this->data['uloga']) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminRoles')->with('poruka', 'Upesno ste uneli izmenu!');  
  
  
 } else {  
 return redirect()->route('adminRoles\_update', ['id\_role' => $id\_role])->with('poruka', 'Izmena nije uspela!');  
 }  
  
// } else {  
// return redirect()->route('adminRoles')->with('poruka', 'Upesno ste uneli izmenu!');  
// }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
 public function adminRoles\_delete($id\_role) {  
 $uloga = new Uloga();  
 $uloga->id\_role = $id\_role;  
 $query = 'DELETE\_roles';  
 $id\_user = 1;  
 try {  
 $rez = $uloga->delete\_role();  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminRoles')->with('poruka', 'Uspesno ste izbrisali ulogu!');  
 } else {  
 return redirect()->route('adminRoles')->with('poruka', 'Brisanje nije izvreseno!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessaage());  
 }  
 }  
  
}

**3.6.9 – UserController.php**

<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Activity;  
use App\Models\Korisnik;  
use App\Models\Uloga;  
use Illuminate\Http\Request;  
use App\Http\Requests\UserRequest;  
class UserController extends Controller  
{  
 public function adminUsers() {  
 $korisnik = new Korisnik();  
  
 try {  
 $this->data['korisnik'] = $korisnik->getAll();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminUsers', $this->data);  
 }  
  
 public function adminUsers\_insert() {  
 $uloga = new Uloga();  
 try {  
 $this->data['uloga'] = $uloga->getRoles();  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 return view('pages.adminUsers\_insert', $this->data);  
 }  
  
 public function doAdminUsers\_insert(UserRequest $request) {  
 if ($request->has('btnAddUser')) {  
  
// $rules = [  
// 'ime' => 'required|regex:/^[A-Z][a-z]{2,}(\s[\w\d]+)\*$/',  
// 'prezime' => 'required|regex:/^[A-Z][a-z]+(\s[\w\d]+)\*$/',  
// 'korisnicko\_ime' => 'required|unique:users,user\_name',  
// 'loznika' => 'required|regex:/^[\S]{4,10}$/'  
// ];  
// $custom\_messages = [  
// 'required' => 'Polje :attribute je obavezno!',  
// 'ime.regex' => 'Ime nije dobrog formata, mora sadrzati najmanje tri slova!',  
// 'prezime.regex' => 'Prezime nije dobrog formata!',  
// 'korisnicko\_ime.unique' => 'To korisnicko ime vec postoji u bazi!',  
// 'lozinka.regex' => 'Polje lozinka nije u ispravnom formatu!'  
// ];  
// $request->validate($rules, $custom\_messages);  
  
 $ime = $request->input('ime');  
 $prezime = $request->input('prezime');  
 $korisnicko\_ime = $request->input('korisnicko\_ime');  
 $lozinka = $request->input('loznika');  
 $uloga\_id = $request->input('ddluloga');  
 $query = $request->input('query');  
 $id\_user = $request->input('id\_user');  
  
 $korisnik = new Korisnik();  
  
 $korisnik->first\_name = $ime;  
 $korisnik->last\_name = $prezime;  
 $korisnik->user\_name = $korisnicko\_ime;  
 $korisnik->password = $lozinka;  
 $korisnik->role\_id = $uloga\_id;  
  
 try {  
 $rez = $korisnik->adminUser\_insert();  
 if ($rez) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminUsers')->with('poruka', 'Uspesno ste uneli korisnika!');  
 } else {  
 return redirect()->route('adminUsers\_insert')->with('poruka', 'Niste uneli korisnika!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
  
 public function adminUsers\_update($id\_user) {  
 $korisnik = new Korisnik();  
 $korisnik->id\_user = $id\_user;  
  
 $uloga = new Uloga();  
 $this->data['korisnik'] = $korisnik->getOne($id\_user);  
 $this->data['uloga'] = $uloga->getAll();  
 return view('pages.adminUsers\_update', $this->data);  
 }  
  
 public function doAdminUsers\_update(UserRequest $request, $id\_user) {  
 if ($request->has('btnUpdateUser')) {  
 $first\_name = $request->input('ime');  
 $last\_name = $request->input('prezime');  
 $user\_name = $request->input('korisnicko\_ime');  
 $password = $request->input('loznika');  
 $role\_id = $request->input('ddluloga');  
 $query = $request->input('query');  
 $id\_user1 = $request->input('id\_user');  
  
 $korisnik = new Korisnik();  
 $korisnik->id\_user = $id\_user;  
 $korisnik->first\_name = $first\_name;  
 $korisnik->last\_name = $last\_name;  
 $korisnik->user\_name = $user\_name;  
 $korisnik->role\_id = $role\_id;  
  
 if ($password == null) {  
 return redirect()->route('adminUsers\_update', ['id\_user' => $korisnik->id\_user])->with('poruka', 'Morate uneti lozinku!');  
 } else {  
 $korisnik->password = $password;  
 }  
 try {  
 $this->data['user'] = $korisnik->update\_User($id\_user);  
// dd($rez);  
  
 if ($this->data['user']) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user1);  
 return redirect()->route('adminUsers')->with('poruka', 'Uspesno ste uneli izmenu za korisnika!');  
 } else {  
 return redirect()->route('adminUsers\_update')->with('poruka', 'Nije uspela izmena!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessage());  
 }  
 }  
 }  
  
 public function adminUsers\_delete($id\_user) {  
 $korisnik = new Korisnik();  
 $korisnik->id\_user = $id\_user;  
 $query = 'DELETE\_user';  
 $id\_user = 1;  
 try {  
 $rez = $korisnik->delete\_User($id\_user);  
 if ($rez == true) {  
 $activity = new Activity();  
 $write = $activity->insertActivity($query, $id\_user);  
 return redirect()->route('adminUsers')->with('poruka', 'Uspesno ste izbrisali korisnika!');  
 } else {  
 return redirect()->route('adminUsers')->with('poruka', 'Brisanje nije izvrseno!');  
 }  
 } catch (\Exception $ex) {  
 \Log::*error*('Greska: ' . $ex->getMessaage());  
 }  
 }  
}

**3.7. PHP - Model (app/http/Models)**

**3.7.1 Activity**

<?php  
  
namespace App\Models;  
use Illuminate\Support\Facades\DB;  
use Illuminate\Database\Eloquent\Model;  
  
class Activity extends Model  
{  
 public function getAll(){  
 return $rez = DB::table('activities')  
 ->join('users','activities.user\_id','=','users.id\_user')  
 ->select('\*')  
 ->get();  
 }  
 public function insertActivity($query,$user\_id){  
 return $rez = DB::table('activities')  
 ->insert([  
 'time'=>time(),  
 'query'=>$query,  
 'user\_id'=>$user\_id  
 ]);  
 }  
 public function insertActivityFront($query,$user){  
 return $rez = DB::table('activities')  
 ->insert([  
 'time'=>time(),  
 'query'=>$query,  
 'user\_id'=>$user  
 ]);  
 }  
 public function doSortByDate($unos){  
 return $rez = DB::table('activities')  
 ->select('\*')  
 ->where('time' ,'=',$unos)  
 ->get();  
 }  
}

**3.7.2 Author**

<?php  
  
  
namespace App\Models;  
use Illuminate\Support\Facades\DB;  
use Illuminate\Database\Eloquent\Model;  
  
  
class Author extends Model  
{  
 public function getAll(){  
 return $rez = DB::table('author')  
 ->select('\*')  
 ->first();  
 }  
}

**3.7.3. Comment**

<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Support\Facades\DB;  
class Comment extends Model  
{  
 public $id\_comment;  
 public $text\_comment;  
 public $posted\_at;  
 public $user\_id;  
 public $news\_id;  
  
 public function get\_comments(){  
 return $rez = DB::table('comments')  
 ->join('users','comments.user\_id','=','users.id\_user')  
 ->join('news','comments.news\_id','=','news.id\_news')  
 ->select('\*')  
 ->get();  
 }  
  
 public function insert\_comment(){  
 return $rez = DB::table('comments')  
 ->insertGetId([  
 'text\_comment'=>$this->text\_comment,  
 'posted\_at'=>$this->posted\_at,  
 'user\_id'=>$this->user\_id,  
 'news\_id'=>$this->news\_id  
 ]);  
 }  
  
 public function delete\_comment($id\_comment){  
 return $rez = DB::table('comments')  
 ->where('id\_comment','=',$this->id\_comment)  
 ->delete();  
 }  
  
 public function getOne($id\_comment){  
 return $rez = DB::table('comments')  
 ->join('users','comments.user\_id','=','users.id\_user')  
 ->join('news','comments.news\_id','=','news.id\_news')  
 ->select('\*')  
 ->where('id\_comment','=',$this->id\_comment)  
 ->first();  
 }  
}

**3.7.4. Contact**

<?php  
  
  
namespace App\Models;  
use Illuminate\Support\Facades\DB;  
use Illuminate\Database\Eloquent\Model;  
  
class Contact extends Model  
{  
 public $id\_message;  
 public $title;  
 public $text;  
 public $email;  
 public $first\_last\_name;  
 public $posted;  
  
 public function getAll(){  
 return $rez = DB::table('contacts')  
 ->select('\*')  
 ->get();  
 }  
 public function insert\_message(){  
 return $rez = DB::table('contacts')  
 ->insert([  
 'title'=>$this->title,  
 'text'=>$this->text,  
 'email'=>$this->email,  
 'first\_last\_name'=>$this->first\_last\_name,  
 'posted'=>time()  
 ]);  
 }  
 public function delete\_contact($id\_contact){  
 return $rez = DB::table('contacts')  
 ->where('id\_contact','=',$this->id\_contact)  
 ->delete();  
 }  
  
}

**3.7.5. Kategorija**

<?php  
*/\*\*  
 \* Created by PhpStorm.  
 \* User: Mladjica  
 \* Date: 3/24/2019  
 \* Time: 8:02 PM  
 \*/*namespace App\Models;  
use Illuminate\Support\Facades\DB;  
  
class Kategorija  
{  
 public $id\_category;  
 public $name\_category;  
  
 public function getAll(){  
 return $rez = DB::table('categories')  
 ->select('\*')  
 ->get();  
 }  
  
  
 public function insertCategory(){  
 return $rez = DB::table('categories')  
 ->insert([  
 'name\_category'=>$this->name\_category  
 ]);  
 }  
  
 public function getOne(){  
 return $rez = DB::table('categories')  
 ->select('\*')  
 ->where('id\_category',$this->id\_category)  
 ->get();  
 }  
  
 public function update\_category(){  
 return $rez = DB::table('categories')  
 ->where('id\_category', $this->id\_category)  
 ->update(['name\_category'=>$this->name\_category]);  
 }  
  
 public function delete\_category(){  
 return $rez = DB::table('categories')  
 ->where('id\_category',$this->id\_category)  
 ->delete();  
 }  
}

**3.7.6. Korisnik**

<?php  
*/\*\*  
 \* Created by PhpStorm.  
 \* User: Mladjica  
 \* Date: 3/26/2019  
 \* Time: 1:35 AM  
 \*/*namespace App\Models;  
use Illuminate\Support\Facades\DB;  
class Korisnik  
{  
 public $id\_user;  
 public $first\_name;  
 public $last\_name;  
 public $email;  
 public $password;  
 public $user\_name;  
 public $name\_role;  
 public $role\_id;  
  
 public function insertRegistracija(){  
 return $rez = DB::table('users')  
 ->insertGetId([  
 'first\_name'=>$this->first\_name,  
 'last\_name'=>$this->last\_name,  
 'password'=>md5($this->password),  
 'user\_name'=>$this->user\_name,  
 'role\_id'=>$this->role\_id = 2  
 ]);  
 }  
  
 public function checkUser($user\_name,$password){  
 return $rez = DB::table('users as u')  
 ->join ('roles as r','u.role\_id','=','r.id\_role')  
 ->where([  
 ['user\_name','=',$user\_name],  
 ['password','=',$password]  
 ])  
 ->select('u.\*','r.\*')  
 ->first();  
 }  
 public function getAll(){  
 return $rez = DB::table('users as u')  
 ->join('roles as r','u.role\_id','=','r.id\_role')  
 ->select('u.\*','r.\*')  
 ->get();  
 }  
  
 public function adminUser\_insert(){  
 return $rex = DB::table('users')  
 ->insert([  
 'first\_name'=>$this->first\_name,  
 'last\_name'=>$this->last\_name,  
 'user\_name'=>$this->user\_name,  
 'password'=>md5($this->password),  
 'role\_id'=>$this->role\_id  
 ]);  
 }  
  
 public function getOne($id\_user){  
 return $rez = DB::table('users as u')  
 ->join ('roles as r','u.role\_id','=','r.id\_role')  
 ->where([  
 ['id\_user','=',$id\_user]  
 ])  
 ->select('u.\*','r.\*')  
 ->first();  
 }  
  
 public function update\_User($id\_user){  
 return $rez = DB::table('users')  
 ->where('id\_user',$this->id\_user)  
 ->update(  
 ['first\_name'=>$this->first\_name,  
 'last\_name'=>$this->last\_name,  
 'user\_name'=>$this->user\_name,  
 'password'=>md5($this->password),  
 'role\_id'=>$this->role\_id]  
 );  
 }  
  
 public function delete\_User(){  
 return $rez = DB::table('users')  
 ->where('id\_user',$this->id\_user)  
 ->delete();  
 }  
}

**3.7.7. News**

<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Support\Facades\DB;  
use Illuminate\Pagination\Paginator;  
use Illuminate\Pagination\LengthAwarePaginator;  
class News extends Model  
{  
 public $id\_news;  
 public $title\_news;  
 public $subtitle;  
 public $text\_news;  
 public $picture\_path;  
 public $category\_id;  
 public $posted\_date;  
 public $status=1;  
  
 public function getAllNews(){  
 return $rez = DB::table('news as n')  
 ->join('categories as c','n.category\_id','=','c.id\_category')  
 ->select('n.\*','c.\*')  
 ->orderBy('posted\_date','desc')  
 ->paginate(6);  
 }  
 public function showAllNewsFromCategory($category\_id)  
 {  
 return $rez = DB::table('news')  
 ->join('categories', 'news.category\_id', '=', 'categories.id\_category')  
 ->select('\*')  
 ->where('categories.id\_category', '=', $category\_id)  
 ->paginate(6);  
 }  
 public function oneNews($id){  
 return $rez = DB::table('news as n')  
 ->join('categories as c', 'n.category\_id', '=', 'c.id\_category')  
 ->select('\*')  
 ->where('id\_news','=',$id)  
 ->get();  
 }  
  
 public function getComment($id){  
 return $rez = DB::table('news as n')  
 ->join('comments as c','c.news\_id','=','n.id\_news')  
 ->join('users as u','c.user\_id','=','u.id\_user')  
 ->select('\*')  
 ->where('id\_news','=',$id)  
 ->get();  
 }  
  
 public function delete\_news(){  
 return $rez = DB::table('news')  
 ->where('id\_news',$this->id\_news)  
 ->delete();  
 }  
  
 public function insert\_News(){  
 return $rez = DB::table('news')  
 ->insertGetId([  
 'title\_news'=>$this->title\_news,  
 'subtitle'=>$this->subtitle,  
 'text\_news'=>$this->text\_news,  
 'picture\_path'=>$this->picture\_path,  
 'category\_id'=>$this->category\_id,  
 'posted\_date'=>$this->posted\_date,  
 //'status'=>$this->status  
 ]);  
 }  
  
 public function update\_news($id\_news){  
 return $rez = DB::table('news')  
 ->where('id\_news',$this->id\_news)  
 ->update([  
 'title\_news'=>$this->title\_news,  
 'subtitle'=>$this->subtitle,  
 'text\_news'=>$this->text\_news,  
 'picture\_path'=>$this->picture\_path,  
 'category\_id'=>$this->category\_id,  
 'posted\_date'=>$this->posted\_date,  
 ]);  
 }  
  
 public function getBySearch($unos){  
 return $rez = DB::table('news')  
 ->join('categories','news.category\_id','=','categories.id\_category')  
 ->select('\*')  
 ->where('news.title\_news','LIKE','%'.$unos.'%')  
 ->orWhere('news.subtitle','LIKE','%'.$unos.'%')  
 ->orWhere('news.text\_news','LIKE','%'.$unos.'%')  
 ->orderBy('news.posted\_date','desc')  
 ->get();  
 }  
  
 public function getImages(){  
 return $rez = DB::table('news')  
 ->select('picture\_path','title\_news')  
 ->orderBy('posted\_date','desc')  
 ->simplePaginate(3);  
 }  
  
 public function getPicturePath($id\_news){  
 return $rez = DB::table('news')  
 ->select('picture\_path')  
 ->where('id\_news','=', $id\_news)  
 ->get();  
 }  
}

**3.7.8. Uloga**

<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Support\Facades\DB;  
use PHPUnit\Util\RegularExpressionTest;  
  
class Uloga extends Model  
{  
  
 public $id\_role;  
 public $name\_role;  
  
 public function getAll(){  
 return $rez = DB::table('roles')  
 ->select('\*')  
 ->get();  
 }  
 public function getRoles(){  
 return $rez = DB::table('roles')  
 ->select('\*')  
 ->get();  
 }  
  
 public function getOne(){  
 return $rez = DB::table('roles')  
 ->select('\*')  
 ->where('id\_role',$this->id\_role)  
 ->first();  
 }  
  
 public function insert\_role(){  
 return $rez = DB::table('roles')  
 ->insert([  
 'name\_role'=>$this->name\_role  
 ]);  
 }  
  
 public function update\_Role(){  
 return $rez = DB::table('roles')  
 ->where('id\_role',$this->id\_role)  
 ->update(['name\_role'=>$this->name\_role]);  
 }  
  
 public function delete\_role(){  
 return $rez = DB::table('roles')  
 ->where('id\_role',$this->id\_role)  
 ->delete();  
 }  
  
}

3.8. **Middelware (app/http/Middleware)**

**3.8.1. AdminMiddleware.php**

<?php  
  
namespace App\Http\Middleware;  
  
use Closure;  
  
class AdminMiddleware  
{  
 */\*\*  
 \* 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')){  
 $user = $request->session()->get('user');  
  
 if($user -> name\_role == 'admin'){  
  
 return $next($request);  
 }  
 return redirect()->route('index')->with('poruka','Niste administrator!');  
 }  
 }  
}

**3.8.2. KorisnikMiddleware.php**

<?php  
  
namespace App\Http\Middleware;  
  
use Closure;  
  
class KorisnikMiddleware  
{  
 */\*\*  
 \* 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')) {  
 $user = $request->session()->get('user');  
  
 if ($user->name\_role == 'korisnik') {  
  
 return $next($request);  
 }  
 return redirect()->route('index')->with('poruka', 'Morate se ulogovati!');  
 }  
 }  
}

**3.9. Kernel.php (app/http/Kernel.php)**

<?php  
  
namespace App\Http;  
  
use Illuminate\Foundation\Http\Kernel as HttpKernel;  
  
class Kernel extends *HttpKernel*{  
 */\*\*  
 \* The application's global HTTP middleware stack.  
 \*  
 \* These middleware are run during every request to your application.  
 \*  
 \** ***@var*** *array  
 \*/* protected $middleware = [  
 \App\Http\Middleware\CheckForMaintenanceMode::class,  
 \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,  
 \App\Http\Middleware\TrimStrings::class,  
 \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,  
 \App\Http\Middleware\TrustProxies::class,  
 ];  
  
 */\*\*  
 \* The application's route middleware groups.  
 \*  
 \** ***@var*** *array  
 \*/* protected $middlewareGroups = [  
 'web' => [  
 \App\Http\Middleware\EncryptCookies::class,  
 \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,  
 \Illuminate\Session\Middleware\StartSession::class,  
 // \Illuminate\Session\Middleware\AuthenticateSession::class,  
 \Illuminate\View\Middleware\ShareErrorsFromSession::class,  
 \App\Http\Middleware\VerifyCsrfToken::class,  
 \Illuminate\Routing\Middleware\SubstituteBindings::class,  
 ],  
  
 'api' => [  
 'throttle:60,1',  
 'bindings',  
 ],  
 ];  
  
 */\*\*  
 \* The application's route middleware.  
 \*  
 \* These middleware may be assigned to groups or used individually.  
 \*  
 \** ***@var*** *array  
 \*/* protected $routeMiddleware = [  
 'auth' => \App\Http\Middleware\Authenticate::class,  
 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,  
 'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,  
 'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,  
 'can' => \Illuminate\Auth\Middleware\Authorize::class,  
 'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,  
 'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,  
 'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,  
 'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,  
 'admin'=>\App\Http\Middleware\AdminMiddleware::class,  
 'korisnik'=>\App\Http\Middleware\KorisnikMiddleware::class  
 ];  
  
 */\*\*  
 \* The priority-sorted list of middleware.  
 \*  
 \* This forces non-global middleware to always be in the given order.  
 \*  
 \** ***@var*** *array  
 \*/* protected $middlewarePriority = [  
 \Illuminate\Session\Middleware\StartSession::class,  
 \Illuminate\View\Middleware\ShareErrorsFromSession::class,  
 \App\Http\Middleware\Authenticate::class,  
 \Illuminate\Session\Middleware\AuthenticateSession::class,  
 \Illuminate\Routing\Middleware\SubstituteBindings::class,  
 \Illuminate\Auth\Middleware\Authorize::class,  
 ];  
}

4.1 **Requests (app/http/Requests)**

**4.1. CategoryUpdateRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class CategoryUpdateRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'ime' => 'required|min:4'  
 ];  
 }  
 public function messages()  
 {  
 return [  
 'ime.required' => 'Unesite naziv nove kategorije!',  
 'ime.min' => 'Naziv kategorije moja imati minimum četiri karaktera!'  
 ];  
 }  
}

**4.2. KorisnikRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class KorisnikRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'ime' => 'required|regex:/^[A-Z][a-z]{2,}(\s[\w\d]+)\*$/',  
 'prezime' => 'required|regex:/^[A-Z][a-z]+(\s[\w\d]+)\*$/',  
 'korisnicko\_ime' => 'required',  
 'loznika' => 'required|regex:/^[\S]{4,10}$/'  
 ];  
 }  
 public function messages()  
 {  
 return [  
 'ime.required' => 'Ime je obavezno!',  
 'ime.regex' => 'Ime nije dobrog formata, mora sadrzati najmanje tri slova!',  
 'prezime.required' => 'Prezime je obavezno!',  
 'prezime.regex' => 'Prezime nije dobrog formata!',  
 'korisnicko\_ime.required' => 'Korisnicko ime je obavezno!',  
 'loznika.required' => 'Lozinka je obavezna!',  
 'lozinka.regex' => 'Lozinka mora imati od cetiri do deset karaktera!'  
 ];  
 }  
}

**4.3. NewsRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class NewsRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'naslov' => 'required',  
 'podnaslov' => 'required',  
 'tekst' => 'required',  
 'slika' => 'mimes:jpeg,jpg,bmp,png,JPEG,JPG,BMP,PNG',  
 'ddlKategorija' => 'required|not\_in:0',  
 ];  
 }  
 public function messages()  
 {  
 return[  
 'required.naslov' => 'Unesite naslov!',  
 'required.podnaslov.' => 'Unesite podnaslov!',  
 'required.tekst.' => 'Unesite tekst!',  
 'required.slika.' => 'Unesite sliku!',  
 'ddlKategorija.required' => 'Odaberite kategoriju!',  
 'ddlKategorija.not\_in' => 'Morate odabrati kategoriju!'  
  
 ];  
  
 }  
}

**4.4. RegistracijaRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class RegistracijaRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'ime' => 'required|regex:/^[A-Z][a-z]{2,}(\s[\w\d]+)\*$/',  
 'prezime' => 'required|regex:/^[A-Z][a-z]+(\s[\w\d]+)\*$/',  
 'korisnicko\_ime' => 'required|unique:users,user\_name',  
 'lozinka' => 'required|regex:/^[\S]{4,10}$/'  
 ];  
 }  
 public function messages()  
 {  
  
 return[  
 'ime.required' => 'Ime je obavezno!',  
 'ime.regex' => 'Ime nije dobrog formata, mora sadrzati najmanje tri slova!',  
 'prezime.required' => 'Prezime je obavezno!',  
 'prezime.regex' => 'Prezime nije dobrog formata!',  
 'korisnicko\_ime.required' => 'Korisničko ime je obavezno!',  
 'korisnicko\_ime.unique' => 'Ovo korisničko ime je već zauzeto!',  
 'lozinka.required' => 'Lozinka je obavezna!',  
 'lozinka.regex' => 'Lozinka nije u ispravnom formatu!'  
 ];  
 }  
}

**4.5. RoleRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class RoleRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'uloga' => 'required|regex:/^[A-Z][a-z]{4,}$/'  
 ];  
 }  
 public function messages()  
 {  
 return [  
 "uloga.required" => "Polje za naziv uloge je obavezno!",  
 "uloga.regex" => "Naziv uloge nije dobrog formata, mora sadrzati najmanje cetiri slova!"  
 ];  
 }  
}

**4.6. UserRequest.php**

<?php  
  
namespace App\Http\Requests;  
  
use Illuminate\Foundation\Http\FormRequest;  
  
class UserRequest extends FormRequest  
{  
 */\*\*  
 \* Determine if the user is authorized to make this request.  
 \*  
 \** ***@return*** *bool  
 \*/* public function authorize()  
 {  
 return true;  
 }  
  
 */\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array  
 \*/* public function rules()  
 {  
 return [  
 'ime' => 'required|regex:/^[A-Z][a-z]{2,}(\s[\w\d]+)\*$/',  
 'prezime' => 'required|regex:/^[A-Z][a-z]+(\s[\w\d]+)\*$/',  
 'korisnicko\_ime' => 'required|unique:users,user\_name',  
 'loznika' => 'required|regex:/^[\S]{4,10}$/'  
 ];  
 }  
 public function messages()  
 {  
 return [  
 'ime.required' => 'Ime je obavezno!',  
 'ime.regex' => 'Ime nije dobrog formata, mora sadrzati najmanje tri slova!',  
 'prezime.required' => 'Prezime je obavezno!',  
 'prezime.regex' => 'Prezime nije dobrog formata!',  
 'korisnicko\_ime.required' => 'Korisnicko ime je obavezno!',  
 'korisnicko\_ime.unique' => 'To korisnicko ime vec postoji u bazi!',  
 'loznika.required' => 'Lozinka je obavezna!',  
 'lozinka.regex' => 'Lozinka mora imati od cetiri do deset karaktera!'  
 ];  
 }  
}

**4.BAZA PODATAKA**

4.1. Baza podataka **php2sajt2**

