Висока школа за информационе и телекомуникационе технологије, Београд

Vasemisljenje

https://drive.google.com/drive/folders/1ZocE9-YOtCIdlPJ8xAHs4RRiDWKJpbHR

Образовни профил: Интернет технологије

Предмет: Web програмирање PHP 2

Студент: Александар Милијановић

Индекс: 91/19

Садржај

1.0 Домен апликације

Апликација покрива домен друштвене мреже. На овој друштвеној мрежи је поред пријаве и регистрације омогућено праћење и отпраћивање других корисника, претрага пратилаца, објављивање садржаја у текстуалном облику, његова измена и брисање, лајковање, одлајковање, коментарисање, измена и брисање коментара и измена личних података.

2.0 Опис функционалности

Функционалности су следеће: пријављивање на сајт, измена корисничких података (имена, презимена, e-mail адресе, лозинке и профилне фотографије), објављивање садржаја, његова измена и брисање, објавиљивање коментара, њихова измена, брисање и лајковање, измена боје тастера услед преласка миша преко њега, промена боје линка услед преласка миша,промена боје код активног линка у навигацији, претрага пратилаца на профилној страници, слање e-mail-a администратору, динамичко исписивање линкова у навигацији, коментара, објава и категорија којима припадају објаве, користи се модал приликом приказа одређене порука која указује на исход радње која је претходно обављена и преусмеравање корисника у зависности од исхода обављене радње. Такође је омогућена и валидација података како на клијенту тако и на серверу и чишћење формулара након успешног уноса или измене података.

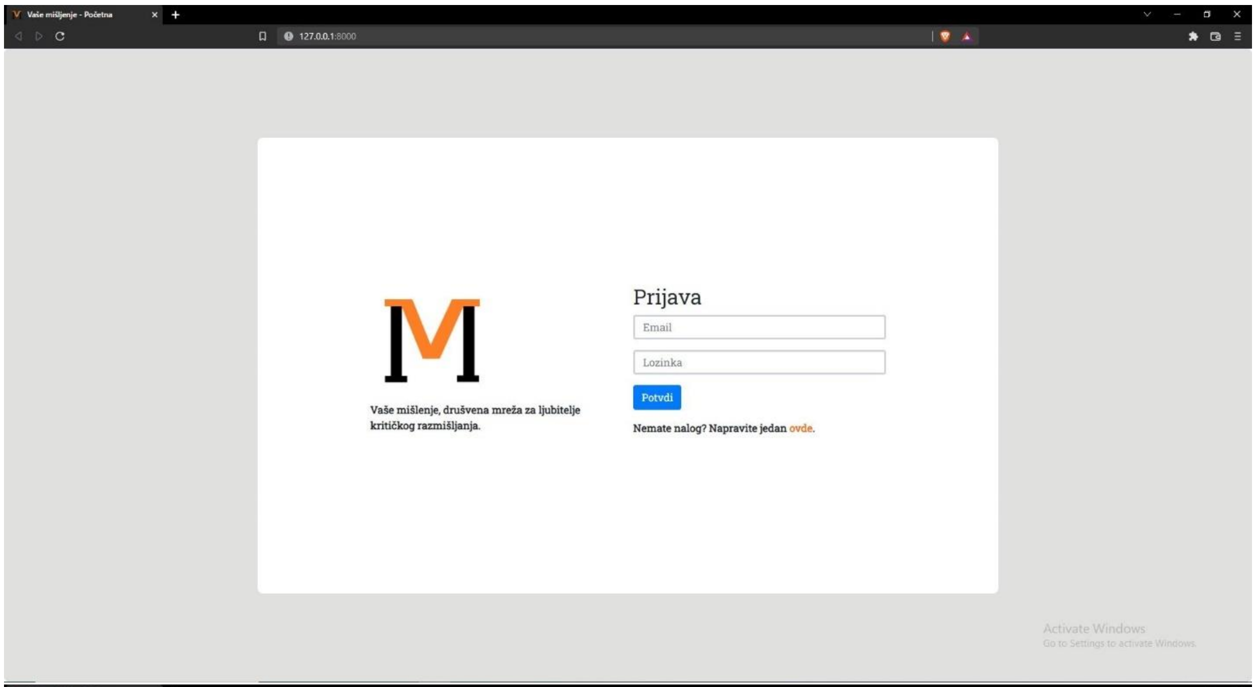
3.0 Коришћене технологије

- Laravel framework (8.0)
- jQuery
- Bootrap
- Blade engine
- HTML 5
- CSS 3
- PHP Storm
- Sass
- Javascript

4.0 Странице и опис функционалности

4.1 Почетка страна

На почетној страни се налази формулар за пријаву. У случају да корисник нема налог може приступити формулару за регистрацију преко линка који се налази испод формулара за пријаву. Након уноса података приказује се модел који извештава корисника и самом иходу и уколико је исход успешан онда се преусмерава на профилну страницу.


Vaše mišljenje - Početna

127.0.0.1:8000

Vaše mišljenje, društvena mreža za ljubitelje kritičkog razmišljanja.

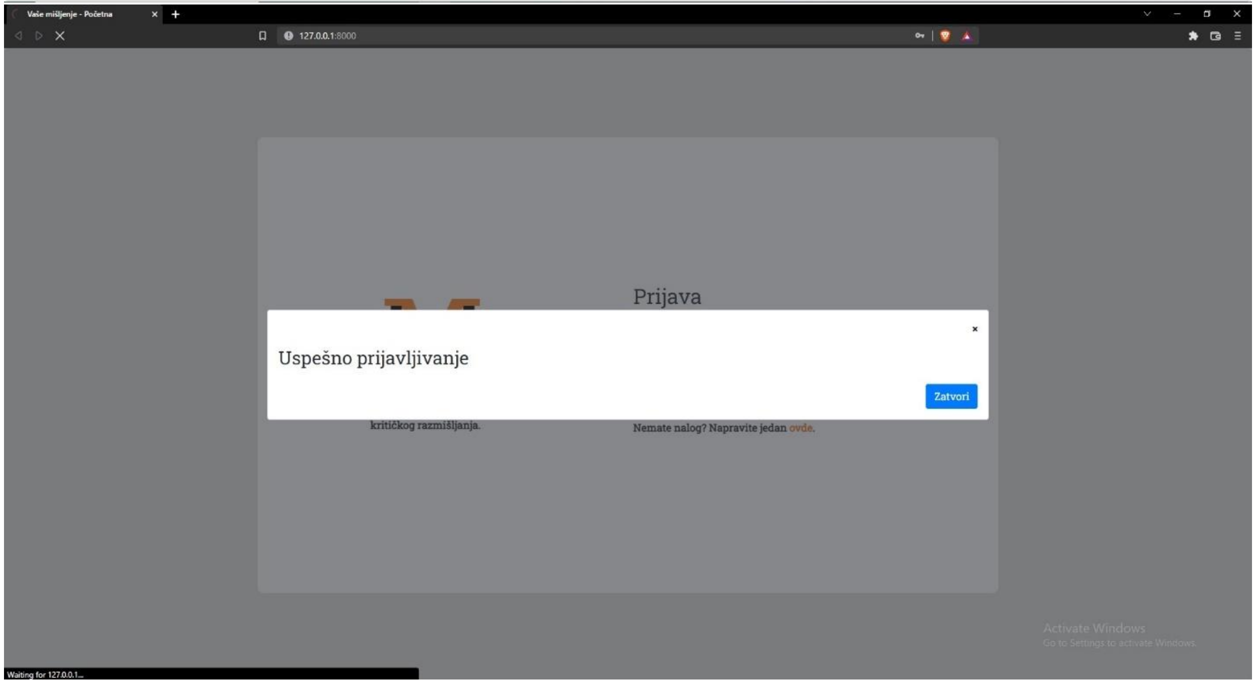
Prijava

Email

Lozinka

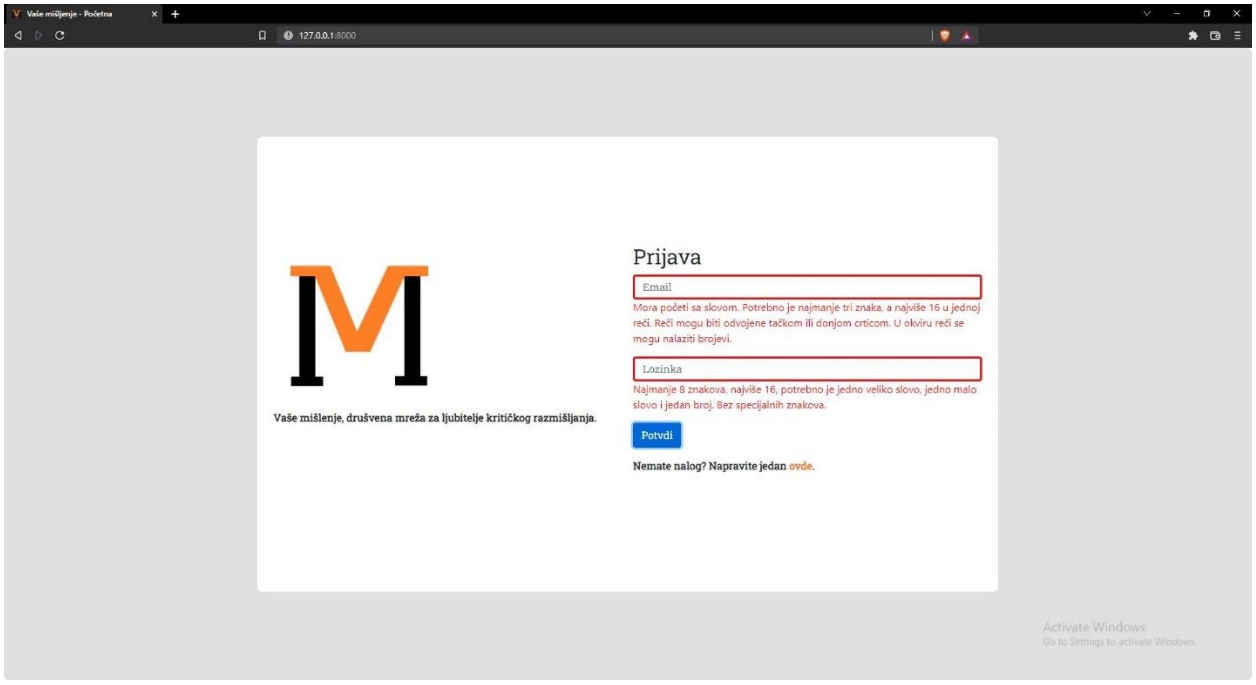
Potvdi

Nemate nalog? Napravite jedan ovde.


Uspešno prijavljivanje

Zatvori

Уколико подаци нису приликом уноса прихватљиви онда је приказ следећи:



Vaše mišljenje, društvena mreža za ljubitelje kritičkog razmišljanja.

Prijava

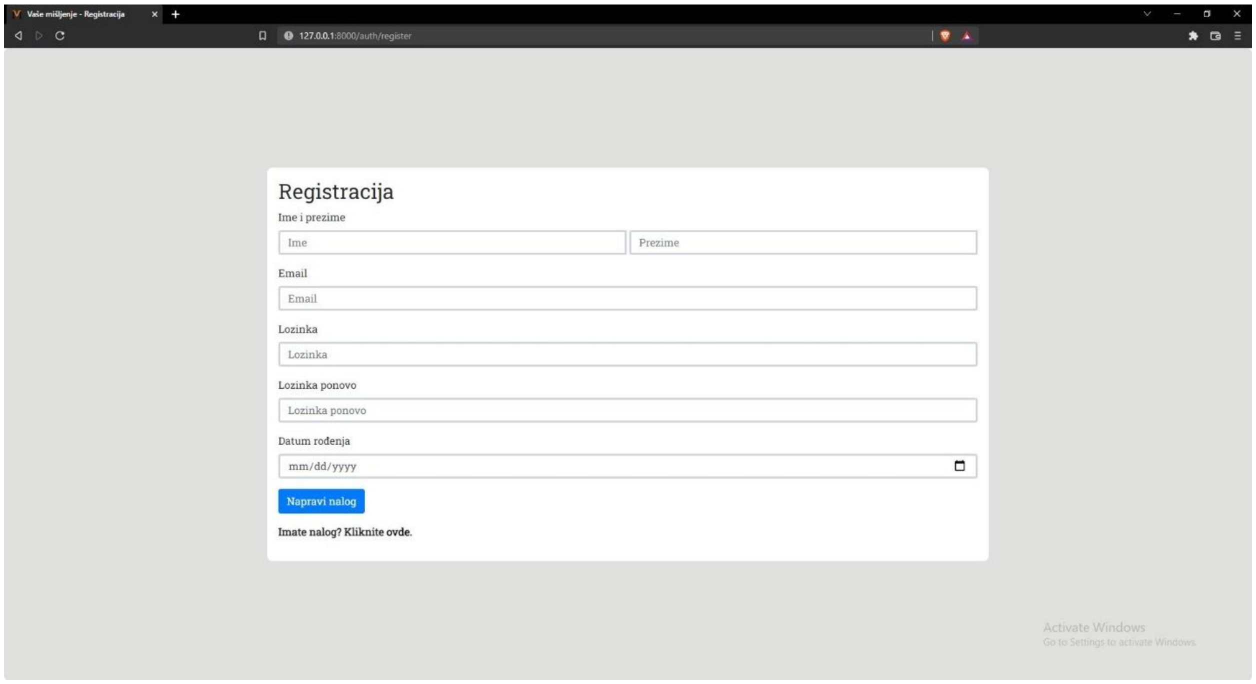
Mora početi sa slovom. Potrebno je najmanje tri znaka, a najviše 16 u jednoj reči. Reči mogu biti odvojene tačkom ili donjom crticom. U okviru reči se mogu nalaziti brojevi.

Najmanje 8 znakova, najviše 16, potrebno je jedno veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih znakova.

Nemate nalog? Napravite jedan ovde.

4.2 Регистрација

Ова страница омогућава корисницима да направе своје налоге, услед уноса неприхватљивих података приказаће се порука која указује како се унета вредност мора разобличити. Након потврђивања формулара, приказује се модал са одређеном поруком. На послетку је могућ повраћај на почетну страну преко линка испод формулара.



Registracija

Ime i prezime

Ime | Prezime

Email

Email

Lozinka

Lozinka

Lozinka ponovo

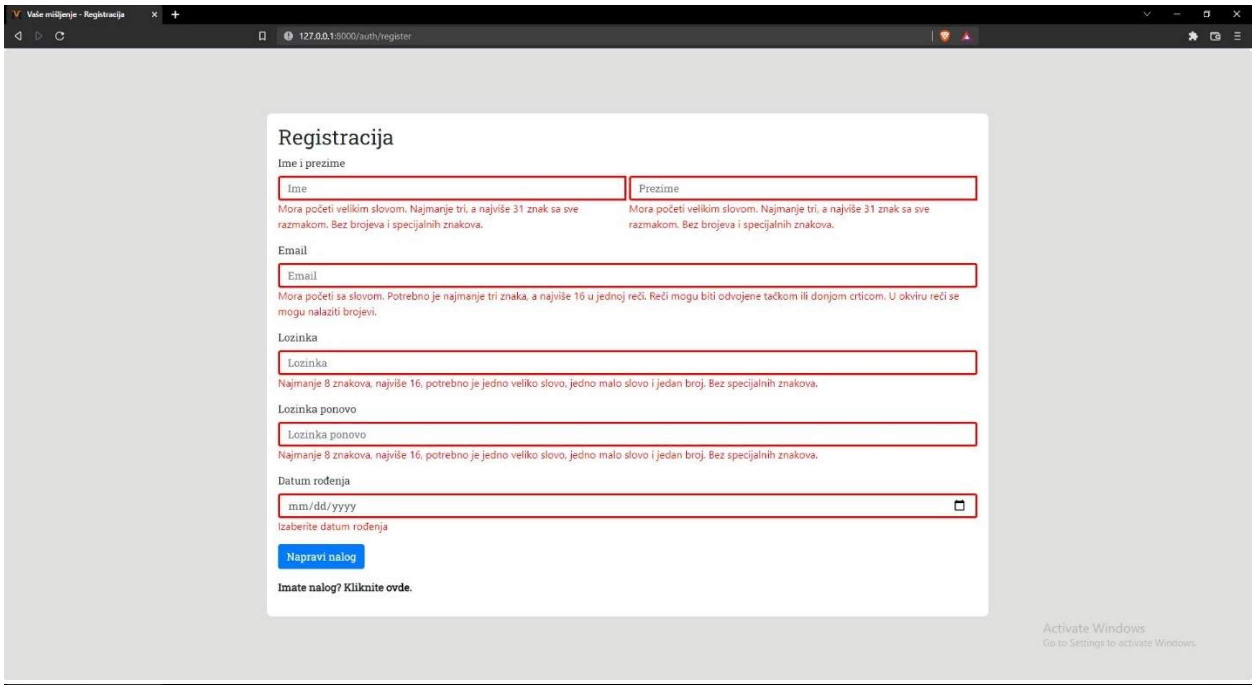
Lozinka ponovo

Datum rođenja

mm/dd/yyyy

Vaše mišljenje - Registracija
127.0.0.1:8000/auth/register
Napravi nalog
Imate nalog? Kliknite ovde.
Activate Windows
Go to Settings to activate Windows.

Приметићете да се услед неправилног уноса података приказују и модал и поруке у исти мах, овде сам доставио две слике како би се што боље виделе поруке исписане испод уносних поља.



Registracija

Ime i prezime

Mora početi velikim slovom. Najmanje tri, a najviše 31 znak sa sve razmakom. Bez brojeva i specijalnih znakova.

Mora početi velikim slovom. Najmanje tri, a najviše 31 znak sa sve razmakom. Bez brojeva i specijalnih znakova.

Email

Mora početi sa slovom. Potrebno je najmanje tri znaka, a najviše 16 u jednoj reči. Reči mogu biti odvojene tačkom ili donjom crticom. U okviru reči se mogu nalaziti brojevi.

Lozinka

Najmanje 8 znakova, najviše 16, potrebno je jedno veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih znakova.

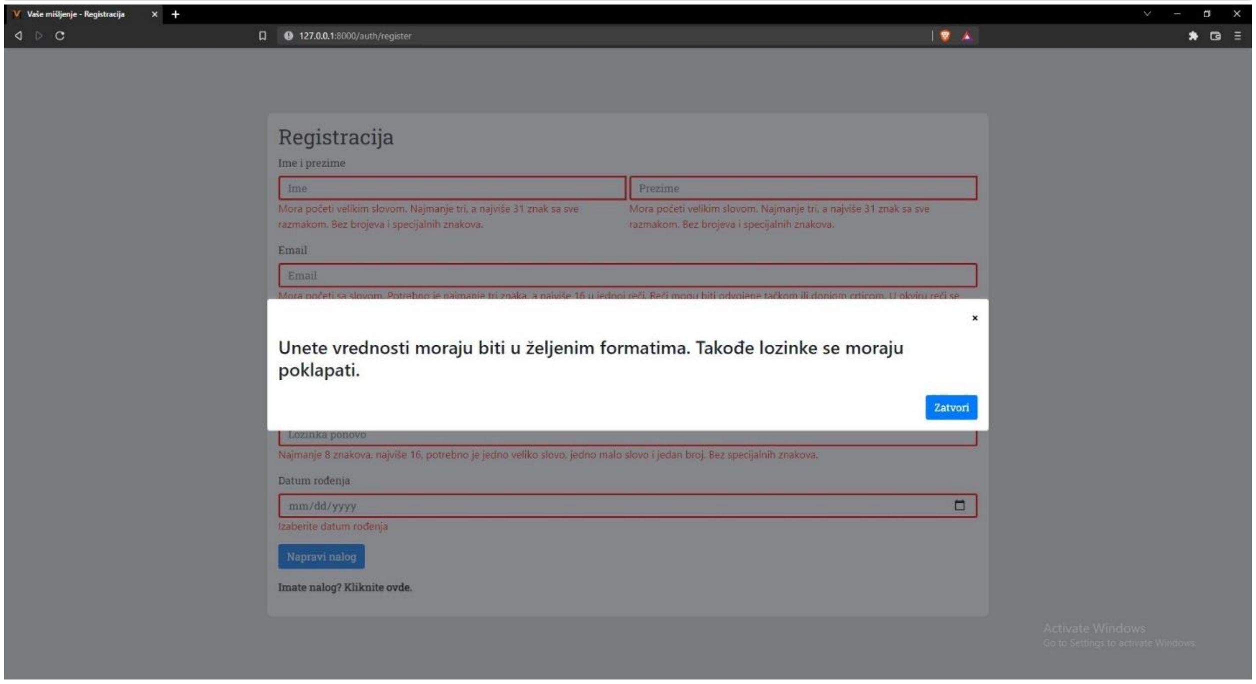
Lozinka ponovo

Najmanje 8 znakova, najviše 16, potrebno je jedno veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih znakova.

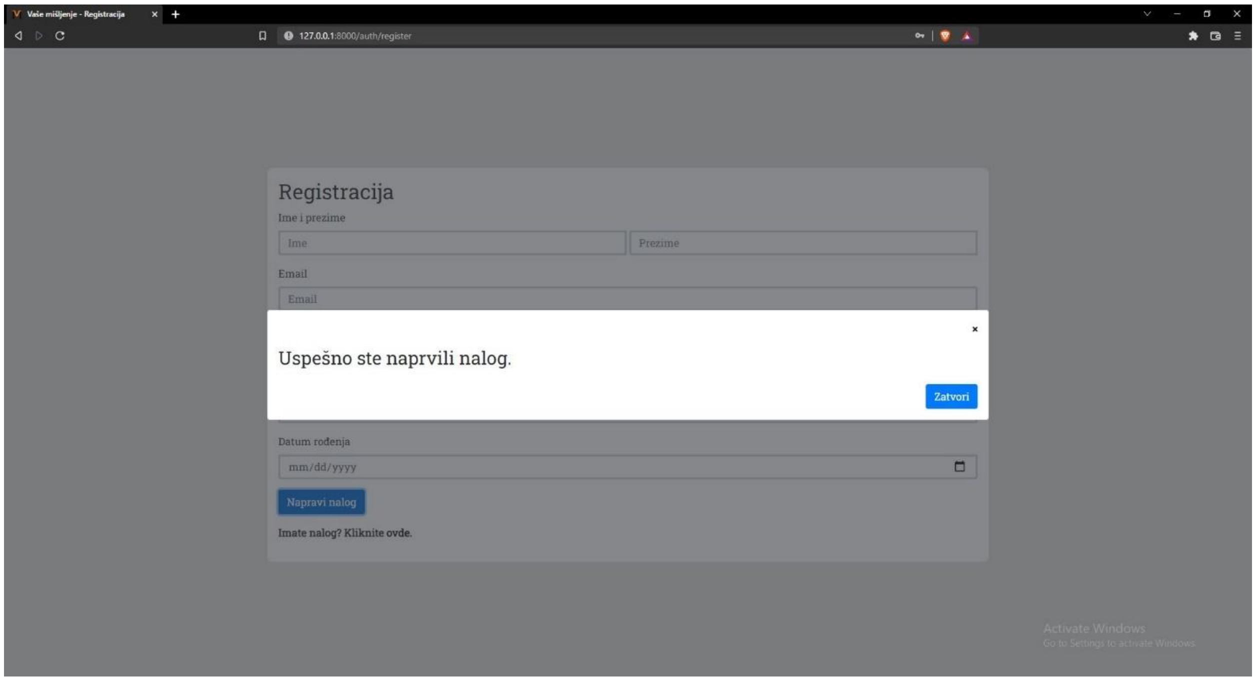
Datum rođenja

Izaberite datum rođenja

Imate nalog? Kliknite ovde.



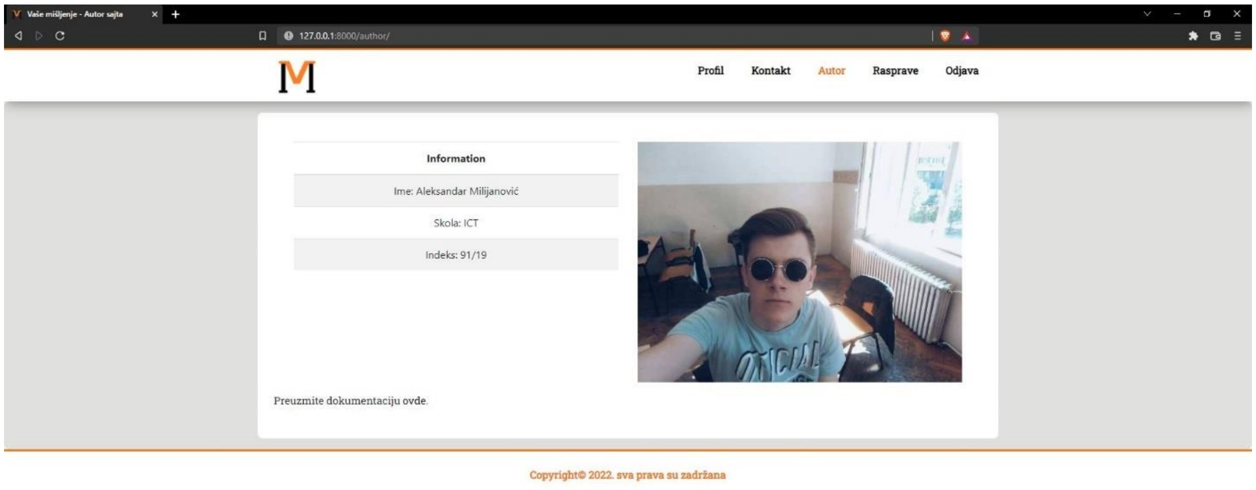
Unete vrednosti moraju biti u željenim formatima. Takođe lozinke se moraju poklapati.


Vaše mišljenje - Registracija
127.0.0.1:8000/auth/register
Registracija
Ime i prezime
Ime
Prezime
Email
Email
Datum rođenja
mm/dd/yyyy
Napravi nalog
Imate nalog? Kliknite ovde.
Zatvori
Activate Windows
Go to Settings to activate Windows.

Uspešno ste naprvili nalog.

4.3 Аутор

На овој страници је представљена кратка биографија аутора сајта, која је дохваћена из базе података, као и његова фотографија. Документација је доступна на овој страници.



Information

Ime: Aleksandar Milijanović

Skola: ICT

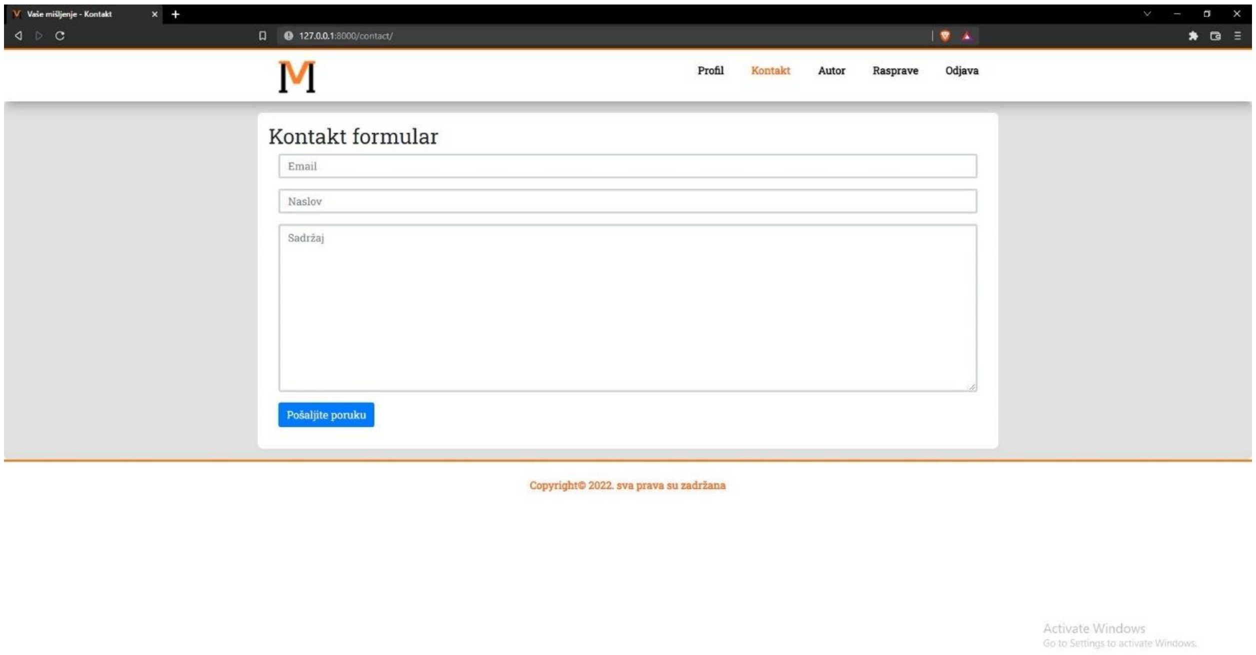
Indeks: 91/19

Preuzmite dokumentaciju ovde.

Copyright© 2022. sva prava su zadržana

4.4 Контакт

Страница омогућава слање e-mail-a преко датог формулара који се шаље администратору. Уколико подаци нису прихватљуви корисник се извештава о томе, као и о исходу слања.



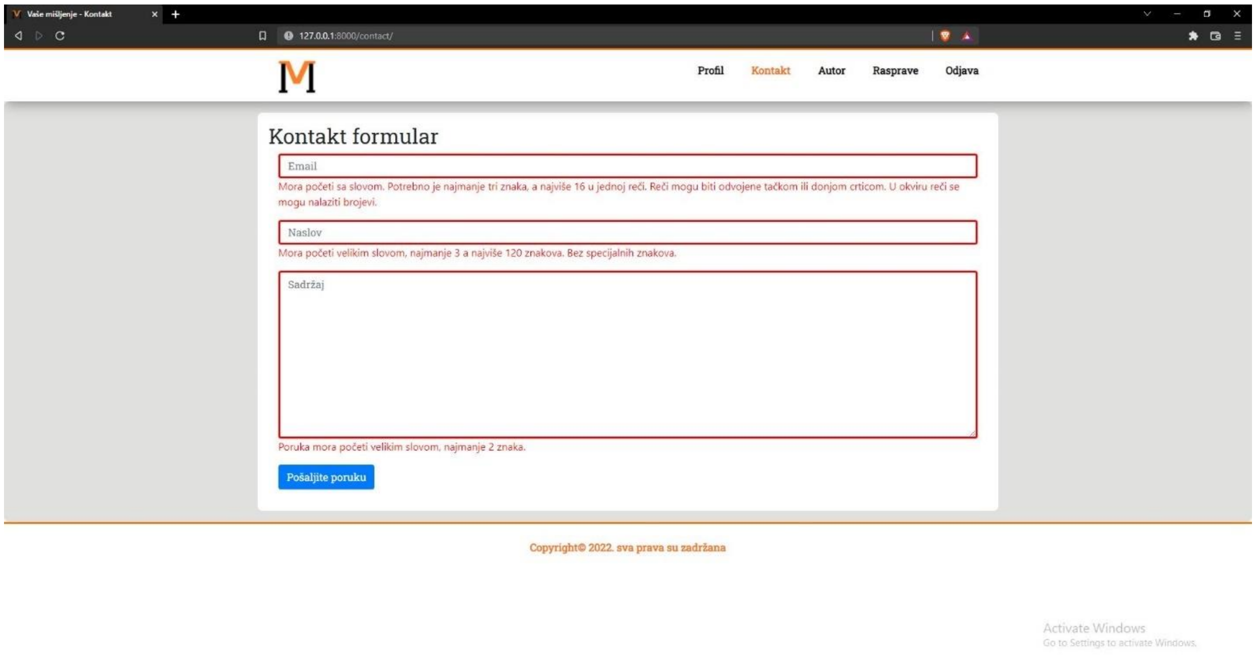
Kontakt formular

Email

Naslov

Sadržaj

Copyright© 2022. sva prava su zadržana



Kontakt formular

Email

Mora početi sa slovom. Potrebno je najmanje tri znaka, a najviše 16 u jednoj reči. Reči mogu biti odvojene tačkom ili donjom crticom. U okviru reči se mogu nalaziti brojevi.

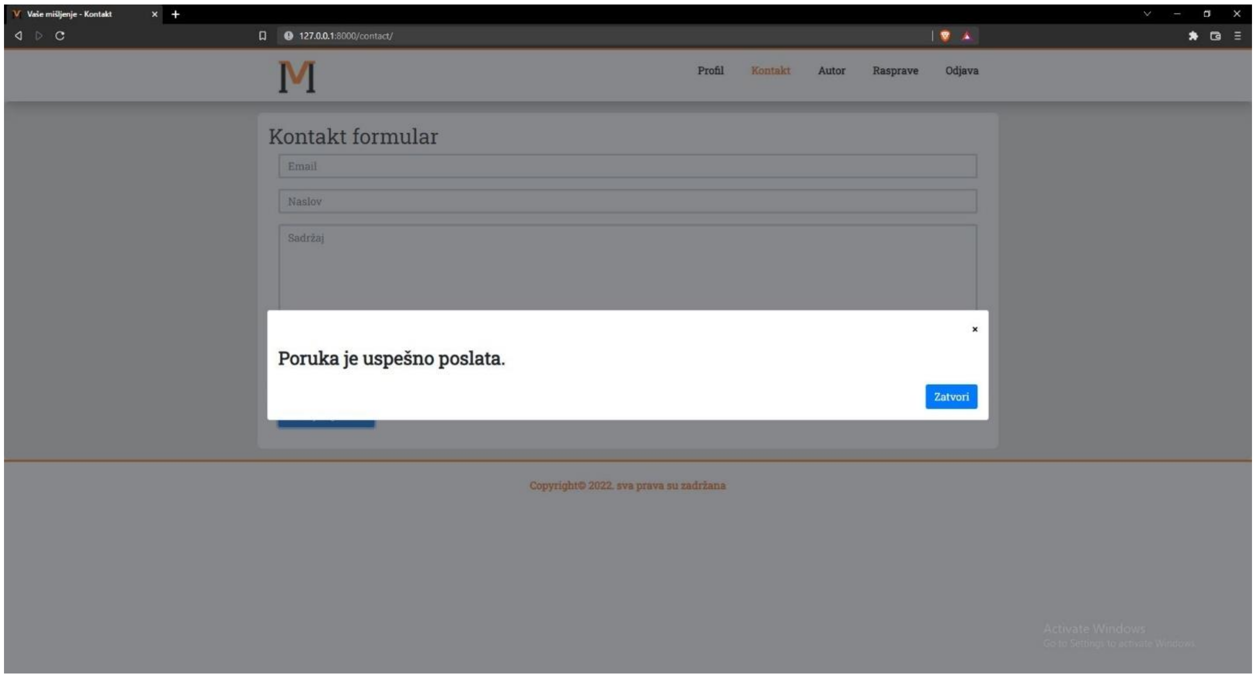
Naslov

Mora početi velikim slovom, najmanje 3 a najviše 120 znakova. Bez specijalnih znakova.

Sadržaj

Poruka mora početi velikim slovom, najmanje 2 znaka.

Copyright© 2022. sva prava su zadržana


Kontakt formular

Email

Naslov

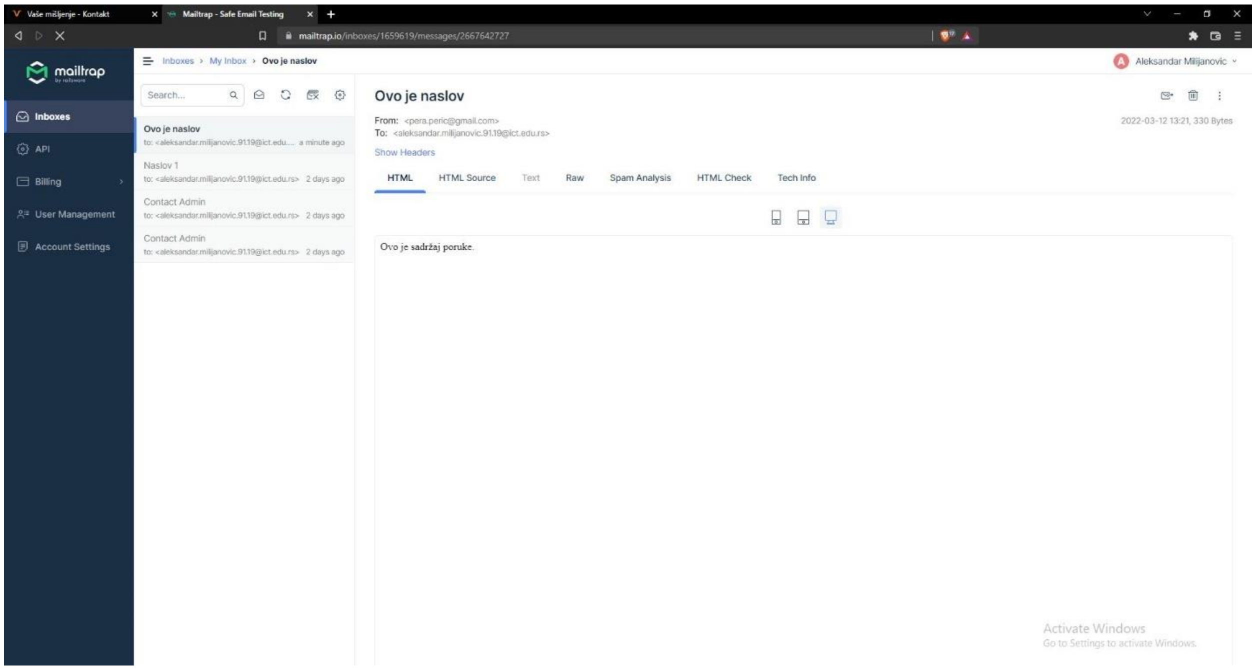
Sadržaj

Poruka je uspešno poslata.

Zatvori

Copyright© 2022. sva prava su zadržana

Овом приликом је коришћен MailTrap ([https://mailtrap.io/)](https://mailtrap.io/) преко кога сам сагледавао послате поруке.


Ovo je naslov

From: <pera.peric@gmail.com>

To: <aleksandar.miljanovic.91.19@ict.edu.rs>

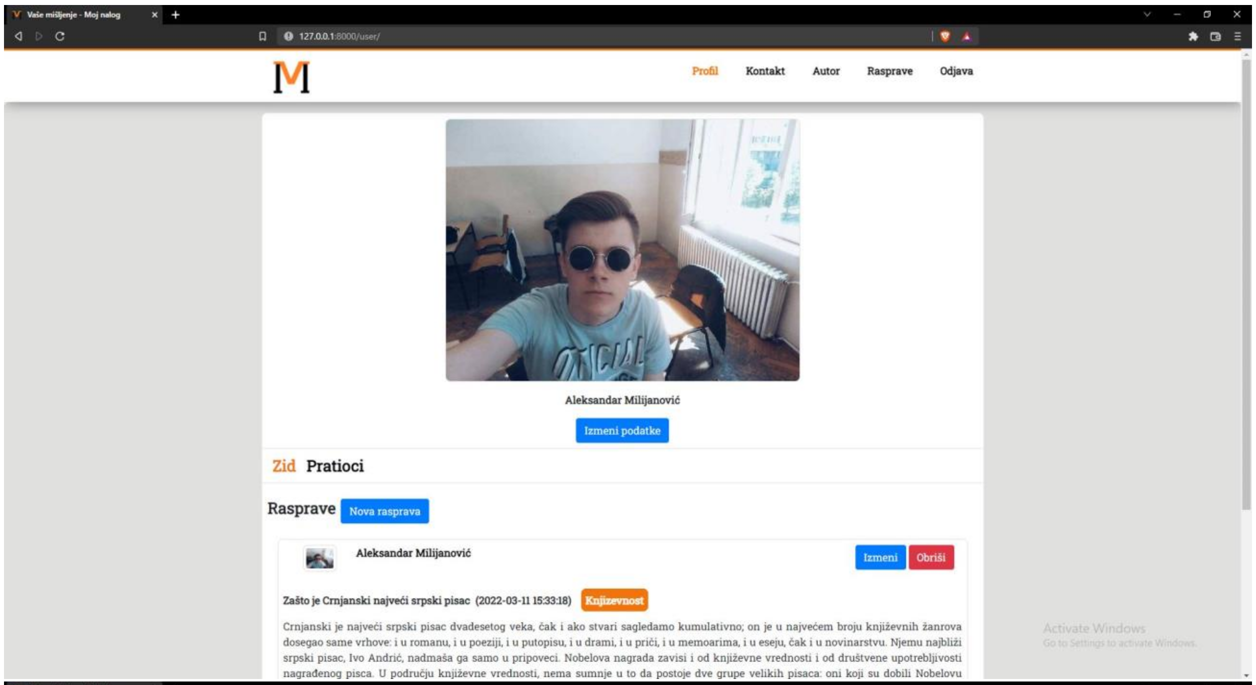
2022-03-12 13:21, 330 Bytes

Ovo je sadržaj poruke.

4.5 Профил

Преко ове странице корисник може увидети своју целокупну активност на апликацији у смислу на објаве које је објавио, њихове коментаре, лајкове као и друге кориснике које је запратио.

По услешном пријављивању, корисник се преусмерава директно на ову страницу. Могуће је направити нову објаву, изменити је или обрисати. Исто важи и за коментаре. Могуће је лајковање и одлајковање, одлазак на профил запраћеног корисника или отпратити га и додати коментар.



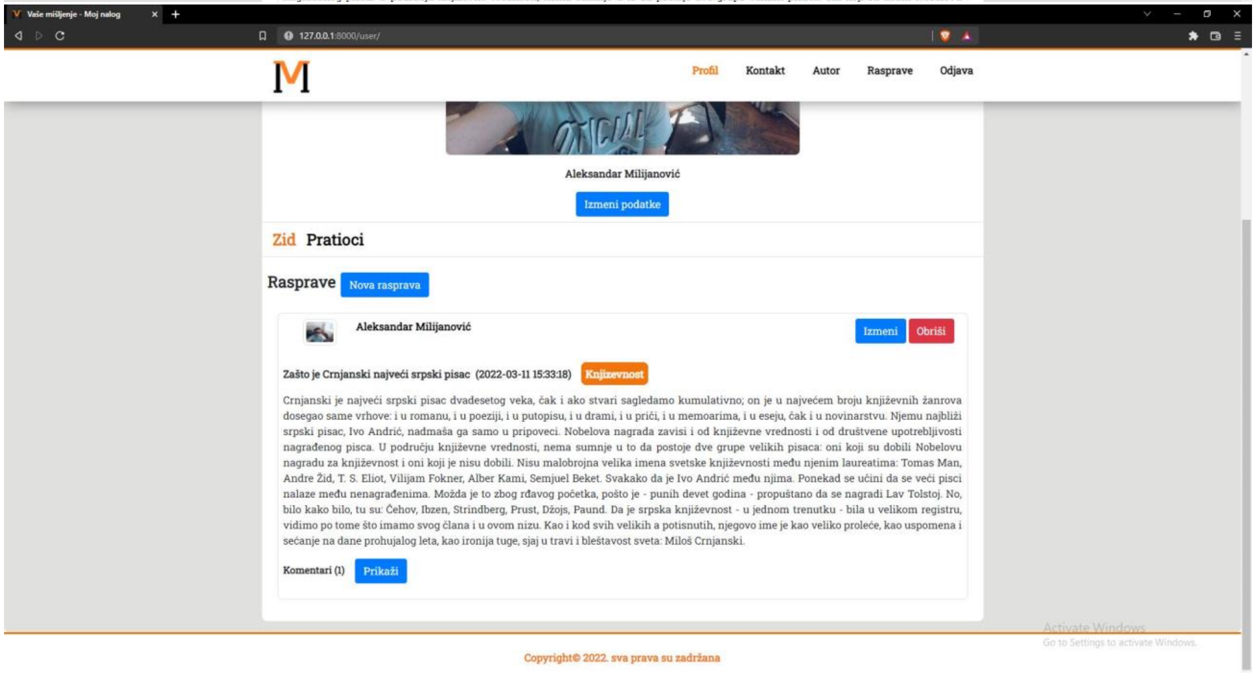
Aleksandar Milijanović

Rasprave

Aleksandar Milijanović

Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18)

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu



Aleksandar Milijanović

Rasprave

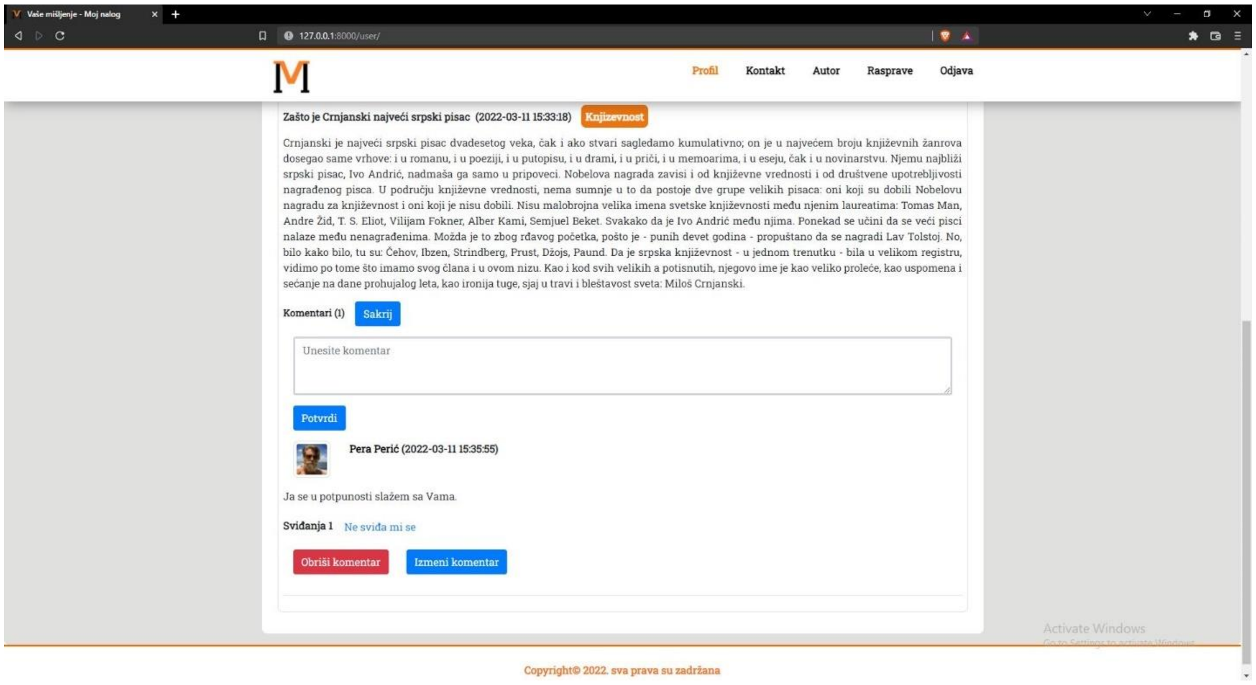
Aleksandar Milijanović

Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18)

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Vilijam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci nalaze među nenagrađenima. Možda je to zbog rđavog početka, pošto je - punih devet godina - propuštano da se nagradi Lav Tolstoj. No, bilo kako bilo, tu su: Čehov, Ibzen, Strindberg, Prust, Džojs, Paund. Da je srpska književnost - u jednom trenutku - bila u velikom registru, vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1)

Copyright© 2022. sva prava su zadržana



Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18) Knjizevnost

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Vilijam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci nalaze među nenagrađenima. Možda je to zbog rđavog početka, pošto je - punih devet godina - propuštano da se nagradi Lav Tolstoj. No, bilo kako bilo, tu su: Čehov, Ibzen, Strindberg, Prust, Džojs, Paund. Da je srpska književnost - u jednom trenutku - bila u velikom registru, vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1) Sakrij

Unesite komentar

Potvrdi

Pera Perić (2022-03-11 15:35:55)

Ja se u potpunosti slažem sa Vama.

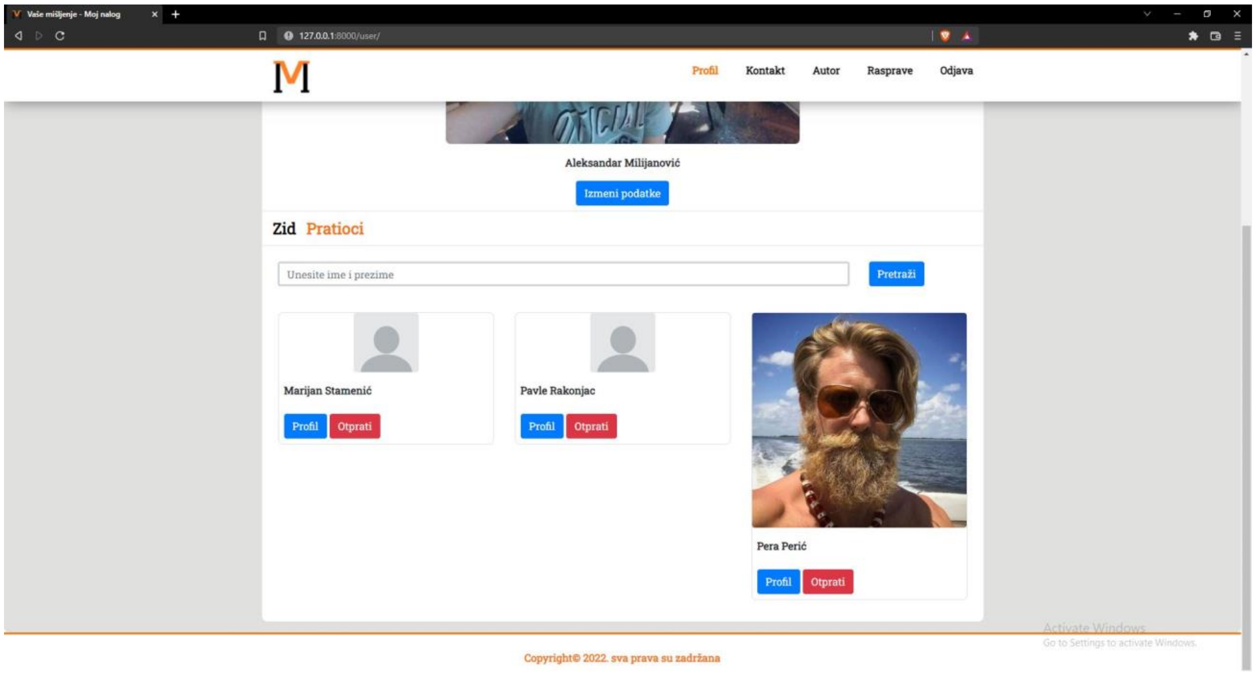
Sviđanja 1 Ne sviđa mi se

Obriši komentar Izmeni komentar

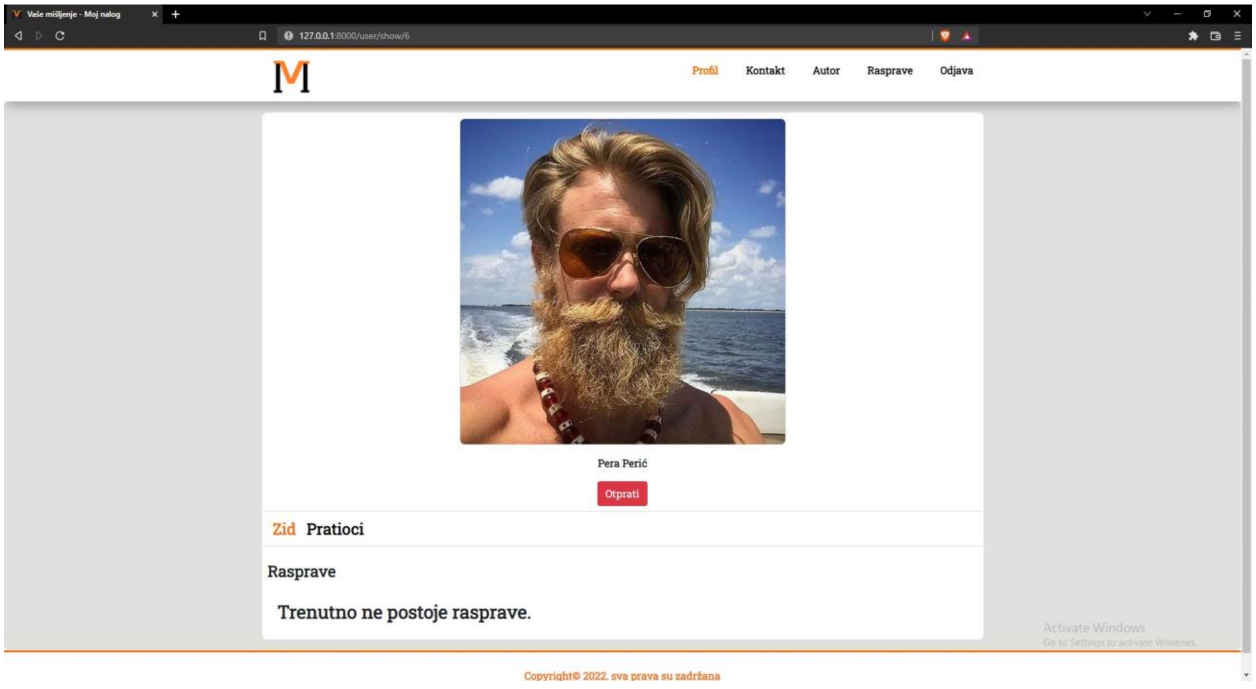
Activate Windows
Go to Settings to activate Windows.

Copyright© 2022. sva prava su zadržana

Преко ове секције се претражују пратиоци, преусмеравање на њихове профиле или отпраћивање.



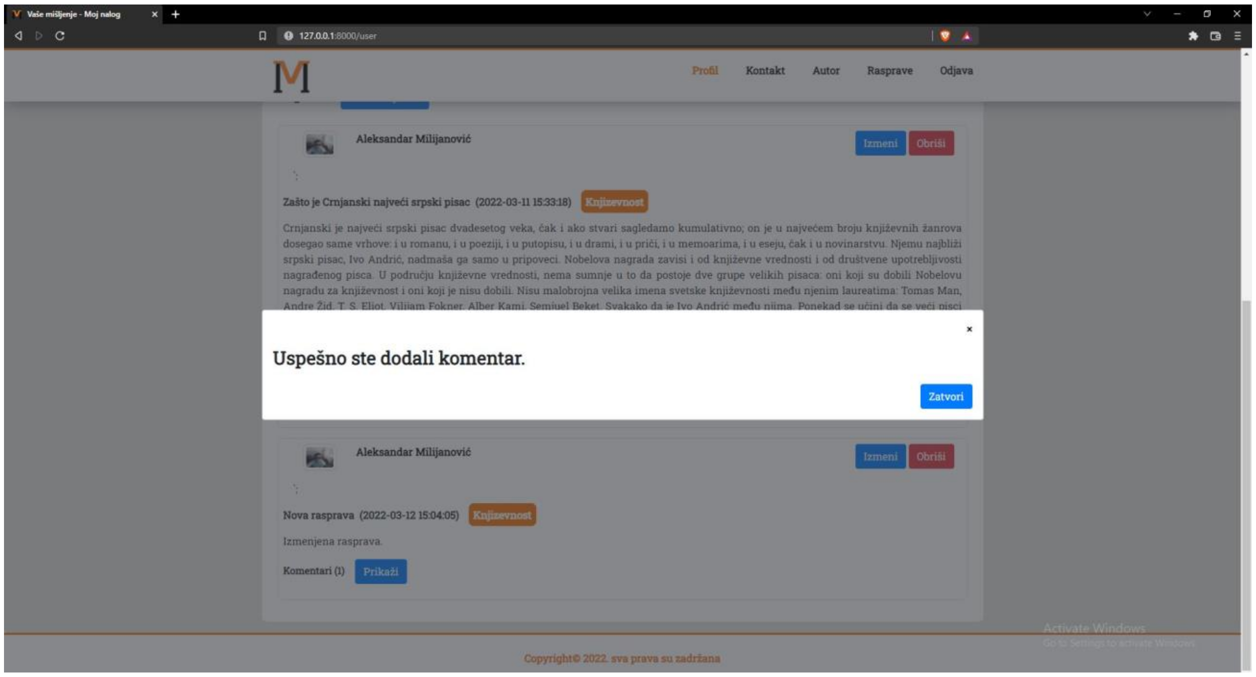
Уколико посетимо пратиочев профил имамо следећи приказ:



Rasprave

Trenutno ne postoje rasprave.

Copyright© 2022. sva prava su zadržana



Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18)

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Viljiam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci

Uspešno ste dodali komentar.

Nova rasprava (2022-03-12 15:04:05)

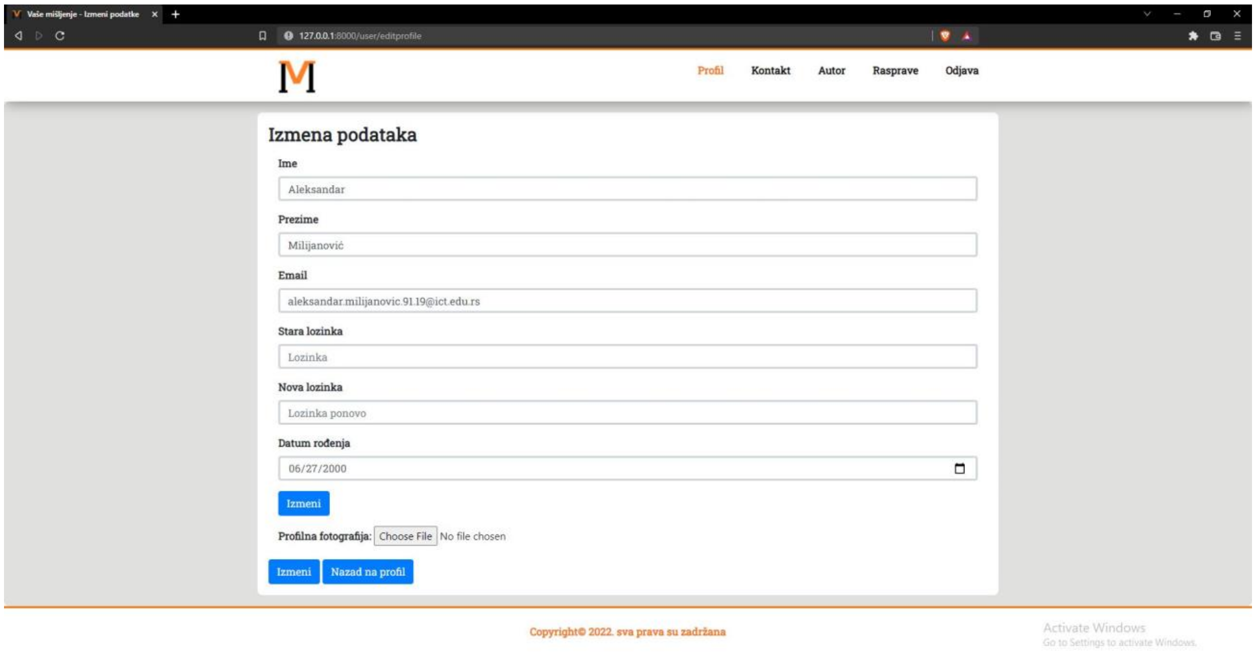
Izmenjena rasprava.

Komentari (1)

Copyright© 2022. sva prava su zadržana

4.6 Измена корисникових података

На овој страници је могућа промена података (имена, презимена, лозинке, e-mail-a и профилне фотографије). Можете приметити да постоје два тастера за потврду. То сам урадио зато што се сви подаци изузев фотографије шаљу путем Ajax-a. Услед уноса недозвољених вредности се приказују поруке испод уносних поља и приказује модал са одређеном поруком. Потребно је истаћи да се приликом учитавања странице формулар попуњава са корисниковим подацима и повраћај на профил се чини кликом на линк у подножу формулара.


Vaše mišljenje - Izmeni podatke
127.0.0.1:8000/user/editprofile
Profil
Kontakt
Autor
Rasprave
Odjava

Izmena podataka

Ime: Aleksandar

Prezime: Milijanović

Email: aleksandar.milijanovic.91.19@ict.edu.rs

Stara lozinka: Lozinka

Nova lozinka: Lozinka ponovo

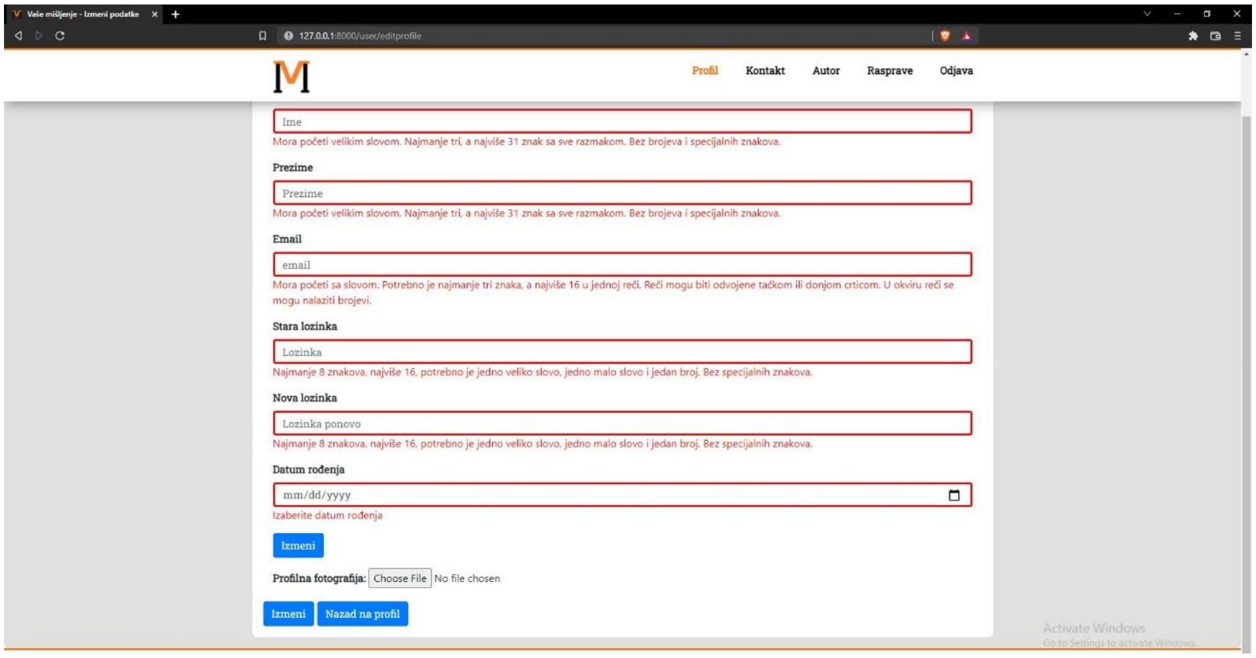
Datum rođenja: 06/27/2000

Izmeni

Profilna fotografija: Choose File No file chosen

Izmeni Nazad na profil

Copyright© 2022. sva prava su zadržana


Vaše mišljenje - Izmeni podatke
127.0.0.1:8000/user/editprofile
Profil
Kontakt
Autor
Rasprave
Odjava

Ime

Mora početi velikim slovom. Najmanje tri, a najviše 31 znak sa sve razmakom. Bez brojeva i specijalnih znakova.

Prezime: Prezime

Mora početi velikim slovom. Najmanje tri, a najviše 31 znak sa sve razmakom. Bez brojeva i specijalnih znakova.

Email: email

Mora početi sa slovom. Potrebno je najmanje tri znaka, a najviše 16 u jednoj reči. Reči mogu biti odvojene tačkom ili donjom crticom. U okviru reči se mogu nalaziti brojevi.

Stara lozinka: Lozinka

Najmanje 8 znakova, najviše 16, potrebno je jedno veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih znakova.

Nova lozinka: Lozinka ponovo

Najmanje 8 znakova, najviše 16, potrebno je jedno veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih znakova.

Datum rođenja: mm/dd/yyyy

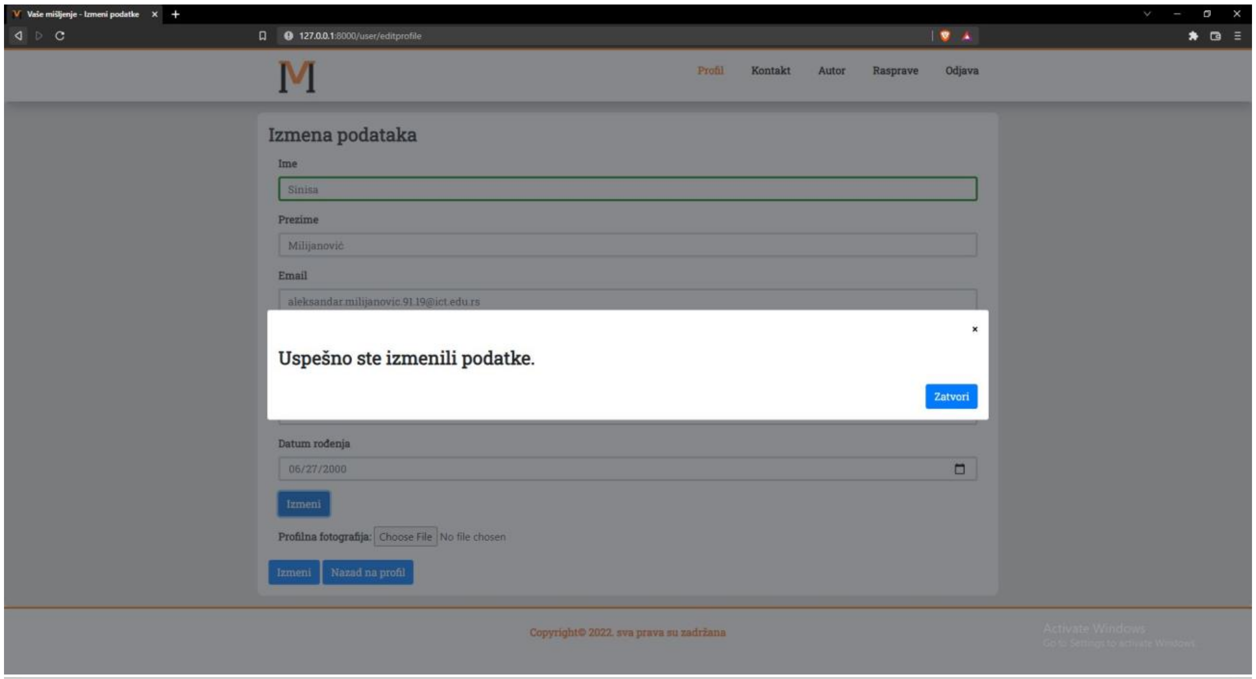
Izaberite datum rođenja

Izmeni

Profilna fotografija: Choose File No file chosen

Izmeni Nazad na profil

Уколико је измена прошла успешно приказује се следећа порука.


Vaše mišljenje - Izmeni podatke
127.0.0.1:8000/user/editprofile
Profil
Kontakt
Autor
Rasprave
Odjava

Izmena podataka

Ime

Sinisa

Prezime

Milijanović

Email

aleksandar.milijanovic.91.19@ict.edu.rs

Uspešno ste izmenili podatke.

Zatvori

Datum rođenja

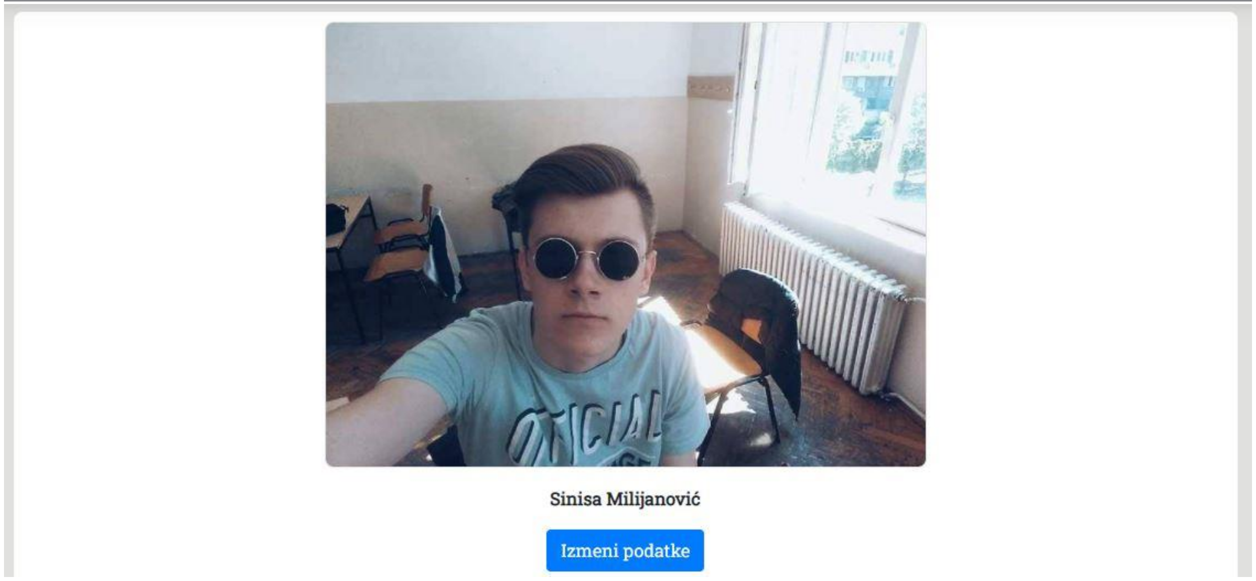
06/27/2000

Izmeni

Profilna fotografija: Choose File No file chosen

Izmeni Nazad na profil

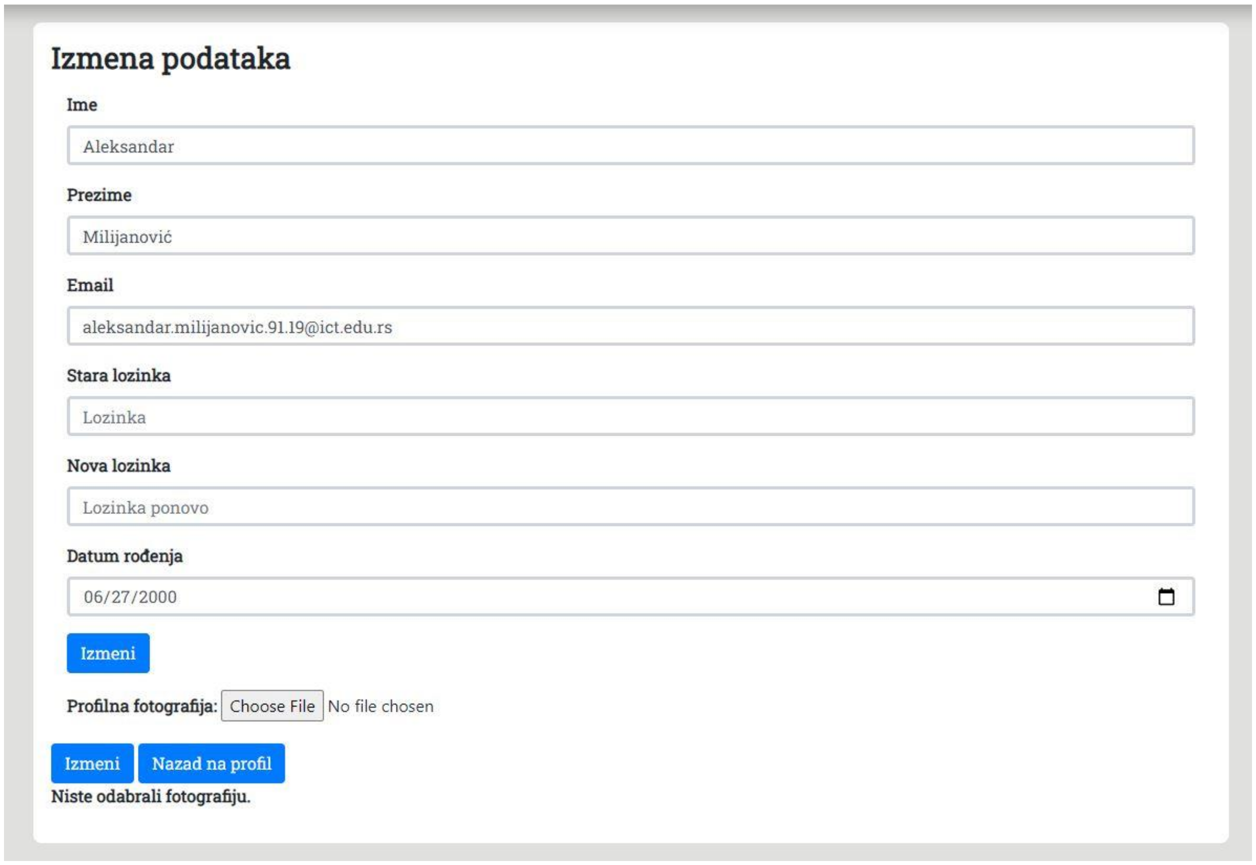
Copyright© 2022. sva prava su zadržana



Sinisa Milijanović

Izmeni podatke

Када је реч о измени слике, порука се исписује испод тастера за потврду.



Izmena podataka

Ime

Aleksandar

Prezime

Milijanović

Email

aleksandar.milijanovic.91.19@ict.edu.rs

Stara lozinka

Lozinka

Nova lozinka

Lozinka ponovo

Datum rođenja

06/27/2000

Izmeni

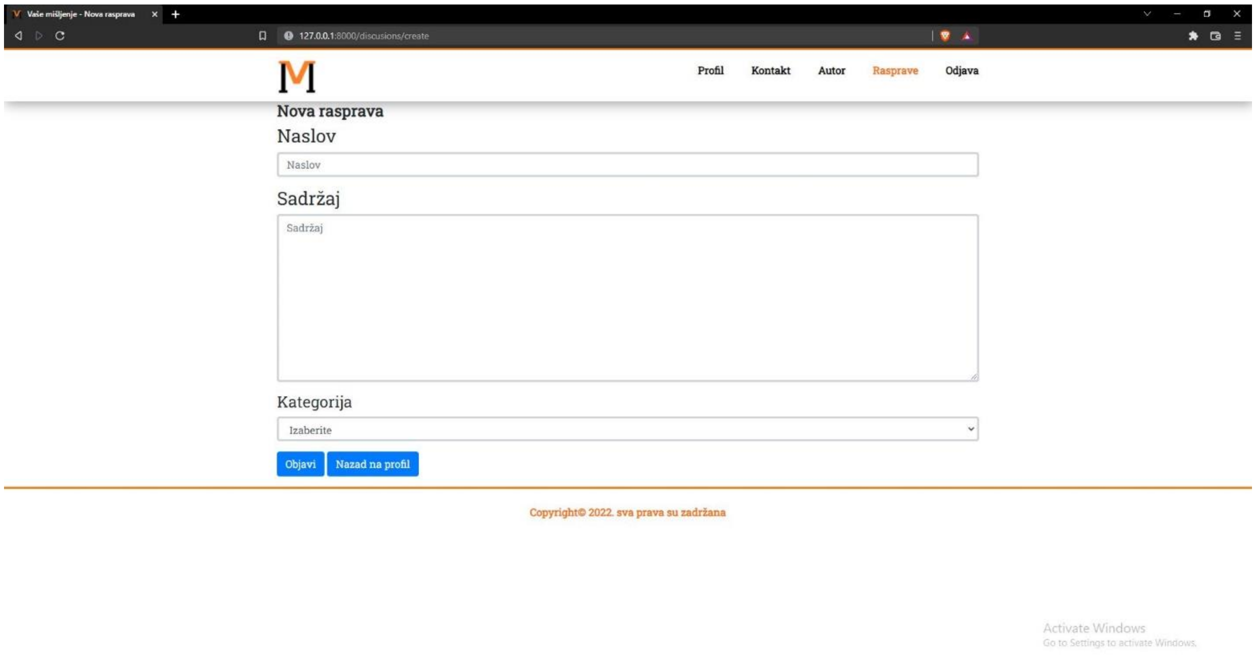
Profilna fotografija: Choose File No file chosen

Izmeni Nazad na profil

Niste odabrali fotografiju.

4.7 Креирање објаве

Ова страница пружа могућност прављења нове објаве преко доступног формулара. Могуће је и враћање на профил преко тастера који је поред тастера за потврду. Приликом уноса неприхватљивих вредности поруке се приказују као и модал са одређеном поруком.


Vaše mišljenje - Nova rasprava
127.0.0.1:8000/discusions/create
Profil
Kontakt
Autor
Rasprave
Odjava

Nova rasprava

Naslov

Naslov

Sadržaj

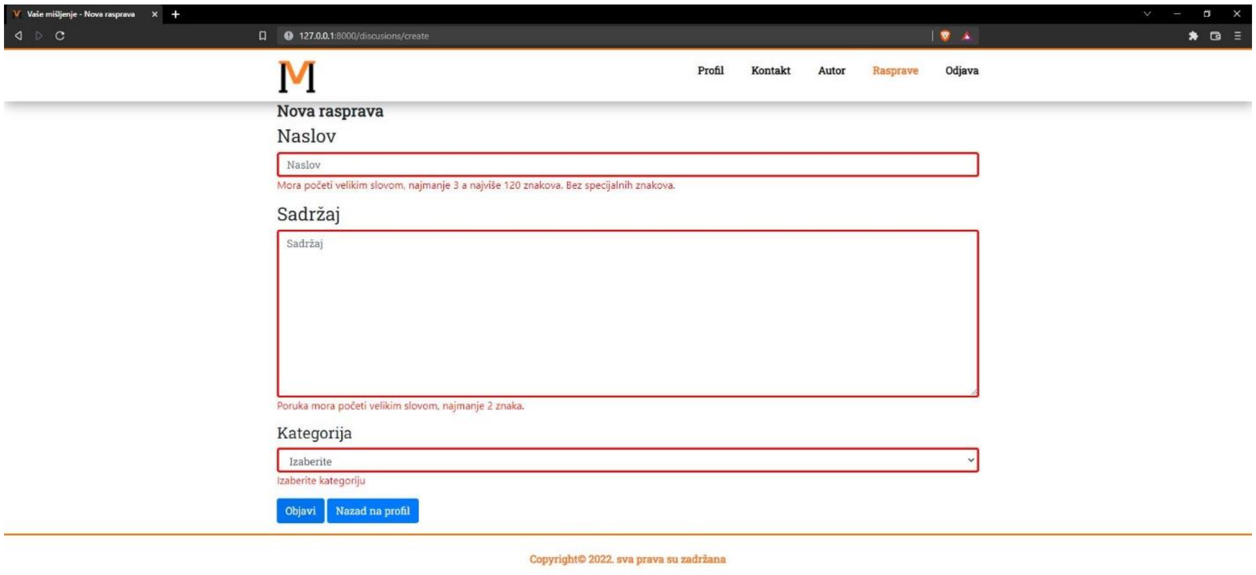
Sadržaj

Kategorija

Izaberite

Objavi Nazad na profil

Copyright© 2022. sva prava su zadržana


Vaše mišljenje - Nova rasprava
127.0.0.1:8000/discusions/create
Profil
Kontakt
Autor
Rasprave
Odjava

Nova rasprava

Naslov

Naslov

Mora početi velikim slovom, najmanje 3 a najviše 120 znakova. Bez specijalnih znakova.

Sadržaj

Sadržaj

Poruka mora početi velikim slovom, najmanje 2 znaka.

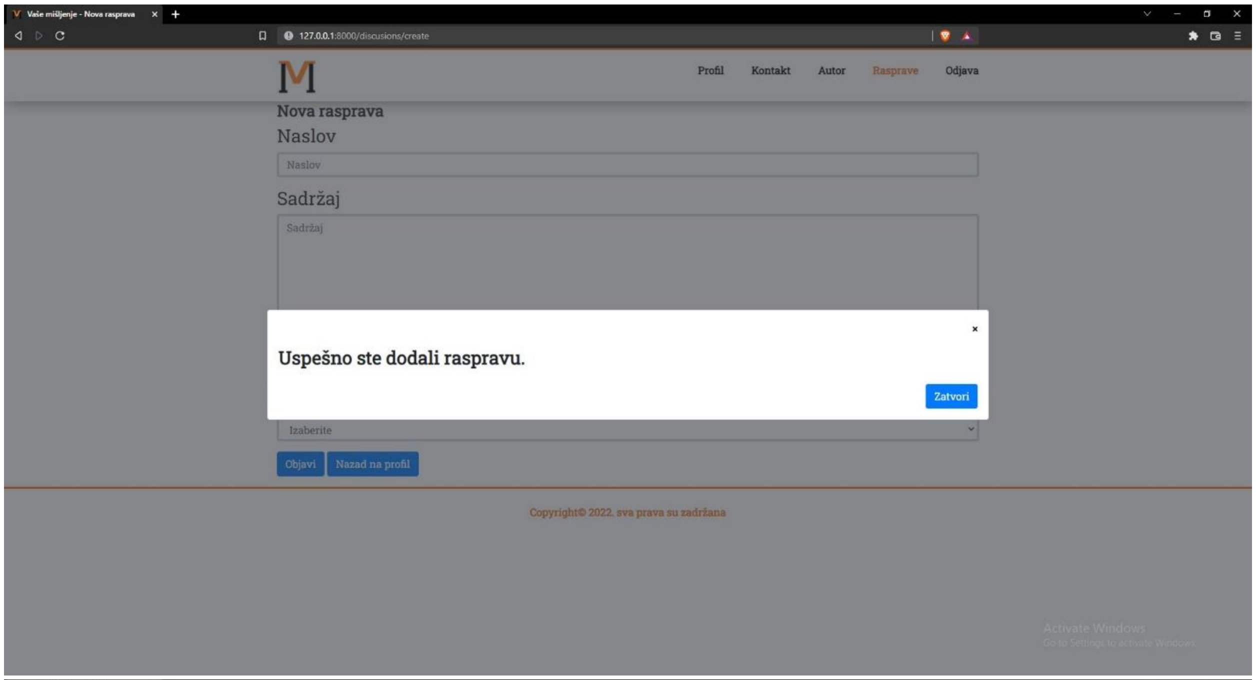
Kategorija

Izaberite

Izaberite kategoriju

Objavi Nazad na profil

Copyright© 2022. sva prava su zadržana


Vaše mišljenje - Nova rasprava
127.0.0.1:8000/discusions/create
Profil
Kontakt
Autor
Rasprave
Odjava

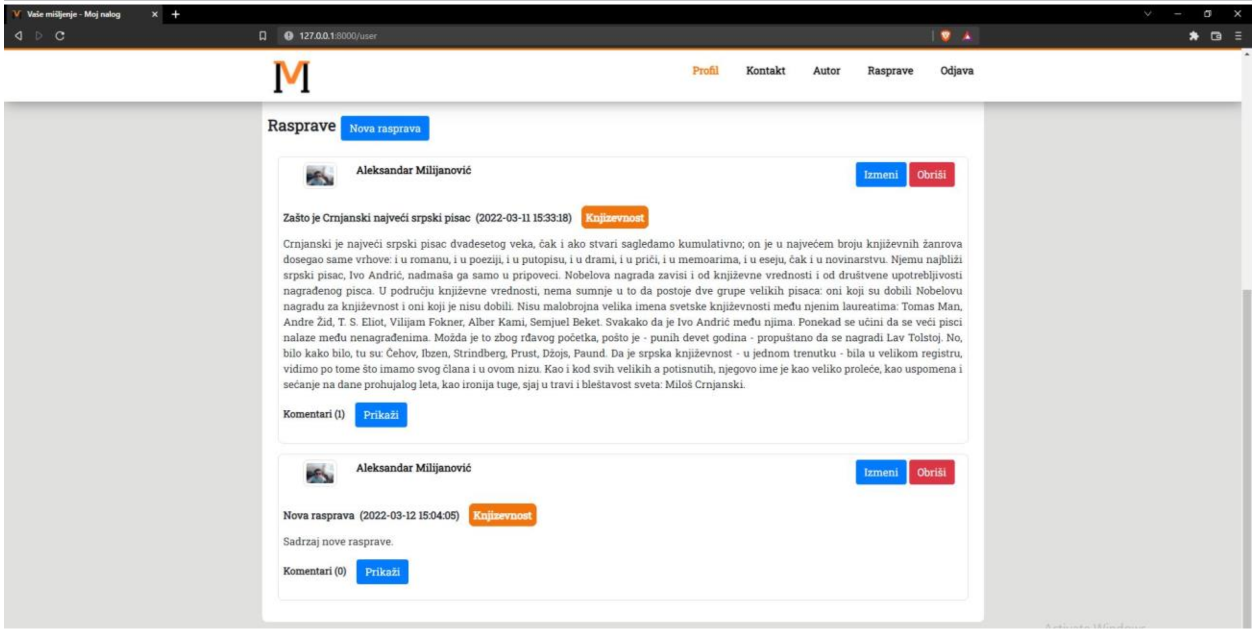
Nova rasprava

Naslov

Sadržaj

Uspešno ste dodali raspravu.

Copyright© 2022. sva prava su zadržana



Rasprave

Aleksandar Milijanović

Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18) Knjizevnost

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Vilijam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci nalaze među nenagrađenima. Možda je to zbog rđavog početka, pošto je - punih devet godina - propuštano da se nagradi Lav Tolstoj. No, bilo kako bilo, tu su: Čehov, Ibzen, Strindberg, Prust, Džojs, Paund. Da je srpska književnost - u jednom trenutku - bila u velikom registru, vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1)

Aleksandar Milijanović

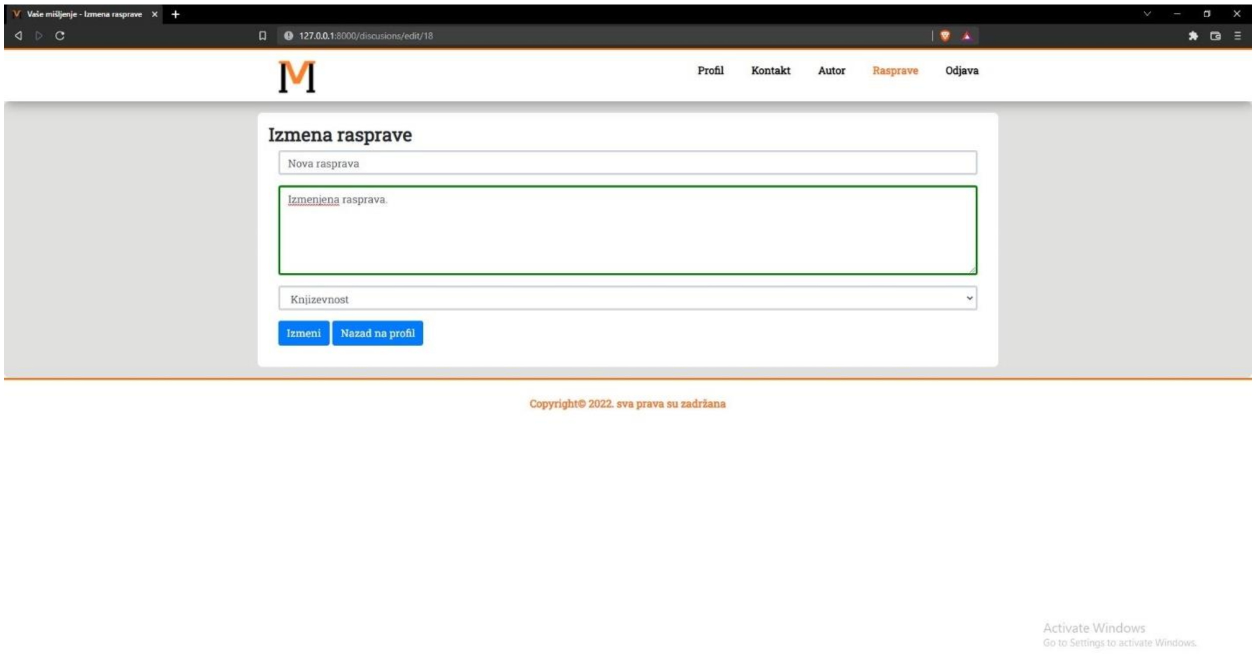
Nova rasprava (2022-03-12 15:04:05) Knjizevnost

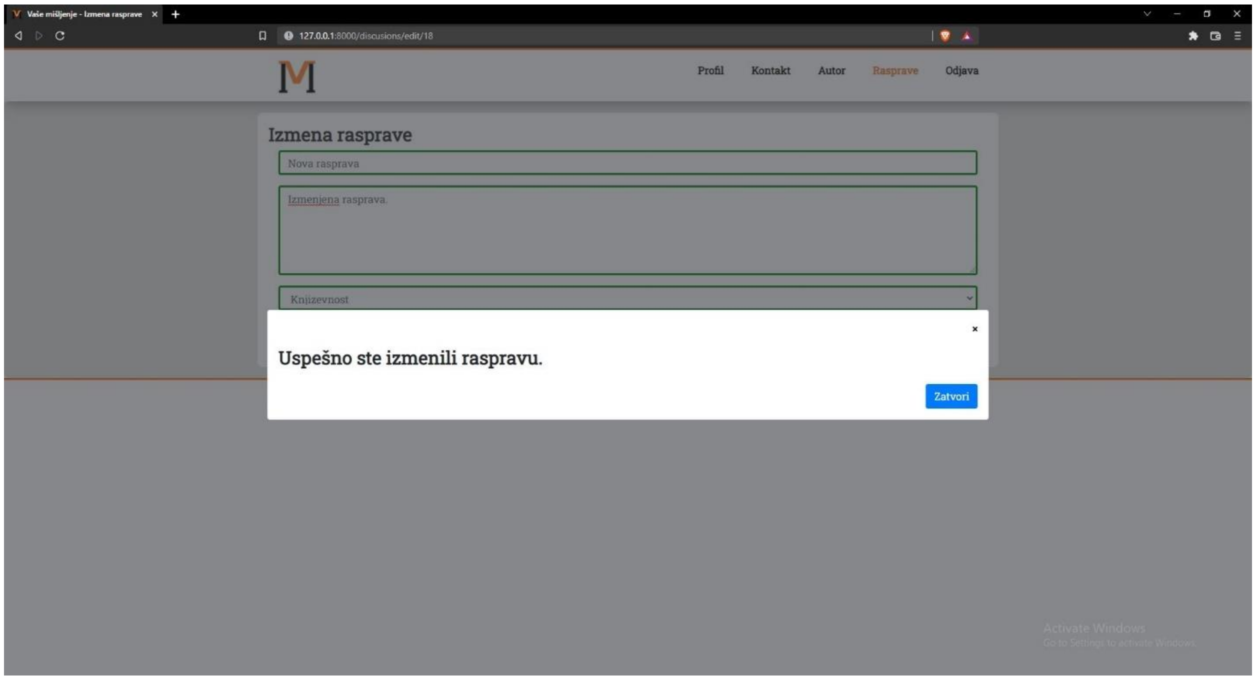
Sadrzaj nove rasprave.

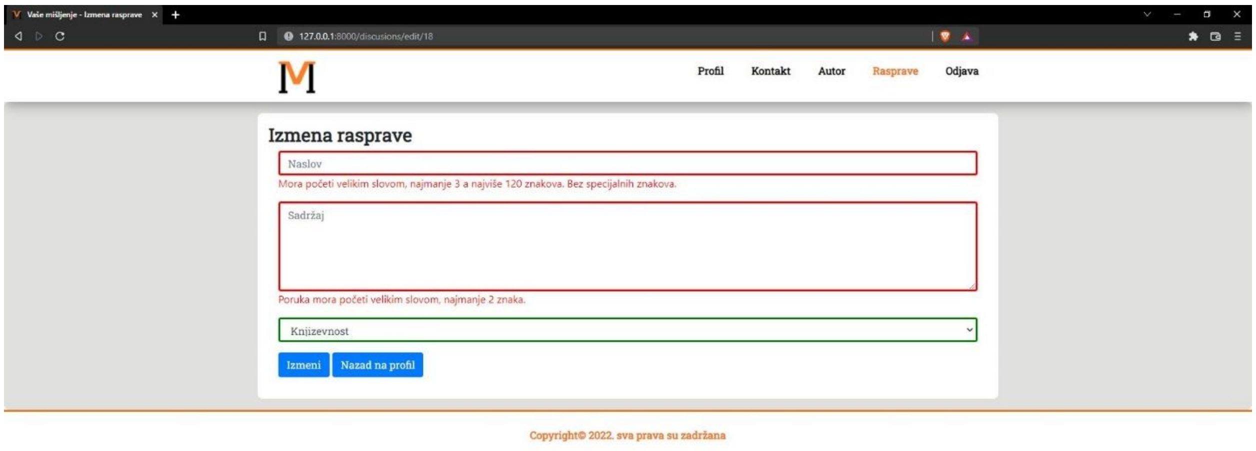
Komentari (0)

4.8 Измена објаве

На овој страници је могуће изменити објаву, вратити се на профил преко споредног линка. Услед уноса неприхватљивих вредности корисник се извештава, као и за исход измене преко модала. Вредности које објава поседује се упису унутар уносних поља приликом учитавања странице.




Vaše mišljenje - Izmena rasprave
127.0.0.1:8000/discusions/edit/18
Izmena rasprave
Nova rasprava
Izmenjena rasprava.
Knjizevnost
Uspešno ste izmenili raspravu.
Zatvori


Vaše mišljenje - Izmena rasprave
127.0.0.1:8000/discusions/edit/18
Profil
Kontakt
Autor
Rasprave
Odjava

Izmena rasprave

Naslov

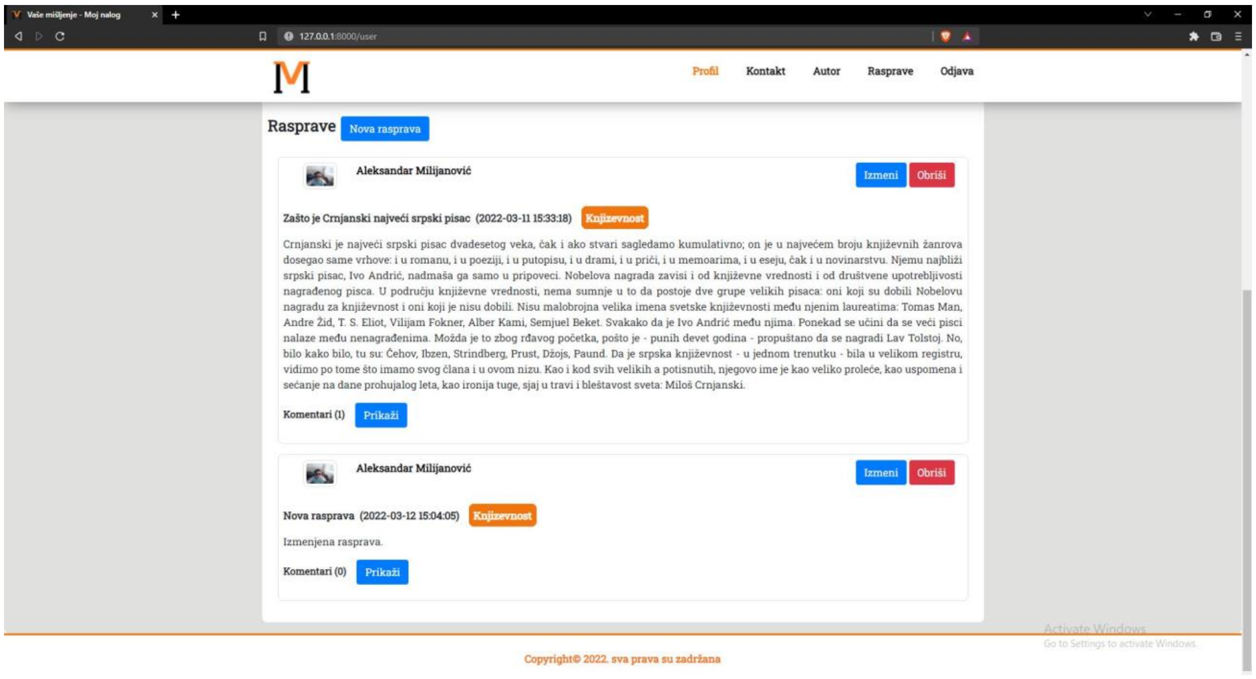
Mora početi velikim slovom, najmanje 3 a najviše 120 znakova. Bez specijalnih znakova.

Sadržaj

Poruka mora početi velikim slovom, najmanje 2 znaka.

Knjizevnost

Copyright© 2022. sva prava su zadržana



Rasprave

Aleksandar Milijanović

Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18) Knjizevnost

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Vilijam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci nalaze među nenagrađenima. Možda je to zbog rđavog početka, pošto je - punih devet godina - propuštano da se nagradi Lav Tolstoj. No, bilo kako bilo, tu su: Čehov, Ibzen, Strindberg, Prust, Džojs, Paund. Da je srpska književnost - u jednom trenutku - bila u velikom registru, vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1)

Aleksandar Milijanović

Nova rasprava (2022-03-12 15:04:05) Knjizevnost

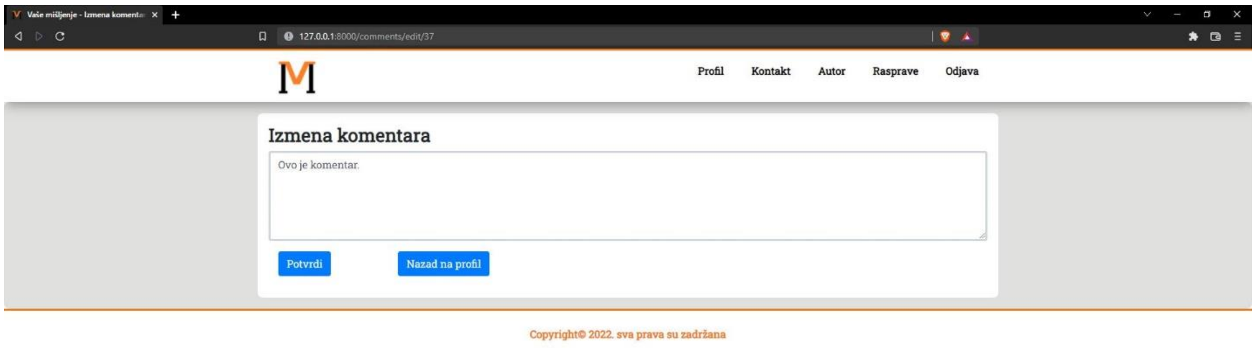
Izmenjena rasprava.

Komentari (0)

Copyright© 2022. sva prava su zadržana

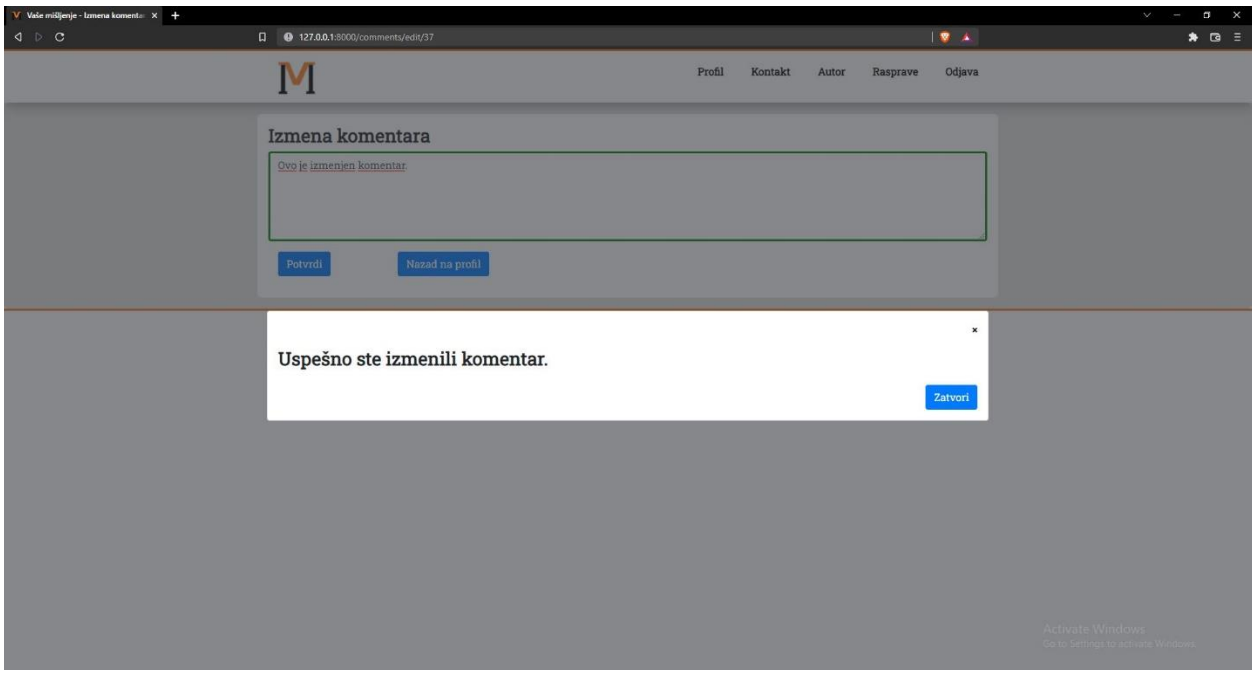
4.9 Измена коментара

Ова страница омогућава измену коментара преко доступног формулара. Могуће је и враћање на профил преко тастера који је поред тастера за потврду. Приликом уноса неприхватљивих вредности поруке се приказују као и модал са одређеном поруком.


Izmena komentara

Ovo je komentar.

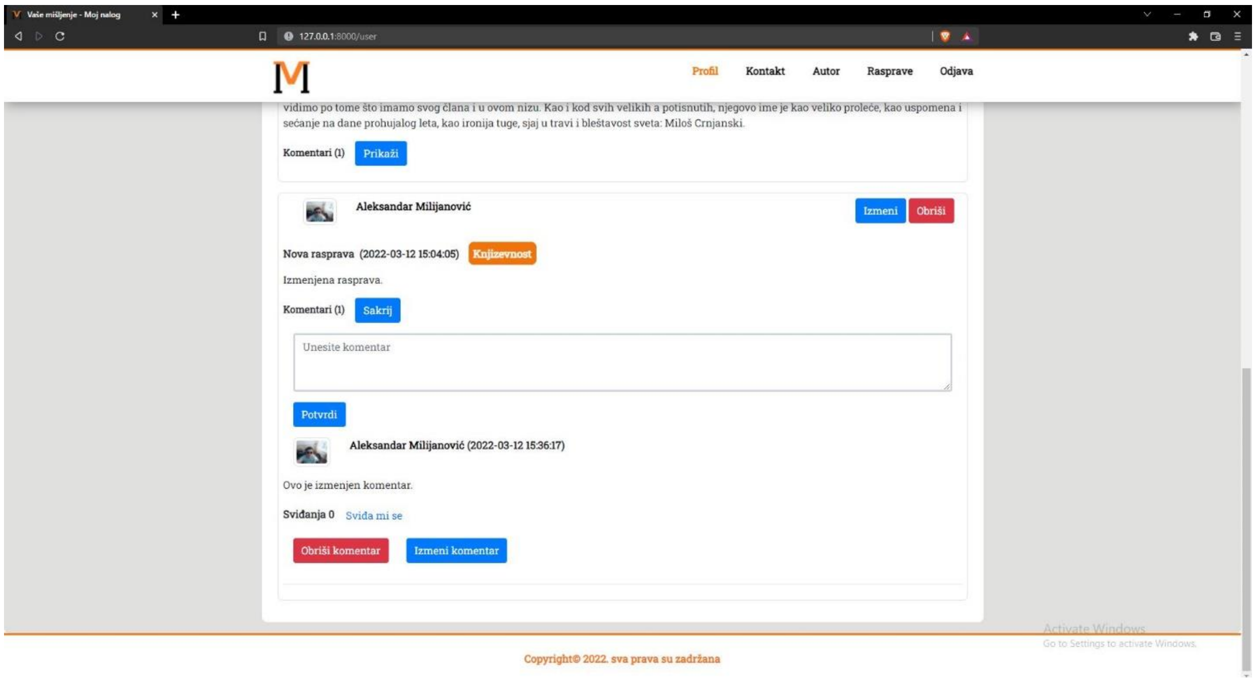
Vaše mišljenje - Izmena komentara
127.0.0.1:8000/comments/edit/37
Profil
Kontakt
Autor
Rasprave
Odjava
Potvrdi
Nazad na profil
Copyright© 2022. sva prava su zadržana
Activate Windows
Go to Settings to activate Windows.


Izmena komentara

Ovo je izmenjen komentar.

Uspešno ste izmenili komentar.

Vaše mišljenje - Izmena komentara
127.0.0.1:8000/comments/edit/37
Profil
Kontakt
Autor
Rasprave
Odjava
Potvrdi
Nazad na profil
×
Zatvori


Vaše mišljenje - Moj nalog

127.0.0.1:8000/user

vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1) Prikaži

Aleksandar Milijanović Izmeni Obriši

Nova rasprava (2022-03-12 15:04:05) Knjizevnost

Izmenjena rasprava.

Komentari (1) Sakrij

Unesite komentar

Potvrdi

Aleksandar Milijanović (2022-03-12 15:36:17)

Ovo je izmenjen komentar.

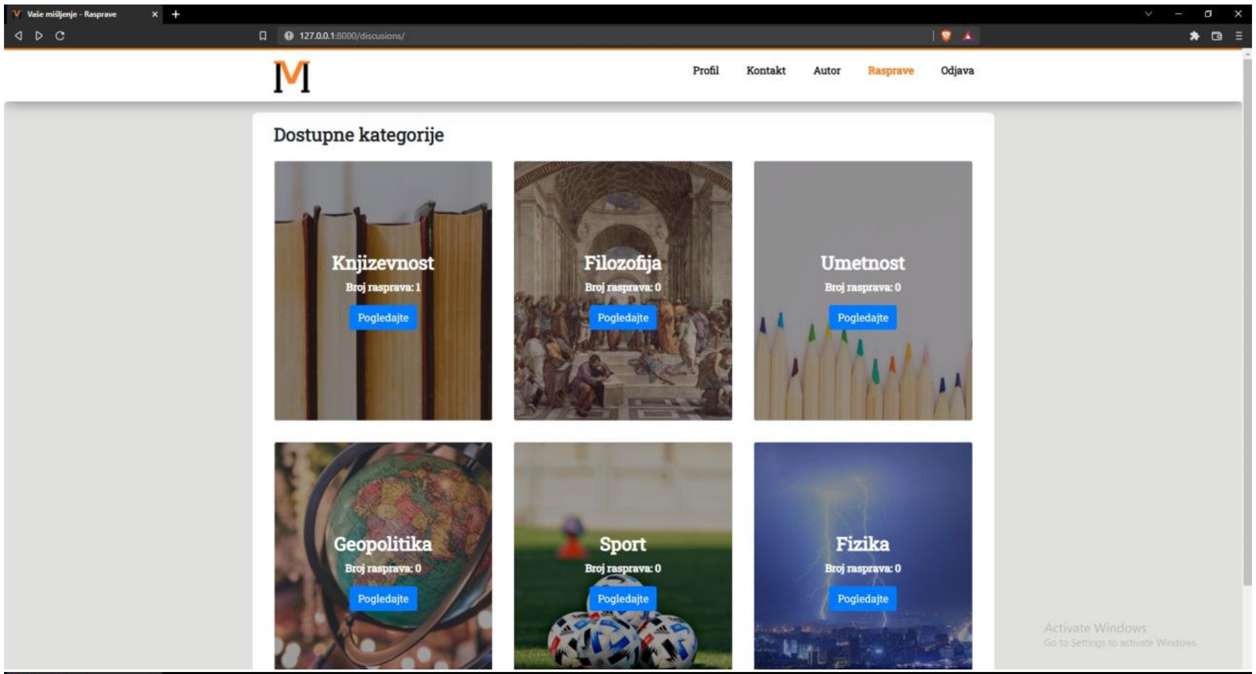
Sviđanja 0 Sviđa mi se

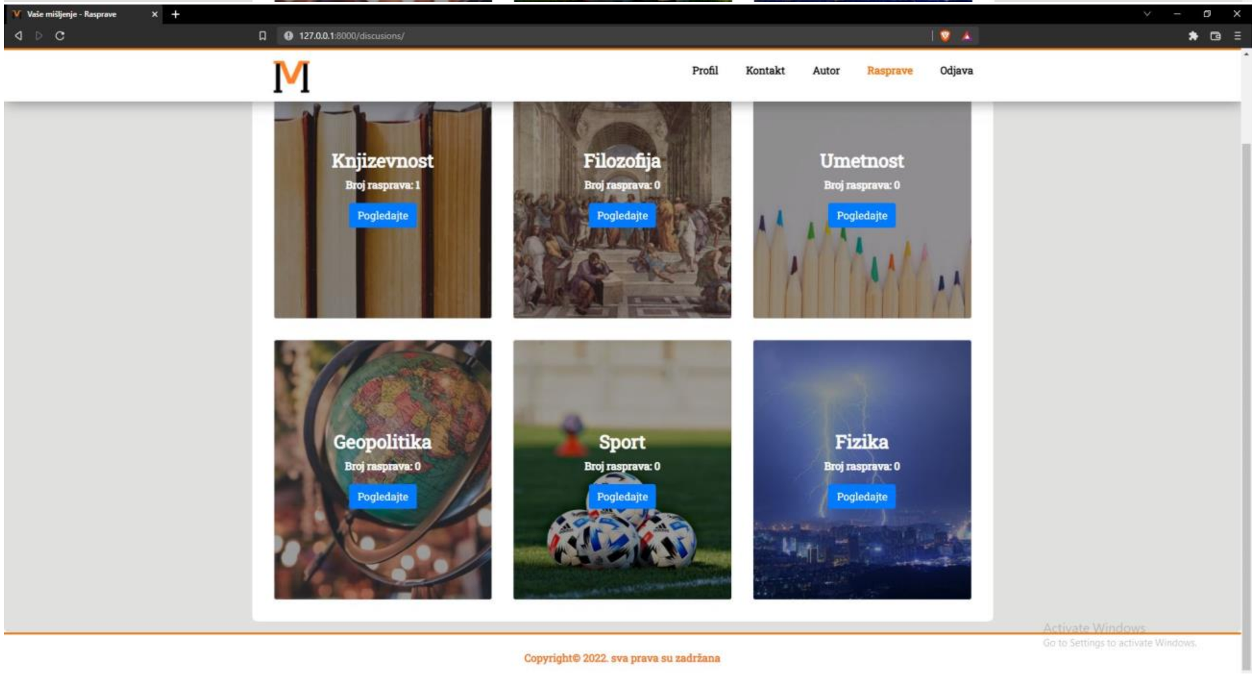
Obriši komentar Izmeni komentar

Copyright© 2022. sva prava su zadržana

4.10 Расправе

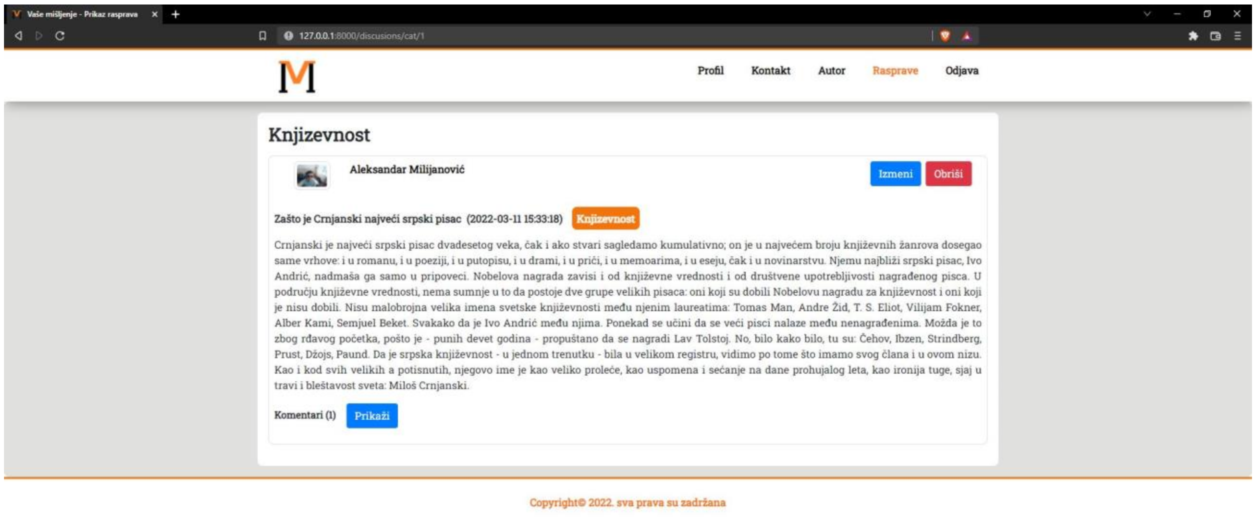
На овој страници корисник се може упознати са могућим категоријама, односно врстама расправа које се на друштвеној мрежи одвијају. Приказан је број расправа који припадају свакој категорији понаособ. Кликом на тастер, корисник бива преусмерен на страницу где су доступне само објаве које припадају датој категорији.





4.11 Расправе одређене категорије

На овој страници је могуће коментарисати и лајковати објаве других, као и приступити станицама за измену и обрисати како објаве тако и коментаре. Могуће је отићи на профиле других корисника који су у одабраној секцији нешто објавили.



Vaše mišljenje - Prikaz rasprava
127.0.0.1:8000/discusions/cat/1
Profil
Kontakt
Autor
Rasprave
Odjava
Izmeni
Obriši
Knjizevnost
Prikaži

Knjizevnost

Aleksandar Milijanović

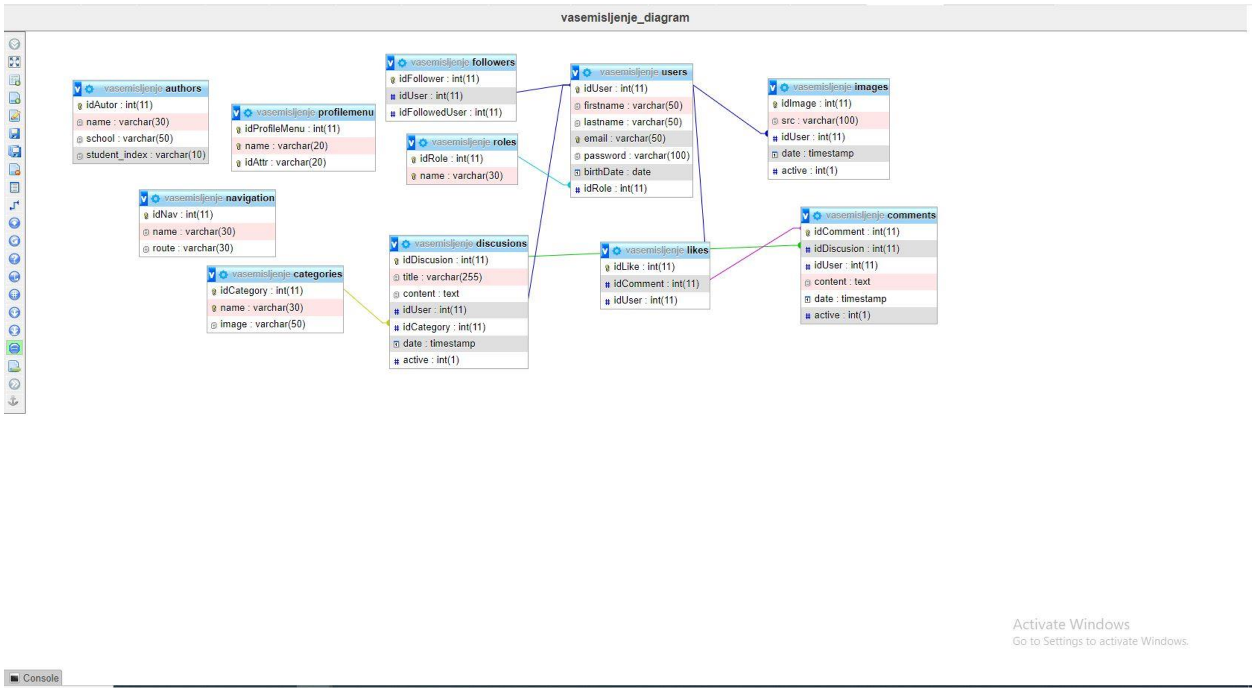
Zašto je Crnjanski najveći srpski pisac (2022-03-11 15:33:18)

Crnjanski je najveći srpski pisac dvadesetog veka, čak i ako stvari sagledamo kumulativno; on je u najvećem broju književnih žanrova dosegao same vrhove: i u romanu, i u poeziji, i u putopisu, i u drami, i u priči, i u memoarima, i u eseju, čak i u novinarstvu. Njemu najbliži srpski pisac, Ivo Andrić, nadmaša ga samo u pripoveci. Nobelova nagrada zavisi i od književne vrednosti i od društvene upotrebljivosti nagrađenog pisca. U području književne vrednosti, nema sumnje u to da postoje dve grupe velikih pisaca: oni koji su dobili Nobelovu nagradu za književnost i oni koji je nisu dobili. Nisu malobrojna velika imena svetske književnosti među njenim laureatima: Tomas Man, Andre Žid, T. S. Eliot, Vilijam Fokner, Alber Kami, Semjuel Beket. Svakako da je Ivo Andrić među njima. Ponekad se učini da se veći pisci nalaze među nenagrađenima. Možda je to zbog rđavog početka, pošto je - punih devet godina - propuštano da se nagradi Lav Tolstoj. No, bilo kako bilo, tu su: Čehov, Ibzen, Strindberg, Prust, Džojs, Paund. Da je srpska književnost - u jednom trenutku - bila u velikom registru, vidimo po tome što imamo svog člana i u ovom nizu. Kao i kod svih velikih a potisnutih, njegovo ime je kao veliko proleće, kao uspomena i sećanje na dane prohujalog leta, kao ironija tuge, sjaj u travi i bleštavost sveta: Miloš Crnjanski.

Komentari (1)

Copyright© 2022. sva prava su zadržana

5.0 Дизајн базе података



6.0 Кодови

6.1 Путање (web.php)

```
<?php

use Illuminate\Support\Facades\Route;


Route::middleware("userAlreadySignedIn")->group(function(){
// HOME
    Route::get("/", [App\Http\Controllers\HomeController::class,
"index"])->name("home");
// AUTH
    Route::get("/auth/register",
[App\Http\Controllers\AuthController::class, "register"])-
>name("register");
    Route::post("/auth/create",
[App\Http\Controllers\AuthController::class, "createAccount"])-
>name("createAccount")->middleware("ajaxDataValid");
    Route::post("/auth/login",
[App\Http\Controllers\AuthController::class, "login"])-
>name("login")->middleware("ajaxDataValid");
});

Route::middleware([\App\Http\Middleware\EnsureUserSignedIn::clas
s])->group(function(){
// USER
    Route::get("/user/",
[App\Http\Controllers\UserController::class, "index"])-
>name("profile");
    Route::get("/user/show/{id}",
[App\Http\Controllers\UserController::class, "show"])-
>name("showOtherProfiles");
    Route::get("/user/editprofile",
[App\Http\Controllers\UserController::class, "editProfile"])-
>name("editprofile");
    Route::put("/user/updateProfilePicture",
[App\Http\Controllers\UserController::class,
"updateProfilePicture"])->name("updateProfilePicture");
    Route::patch("/user/updateprofile",
[App\Http\Controllers\UserController::class, "update"])-
>name("updateProfile");
    Route::get("/user/followers",
[App\Http\Controllers\UserController::class, "followers"])-
>name("followers");
    Route::delete("/user/unfollow/",
[App\Http\Controllers\UserController::class, "unfollow"])-
>name("unfollow");
    Route::post("/user/follow/",
[App\Http\Controllers\UserController::class, "follow"])-
>name("follow");
// CONTACT
    Route::get("/contact",
[App\Http\Controllers\ContactController::class, "index"])-
>name("contact");
    Route::post("/contact/send",
[App\Http\Controllers\ContactController::class, "sendemail"])-
>name("sendEmail");
// AUTHOR
    Route::get("/author",
[App\Http\Controllers\AuthorController::class, "index"])-
>name("author");
    Route::get("/author/get",
[App\Http\Controllers\AuthorController::class,
"getAuthorInfo"]);
// DISCUSIONS
    Route::get("/discusions",
[App\Http\Controllers\DuscusionController::class, "index"])-
>name("discusions");
    Route::get("/discusions/cat/{id}",
[App\Http\Controllers\DuscusionController::class,
"showFromCat"])->name("showFromCat");
    Route::get("/discusions/create",
[App\Http\Controllers\DuscusionController::class, "create"])-
>name("newDiscusion");
    Route::post("/discusions/store",
[App\Http\Controllers\DuscusionController::class, "store"])-
>name("storeDiscusion");
    Route::delete("/discusions/delete",
[App\Http\Controllers\DuscusionController::class,
"deleteDiscusion"])->name("deleteDiscusion");
    Route::get("/discusions/edit/{id}",
[App\Http\Controllers\DuscusionController::class, "edit"])-
>name("editDiscusion");
    Route::put("/discusions/update/",
[App\Http\Controllers\DuscusionController::class, "update"])-
>name("updateDiscsusion");
// COMMENTS
    Route::post("/comments/store",
[App\Http\Controllers\CommentController::class, "store"])-
>name("insertComment");
    Route::delete("/comments/delete/",
[App\Http\Controllers\CommentController::class, "delete"])-
>name("deleteComment");
    Route::get("/comments/edit/{id}",
[App\Http\Controllers\CommentController::class, "edit"])-
>name("editComment");

Route::put("/comments/update",[App\Http\Controllers\CommentContr
oller::class, "updateComment"])->name("updateComment");
//  Likes
    Route::post("/likes/store",
[App\Http\Controllers\LikesController::class, "store"])-
>name("storeLike");
    Route::post("/likes/delete",
[App\Http\Controllers\LikesController::class, "deleteLike"])-
>name("deleteLike");
//   AUTH
    Route::get("/auth/logout",
[App\Http\Controllers\AuthController::class, "logout"])-
>name("logout");
});
```

6.2 Controllers/

6.2.1 authController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\UserModel;
use Illuminate\Support\Facades\Log;
use App\Regex;
class AuthController extends Controller
{
    //
    public function index(){

    }
    public function login(Request  $request){
//        dd($request->all());
        $data = $request->all();
        $errorNum = 0;
        try{
            if(!preg_match(Regex::email(), $data["email"])){
                $errorNum++;
            }
            if(!preg_match(Regex::password(),
$data["password"])){
                $errorNum++;
            }
            // pass data to model
            if($errorNum > 0){
                return redirect()->route("home");
            }
            $model = new UserModel();
            $user = $model->authentification($data["email"],
$data["password"]);
            if($user){
                // begin with session
                session()->put("user", $user);
                return response()->json(["success"=>true]);
            }else {
                return response()->json(["error"=>"Pogrešno
korisnicko ime ili lozinka."]);
```

```
            }
        }catch(Exception $sql){
            Log::error($sql->getMessage());
            return response()->status(500);
        }
    }
    public function register(){
        return view("pages/registration");
    }
    public function createAccount(Request $request){
        $data = $request->all();
        $errorNum = 0;
        if(!preg_match(Regex::firstLastName(),
$data["firstname"])){
            $errorNum++;
        }
        if(!preg_match(Regex::firstLastName(),
$data["lastname"])){
            $errorNum++;
        }
        if(!preg_match(Regex::email(), $data["email"])){
            $errorNum++;
        }

        if(!preg_match(Regex::password(), $data["password"])){
            $errorNum++;
        }
        if(!preg_match(Regex::password(),
$data["passwordAgain"])){
            $errorNum++;
        }
        if($data["password"] != $data["passwordAgain"]){
            $errorNum++;
        }
        if($errorNum == 0) {
            try{
                $model = new UserModel();
                $user  = $model-
>authentification($data["email"], $data["password"]);
                if($user == null){
                    if($model->store(
                        $data["firstname"],
                        $data["lastname"],
                        $data["email"],
                        $data["date"],
                        $data["password"]
                    )){
                        return response()->json([
                            "success"  => true,
                            "message" => "Uspešno ste naprvili
nalog."
                        ]);
                    }else {
                        return response()->json([
                            "message"=> "Nastao je problem sa
serverom, molimo Vas pokušajte kasnije."
                        ]);
                    }
                }else {
                    return response()->json([
                        "message"=>"Već postoji korisnik sa istim
email-om i lozinkom."
                    ]) ;
                }
            }catch(Exception $ex){
                Log::error($ex->getMessage());
            }
        }else {
            // not sent from the client
            return redirect()->route("home");
        }


    }
    public function logout(){
        session()->forget("user");
        return redirect()->route("home");
    }
}
```

6.2.2 authorController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\AuthorModel;
class AuthorController extends Controller
{
    //
    public function index(){
        return view("pages.author");
    }
    public function getAuthorInfo(){
        $model = new AuthorModel();
        return $model->get();
    }

}
```

6.2.3 commentController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\CommentModel;
use App\Regex;
use Illuminate\Support\Facades\Log;
use App\Models\DiscusionModel;
use App\Models\ImageModel;
class CommentController extends DataManagerController
{
    //
    public function store(Request $request){
        $content = $request->get("content");
        $idDiscusion = $request->get("idDiscusion");
        $idUser = session()->get("user")->idUser;
        if(preg_match(Regex::message(), $content)){
            // insert comment
            try{
                $model = new CommentModel();
                if($model->insertComment(
                    $content,
                    $idDiscusion,
                    $idUser
                )){
                    if($request->has("idCategory")){
                        $model = new DiscusionModel();
                        $id = $request->get("idCategory");
                        $data = $model-
>getDiscusionFromCategory($id);
//                        dd($data);
                        return response()->json([
                            "success"=>true,
                            "discusions" =>$data
                        ]);
                    }else {
                        $id = $request->get("idUser");
                        $data = $this-
>getUserDiscusionsAndComments($id);
                        return response()->json([
                            "success"=>true,
                            "discusions" =>$data["discusions"]
                        ]);
                    }
                }else {
                    return response()->json([
                        "error"=>true
                    ]);
                }
            }catch(Exception $ex){
                Log::error($ex->getMessage());
                return response()->json([
                    "error"=>true
                ]);
            }
        }else {
            return redirect()->route("home");
        }
//          return $request->all();
    }
    public function delete(Request $request){
        try{
            $id = $request->get("idComment") ;
            $model = new CommentModel();
            if($model->deleteComment($id)){
                if($request->has("idCategory")){
                    $model = new DiscusionModel();
                    $id = $request->get("idCategory");
                    $data = $model-
>getDiscusionFromCategory($id);
                    return response()->json([
                        "success"=>true,
                        "discusions" =>$data
                    ]);
                }else {
                    $idUser = $request->get("idUser");
                    $data = $this-
>getUserDiscusionsAndComments($idUser);
```

```
                   return response()->json([
                       "success"=>true,
                       "userImage" =>$data["image"],
                       "discusions"=>$data["discusions"]
                   ]);
               }
           }else {
               return response()->json([
                   "error"=>true
               ]);
           }
        }
        catch(Exception $ex){
            Log::error($ex->getMessage());
            return response()->json([
                "error"=>true
            ]);

        }
    }
    public function edit($id){
        $model = new CommentModel();
        $comment = $model->find($id) ;
        return view("pages.editcomment", ["comment"=>$comment]);
    }
    public function updateComment(Request  $request){
        $content = $request->get("content");
        if(preg_match(Regex::message(), $content) && $request-
>get("id")){
            try{
                $model  = new CommentModel();
                if($model->updateComment([
                    "content"=>$content,
                    "idComment"=>$request->get("id")
                ])){
                    return response()->json([
                        "success"=>true
                    ]);
                }else {
                    return response()->json([
                        "nothingUpdated"=>true
                    ]);
                }
            }catch(Exception $ex){
                Log::error($ex->getMessage());
                return response()->json([
                    "error"=>true
                ]);
            }
        }else {
            return redirect()->route("home");
        }
    }

}
```

6.2.4 ContactController.php

```
<?php

namespace App\Http\Controllers;

use App\Regex;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;
use App\Mail\ContactAdmin;
class ContactController extends Controller
{
    //
    public function index(){
        return view("pages.contact") ;
    }
    public function sendEmail(Request $request){
        $email = $request->get('email');
        $subject = $request->get("subject");
        $message = $request->get("message");
        $errorNum = 0;
        if(!preg_match(Regex::email(), $email)){
            $errorNum ++;
        }
        if(!preg_match(Regex::subject(), $subject)){
            $errorNum++;
        }
        if(!preg_match(Regex::message(), $message)){
            $errorNum++;
        }
        try{
            if($errorNum == 0){
                //send email
                $contact = new \stdClass();
                $contact->email = $email;
                $contact->subject =$subject;
                $contact->message = $message;

Mail::to("aleksandar.milijanovic.91.19@ict.edu.rs")->send(new
ContactAdmin($contact));
                return response()->json([
                    "success"=>true
                ]);
            }else {
                return redirect()->route("home");
            }
        }catch(Exception $ex){
            Log::error($ex->getMessage());
            return response()->json([
                "error"=>true
            ]);
        }
    }
}
```

6.2.5 Controller.php

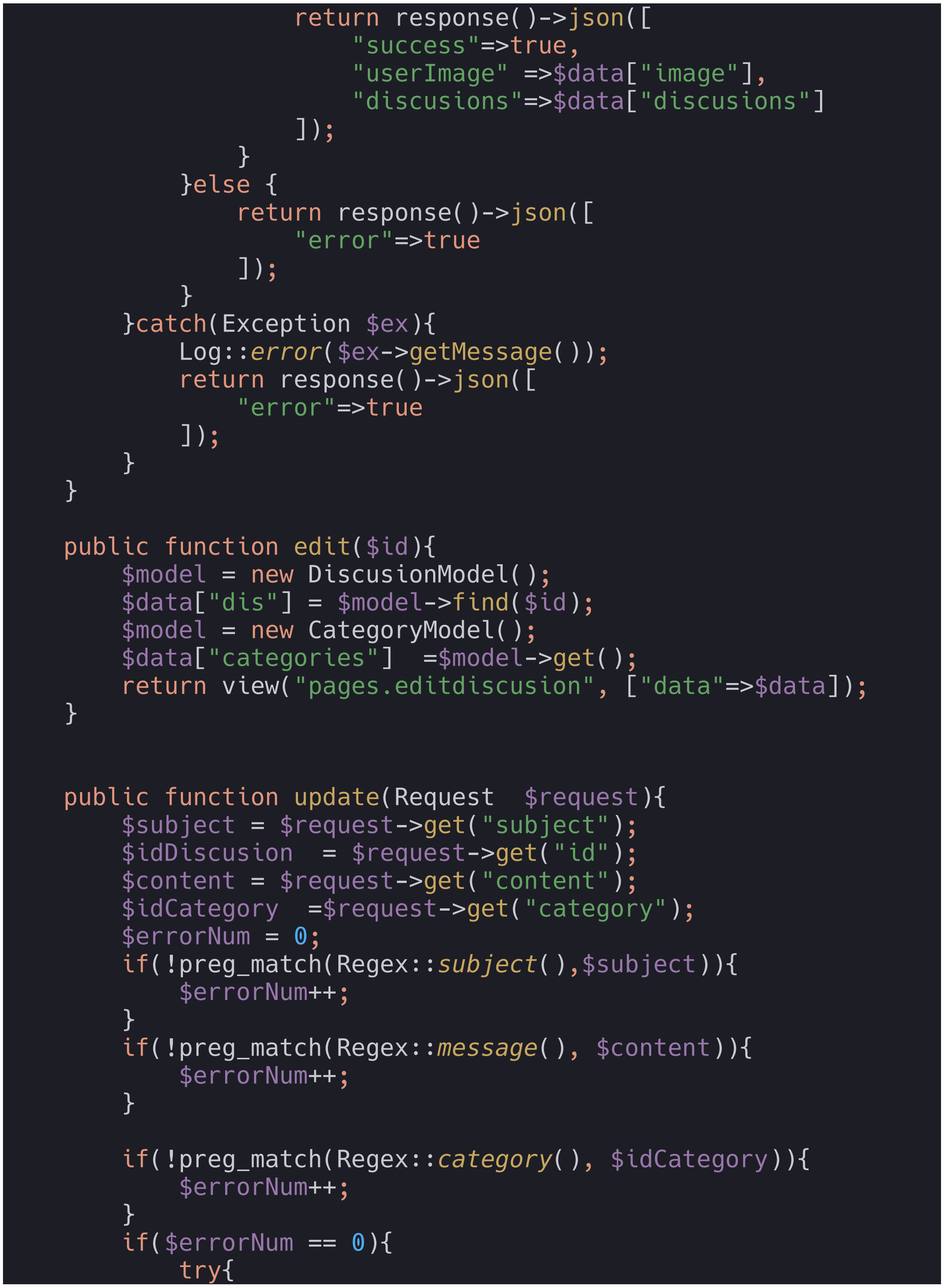
```
<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;
use App\Models\NavigationModel;
abstract class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
    protected $initialData = [];
}
```

6.2.6 DataManagerController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\DiscusionModel;
use App\Models\ImageModel;
class DataManagerController extends Controller
{
    //

    public function getUserDiscusionsAndComments($id = null){
       try{
           if($id == null){
               $id = session()->get("user")->idUser;
           }
           $model = new DiscusionModel();
           $imageModel = new ImageModel();
           $userImage =$imageModel->getForUser($id)->src;
           $discusions = $model->getUsersDiscusions($id);
           return [
               "image"=>$userImage,
               "discusions"=>$discusions
           ];
       } catch(Exception $ex){
           Log::error($ex->getMessage());
       }
    }
}
```

6.2.7 DuscusionController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\CategoryModel;
use App\Models\DiscusionModel;
use Illuminate\Http\Request;
use App\Regex;
class DuscusionController extends DataManagerController
{
    public function index(){
        $model = new CategoryModel();
        $data["categories"] = $model->getWithDebateCount();
        return view("pages.discusions", ["data" => $data]);
    }
    public function create(){
        $model = new CategoryModel() ;
        return view("pages.newdiscusion", ["categories"=>$model-
>get()]);
    }

    public function showFromCat($id){
        $model =  new DiscusionModel();
        $data["discusions"] = $model-
>getDiscusionFromCategory($id) ;
        return view("pages.discusionsfromcategory",
["data"=>$data]);
    }
    public function store(Request $request){
       $subject = $request->get("subject") ;
       $message = $request->get("message");
       $categoryId = $request->get("categoryId");
       $errorNum = 0;
       if(!preg_match(Regex::subject(), $subject )){
            $errorNum++;
       }
       if(!preg_match(Regex::message(), $message)){
           $errorNum++;
       }
       if(!preg_match(Regex::category(), $categoryId)){
           $errorNum++;
       }
       if($errorNum == 0){
            // insert post into table
           $model = new DiscusionModel();
           try{
               if($model->insertDiscusion($subject, $message,
$categoryId)){
                   return response()->json([
                       "success"=>true
                   ]);
               }else {
                return response()->json([
                    "error"=>true
                ]);
               }
           }catch(Exception $ex){
                Log::error($ex->getMessage());
                return response()->json([
                    "errorServer"=>true
                ]);
           }
       }else {
            return redirect()->route('home') ;
       }

//        return $request->all();
    }

    public function deleteDiscusion(Request $request){
//        return $request->all();
        $idDiscusion = $request->get("idDiscusion");
        try{
            $model = new DiscusionModel();
            if($model->deleteDiscusion($idDiscusion)){
                if($request->has("idCategory")){
                    $model = new DiscusionModel();
                    $idCategory = $request->get("idCategory");
                    $data = $model-
>getDiscusionFromCategory($idCategory);
                    return response()->json([
                        "success"=>true,
                        "discusions" =>$data
                    ]);
                }else {
                    $idUser = $request->get("idUser");
                    $data = $this-
>getUserDiscusionsAndComments($idUser);
```



```
                return response()->json([
                    "success"=>true,
                    "userImage" =>$data["image"],
                    "discusions"=>$data["discusions"]
                ]);
            }
        }else {
            return response()->json([
                "error"=>true
            ]);
        }
    }catch(Exception $ex){
        Log::error($ex->getMessage());
        return response()->json([
            "error"=>true
        ]);
    }
}

public function edit($id){
    $model = new DiscusionModel();
    $data["dis"] = $model->find($id);
    $model = new CategoryModel();
    $data["categories"]  =$model->get();
    return view("pages.editdiscusion", ["data"=>$data]);
}


public function update(Request  $request){
    $subject = $request->get("subject");
    $idDiscusion  = $request->get("id");
    $content = $request->get("content");
    $idCategory  =$request->get("category");
    $errorNum = 0;
    if(!preg_match(Regex::subject(),$subject)){
        $errorNum++;
    }
    if(!preg_match(Regex::message(), $content)){
        $errorNum++;
    }

    if(!preg_match(Regex::category(), $idCategory)){
        $errorNum++;
    }
    if($errorNum == 0){
        try{
                $model = new DiscusionModel();
                if($model->updateDiscusion([
                    "content" =>$content,
                    "title"=>$subject,
                    "idCategory"=>$idCategory,
                    "idDiscusion"=>$idDiscusion
                ])){
                    return response()->json([
                        "success"=>true
                    ]);
                }else {
                    return response()->json([
                        "nothingUpdated"=>true
                    ]);
                }
            }catch(Exception $ex){
                Log::error($ex->getMessage());
                return response()->json([
                    "error"=>true
                ]);
            }
        }else {
            return redirect()->route("home");
        }



    }

}
```

6.2.8 HomeController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return view("pages/index");
    }
}
```

6.2.9 LikesController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\LikeModel;
use Illuminate\Support\Facades\Log;
class LikesController extends Controller
{
    public function store(Request $request){
        $id = $request->get("id");
        $idUser =$request->session()->get("user")->idUser;
        try{
            $model = new LikeModel();
            if($model->insertLike($id, $idUser)){
                return response()->json([
                    "success" =>true,
                    "idComment"=>$id,
                    "numberOfLikes" =>$model-
>getNumberOfLikes($id)
                ]);
            }else {
                return response()->json([
                   "error" =>true
                ]);
            }
        }catch(Exception $ex){
            Log::error($ex->getMessage());
            return response()->json([
                "error" =>true
            ]);
        }
    }
    public function deleteLike(Request $request){
        $id = $request->get("id");
        $idUser =$request->session()->get("user")->idUser;
        try{
            $model = new LikeModel();
            if($model->deleteLike($id, $idUser)){
                return response()->json([
                    "success" =>true,
                    "idComment"=>$id,
                    "numberOfLikes" =>$model-
>getNumberOfLikes($id)
                ]);
            }else {
                return response()->json([
                    "error" =>true
                ]);
            }
        }catch(Exception $ex){
            Log::error($ex->getMessage());
            return response()->json([
                "error" =>true
            ]);
        }
    }
}
```

6.2.10 UserController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\ProfileMenuModel;
use Illuminate\Http\Request;
use App\Models\UserModel;
use App\Models\DiscusionModel;
use App\Models\ImageModel;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Log;
use App\Regex;
class UserController extends DataManagerController
{
    //
    public function index(){
        $id = session()->get("user")->idUser;
        return view("pages.profile", ["data"=>$this-
>fetchUserData($id)]);
    }
    public function show($id)
    {
        if($id == session()->get("user")->idUser){
            return redirect()->route("profile");
        }
        $model = new UserModel();
        $isAFollower = $model->isFollowed($id) != null ;
        return view("pages.profile", ["data"=>$this-
>fetchUserData($id), "isAFollower"=>$isAFollower]);
    }
    public function create(){
        return view("pages.newdiscusion");
    }
    public function editProfile(){
        return view("pages.editprofile");
    }
    public function follow(Request $request){
        $id = $request->get("id");
        try{
            $model = new UserModel();
            if($model->follow($id)){
                return response()->json([
                    "success"=>true,
                    "id"=>$id
                ]);
            }else {
                return response()->json([
                    "error"=>true
                ]);
            }
        }catch(Exception $ex){
           Log::error($ex->getMessage()) ;
           return response()->json([
               "error"=>true
           ]);
        }

    }
    public function unfollow(Request  $request){
        $id = $request->get("id");
        try{
            $model = new UserModel();
            $isFollowing = $model->isFollowed($id) != null;
            if($isFollowing){
                if($model->unfollow($id)){
                    if($request->has("returnFollowers")){
                        $search = $request->get("search");
                        $idAuth = session()->get("user")-
>idUser;
                        if(preg_match(Regex::subject(),
$search)){
                            $followers = $model-
>getUserFollowers($idAuth, $search);
                        }else {
                            $followers = $model-
>getUserFollowers($idAuth);
                        }

                        return response()->json([
                            "success"=>true,
                            "followers" =>$followers
                        ]);
                    }
                    return response()->json([
                        "success"=>true,
                        "id"=>$id
                    ]);
                }else {
                    return response()->json([
                        "error"=>true
                    ]);
                }
            }
        }catch(Exception $ex){
           Log::error($ex->getMessage()) ;
            return response()->json([
                "error"=>true
            ]);
        }

    }
    // it's called whenever a follower search is commited
    public function followers(Request $request){
        $id = session()->get("user")->idUser;
        $search = $request->query("search");
        if(preg_match(Regex::search(), $search)){
            try{
                $model = new UserModel();
                return $model->getUserFollowers($id, $search);
            }catch(Exception $ex){
                Log::error($ex->getMessage()) ;
                return null;
            }
        }else {
            return redirect()->route("home") ;
        }

    }
    private function fetchUserData($id){
        $initialData = [];
        $userModel = new UserModel();
        $data = $this->getUserDiscusionsAndComments($id);
        $initialData["userImage"] = $data["image"];
        $initialData["userData"] = $userModel->getUser($id);
        $initialData["discusions"] = $data["discusions"];
        $initialData["followers"] = $userModel-
>getUserFollowers($id);
        $profileMenuModel  = new ProfileMenuModel();
        $initialData["menu"] = $profileMenuModel->get();
        return $initialData;
    }
    public function update(Request  $request){
        $data = $request->all();

        $errorNum = 0;
        if(isset($data["firstname"]) &&
!preg_match(Regex::firstLastName(), $data["firstname"])){
            $errorNum++;
        }
        if(isset($data["lastname"]) &&
!preg_match(Regex::firstLastName(), $data["lastname"])){
            $errorNum++;
        }


        if(isset($data["email"]) && !preg_match(Regex::email(),
$data["email"])){
            $errorNum++;
        }
        if(
            (isset($data["passwordOld"]) &&
isset($data["password"])) && (!preg_match(Regex::password(),
$data["passwordOld"]) || !preg_match(Regex::password(),
$data["password"]) || $data["passwordOld"] == $data["password"]
|| md5($data["passwordOld"])  != session()->get("user")-
>password)
        ){
            $errorNum++;
        }
        if(isset($data["birthDate"]) &&
!preg_match(Regex::date(), $data["birthDate"])){
            $errorNum++;
        }
        $model = new UserModel();
        if($errorNum == 0){
            try{
                if(isset($data["passwordOld"])){
                    unset($data["passwordOld"]);
                    $data["password"] = md5($data["password"]);
                }
                if($model->updateUserData($data)){
                    session()->put("user", $model-
>getUser(session()->get("user")->idUser));
                    return response()->json([
                        "success"=>true
                    ]);
                }
                else{
                    return response()->json([
                        "success"=>false
                    ]);
                }
            }catch(Exception $ex){
               Log::error($ex->getMessage()) ;
               return response()->status(500);
            }
        }else {
           return response()->json([
               "error"=>true
           ]);
        }
    }

    public function updateProfilePicture(Request $request)
    {
        if(!$request->has("pictureEditProfile")){
            return redirect()->back()->with("error", "Niste
odabrali fotografiju.");
        }
//          $image = $_FILES["pictureEditProfile"];
        $image = $request->file("pictureEditProfile");
        // validation
        if (!in_array($image->getMimeType(), ["image/png",
"image/jpg", "image/jpeg"])) {
```

```
            return redirect()->back()->with("message", "Tip
fotografije nije dobar.");
        }
        try{
            $destination = "public/images";
            $filename = time() . $image-
>getClientOriginalName();
            $model = new ImageModel();
            $id = session()->get("user")->idUser;
            if($model->softDeleteImage($id)){
                $model->insertImage($filename, $id) ;
                $path = $image->storeAs($destination,
$filename);
                return redirect()->back()->with("success",
"Uspešno ste promenili profilnu fotografiju.");
            }else {
                return redirect()->back()->with("error",
"Promena profilne fotogragije nije moguća.");
            }
//            echo($filename);
        }catch(Exception $ex){
            Log::error($ex->getMessage());
            return redirect()->back()->with("error", "Došlo je
do greške na serveru, molimo Vas pokušajte kasnije.");
        }

    }
}
```

6.3 Middleware/

6.3.1 EnsureAjaxDataValid.php

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class EnsureAjaxDataValid
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
(\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)
$next
     * @return
\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(!$request->ajax()){
            return response()->redirectToRoute("home");
        }
        $submit = $request->all()["submit"];
        if(!isset($submit) || $submit != true){
            return response()->redirectToRoute("home");
        }
        return $next($request);
    }
}
```

6.3.2 EnsureUserAlreadySignedIn.php

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class EnsureUserAlreadySignedIn
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
(\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)
$next
     * @return
\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(session()->has("user")){
            return redirect()->route("profile");
        }
        return $next($request);
    }
}
```

6.3.3 EnsureUserSignedIn.php

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class EnsureUserSignedIn
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request):
(\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)
$next
     * @return
\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(!session()->has("user")){
            return redirect()->route("home");
        }
        return $next($request);
    }
}
```

6.3.4 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<int, class-string|string>
     */
    protected $middleware = [
        // \App\Http\Middleware\TrustHosts::class,
        \App\Http\Middleware\TrustProxies::class,
        \Fruitcake\Cors\HandleCors::class,

\App\Http\Middleware\PreventRequestsDuringMaintenance::class,

\Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,

\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull
```

```
::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array<string, array<int, class-string|string>>
     */
    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' => [
            //
\Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStatef
ul::class,
            'throttle:api',

\Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];

    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used
individually.
     *
     * @var array<string, class-string|string>
     */
    protected $routeMiddleware = [
        'auth' => \App\Http\Middleware\Authenticate::class,
        'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
        'cache.headers' =>
```

```
\Illuminate\Http\Middleware\SetCacheHeaders::class,
        'can' => \Illuminate\Auth\Middleware\Authorize::class,
        'guest' =>
\App\Http\Middleware\RedirectIfAuthenticated::class,
        'password.confirm' =>
\Illuminate\Auth\Middleware\RequirePassword::class,
        'signed' =>
\Illuminate\Routing\Middleware\ValidateSignature::class,
        'throttle' =>
\Illuminate\Routing\Middleware\ThrottleRequests::class,
        'verified' =>
\Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
        "ajaxDataValid"=>
\App\Http\Middleware\EnsureAjaxDataValid::class,
        "userSignedIn"=>
\App\Http\Middleware\EnsureUserSignedIn::class,

"userAlreadySignedIn"=>\App\Http\Middleware\EnsureUserAlreadySig
nedIn::class
    ];
}
```

6.4 Providers /

6.4.1 AppServiceProvider.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<int, class-string|string>
     */
    protected $middleware = [
        // \App\Http\Middleware\TrustHosts::class,
        \App\Http\Middleware\TrustProxies::class,
        \Fruitcake\Cors\HandleCors::class,

\App\Http\Middleware\PreventRequestsDuringMaintenance::class,

\Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,

\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull
::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array<string, array<int, class-string|string>>
     */
    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' => [
            //
\Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStatef
ul::class,
            'throttle:api',

\Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];

    /**
     * The application's route middleware.
     *
     * These middleware may be assigned to groups or used
individually.
     *
     * @var array<string, class-string|string>
     */
    protected $routeMiddleware = [
        'auth' => \App\Http\Middleware\Authenticate::class,
        'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
        'cache.headers' =>
\Illuminate\Http\Middleware\SetCacheHeaders::class,
        'can' => \Illuminate\Auth\Middleware\Authorize::class,
        'guest' =>
\App\Http\Middleware\RedirectIfAuthenticated::class,
        'password.confirm' =>
\Illuminate\Auth\Middleware\RequirePassword::class,
        'signed' =>
\Illuminate\Routing\Middleware\ValidateSignature::class,
        'throttle' =>
\Illuminate\Routing\Middleware\ThrottleRequests::class,
        'verified' =>
\Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
        "ajaxDataValid"=>
\App\Http\Middleware\EnsureAjaxDataValid::class,
        "userSignedIn"=>
```

```
\App\Http\Middleware\EnsureUserSignedIn::class,

"userAlreadySignedIn"=>\App\Http\Middleware\EnsureUserAlreadySig
nedIn::class
    ];
}
```

6.5 Mail

6.5.1 ContactAdmin.php

```php
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class ContactAdmin extends Mailable
{
    use Queueable, SerializesModels;

    /**
     * Create a new message instance.
     *
     * @return void
     */
    public $contact;
    public function __construct(\stdClass $contact)
    {
        //
        $this->contact = $contact;
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->from($this->contact->email)-
>subject($this->contact->subject)->view('emails.contactadmin');
    }
}
```

6.6 Models/

6.6.1 AuthorModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class AuthorModel extends Model
{
    use HasFactory;
    public function getAll(){

    }
    public function get(){
        return DB::table("authors")->select("name", "school",
"student_index")->first();
    }
}
```

6.6.2 CategoryModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class CategoryModel extends Model
{
    use HasFactory;
    public function get(){
       return DB::table("categories")->get();
    }
    public function getWithDebateCount(){
        // moguca resenja:
        // 1) da prvo dohvatim sve kategorije pa da svakoj
ponaosob dodajem broj rasprava
        // 2) da osmislim kako preko jednog upita da dohvatim
```

```
sve sto mi treba
        // 3) da primenjujem nepovratno brisanje sto ne bih
zeleo uraditi
        $categories =  DB::table("categories")
            ->get();
        foreach($categories as $c){
            $c->discusionsNum = DB::table("discusions")
                                ->where([
                                    ["active", 1],
                                    ["idCategory", $c-
>idCategory]
                                ])->count("idDiscusion");
        }
       return $categories;
    }
}
```

6.6.3 CommentModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class CommentModel extends Model
{
    use HasFactory;
    public function getCommentsForDiscusion($id){
        $comments =  DB::table("comments AS com")->
        select(
            "com.idComment",
            "com.content",
            "com.date",
            "images.src as profilePic",
            "users.firstname",
            "users.lastname",
            "users.idUser"
        )
            ->join("users", "users.idUser" , "=", "com.idUser")
            ->join("images", "images.idUser", "=", "com.idUser")
            ->where([
                ["com.idDiscusion", "=",  $id],
                ["com.active", "=", 1],
                ["images.active", "=", 1]
            ])
            ->get();
        $idAuth = session()->get("user")->idUser;
        foreach($comments as $c){
            $c->likes = DB::table("likes")
                ->where("idComment", $c->idComment)
                ->count("idLike");
            $c->AuthUserLiked = DB::table("likes")
                ->where("idComment", $c->idComment)
                ->where("idUser", $idAuth)
                ->first() != null;
        }
        return $comments;
    }
    public function find($id){
        return DB::table("comments")
            ->where("idComment", "=", $id)
            ->first();
    }
    public function insertComment(
        $content,
        $idDiscusion,
        $idUser
    ){
        return DB::table("comments")
            ->insert([
                "idDiscusion"=>$idDiscusion,
                "idUser"=>$idUser,
                "content"=>$content
            ]);
    }

    public function deleteComment($id){
        return DB::table("comments")
            ->where("idComment" ,"=" ,$id)
            ->update([
                "active"=> 0
            ]);
    }

    public function updateComment($data){
        return DB::table("comments")
            ->where("idComment", $data["idComment"])
            ->update([
                "content"=>$data["content"]
            ]);
    }
}
```

6.6.4 DiscusionModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
use App\Models\CommentModel;
class DiscusionModel extends Model
{
    use HasFactory;
    public function getUsersDiscusions($id){
       $discusions = DB::table("discusions")
           ->select(
               "discusions.*",
               "users.firstname",
               "users.lastname",
               "users.idUser",
               "categories.name as category",
               "images.src"
           )
           ->join("users", "users.idUser",
"=","discusions.idUser")
           ->join("categories", "categories.idCategory", "=",
"discusions.idCategory")
           ->join("images", "images.idUser", "=",
"users.idUser")
           ->where([
               ["discusions.active", 1],
               ["users.idUser", "=", $id],
               ["images.active", "=", 1]
           ])
           ->get();
        // after fetching all discusions, fetch all comemnts for
fetched posts
        $commentModel = new CommentModel();
        foreach($discusions as $dis){
            $dis->comments = $commentModel-
>getCommentsForDiscusion($dis->idDiscusion);
```

```
        }
        return $discusions;
    }
    public function get(){
        $discusions = DB::table("discusions")
            ->select(
                "discusions.*",
                "users.firstname",
                "users.lastname",
                "users.idUser",
                "categories.name as category",
                "images.src"
            )
            ->join("users", "users.idUser",
"=","discusions.idUser")
            ->join("categories", "categories.idCategory", "=",
"discusions.idDiscusion")
            ->join("images", "images.idUser", "=",
"users.idUser")
            ->where([
                ["discusions.active", 1]
            ])
            ->get();
        // after fetching all discusions, fetch all comemnts for
fetched posts
        $commentModel = new CommentModel();
        foreach($discusions as $dis){
            $dis->comments = $commentModel-
>getCommentsForDiscusion($dis->idDiscusion);
        }
        return $discusions;
    }

    public function find($id){
        return DB::table("discusions")
            ->where("idDiscusion", $id)
            ->first();
    }


    public function getDiscusionFromCategory($idCategory)
    {
        $discusions = DB::table("discusions")
            ->select(
                "discusions.*",
                "users.firstname",
                "users.lastname",
                "users.idUser",
                "categories.name as category",
                "categories.idCategory",
                "images.src"
            )
            ->join("users", "users.idUser",
"=","discusions.idUser")
            ->join("categories", "categories.idCategory", "=",
"discusions.idCategory")
            ->join("images", "images.idUser", "=",
"users.idUser")
            ->where([
                ["discusions.active", "=", 1],
                ["discusions.idCategory", "=",$idCategory],
                ["images.active", "=", 1]
            ])
            ->get();
        // after fetching all discusions, fetch all comemnts for
fetched posts
        $commentModel = new CommentModel();
        foreach($discusions as $dis){
            $dis->comments = $commentModel-
>getCommentsForDiscusion($dis->idDiscusion);
        }
        return $discusions;
    }

    public function insertDiscusion(
        $subject,
        $message,
        $categoryId
    ){
        return DB::table("discusions")
            ->insert([
                "title" => $subject,
                "content"=>$message,
                "idCategory"=>$categoryId,
                "idUser"=>session()->get("user")->idUser
            ]);
    }

    public function deleteDiscusion($id){

        if( DB::table("discusions")
            ->where("idDiscusion", $id)
            ->update([
                "active"=>0
            ])){
             DB::table("comments")
                ->where("idDiscusion", $id)
                ->update([
                    "active"=>0
                ]);
             return true;
        }
        return false;
    }

    public function updateDiscusion($data){
        return DB::table("discusions")
            ->where("idDiscusion", $data["idDiscusion"])
            ->update([
                "title"=>$data["title"],
                "content"=>$data["content"],
                "idCategory"=>$data["idCategory"]
            ]);
    }
}
```

6.6.5 ImageModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;
class ImageModel extends Model
{
    use HasFactory;

    public function insertImage($filename, $idUser){
        return DB::table("images")
            ->insert([
                "src" =>$filename,
                "idUser"=>$idUser
            ]);
    }
    public function getForUser($id){
        return DB::table("images")
            ->select("src")
            ->where([
                ["idUser" , "=" , $id],
                ["active", "=", 1]
            ])->first();
    }
    public function softDeleteImage($id){
        return DB::table("images")
            ->where([
                ["idUser", $id],
                ["active" , 1]
                ])
            ->update([
                    "active"=>0
                ]);
    }
    public function hardDeleteImage($id){
        $destination = "public/images/";
        $filename = DB::table("images")
            ->select("src")
            ->where([
                ["idUser", $id],
                ["active" , 1]
            ])->first();
        Storage::delete($destination . $filename);
        return DB::table("images")
            ->where([
                ["idUser", $id],
                ["active" , 1]
            ])
            ->delete();
    }
}
```

6.6.6 LikeModel.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class LikeModel extends Model
{
    use HasFactory;

    public function insertLike($idComment, $idUser){
        return DB::table("likes")
            ->insert([
                "idUser"=>$idUser,
                "idComment"=>$idComment
            ]);
    }
    public function deleteLike($idComment, $idUser){
        return DB::table("likes")
            ->where([
                ["idComment", "=", $idComment],
                ["idUser", "=", $idUser]
            ])
            ->delete();
    }
    public function getNumberOfLikes($idComment){
        return DB::table("likes")
            ->where("idComment", $idComment)
            ->count("idLike");

    }
}
```

6.6.7 NavigationModel.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;

class NavigationModel extends Model
{
    use HasFactory;
    public function get(){
        $links = DB::table("navigation")->get();
        $i = 0;
        $explodedURLNames = [
            "user",
            "contact",
            "author",
            "discusions",
            "auth"
        ];
        foreach($links as $link)     {
            $link->explodedURLName = $explodedURLNames[$i++];
        }
        return $links;
    }
}
```

6.6.8 ProfileMenuModel.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class ProfileMenuModel extends Model
{
    use HasFactory;

    public function get(){
        return DB::table("profileMenu")    ->get();
    }
}
```

6.6.9 UserModel.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;
class UserModel extends Model
{
    use HasFactory;

    public function authentification($email, $password){
        $password = md5($password);
        return DB::table("users")->where([
            ["email" , "=",  $email],
            ["password", "=", $password]
        ])->first();
    }

    public function store(
        $firstname,
        $lastname,
        $email,
        $date,
        $password
    ){
       $id =  DB::table("users")->insertGetId([
            "firstname" => $firstname,
            "lastname" => $lastname,
            "email" => $email,
            "password"=>md5($password),
            "birthDate" => $date
        ]);

        DB::table("images")->insert([
            "src"=>"profilePicture1.jpg",
            "idUser"=>$id
        ]);
        return true;
    }
    public function getUser($id){
        $active = 1;
        return DB::table("users")
            ->select("users.idUser", "users.firstname",
"users.lastname", "images.src", "users.email",
"users.birthDate", "users.password")
            ->join("images", "images.idUser", "=",
"users.idUser")
            ->where([
                ["users.idUser" , $id],
                ["images.active", $active]
            ])->first();
    }
    public function isFollowed($id){
        return DB::table("followers")
            ->where("idFollowedUser", "=", $id)
            ->first();
    }
    public function unfollow($id){
        $idAuth = session()->get("user") ->idUser;
       return DB::table("followers")
            ->where([
                ["idFollowedUser", "=", $id],
                ["idUser" , "=", $idAuth]
        ])
            ->delete();
    }
    public function follow($id){
        $idAuth = session()->get("user")->idUser;
        return DB::table("followers")
            ->insert([
                "idUser"=>$idAuth,
                "idFollowedUser"=>$id
            ]);
    }
    public function getUserFollowers($id, $search= null){
        if($search == null){
            return DB::table("users")
                ->select("users.firstname", "users.lastname",
"users.idUser", "images.src")
                ->join("followers", "followers.idFollowedUser",
"=", "users.idUser")
                ->join("images", "images.idUser" , "="
,"users.idUser")
                ->where([
                    ["followers.idUser", $id],
                    ["images.active", 1]
                ])->get();
        }
        return DB::table("users AS u1")
            ->select("u2.firstname", "u2.lastname", "u2.idUser",
"images.src")
            ->join("followers", "followers.idUser", "=",
"u1.idUser")
            ->join("users AS u2", "followers.idFollowedUser",
"=", "u2.idUser")
            ->join("images", "images.idUser", "=", "u2.idUser")
            ->where([
                ["images.active", "=",1],
                ["u1.idUser", "=", $id],
            ])
            ->where(function($query) use($search){
                $query->where("u2.firstname", "like",
"%$search%")
                        ->orWhere("u2.lastname", "like",
"%$search%")
                        ->orWhere("u2.email", "like", "%$search%");
            })
            ->get();
    }
    public function updateUserData($data){
       return  DB::table("users")->where("users.idUser",
session()->get("user")->idUser)
            ->update($data);

    }
}
```

6.7 Regex.php

```
<?php

namespace App;
abstract class Regex
{

    public static function firstLastName()
    {
        return "/^[A-ZŽĐŠĆČ][a-zžđšćč]{2,14}(\s[A-ZŽĐŠĆČ][a-
zžđšćč]{2,14}){0,1}$/";
    }

    public static function email()
    {
        return "/^[a-z][a-z0-9]{2,14}([\._][a-z0-
9]{1,14}){0,4}\@([a-z]{3,5}\.){1,2}[a-z]{2,3}$/";
    }

    public static function password()
    {
        return "/^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,16}$/";
    }

    public static function subject()
    {
        return "/^[A-ZŽĐŠĆČ][a-zžđšćč]{2,19}(\s[A-ZŽĐŠĆČa-
zžđšćč0-9\@]{1,19}){0,5}$/";
    }

    public static function message()
    {
        return "/^[A-ZŽĐŠĆČ@][A-zŽĐŠĆČžđšćč0-
9\(\)\.,?!:\/\;\s\n\*-_@]{1,}$/";
    }

    public static function category()
    {
        return "/^[0-9]$/";
    }

    public static function date()
    {
        return "/^[0-9]{4}\-[0-9]{2}\-[0-9]{2}$/";
    }

    public static function search(){
        return "/^[A-zŽĐŠĆČžđšćč\s\@\-\_]{0,50}$/";
    }
}
```

6.8 Views/Components/

6.8.1 category.blade.php

```
<section class="col-12 col-md-4   p-3" >
    <section style="background: url('{{asset("assets/img/".
$category->image)}}') rgba(90, 90, 90, .6); background-size:
cover; background-blend-mode: multiply; background-position:
center;" class="d-flex justify-content-center  flex-column
white-forecolor height-400 border rounded ">
        <article class="">
            <h3 class="roboto-slab-bold text-center category-
name">{{$category->name}}</h3>
        </article>
        <article>
            <p class="roboto-slab-bold text-center">Broj
rasprava: {{$category->discusionsNum}}</p>
        </article>
        <article class="text-center">
            <a href="{{route("showFromCat", ["id"=>$category-
>idCategory])}}" class="btn btn-primary">Pogledajte</a>
        </article>
    </section>
</section>
```

6.8.2 comment.blade.php

```
<section class="comment mb-3">
    <section class="row ">
        <figure class="col-12 col-md-1"><img
src="{{asset("storage/images/" . $comment->profilePic)}}"
class="profile-pic-small img-fluid img-thumbnail" alt="profile
picture"/></figure>
        <article class="col-11">
            <p   class="roboto-slab-bold"><a
href="{{route("showOtherProfiles", ["id"=>$comment-
>idUser])}}">{{$comment->firstname}} {{$comment->lastname}}</a>
({{$comment->date}})</p>
        </article>
    </section>
    <article>
        <p class="roboto-slab">{{$comment->content}}</p>
    </article>
    <article class="like-comment-link-wrapper">
        <span class="roboto-slab-bold">Sviđanja <span
class="like-count">{{$comment->likes}}</span></span>
        @if($comment->AuthUserLiked)
            <a href="#" class="btn btn-link dislike-comment"
data-id="{{$comment->idComment}}">Ne sviđa mi se</a>
            @else
        <a href="#" class="btn btn-link like-comment" data-
id="{{$comment->idComment}}">Sviđa mi se</a>
            @endif
    </article>
    @if(\Request::is("user") || session()->get("user")->idUser
== $comment->idUser || $discusionIdUser == session()-
>get("user")->idUser)
        <section class=" row my-3">
            <article class="col-12 col-md-2 mb-3">
                <form class="delete-comment-form" name=""
action="#" data-id="{{$comment->idComment}}" method="POST" >
                    <input type="submit" value="Obriši komentar"
class="btn btn-danger" />
                </form>
            </article>
            <article class="col-12 col-md-3"><a
href="{{route("editComment", ["id"=>$comment->idComment])}}"
class="btn btn-primary">Izmeni komentar</a></article>
        </section>

        @endif
</section>
```

6.8.3 discusion.blade.php

```
<section class="w-100  mb-3 p-2 discusions">
    <!-- prvo ide ime i datum u jednom redu-->
    <section class=" mb-3  mx-0 row justify-content-between">
        <section class="row col-12 col-md-6">
            <figure class="col-12 col-md-2 "><img
src="{{asset("storage/images/" . $userImage)}}"  class="profile-
pic-small img-fluid img-thumbnail" alt="profile
picture"/></figure>
            <article class="col-12 col-md-10"><a
href="{{route("showOtherProfiles", ["id"=>$dis->idUser])}}"
class="roboto-slab-bold ">{{$dis->firstname}} {{$dis-
>lastname}}</a></article>
        </section>
        @if(\Request::is("user") || session()->get("user")-
>idUser == $dis->idUser)
            <section class="row col-12 col-md-6 justify-content-
end">
                <article class="col-12 col-md-2">
                    <a href="{{route("editDiscusion",
["id"=>$dis->idDiscusion])}}" class="btn btn-primary"
>Izmeni</a>
                </article>
                <article class="col-12 col-md-2"><a href="#"
class="btn btn-danger delete-discusion-link" data-id="{{$dis-
>idDiscusion}}">Obriši</a></article>
            </section>
            @endif
    </section>
    <section class="row justify-content-start">
        <article class="   mr-2">
            <p class="roboto-slab-bold mb-0">{{$dis->title}}</p>
        </article>
        <article class=" ">
            <p class="roboto-slab-bold mb-0">({{$dis-
>date}})</p>
        </article>
        <article class="col-12 col-md-2 my-2 my-md-0">
            <span class="roboto-slab-bold category">{{$dis-
>category}}</span>
        </article>
    </section>
        <article class="my-3 ">
            <p class="roboto-slab text-justify">{{$dis-
>content}}</p>
        </article>
   @if($dis->comments )
    <article>
        <p class="roboto-slab-bold title-1">Komentari
({{count($dis->comments)}})<a href="#" class="showhide-comment-
section ml-3 btn btn-primary" >Prikaži</a></p>
        <section class="comment-section">
            <section >
                <form action="{{route("insertComment")}}"data-
id="{{$dis->idDiscusion}}" method="POST"
name="formInsertComment" class="form-insert-comment">
                    <section class="row">
                        <article class="col-12 mb-3">
                            <textarea rows="3" cols="20"
class="comment-content form-control" placeholder="Unesite
komentar"></textarea>

@include("fixed.errorMessages.message")
                        </article>
                        <article class="col-12 mb-3">
                            <input type="submit" value="Potvrdi"
class="btn btn-primary" />
                        </article>
                    </section>
                </form>
            </section>
            @foreach($dis->comments as $comment)
                @component("components.comment",
["comment"=>$comment, "discusionIdUser"=>$dis->idUser])
                @endcomponent
            @endforeach
        </section>
    </article>
    @endif
</section>
```

6.8.4 follower-card.blade.php

```
<section class="col-12 col-md-4 p-3   ">
   <section class="follower-card ">
       <figure class="w-100 d-flex justify-content-center">
           <img src="{{asset("storage/images/". $follower->src)
}}" class="img-fluid rounded" alt="user image"/>
       </figure>
       <article class="px-2">
           <p class="roboto-slab-bold">{{$follower->firstname}}
{{$follower->lastname}}</p>
       </article>
       <article class="p-2">
           <a class="btn btn-primary go-to-profile-link"
href="{{route("showOtherProfiles", ["id"=>$follower->idUser])}}"
>Profil</a>
            @if (session()->get("user")->idUser != $follower-
>idUser)
           <a href="#" class="btn btn-danger unfollow-link"
data-id="{{$follower->idUser}}">Otprati</a>
                @endif
       </article>
   </section>
</section>
```

6.8.5 nav-link.blade.php

```
    @if(!empty($explodedURL[2]) && $explodedURL[2] == $link-
>explodedURLName)
    <li>
        <a href="{{$link->route}}" class="orange-forecolor
roboto-slab-bold">{{$link->name}}</a>
    </li>
@else
    <li>
        <a href="{{$link->route}}" class="roboto-slab-
bold">{{$link->name}}</a>
    </li>
@endif
```

6.9 Views/Email/

6.9.1 contactadmin.php

```
<p >{{$contact->message}}</p>
```

6.10 Views/Fixed/

6.10.1 errorMessages/

6.10.1.1 category.blade.php

```
<span class="form-element-message error-message"
id="category">Izaberite kategoriju</span>
```

6.10.1.2 date.blade.php

```
<span class="form-element-message error-message"
id="date">Izaberite datum rođenja</span>
```

6.10.1.3 email.blade.php

```
<span class="form-element-message error-message "
id="email">Mora početi sa slovom. Potrebno je najmanje tri
znaka, a najviše 16 u jednoj reči. Reči mogu biti odvojene
tačkom ili donjom crticom. U okviru reči se mogu nalaziti
brojevi.</span>
```

6.10.1.4 firstlastName.blade.php

```
<span class="form-element-message error-message"
id="{{$id}}">Mora početi velikim slovom. Najmanje tri, a najviše
31 znak sa sve razmakom. Bez brojeva i specijalnih
znakova.</span>
```

6.10.1.5 message.blade.php

```
<span class="form-element-message error-message"
id="message">Poruka mora početi velikim slovom, najmanje 2
znaka.</span>
```

6.10.1.6 password.blade.php

```
<span class="form-element-message error-message"
id="{{$id}}">Najmanje 8 znakova, najviše 16, potrebno je jedno
veliko slovo, jedno malo slovo i jedan broj. Bez specijalnih
znakova.</span>
```

6.10.1.7 search.blade.php

```
<span class="form-element-message error-message "
id="search">Može sadržati slova i određene specijalne karatkere
(@, - i  _) sa razmacima. Najviše je dozvoljeno 50
znakova.</span>
```

6.10.1.8 subject.blade.php

```
<span class="form-element-message error-message"
id="subject">Mora početi velikim slovom, najmanje 3 a najviše
120 znakova. Bez specijalnih znakova.</span>
```

6.10.2 footer.blade.php

```
<footer class="py-4 container-fluid d-flex align-items-center
justify-content-center">
    <p class="orange-forecolor text-center m-0 roboto-slab-
bold">Copyright&copy; {{ date("Y.") }} sva prava su zadržana</p>
</footer>
```

6.10.3 head.blade.php

```
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial-
scale=1"/>
<title>Vaše mišljenje - @yield("title")</title>
<meta name="csrf-token" content="{{ csrf_token() }}">
<meta  name="keywords"  content="@yield("keywords")"/>
<meta name="description" content="@yield("description")"/>
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.11.2/css/all.css"/
>
<link rel="stylesheet"
href="{{asset("assets/vendor/bootstrap/css/bootstrap.css")}}"/>
<link  rel="stylesheet"
href="{{asset("assets/css/style.css")}}"/>
<link rel="shortcut icon"
href="{{asset("assets/img/favicon.ico")}}" type="image/x-icon"/>
```

6.10.3 logo.blade.php

```
<a href="{{route("home")}}" class="">
    <img src="{{asset("assets/img/logo.png")}}" title="Vaše
mišljenje" class="img-fluid" alt="logo"/>
</a>
```

6.10.4 modal.blade.php

```
<section class="container-fluid my-modal d-none justify-content-
center align-items-center" id="modal">
    <section class="container ">
       <section class="row white-backcolor border rounded p-3">
            <article class="d-flex justify-content-end w-100">
                <a href="#" id="close-modal" class="roboto-slab-
bold ">&times;</a>
            </article>
           <article id="modal-body" class="my-3">

           </article>
           <article id="modal-footer" class="d-flex justify-
content-end w-100">
                <a href="#" id="close-modal-btn" class="btn btn-
primary">Zatvori</a>
           </article>
       </section>
    </section>
</section>
```

6.10.5 navigation.blade.php

```
<?php
    $explodedURL = explode("/",$_SERVER["PHP_SELF"]);
?>
<article id="bars-menu" class=" white-backcolor position-fixed">
    <article class="d-flex justify-content-end px-2">
        <a href="#" id="bars-menu__close" class="">&times;</a>
    </article>
    <article class="d-flex justify-content-center">
        <ul class=" text-center w-100">
            @foreach($menu as $link)
                @component("components.nav-link",
["explodedURL"=>$explodedURL, "link"=>$link])
                @endcomponent
            @endforeach
        </ul>
    </article>
</article>
<header class="container-fluid white-backcolor">
    <section class="container mx-auto py-2">
        <section class="row justify-content-between">
            <article class="col-2 col-md-1">
                @include("fixed.logo")
            </article>
            <article class="col-5 d-none d-md-flex align-items-
center justify-content-center" id="nav_link">
                <ul class="d-flex justify-content-between  w-
100">
                    @foreach($menu as $link)
                        @component("components.nav-link",
["explodedURL"=> $explodedURL, "link"=>$link])
                        @endcomponent
                        @endforeach
                </ul>
            </article>
            <article class="col-1 d-flex  d-md-none align-items-
center">
                <span class=" fas fa-bars " id="bars"></span>
            </article>
        </section>
    </section>
</header>
```

6.10.6 scripts.blade.php

```
<script
src="{{asset("assets/vendor/jquery/jquery.js")}}"></script>
<script src="{{asset("assets/js/main.js")}}"></script>
```

6.11 Views/Layouts

6.11.1 master.blade.php

```
<?php session_start();?>
<!DOCTYPE html>
<html lang="en">
<head>
@include("fixed.head")
</head>
<body >
@include("fixed.modal")
@include("fixed.navigation")
@yield("content")
@include("fixed.footer")
@include("fixed.scripts")
</body>
</html>
```

6.11.2 no-nav-layout.blade.php

```
<?php session_start();?>
    <!DOCTYPE html>
<html lang="en">
<head>
    @include("fixed.head")
</head>
<body >
@include("fixed.modal")
    @yield("content")
@include("fixed.scripts")
</body>
</html>
```

6.12 Views/Pages/

6.12.1 author.blade.php

```
@extends("layouts.master")
@section("title", "Autor sajta")
@section("keywords", "autor, podaci, dokumentacija")
@section("description", "Preko ove stranice se upoznajete sa
autorom sajta, imate uvid u njegove osnovne podatke i
dokumentaciju.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container mx-auto p-4 height-500 d-flex
justify-content-center flex-column  child">
            <section class="row  p-3">
                <article class="col-12 col-md-6">
                    <article>
                        <table class="table table-striped text-
center w-100" id="authorInfoTable">
                            <thead>
                            <tr>
                                <th >Information</th>
                            </tr>
                            </thead>
                            <tbody id="author-data">

                            </tbody>
                        </table>
                    </article>
                </article>
                <article class="col-12 col-md-6">
                    <img
src="{{asset("storage/images/autor.jpg")}}" alt="autor"
class="img-fluid"/>
                </article>
            </section>
            <article>
                <p class="roboto-slab">Preuzmite dokumentaciju
<a href="{{asset("documentation.pdf")}}">ovde</a>.</p>
            </article>
        </section>
    </section>
    @endsection
```

6.12.2 contact.blade.php

```
@extends("layouts.master")
@section("title", "Kontakt")
@section("keywords", "kontakt, pomoć, pitanja")
@section("description", "Preko ove stranice možete doći u dodir, 
postaviti pitanja i zatražiti pomoć  od našeg Administratora.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container child  p-3 d-flex flex-column 
justify-content-center mx-auto py-3 ">
            <article>
                <h1 class="title">Kontakt formular</h1>
            </article>
            <form name="formContact" id="formContact" 
method="POST" action="{{route("sendEmail")}}">
                <section class="row">
                    <article class="col-12 mb-3">
                        <input type="email" placeholder="Email" 
class="form-control" id="emailContact" />
                        @include("fixed.errorMessages.email")
                    </article>
                    <article class="col-12 mb-3">
                        <input type="text" placeholder="Naslov" 
class="form-control" id="subjectContact" />
                        @include("fixed.errorMessages.subject")
                    </article>
                    <article class="col-12 mb-3">
                        <textarea rows="10" cols="20" class="w-
100 form-control" id="messageContact" 
placeholder="Sadržaj"></textarea>
                        @include("fixed.errorMessages.message")
                    </article>
                    <article class="col-12 mb-3">
                        <input type="submit" value="Pošaljite 
poruku" class="btn btn-primary" />
                    </article>
                </section>
            </form>
        </section>
    </section>
    @endsection
```

6.12.3 discusions.blade.php

```
@extends("layouts.master")
@section("title", "Rasprave")
@section("keywords", "Rasprave, stav, teme")
@section("description", "Uzmite učešća u brojnim raspravama,
izaberite teme i jasno pokažite svoj stav.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container mx-auto child p-3">
            <article>
                <h3 class="roboto-slab-bold px-3">Dostupne
kategorije</h3>
            </article>
            <section class="row justify-content-between">
                @foreach($data["categories"] as $category)
                    @component("components.category",
["category"=> $category])
                    @endcomponent
                @endforeach
            </section>
        </section>
    </section>
@endsection
```

6.12.4 discusionsfromcategory.blade.php

```
@extends("layouts.master")
@section("title", "Prikaz rasprava")
@section("keywords", "kategorija, rasprava, oblast")
@section("description", "Preko ove stranice pristupate samo onim
raspravama koje pripadaju odabranoj kategoriji, odnosno
oblasti.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container mx-auto p-3 child">
            <article class="mb-3">
                @if(isset($data["discusions"][0]))
                <h3 class="roboto-slab-bold" data-
id="{{$data["discusions"][0]->idCategory}}" id="category-
name">{{$data["discusions"][0]->category}}</h3>
                    @endif
            </article>
            <section class="row" id="all-discusions">
```

```
                @if(count($data["discusions"]) > 0)
                @foreach($data["discusions"] as $discusion)
                    @component("components.discusion", ["dis" =>
$discusion, "userImage"=>$discusion->src])
                        @endcomponent
                    @endforeach
                @else
                    <article class="col-12">
                        <p class="roboto-slab-bold">Trenutno ne
postoje rasprave u ovoj kategoriji.</p>
                    </article>
                    <article class="col-12">
                        <p class="roboto-slab-bold">Možete
pregledati dostupne rasprave <a class=""
href="{{route("discusions")}}">ovde</a>, ili objaviti svoju iz
odabrane kategorije.</p>
                    </article>
                @endif
            </section>
        </section>
    </section>
    @endsection
```

6.12.5 editcomment.blade.php

```
@extends("layouts.master")
@section("title", "Izmena komentara")
@section("keywords", "izmena komentara, komentar, promena
sadržaja")
@section("description", "Uz pomoć ove stranice možete promeniti
Vaš komentar.")
@section("content")
    <section class="container-fluid py-3 wrapper">
       <section class="container mx-auto child p-3">
           <article>
               <h3 class="roboto-slab-bold"> Izmena
komentara</h3>
           </article>
            <section>
                <form method="POST" name="updateCommentForm"
id="updateCommentForm" action="#">
                    <article class="mb-3">
                        <textarea rows="5"
cols="20"id="messageUpdateComment" data-id="{{$comment-
>idComment}}" class="form-control">{{$comment-
>content}}</textarea>
```

```
                        @include("fixed.errorMessages.message")
                    </article>
                    <section class="row">
                        <article class="col-12 col-md-2 mb-3">
                            <input type="submit"  class="btn
btn-primary"value="Potvrdi"/>
                        </article>
                        <article class="col-12 col-md-2 mb-3">
                            @if(strpos(url()->previous(),
"user"))
                                <a href="{{route("profile")}}"
class="btn btn-primary">Nazad na profil</a>
                                @elseif(strpos(url()-
>previous(), "discusions"))
                                <a href="{{route("showFromCat",
["id"=>url()->previous()[strlen(url()->previous())-1]])}}"
class="btn btn-primary">Nazad na raspravu</a>
                                @endif
                        </article>
                    </section>
                </form>
            </section>
        </section>
    </section>
    @endsection
```

6.12.6 editdiscusion.blade.php

```
@extends("layouts.master")
@section("title", "Izmena rasprave")
@section("keywords", "izmena, izmena rasprave, izmena sadržaja")
@section("description", "Izmenite raspravu shodno Vašoj
zamisli.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container mx-auto child p-3">
            <article>
                <h3 class="roboto-slab-bold ">Izmena
rasprave</h3>
            </article>
            <section>
                <form name="updateDiscusionForm" action="#"
method="POST">
                    <section class="row">
                        <article class="col-12 mb-3">
```

```
                            <input type="text" class="form-
control" placeholder="Naslov" data-id="{{$data["dis"]-
>idDiscusion}}" id="subjectUpdateDiscusion"
value="{{$data["dis"]->title}}" />

@include("fixed.errorMessages.subject")
                        </article>
                        <article class="col-12 mb-3">
                            <textarea rows="5" cols="20"
class="form-control" placeholder="Sadržaj"
id="messageUpdateDiscusion">{{$data["dis"]->content}}</textarea>

@include("fixed.errorMessages.message")
                        </article>
                        <article class="col-12 mb-3">
                            <select id="categoryUpdateDiscusion"
class="form-control">
                            @foreach($data["categories"] as $c)
                                @if($c->idCategory ==
$data["dis"]->idCategory)
                                    <option value="{{$c-
>idCategory}}" selected>{{$c->name}}</option>
                                @else
                                    <option value="{{$c-
>idCategory}}">{{$c->name}}</option>
                                @endif
                            @endforeach
                            </select>

@include("fixed.errorMessages.category")
                        </article>
                        <article class="col-12 mb-3">
                            <input type="submit" class="btn btn-
primary" value="Izmeni" />
                            @if(strpos(url()->previous(),
"user"))
                                <a href="{{route("profile")}}"
class="btn btn-primary">Nazad na profil</a>
                                @elseif(strpos(url()-
>previous(), "discusions"))
                                <a href="{{route("showFromCat",
["id"=>url()->previous()[strlen(url()->previous())-1]])}}"
class="btn btn-primary">Nazad na raspravu</a>
                            @endif
                        </article>
                    </section>
```

```
                </form>
            </section>
        </section>
    </section>
    @endsection
```

6.12.7 editprofile.blade.php

```
@extends("layouts.master")
@section("title", "Izmeni podatke")
@section("keywords", "izmena podataka, izmena, podataka, izmena
lozinke")
@section("description", "Izmenite podatke po potrebi.")
@section("content")
    <section class="container-fluid wrapper py-3">
        <section class="container mx-auto child p-3">
{{--            Edit firstname, lastname, email, password, date
of birth and the profile picture --}}
            <article class="mb-3">
                <h3 class="roboto-slab-bold">Izmena
podataka</h3>
            </article>
            <form name="editProfileForm" id="editProfileForm"
method="POST" action="#">
                <section class="row">
                    <article class="col-12 mb-3">
                        <label for="firstnameEditProfile"
class="roboto-slab-bold">Ime</label>
                        <input type="text" placeholder="Ime"
id="firstnameEditProfile" value="{{session()->get('user')-
>firstname}}" class="form-control" />

@include("fixed.errorMessages.firstlastName",
["id"=>"firstname"])
                    </article>
                    <article class="col-12 mb-3">
                        <label
for="lastnameEditProfile"class="roboto-slab-
bold">Prezime</label>
                        <input type="text" placeholder="Prezime"
id="lastnameEditProfile" value="{{session()->get("user")-
>lastname}}"class="form-control"/>

@include("fixed.errorMessages.firstlastName",
["id"=>"lastname"])
```

```
                    </article>
                    <article class="col-12 mb-3">
                        <label 
for="emailEditProfile"class="roboto-slab-bold">Email</label>
                        <input type="email" placeholder="email" 
id="emailEditProfile" value="{{session()->get("user")-
>email}}"class="form-control"/>
                        @include("fixed.errorMessages.email")
                    </article>
                    <article class="col-12 mb-3">
                        <label 
for="passwordEditProfile"class="roboto-slab-bold">Stara 
lozinka</label>
                        <input type="password" 
placeholder="Lozinka" id="passwordEditProfile" class="form-
control"/>
                        @include("fixed.errorMessages.password", 
["id"=>"password"])
                    </article>
                    <article class="col-12 mb-3">
                        <label 
for="passwordAgainEditProfile"class="roboto-slab-bold">Nova 
lozinka</label>
                        <input type="password" 
placeholder="Lozinka ponovo" id="passwordNewEditProfile" 
class="form-control"/>
                        @include("fixed.errorMessages.password", 
["id"=>"passwordNew"])
                    </article>
                    <article class="col-12 mb-3">
                        <label 
for="dateEditProfile"class="roboto-slab-bold">Datum 
rođenja</label>
                        <input type="date" placeholder="Datum 
rođenja" value="{{session()->get("user")-
>birthDate}}"id="dateEditProfile" class="form-control"/>
                        @include("fixed.errorMessages.date")
                    </article>
                    <article class="col-12 mb-3">
                        <input type="submit" class="btn btn-
primary" name="submitUserData" value="Izmeni"  />
                    </article>
                </section>
            </form>
            <section>
                <form name="formChangeProfilePicture"
```

```
action="{{route("updateProfilePicture")}}" method="POST"
enctype="multipart/form-data" >
                    @csrf
                    @method("PUT")
                    <article class="col-12 mb-3">
                        <label for="pictureEditProfile"
class="roboto-slab-bold">Profilna fotografija:</label>
                        <input type="file"
name="pictureEditProfile" id="pictureEditProfile" />
                        @if(session()->get("message"))
                            <span class="error-message
">{{session()->get("message")}}</span>
                        @endif
                    </article>
                    <article>
                        <input type="submit"class="btn btn-
primary" name="submitProfilePicture" value="Izmeni" />
                        <a href="{{route("profile")}}"
class="btn btn-primary">Nazad na profil</a>
                    </article>
                </form>
            </section>
            @if(session()->has("success"))
            <section>
                <p class="roboto-slab-bold">{{session()-
>get("success")}}</p>
            </section>
            @elseif(session()->has('error'))
                <section>
                    <p class="roboto-slab-bold">{{session()-
>get("error")}}</p>
                </section>
            @endif
        </section>
    </section>
    @endsection
```

6.12.8 index.blade.php

```
@extends("layouts.no-nav-layout")
@section("title", "Početna")
@section("keywords", "mreža, prijava, razmišanje, kritičko,
kritika, kritićko razmišljanje")
@section("description", "Društvena mreža  za ljubitelje
kritičkog razmišljanja.")
@section("content")
    <section class="container-fluid wrapper py-3 vheight-100 d-
flex align-items-center">
        <section class="container child p-3 mx-auto py-3 d-flex
justify-content-center align-items-center height-700">
            <section class="row justify-content-center">
                <section class="col-12 col-md-6 p-2">
                    <figure class="col-6 d-flex ">
                        @include("fixed.logo")
                    </figure>
                    <article>
                        <p class="roboto-slab-bold">Vaše
mišlenje, drušvena mreža za ljubitelje kritičkog razmišljanja.
</p>
                    </article>
                </section>
                <article class="col-12 col-md-6 p-2">
                    <article>
                        <h1 class="hack title">Prijava</h1>
                    </article>
                    <form name="formLogin" id="formLogin"
method="POST" action="{{route("login")}}">
                        @csrf
                        <article class="mb-3">
                            <input type="text"id="emailLogin"
class="form-control hack"placeholder="Email"/>

@include("fixed.errorMessages.email")
                        </article>
                        <article class="mb-3">
                            <input
type="password"id="passwordLogin" class="form-control "
placeholder="Lozinka"/>

@include("fixed.errorMessages.password", ["id"=> "password"])
                        </article>
                        <article class="mb-3">
                            <input type="submit" class="btn btn-
primary " value="Potvdi"/>
```

```
                        </article>
                    </form>
                    <article>
                        <p class="roboto-slab-bold">Nemate
nalog? Napravite jedan <a
href="{{route("register")}}"class="orange-forecolor roboto-slab-
bold" >ovde</a>.</p>
                    </article>
                </article>
            </section>
        </section>
    </section>
@endsection
```

6.12.9 newdiscusion.blade.php

```
@extends("layouts.master")
@section("title", "Nova rasprava")
@section("keywords", "Nova rasprava, nova objava, nova,
mišljenje")
@section("description", "Objavite novu raspravu, izmesite Vaše
ideje i poglede po brojnim pitanjima.")
@section("content")
    <section class="container mx-auto">
        <section class="row">
            <section class="col-12">
                <article>
                    <h3 class="sub-title roboto-slab-bold">Nova
rasprava</h3>
                </article>
                <section class="w-100">
                    <form name="formNewDescusion"
id="formNewDiscusion" action="" method="POST">
                        <article>
                            <h3 class="sub-tittle roboto-
slab">Naslov</h3>
                        </article>
                        <article class="mb-3">
                            <input type="text"
placeholder="Naslov" id="subjectNewDiscusion" class="form-
control" />

@include("fixed.errorMessages.subject")
                        </article>
                        <article>
```

```
                            <h3 class="sub-tittle roboto-
slab">Sadržaj</h3>
                        </article>
                        <article class="mb-3">
                            <textarea rows="10" cols="20"
class="form-control w-100" placeholder="Sadržaj"
id="messageNewDiscusion"></textarea>

@include("fixed.errorMessages.message")
                        </article>
                        <article class="mb-3">
                            <h3 class="sub-title roboto-
slab">Kategorija</h3>
                            <select id="categoryNewDiscusion"
class="form-control">
                                <option value="-
1">Izaberite</option>
                                @for($i = 0; $i <
count($categories); $i++)
                                <option
value="{{$categories[$i]->idCategory}}"> {{$categories[$i]-
>name}}</option>
                                    @endfor
                            </select>

@include("fixed.errorMessages.category")
                        </article>
                        <article class="mb-3">
                            <input type="submit" class="btn btn-
primary" value="Objavi" />
                            <a href="{{route("profile")}}"
class="btn btn-primary">Nazad na profil</a>
                        </article>
                    </form>
                </section>
            </section>
        </section>
    </section>
@endsection
```

6.12.10 profile.blade.php

```
@extends("layouts.master")
@section("title", "Moj nalog")
@section("keyword", "nalog, pratioci, objava, izmeni podatke")
@section("description", "Preko ove stranice imate potpun pregled
u Vaše aktivnosti, Vaše objave, komentare i pratioce.")
@section("content")
    <section class="container-fluid  py-3 wrapper">
        <section class="container mx-auto">
            <section class="row child row-column">
                <section class="col-12 p-2 d-flex justify-
content-center align-items-center  flex-column ">
                    <figure class="d-flex  align-items-end
justify-content-center w-50  " id="user-profile-pic">
{{--                        <img src="{{asset("assets/img/" .
$data["userData"]->src)}}" class="img-fluid profile-pic"
alt="profile picture"/>--}}
                        <img src="{{asset("/storage/images/" .
$data["userImage"])}}" class="img-fluid profile-pic"
alt="profile picture"/>
                    </figure>
                    <section  class="w-100">
                        <p id="users-firstlastname" class="text-
center roboto-slab-bold"data-sess="{{session()->get("user")-
>idUser}}" data-id="{{$data["userData"]-
>idUser}}">{{$data["userData"]->firstname}} {{$data["userData"]-
>lastname}}</p>
                    </section>
                    <section class="w-100 text-center"
id="followUnfollow-wrapper">
                        @if(session()->get("user")->idUser ==
$data["userData"]->idUser)
                        <a href="{{route("editprofile")}}"
class="btn btn-primary">Izmeni podatke</a>
                            @elseif($isAFollower  &&
$isAFollower == true)
                                <a href="{{route("unfollow",
["id"=>$data["userData"]->idUser])}}" data-
id="{{$data["userData"]->idUser}}" id="unfollow-link"class="btn
btn-danger ">Otprati</a>
                            @else
{{--                            later add logic for separating
followed from unfollowed user--}}
                                <a href="#" data-
id="{{$data["userData"]->idUser}}"id="follow-link" class="btn
btn-primary">Zaprati</a>
```

```
                        @endif
                    </section>
                </section>
                <!-- NA OVOM MESTU SE NALAZI NAVIGACIJA SA
LINKOVIMA KA ZIDU I PRIJATELJJIMA-->
                <article class="p-2 w-100" id="users-profile-
menu">
                    <ul class="d-flex mb-0">
                    @foreach($data["menu"] as $link)
                        @if($link->name == "Zid")
                            <li class="mx-1">
                                <a href="#" data-id="{{$link-
>idAttr}}"class="roboto-slab-bold sub-title orange-forecolor
users-profile-menu-link mx-1">{{$link->name}}</a>
                            </li>
                        @else
                            <li class="mx-1">
                                <a href="#" data-id="{{$link-
>idAttr}}"class="roboto-slab-bold sub-title users-profile-menu-
link mx-1">{{$link->name}}</a>
                            </li>
                        @endif
                    @endforeach
                    </ul>
                </article>
{{--                Wall--}}
                <section class="col-12 p-2 " id="wall">
                    <article class="d-flex my-3">
                        <h3 class="sub-title roboto-slab-
bold">Rasprave</h3>
                        @if(session()->get("user")->idUser ==
$data["userData"]->idUser)
                        <article class="mx-2">
                            <a href="{{route("newDiscusion")}}"
class="btn btn-primary">Nova rasprava</a>
                        </article>
                            @endif
                    </article>
                    <section id="users-discusions" class="col-12
py-2 ">
                        @if(count($data["discusions"]) > 0)
                        @foreach($data["discusions"] as $dis)
                            @component("components.discusion",
["dis"=>$dis, "userImage"=>$data["userImage"]])
                            @endcomponent
                        @endforeach
                        @else
                            <h3 class="roboto-slab-
bold">Trenutno ne postoje rasprave.</h3>
                        @endif
                    </section>
                </section>
            {{-- Followers--}}
                <section class="col-12 p-2 d-none  py-3"
id="followers">
                    <article>
                        <form name="formSearchFollowers"
id="formSearchFollowers" action="{{route("followers")}}"
method="POST" >
                            <section class="row py-2 white-
backcolor">
                                <article class="col-12 col-md-10
mb-3">
                                    <input type="text"
class="form-control" placeholder="Unesite ime i prezime"
id="searchFollower" />

@include("fixed.errorMessages.search")
                                </article>
                                <article class="col-12 col-md-
2">
                                    <input type="submit"
class="btn btn-primary" value="Pretraži"/>
                                </article>
                            </section>
                        </form>
                    </article>
                    @if(count($data["followers"]) > 0)
                    <section class="row  justify-content-
between" id="followers-parent">
                        @foreach($data["followers"] as
$follower)
                            @component("components.follower-
card", ["follower"=>$follower])
                            @endcomponent
                        @endforeach
                    </section>
                   @else
                       <h3 class="roboto-slab-bold">Trenutno ne
pratite nikoga.</h3>
                    @endif
                </section>
            </section>
        </section>
    </section>
    @endsection
```

6.12.11 registration.blade.php

```
@extends("layouts.no-nav-layout")
@section("title", "Registracija")
@section("keywords", "registracija, nalog, napravite nalog,
vasemisljenje registracija")
@section("description", "Napravite nalog ovde, brzo i
jednostavno")

@section("content")
    <section class="container-fluid d-flex align-items-center
justify-content-center vheight-100 wrapper">
        <article class="container ">
            <section class="row flex-column child p-3">
                <article>
                    <h1 class="title">Registracija</h1>
                </article>
                <form name="formRegistration" id="formRegister"
method="POST" action="{{route("createAccount")}}">
                    <label for="firstnameRegistration"
class="roboto-slab">Ime i prezime</label>
                    <section class="d-flex flex-column flex-md-
row mb-3">
                        <article class="input-group mr-md-1 mb-3
mb-md-0">
                            <input type="text"
placeholder="Ime" id="firstnameRegistration" class="form-
control"/>

@include("fixed.errorMessages.firstlastName", ["id" =>
"firstname"])
                        </article>
                        <article class="input-group">
                            <input type="text"
placeholder="Prezime" id="lastnameRegistration" class="form-
control"/>

@include("fixed.errorMessages.firstlastName", ["id"=>
"lastname"])
                        </article>
                    </section>
                    <article class="mb-3">
                        <label
for="emailRegistration"class="roboto-slab">Email</label>
                        <input type="email" placeholder="Email"
class="form-control"  id="emailRegistration" />
                        @include("fixed.errorMessages.email")
                    </article>
                    <article class="mb-3">
                        <label
for="passwordRegistration"class="roboto-slab">Lozinka</label>
                        <input type="password"
placeholder="Lozinka" class="form-control"
id="passwordRegistration" />
                        @include("fixed.errorMessages.password",
["id"=>"password"])
                    </article>
                    <article class="mb-3">
                        <label
for="passwordRegistration"class="roboto-slab">Lozinka
ponovo</label>
                        <input type="password"
placeholder="Lozinka ponovo" class="form-control"
id="passwordAgainRegistration" />
                        @include("fixed.errorMessages.password",
["id"=>"passwordAgain"])
                    </article>
                    <article class="mb-3">
                        <label
for="dateRegistration"class="roboto-slab">Datum rođenja</label>
                        <input type="date"  class="form-control"
id="dateRegistration" />
                        @include("fixed.errorMessages.date")
                    </article>
                    <article>
                        <input type="submit" class="btn btn-
primary" value="Napravi nalog" />
                    </article>
                </form>
                <article>
                    <p>
                    <p class="roboto-slab-bold">Imate nalog?
Kliknite <a href="{{route("home")}}" >ovde</a>.</p>
                </article>
            </section>
        </section>
```

```
    </section>
    @endsection
```

6.13 Sass

```
    @import
url('https://fonts.googleapis.com/css2?family=Roboto+Slab&displa
y=swap');
    *{
        margin: 0;
        padding: 0;
    }
    .row{
        margin: 0;
    }
    // COLORS
    $orange-color: #f2740d;
    $black-color: #000;
    $orange-light-color: #fee4cd;
    $blue-color: #0b96ff;
    $blue-light-color: #cde9fe;
    $grey-dark-color: #34373c;
    $white: #fff;
    $grey-light:#e0e0de;
    // FONTS
    .roboto-slab{
        font-family: 'Roboto Slab', serif;
    }
    .roboto-slab-bold{
        @extend  .roboto-slab;
        font-weight: bold;
    }

    // VHeight
    .vheight-100{
        height: 100vh;
    }
    // COLORS CLASSES
    //#e0e0de

    //"grey-light": #c9c9c7
    @each $name, $value in (
        "orange": $orange-color,
        "black" : $black-color,
        "orange-light": $orange-light-color,
        "blue" : $blue-color,
        "blue-light" : $blue-light-color,
        "grey-dark": $grey-dark-color,
        "white" : $white,
        "grey-light":$grey-light
    ){
        .#{$name}-forecolor {
            color: $value;
        }
        .#{$name}-backcolor{
            background-color: $value;
        }
    }
    .title{
        font-size: 2rem;
        @extend  .roboto-slab;
    }
    .sub-title{
        font-size: 1.5rem;
    }

    // MIXINS
    @mixin hover{
        @content
    }
    @mixin setBorder($color, $borderWidth : 1px){
        border: $borderWidth solid $color;
        border-radius: 8px;
    }
    @mixin setSpecificBorder($side, $color, $borderWidth: 1px){
        @if($side == top) {
           border-top: $borderWidth solid $color;
        }
        @else if($side == botton){
            border-bottom: $borderWidth solid $color;
        }
        @else if($side == left){
            border-left: $borderWidth solid $color;
        }
        @else if($side == right){
            border-right : $borderWidth solid $color;
        }
    }

    .wrapper{
        @extend .grey-light-backcolor;
        @include setBorder($grey-light);
        .child{
            @include setBorder($white);
            @extend .white-backcolor;
        }
    }
    // Transfrom classes
    .transition-color{
        transition: color .5s ease-in-out;
    }
    .transition-backcolor{
        transition: background-color .5s ease-in-out;
    }
// a and ul
    a{
        text-decoration: none !important;
        @extend  .black-forecolor;
        @extend .transition-color;
        &:hover{
            text-decoration: none;
            color: $orange-color;

        }
    }
    ul{
        list-style-type: none;
    }
// BOX-SHADOW
    .box-shadow{
        box-shadow: 3px 3px 20px rgba(#333, 0.5);
    }
 //BODER CLASSES
    .border-top-orange{
        border-top: 3px solid $orange-color ;
    }
    .border-grey-light {
        @include setBorder($grey-light);
    }
// NAVIGATION
    header{
        @extend .box-shadow;
        @extend .border-top-orange;
        position: sticky;
        top: 0;
        z-index: 10;
    }
// NAVIGATION--END
    .form-element-message{
       display:none;
    }
    .error-message{
        color: red;
    }
    .error-border{
        border-color: red;
    }
    .success-message{
        color: green;
    }
    .success-border{
        border-color: green;
    }
    .form-control{
        transition: border 0.5s ease-in-out;
        border-width: 3px;
        @extend .roboto-slab;
    }
    #username:hover #user-links{
        display:block;
    }
    #user-links{
        display:none;
    }
    .position-relative{
        position:relative;
    }
    .position-absolute{
        position: absolute;
    }
    //FOOTER
    footer{
        @extend .border-top-orange;
    }
    $font-x : 14pt;
    .fa-bars{
        font-size:$font-x;
    }
    .cursor-pointer{
        cursor:pointer;
    }
    #bars{
        @extend .cursor-pointer;
    }
    #bars-menu{
        z-index: 15;
        @extend .box-shadow;
        transition: width .5s ease-in-out;
        overflow: hidden;
        width: 0;
        &__close{
            font-size: $font-x;
        }
    }
    @each $value in (500, 700, 400){
       .height-#{$value}{
           height: #{$value}px;
       }
    }
    .btn{
        @extend .roboto-slab;
    }
    .profile-pic{
        //$borderWidth: 3px;
        //@include setBorder($grey-light);
        @extend .border-grey-light;
    }
    .profile-pic-small{
       //@include setBorder($orange-color);
        @extend .border-grey-light;
    }
    .profile-pic-small-wrapper-width{
       width: 100px;
    }

    .title-1{
        font-size: 15px;
    }
    .comment{
        //@include  setSpecificBorder(bottom, $orange-color, 
3px);
        //border-bottom: 1px solid rgba($orange-color, 0.6);
        border-bottom: 1px solid $grey-light;
    }
    .comment-section{
        display:none;
    }

    #users-profile-menu{
        border-top :1px solid $grey-light;
        border-bottom: 1px solid $grey-light;
    }
    .follower-card{
        //@include setBorder($grey-light);
        @extend .border-grey-light;
    }
    .discusions {
        @extend .border-grey-light;
    }

    .transparent-grey-dark{
    }
    .my-modal{
        background-color: rgba($grey-dark-color, .6);
        position: fixed;
        top:0;
        z-index: 100;
        height: 100vh;
    }
    .category{
        @include setBorder($orange-color, 2px);
        @extend .orange-backcolor;
        @extend .white-forecolor;
        padding: 5px;
        &-name{
            //border-bottom: 2px solid $white;
        }
    }
```

6.14 Javascript/jQuery

```
    const siteRoot = "http://127.0.0.1:8000/";
    const modal = {
        selector : "#modal",
        show :function(){
            $(this.selector).removeClass("d-none");
            $(this.selector).addClass("d-flex");
        } ,
        hide : function(){
            $(this.selector).addClass("d-none");
            $(this.selector).removeClass("d-flex");
        },
        set : function(content){
            $(this.selector + "-body") .html(content);
        },
    };
    function setUpAjaxHeader(){
        $.ajaxSetup({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-
token"]').attr('content')
            }
        });
    }
    setUpAjaxHeader();
    function onPage(name){
        return location.href.indexOf(name) !== -1;
    }
    function valueAcceptable(control){
        control.element.removeClass("error-border");
        control.element.addClass("success-border");
        control.messageSpan.removeClass("d-block");
        control.messageSpan.addClass("d-none");
    }

    function valueUnacceptable(control){
        control.element.removeClass("success-border");
        control.element.addClass("error-border");
        control.messageSpan.removeClass("d-none");
        control.messageSpan.addClass("d-block");
    }

    function addBlurEventToControls(controls){
        for(let control of controls){
            control.element.on("blur", function(){
                if(control.element.val().match(control.regex)){
                    valueAcceptable(control);
                }else {
                    valueUnacceptable(control);
                }
            });
        }
    }

    function formValuesAcceptable(controls){
        let errorNum = 0;
        for(let control of controls){
            if(!control.element.val().match(control.regex)){
                valueUnacceptable(control);
                errorNum++;
            }else {
                valueAcceptable(control);
            }
        }
        return errorNum === 0;
    }

    function clearForm(controls){
        for(let control of controls){
            control.val("");
            control.removeClass("success-border");
        }
    }
    function sendGETRequest(
        url,
        callback
    ){
        $.ajax({
            url:  url,
            method: "GET",
            dataType: "json",
            statusCode: {
                201: function (result) {
                    if (callback != null) {
                        callback();
                    }
                },
                200: function (result) {
                    if (callback != null) {
                        callback(result);
                    }
                },
                422: function () {
                    console.error(422);
                },
                500: function () {
                    console.error(500);
                }
            }
        });
    }
    function sendPOSTRequest(
        url,
        data,
        callback
    ){
        $.ajax({
            url:  url,
            method: "POST",
            data: data,
            dataType: "json",
            statusCode: {
                201: function (result) {
                    if (callback != null) {
                        callback();
                    }
                },
                200: function (result) {
                    if (callback != null) {
                        callback(result);
                    }
                },
                422: function () {
                    console.error(422);
                },
                500: function () {
                    console.error(500);
                }
            }
        });
    }

    // regexes
    const firstLastNameRegex = /^[A-ZŽĐŠĆČ][a-
zžđšćč]{2,14}(\s[A-ZŽĐŠĆČ][a-zžđšćč]{2,14}){0,1}$/;
    const emailRegEx = /^[a-z][a-z0-9]{2,14}([\._][a-z0-
9]{1,14}){0,4}\@([a-z]{3,5}\.){1,2}[a-z]{2,3}$/;
    const passwordRegEx = /^(?=.*[A-Za-z])(?=.*\d)[A-Za-
z\d]{8,16}$/;
    const messageRegEx = /^[A-ZŽĐŠĆČ@][A-zŽĐŠĆČžđšćč0-
9\(\)\.,?!:\/\;\s\n\*-_@]{1,}$/;
    const dateRegEx = /^[0-9]{4}\-[0-9]{2}\-[0-9]{2}$/;
    const subjectRegEx = /^[A-ZŽĐŠĆČ][a-zžđšćč]{2,19}(\s[A-
ZŽĐŠĆČa-zžđšćč0-9\@]{1,19}){0,5}$/;
    const searchRegEx = /^[A-zŽĐŠĆČžđšćč\s\@\-\_]{0,50}$/;
    // const messageRegEx = /^[A-ZŽĐŠĆČ][a-
zžđšćč\(\)\/\.\?\!\.\,]{2,19}(\s[A-ZŽĐŠĆČa-zžđšćč0-
9\(\)\/\.\?\!\.\,]{1,20}){0,}$/;
    const categoryRegEx = /^[0-9]$/;
   if(document.getElementById("modal") != undefined) {
      $("#close-modal") .click(function(e){
          e.preventDefault();
          modal.hide();
      });
      $("#close-modal-btn").click(function(e){
          e.preventDefault();
          modal.hide();
      });
   }


    // bars-menu code
    if(document.getElementById("bars-menu") != undefined){
        $("#bars-menu__close").on("click", function(e){
           e.preventDefault();
            $("#bars-menu").css("width", "0px");
        });
    }
    if(document.getElementById("bars") != undefined){
        $("#bars").on("click", function(){
            $("#bars-menu").css("width", "100%");
        });
    }
    // home page
    if(document.getElementById("formLogin")){
        // add onblur event for each form control
        const loginControls = [
            {
                "element" : $("#emailLogin"),
                "regex" : emailRegEx,
                "messageSpan" : $("#email")
            },
            {
                "element" : $("#passwordLogin"),
                "regex" : passwordRegEx,
                "messageSpan" : $("#password")
            }
        ];
        addBlurEventToControls(loginControls);

document.getElementById("formLogin").addEventListener("submit",
function(e){
            e.preventDefault();
            // validation
            if(formValuesAcceptable(loginControls)){
                // ready to send request
                // complete this jquery code
                // I need correct set of properties for
apropriate request sending
                // $.ajaxSetup({
                //     headers: {
                //         'X-CSRF-TOKEN': $('meta[name="csrf-
token"]').attr('content')
                //     }
                // });
                // setUpAjaxHeader();
                $.post("/auth/login",
                    {email:$("#emailLogin").val(),
password:$("#passwordLogin").val(), submit: true},
                    function(data){
                    if(data.error){
                       let html = `<h3 class="roboto-
slab">${data.error}</h3>`;
                       modal.set(html) ;
                       modal.show();
                    }
                    else if (data.success){
                        let html = `<h3 class="roboto-
slab">Uspešno prijavljivanje</h3>`;
                        clearForm([
                            $("#emailLogin"),
                            $("#passwordLogin")
                        ]);
                        modal.set(html) ;
                        modal.show();
                        location.href = siteRoot + "user/";
                    }
                });
            }
        });
    }
    if(document.formRegistration != undefined){
        const registrationControls = [
            {
                "element" : $("#firstnameRegistration"),
                "regex" : firstLastNameRegex,
                "messageSpan" : $("#firstname")
            },
            {
                "element" : $("#lastnameRegistration"),
                "regex" : firstLastNameRegex,
                "messageSpan" : $("#lastname")
            },
            {
                "element" : $("#emailRegistration"),
                "regex" : emailRegEx,
                "messageSpan" : $("#email")
            },
            {
                "element" : $("#passwordRegistration"),
                "regex" : passwordRegEx,
                "messageSpan" : $("#password")
            },
            {
                "element" : $("#passwordAgainRegistration"),
                "regex" : passwordRegEx,
                "messageSpan" : $("#passwordAgain")
            },
            {
                "element" : $("#dateRegistration"),
                "regex" : dateRegEx,
                "messageSpan" : $("#date")
            }
        ];
        addBlurEventToControls(registrationControls);

document.getElementById("formRegister").addEventListener("submit
", function(e){
            e.preventDefault();
            if(formValuesAcceptable(registrationControls) &&
$("#passwordRegistration").val() ===
$("#passwordAgainRegistration").val()){
                // setUpAjaxHeader();
                $.post(
                    "/auth/create/",
                    {
                        "firstname" :
$("#firstnameRegistration").val(),
                        "lastname" :
$("#lastnameRegistration").val(),
                        "email" : $("#emailRegistration").val(),
                        "password" :
$("#passwordRegistration").val(),
                        "passwordAgain" :
$("#passwordAgainRegistration").val(),
                        "date" : $("#dateRegistration").val(),
                        "submit" : true
                    },
                    function(res){
                        let html = `<h3 class="roboto-
slab">${res.message}</h3>`;
                        modal.set(html);
                        modal.show();
                        if(res.success){
                            clearForm([
                                $("#firstnameRegistration"),
                                $("#lastnameRegistration"),
                                $("#emailRegistration"),
                                $("#passwordRegistration"),
                                $("#passwordAgainRegistration"),
                                $("#dateRegistration")
                            ]);
                        }
                    }
                );
            }else {
                let html = "<h3>Unete vrednosti moraju biti u
željenim formatima. Takođe lozinke se moraju poklapati.</h3>";
                modal.set(html);
                modal.show();
            }
        });
    }
    else if(document.getElementById("formContact") != undefined)
    {
        const contactControls = [
            {
                "element" : $("#emailContact"),
                "regex" : emailRegEx,
                "messageSpan": $("#email")
            },
            {
                "element" : $("#subjectContact"),
                "regex" : subjectRegEx,
                "messageSpan" : $("#subject")
            },
            {
                "element": $("#messageContact"),
                "regex" : messageRegEx,
                "messageSpan" : $("#message")
            }
        ];
        addBlurEventToControls(contactControls  );

document.getElementById("formContact").addEventListener("submit"
, function(e){
            e.preventDefault();
            if(formValuesAcceptable(contactControls)){

                $.post("/contact/send",
                    {
                        "email" : $("#emailContact").val(),
                        "subject" : $("#subjectContact").val(),
                        "message" : $("#messageContact").val()
                    },
                    function(res){
                        let html;
                        if(res.success){
                            html = `<h3 class="roboto-slab-
bold">Poruka je uspešno poslata.</h3>` ;
                            clearForm([
                                $("#emailContact"),
                                $("#subjectContact"),
                                $("#messageContact")
                            ]);
                        }else if(res.error){
                            html = "Došlo je do grekše na
serveru.";
                        }
                        modal.set(html);
                        modal.show();
                    });
            }
        });
    }
    else if(document.getElementById("author-data") !=
undefined){
        // sendGETRequest(
        //     "author/get",
        //     function(res){
        //         let html  = "";
        //         let titles = [
        //             "Ime:",
        //             "Skola:",
        //             "Indeks:"
        //         ];
        //         let i = 0;
        //         for(let p in res){
        //             html += `
        //                 <tr>
        //                     <td>${titles[i++]} ${res[p]}</td>
        //                 </tr>
        //             `;
        //         }
        //         document.getElementById("author-
data").innerHTML = html;
        //         alert("ok");
        //     }
        // );
        $.get("/author/get", function(res){
            let html  = "";
            let titles = [
                "Ime:",
                "Skola:",
                "Indeks:"
            ];
            let i = 0;
            for(let p in res){
                html += `
                        <tr>
                            <td>${titles[i++]} ${res[p]}</td>
                        </tr>
                    `;
            }
            document.getElementById("author-data").innerHTML =
html;
        });
    } if(document.getElementById("users-firstlastname") !=
undefined){
        function setEventForUnfollowLink(){
            $("#unfollow-link").click(function(e){
                e.preventDefault();
                // send ajax request
                $.ajax({
                    "url" : "/user/unfollow",
                    "dataType" : "json",
                    "method" : "DELETE",
                    "data" : {
                        "id" : $(this).data('id')
                    },
                    "success" :function(res) {
                        if(res.success){
                            let html = `
                                <a href="#" data-
id="${res.id}"id="follow-link" class="btn btn-
primary">Zaprati</a>
                            `;
                            $("#followUnfollow-
wrapper").html(html);
                            setEventForFollowLink();
                            // location.href = siteRoot +
"user/show/" + res.id;
                        }else if(res.error){
                            let html = "<h3 class='roboto-slab-
bold'>Došlo je do grepke na serveru.</h3>";
                            modal.set(html);
                            modal.show();
                        }
                    }
                });
            });
        }
        function setEventForFollowLink(){
            $("#follow-link").click(function(e){
                e.preventDefault();
                $.ajax({
                    "url"   : "/user/follow/",
                    "dataType" : "json",
                    "method" : "POST",
                    "data" : {"id": $(this).data("id")},
                    "success" : function(res){
                        if(res.success){
                            let html = `
                                <a href="#" data-id="${res.id}"
id="unfollow-link"class="btn btn-danger ">Otprati</a>
                            `;
                            $("#followUnfollow-
wrapper").html(html);
```

```
                            setEventForUnfollowLink();
                            // location.href= siteRoot +
"user/show/" + res.id;
                        }else if(res.error){
                            let html = "<h3 class='roboto-slab-
bold'>Došlo je do greške na serveru.</h3>"
                            modal.set(html);
                            modal.show();
                        }
                    }
                });
            });
        }
        setEventForLikeCommentLinks();
        setEventForDislikeCommentLinks();
        setEventForFollowLink();


        // DISCUSIONS

        // FOLLOWERS
        $(".users-profile-menu-link").click(function(e){
            e.preventDefault();
            // alert($(this).data("id"));
            let selectedLink =
Array.from(document.getElementsByClassName("users-profile-menu-
link")).filter(l => $(l).data("id") === $(this).data('id'));
            let ids =
Array.from(document.getElementsByClassName("users-profile-menu-
link")).map(l => $(l).data("id"));
            $(ids).each(function(){
                // console.log(this)  ;
                $("#"+ this).addClass("d-none");
            });
            $("#" + $(selectedLink).data("id")).removeClass("d-
none");
            $(".users-profile-menu-link").removeClass("orange-
forecolor");
            $(selectedLink).addClass("orange-forecolor");
        });

        // $("#follow-link").click(function(e){
        //   e.preventDefault();
        // });
        // UNFOLLOW LINKS
        setEventForUnfollowLink();
```

```
            $("#unfollow-link").click(function(e){
                e.preventDefault();
                // send ajax request
                $.ajax({
                    "url" : "/user/unfollow",
                    "dataType" : "json",
                    "method" : "DELETE",
                    "data" : {
                        "id" : $(this).data('id')
                    },
                    "success" :function(res) {
                        if(res.success){

                            location.href = siteRoot +
"user/show/" + res.id;
                        }else if(res.error){
                            let html = "<h3 class='roboto-slab-
bold'>Došlo je do grepke na serveru.</h3>";
                            modal.set(html);
                            modal.show();
                        }
                    }
                });
            });
            setEventForUnfollowLinks()
        const searchFollowerControls = [
            {
                "element" : $("#searchFollower"),
                "regex" : searchRegEx,
                "messageSpan" : $("#search")
            }
        ];
        function setBlurEventForSearchControl(){
            for(let control of searchFollowerControls){
                control.element.on("blur", function(){

if(control.element.val().match(control.regex)){
                        valueAcceptable(control);
                        if(control.element.val() == ""){
                            $.ajax({
                                url: "/user/followers"   ,
                                dataType: "json",
                                method: "GET",
                                data : {
                                    "search" : ""
                                },
                                success: function(followers){
                                    loadFollowers(followers, "Ne
postoje pratioci koji zadovoljavaju kriterijum pretrage.");
                                }
                            });
                        }
                    }else {
                        valueUnacceptable(control);
                    }
                });
            }
        }
        setBlurEventForSearchControl();
        // addBlurEventToControls(searchFollowerControls);
        function setEventForUnfollowLinks(){
            $(".unfollow-link").each(function(){
               $(this).click(function(e){

                   e.preventDefault();
                   $.ajax({
                       "url" : "/user/unfollow",
                       "dataType" : "json",
                       "method" : "DELETE",
                       "data" : {
                           "id" : $(this).data('id'),
                           "returnFollowers" : true,
                           "search" : $("#searchFollower").val()
                       },
                       "success" :function(res) {
                           if(res.success){
                               loadFollowers(res.followers,
"Trenuto ne pratite nikoga.");
                           }else if(res.error){
                               let html = "<h3 class='roboto-
slab-bold'>Došlo je do grepke na serveru.</h3>";
                               modal.set(html);
                               modal.show();
                           }
                       }
                   });
               })
            });
        }
        function loadFollowers(followers, msg = null){
            let html = "";
            for(let follower of followers){
                html += `
<section class="col-12 col-md-4 p-3   ">
   <section class="follower-card ">
       <figure class="w-100 d-flex justify-content-center">
           <img src="/storage/images/${follower.src}"
class="img-fluid rounded" alt="user image"/>
       </figure>
       <article class="px-2">
           <p class="roboto-slab-bold">${follower.firstname}
${follower.lastname}</p>
       </article>
       <article class="p-2">
           <a class="btn btn-primary go-to-profile-link"
href="${siteRoot + "user/show/" + follower.idUser}" >Profil</a>
                `;
            if($("#users-firstlastname").data("sess") !=
follower.idUser){
                html += `
           <a href="#" class="btn btn-danger unfollow-link"
data-id="${follower.idUser}">Otprati</a>
                ` ;
            }
                html += `
       </article>
   </section>
</section>
                `;
            }
            if(html == ""){
                html += `
                <article class="col-12 ">
                <p class="roboto-slab-bold">${msg}</p>
</article>
                   `    ;

                $("#followers-parent").html(html);
            }else {
                $("#followers-parent").html(html);
                setEventForUnfollowLinks();
            }
        }

document.getElementById("formSearchFollowers").addEventListener(
"submit", function(e){
            e.preventDefault();
            if(formValuesAcceptable(searchFollowerControls)){
                $.ajax({
                    url: "/user/followers"   ,
                    dataType: "json",
                    method: "GET",
                    data : {
                        "search" : $("#searchFollower").val()
                    },
                    success: function(followers){
                        loadFollowers(followers, "Ne postoje
pratioci koji zadovoljavaju kriterijum pretrage.");
                    }
                });
            }
        });
    }
    function setEventForDislikeCommentLink(link){
        link.click(function(e){
            e.preventDefault();
            $.ajax({
                "method" : "POST",
                "url" : "/likes/delete",
                "data" : {
                    "id" : link.data("id")
                },
                "success" : function(res){
                    if(res.success){
                        // link.text(res.linkText);
                        //like-comment-link-wrapper
                        const parent = link.parent();
                        parent.html(`
        <span class="roboto-slab-bold">Sviđanja <span
class="like-count">${res.numberOfLikes}</span></span>
        <a href="#" class="btn btn-link like-comment" data-
id="${res.idComment}">Sviđa mi se</a>
                                `);

setEventForLikeCommentLink(parent.find(".like-
comment").first());

                    }else if(res.error){
                        modal.set("Doslo je do greske na
serveru.");
                        modal.show();
                    }
                }
            });
        });
    }
    function setEventForLikeCommentLink(link){
        link.click(function(e){
            e.preventDefault();
            $.ajax({
                "method" : "POST",
                "url" : "/likes/store",
                "data" : {
                    "id" : link.data('id')
                },
                "success" : function(res){
                    if(res.success){
                        const parent = link.parent();
                        parent.html(`
        <span class="roboto-slab-bold">Sviđanja <span
class="like-count">${res.numberOfLikes}</span></span>
        <a href="#" class="btn btn-link dislike-comment" data-
id="${res.idComment}">Ne sviđa mi se</a>
                            `);

setEventForDislikeCommentLink(parent.find(".dislike-
comment").first());
                    }else {
                        modal.set("Doslo je do greske na
serveru.");
                        modal.show();
                    }
                }
            });
        });
    }
    function setEventForLikeCommentLinks(){
        $(".like-comment").each(function(){
            $(this)  .click(function (e){
                e.preventDefault();
                setEventForLikeCommentLink($(this)) ;
            });
        });
    }
    function setEventForDislikeCommentLinks(){
        $(".dislike-comment").each(function(){
            $(this).click(function(e){
                e.preventDefault();
                setEventForDislikeCommentLink($(this));
            });
        });
    }
    function setDiscusionSectionsFunctionalities(){
        setClickEventForCommentSectionButtons();
        setClickEventForDeleteCommentButtons();
        setSubmitEventForAddCommentForm();
        setClickEventForDeleteDiscusionButtons();
        setEventForLikeCommentLinks();
    }
    function setSubmitEventForAddCommentForm(){
        let insertCommentControls = [];
        $(".comment-content").each(function(){
            insertCommentControls.push({
                "element" : $(this),
                "regex" : messageRegEx,
                "messageSpan": $(this).next()
            });
        });
        addBlurEventToControls(insertCommentControls);
        $(".form-insert-comment").each(function(){
            $(this).submit(function(e){
                e.preventDefault();
                commentContent = $(this).find(".comment-
content");
                if(formValuesAcceptable([{
                    "element" : commentContent,
                    "regex" : messageRegEx,
                    "messageSpan": $(this).next()
                }])){
                    let data;
                    if(document.getElementById("users-
discusions") != undefined){
                        data = {
                            "content"  : commentContent.val(),
                            "idDiscusion" : $(this).data('id'),
                            "idUser" : $("#users-
firstlastname").data("id")
                        }
                    }else {
                        data = {
                            "content"  : commentContent.val(),
                            "idDiscusion" : $(this).data('id'),
                            "idUser" : $("#users-
firstlastname").data("id"),
                            "idCategory" : $("#category-name")
.data("id")
```

```
                        }
                    }
                    $.ajax({
                        "url" : "/comments/store",
                        "method" : "POST",
                        "dataType" : "json",
                        "data": data,
                        "success" : function(res){
                           showModalAndLoadDiscusions("Uspešno 
ste dodali komentar.", res) ;
                        }
                    });
                    // send the message to the controller

                }
            });
        });
    }
    function showModalAndLoadDiscusions(
        message,
        res
    ){
        let html ;
        if(res.success){
            html = "<h3 class='roboto-slab-bold'>" + message+ 
"</h3>"  ;
            if(res.discusions.length > 0){
                loadDiscusions(res);
            }else {
                html = "<h3 class='roboto-slab-bold'>Trenutno ne 
postoje objave</h3>"  ;
                $("#users-discusions").html(html);
            }
        }else if(res.error){
            html = "<h3 class='roboto-slab-bold'>Došlo je do 
greške na serveru.</h3>"
        }
        modal.set(html);
        modal.show();
    }
    function setClickEventForDeleteCommentButtons(){
        $(".delete-comment-form").each(function(){
            $(this).on("submit", function(e){
                e.preventDefault();
                // alert($(this).data('id'));
                let data;
                if(document.getElementById("users-discusions")
!= undefined){
                    data = {
                        "idComment" : $(this).data("id"),
                        "idUser" : $("#users-
firstlastname").data("id")
                    }
                }else {
                    data = {
                        "idComment" : $(this).data("id"),
                        "idUser" : $("#users-
firstlastname").data("id"),
                        "idCategory" : $("#category-name")
.data("id")
                    }
                }
                $.ajax({
                    "url" : "/comments/delete",
                    "dataType" : "json",
                    "method" : "DELETE",
                    "data" : data,
                    "success" : function(res){
                        showModalAndLoadDiscusions("Uspešno ste
obrisali komentar", res);
                    }
                })
            });
        })
    }
    function setClickEventForDeleteDiscusionButtons(){
        $(".delete-discusion-link").each(function(){
            $(this)  .on("click", function(e){
                e.preventDefault();
                // alert($(this).data("id"));
                let data;
                if(document.getElementById("users-discusions")
!= undefined){
                    data = {
                        "idDiscusion" : $(this).data('id'),
                        "idUser" : $("#users-
firstlastname").data("id")
                    }
                }else {
                    data = {
                        "idDiscusion" : $(this).data('id'),
                        "idUser" : $("#users-
firstlastname").data("id"),
                        "idCategory" : $("#category-name")
.data("id")
                    }
                }
                $.ajax({
                    "url" : "/discusions/delete",
                    "method" : "DELETE",
                    "data" : data,
                    "success" : function(res){
                        showModalAndLoadDiscusions("Uspešno ste
obrisali raspravu.", res);
                    }
                });
            });
        });
    }
    function setClickEventForCommentSectionButtons(){
        $(".comment-section").css("display", "none");
        $(".showhide-comment-section").each(function(){
            $(this).click(function(e) {
                e.preventDefault();
                let commentSection =
$(this).parent().parent().find(".comment-section");
                let commentSectionDisplay =
$(commentSection).css("display");
                let btn = $(this);
                $(commentSection).slideToggle("slow",
function(){
                    //
console.log($(commentSection).css("display"));
                    if(commentSectionDisplay == "block"){
                        btn .text("Prikaži");
                    }else {
                        btn .text("Sakrij");
                    }
                });

            });
        });
    }
    function loadDiscusions(res){
        let html = "";
        for(let d of res.discusions){
            html += `
<section class="w-100  mb-3 p-2 discusions">
```

```
    <section class="d-flex mb-3  mx-0 row">
        <section class="row col-12 col-md-6">
            <figure class="col-12 col-md-2 "><img
src="/storage/images/${d.src}"  class="profile-pic-small img-
fluid img-thumbnail" alt="profile picture"/></figure>
            <article class="col-12 col-md-10"><a
href="${siteRoot + "user/show/" + d.idUser}" class="roboto-slab-
bold ">${d.firstname} ${d.lastname}</a></article>\`;
        </section>`;

            if($("#users-firstlastname").data("id") ==
$("#users-firstlastname").data("sess")) {
                html += `
            <section class="row col-12 col-md-6 justify-content-
end">
                <article class="col-12 col-md-2">
                    <a href="${siteRoot + "discusions/edit/" +
d.idDiscusion}" class="btn btn-primary" >Izmeni</a>
                </article>
                <article class="col-12 col-md-2"><a href="#"
class="btn btn-danger delete-discusion-link" data-
id="${d.idDiscusion}">Obriši</a></article>
            </section>

           `;

            }

            html += ` </section>
    <section class="row justify-content-start">
        <article class="   mr-2">
            <p class="roboto-slab-bold mb-0">${d.title}</p>
        </article>
        <article class=" ">
            <p class="roboto-slab-bold mb-0">(${d.date})</p>
        </article>
        <article class="col-12 col-md-2 my-2 my-md-0">
            <span class="roboto-slab-bold
category">${d.category}</span>
        </article>
    </section>
        <article class="my-3 ">
            <p class="roboto-slab text-justify">${d.content}</p>
        </article>
                `;
            if(d.comments){
                html += `
    <article>
        <p class="roboto-slab-bold title-1">Komentari
(${d.comments.length})<a href="#" class="showhide-comment-
section ml-3 btn btn-primary" >Prikaži</a></p>
        <section class="comment-section">
            <section >
                <form action="${siteRoot +
"comments/store"}"data-id="${d.idDiscusion}" method="POST"
name="formInsertComment" class="form-insert-comment">
                    <section class="row">
                        <article class="col-12 mb-3">
                            <textarea rows="3" cols="20"
class="comment-content form-control" placeholder="Unesite
komentar"></textarea>
<span class="form-element-message error-message"
id="message">Poruka mora početi velikim slovom, najmanje 2 a
najviše 255 znaka.</span>
                        </article>
                        <article class="col-12 mb-3">
                            <input type="submit" value="Potvrdi"
class="btn btn-primary" />
                        </article>
                    </section>
                </form>
            </section>
                `;
                for(let c of d.comments){
                    // console.log(c);
                    html += getCommentHTML(c);
                }
                html += `

        </section>
    </article>
                `;
            }
            html += `

        </section>
            `;
        }
        if(document.getElementById("users-discusions")
!=undefined){
            $("#users-discusions").html(html);
        }else {
            $("#all-discusions") .html(html);
        }
        // add submit event
        setDiscusionSectionsFunctionalities();
        // setClickEventForCommentSectionButtons();
        // setClickEventForDeleteCommentButtons();
        // setSubmitEventForAddCommentForm();
    }
    function getCommentHTML(comment){
        let html;
        html = `
<section class="comment mb-3">
    <section class="row ">
        <figure class="col-12 col-md-1"><img
src="/storage/images/${comment.profilePic}"   class="profile-
pic-small img-fluid img-thumbnail" alt="profile
picture"/></figure>
        <article class="col-11">
            <p   class="roboto-slab-bold"><a href="${siteRoot +
"user/show/"  + comment.idUser}">${comment.firstname}
${comment.lastname}</a> (${comment.date})</p>
        </article>
    </section>
    <article>
        <p class="roboto-slab">${comment.content}</p>
    </article>
    <article class="like-comment-link-wrapper">
        <span class="roboto-slab-bold">Sviđanja <span
class="like-count">${comment.likes}</span></span>
        <a href="#" class="btn btn-link like-comment" data-
id="${comment.idComment}">Sviđa mi se</a>
    </article>
           `;
        if($("#users-firstlastname").data("sess") == $("#users-
firstlastname").data("id") || comment.idUser == $("#users-
firstlastname").data("sess")){
            html += `

        <section class=" row my-3">
            <article class="col-12 col-md-2 mb-3">
                <form class="delete-comment-form" name=""
action="#" data-id="${comment.idComment}" method="POST" >
                    <input type="submit" value="Obriši komentar"
class="btn btn-danger" />
                </form>
            </article>
            <article class="col-12 col-md-2"><a href="${siteRoot
+ "comments/edit/" + comment.idComment}" class="btn btn-
primary">Izmeni komentar</a></article>
        </section>

            `;
        }
        html += `
    </section>
           `;
        return  html;
    }
    if(document.getElementsByClassName("comment-section").length
> 0){

        setDiscusionSectionsFunctionalities();
    }
    if(document.editProfileForm){
        const editProfileControls = [
            {
                "element" : $("#firstnameEditProfile"),
                "regex" : firstLastNameRegex,
                "messageSpan" : $("#firstname")
            },
            {
                "element" : $("#lastnameEditProfile"),
                "regex" : firstLastNameRegex,
                "messageSpan" : $("#lastname")
            },
            {
                "element" : $("#emailEditProfile"),
                "regex" : emailRegEx,
                "messageSpan" : $("#email")
            },
            {
                "element" : $("#dateEditProfile"),
                "regex" : dateRegEx,
                "messageSpan" : $("#date")
            },
            {
                "element" : $("#passwordEditProfile"),
                "regex" : passwordRegEx,
                "messageSpan" : $("#password")
            },
            {
                "element" : $("#passwordNewEditProfile"),
                "regex" : passwordRegEx,
                "messageSpan" : $("#passwordNew")
            }
        ];
        addBlurEventToControls(editProfileControls);
        document.editProfileForm.onsubmit = function(e){
            e.preventDefault();
            // setUpAjaxHeader();
            let data = {};

if($("#firstnameEditProfile").val().match(firstLastNameRegex) &&
$("#firstnameEditProfile").val()){
                data.firstname =
$("#firstnameEditProfile").val();
            }

if($("#lastnameEditProfile").val().match(firstLastNameRegex) &&
$("#lastnameEditProfile").val()){
                data.lastname = $("#lastnameEditProfile").val();
            }
            if($("#emailEditProfile").val().match(emailRegEx) &&
$("#emailEditProfile").val()){
                data.email = $("#emailEditProfile").val();
            }

if($("#passwordNewEditProfile").val().match(passwordRegEx) &&
$("#passwordEditProfile").val().match(passwordRegEx) &&
$("#passwordNewEditProfile").val() &&
$("#passwordEditProfile").val()){
                data.passwordOld =
$("#passwordEditProfile").val();
                data.password =
$("#passwordNewEditProfile").val();
            }
            if($("#dateEditProfile").val()){
                data.birthDate = $("#dateEditProfile").val();
            }
            $.ajax({
                "url" : "/user/updateprofile",
                "method" : "PATCH",
                "data" : data,
                "success" : function(res){
                    if(res.success){
                        let html = `<h3 class="roboto-slab-
bold">Uspešno ste izmenili podatke.</h3>`;
                        modal.set(html);
                        modal.show();
                        // location.href= siteRoot + "user";
                    }else {
                        let html = `<h3 class="roboto-slab-
bold">Niste izmenili podatke.</h3>`;
                        modal.set(html);
                        modal.show();
                    }
                }
            });
        };
    }
    if(document.formNewDescusion != undefined){
        const newDiscuionControls = [{
                "element"   : $("#subjectNewDiscusion"),
                "regex" : subjectRegEx,
                "messageSpan" : $("#subject")
            },
            {
                "element" : $("#messageNewDiscusion"),
                "regex" : messageRegEx,
                "messageSpan" : $("#message")
            },
            {
                "element" : $("#categoryNewDiscusion"),
                "regex" : categoryRegEx,
                "messageSpan" : $("#category")
            }
        ];
        addBlurEventToControls(newDiscuionControls);
        $("#formNewDiscusion").on("submit", function(e) {
            e.preventDefault();
            if (formValuesAcceptable(newDiscuionControls)) {
                $.post(
                    "/discusions/store",
                    {
                        "subject":
$("#subjectNewDiscusion").val(),
                        "message":
$("#messageNewDiscusion").val(),
                        "categoryId":
$("#categoryNewDiscusion").val()
                    },
                    function (res) {
                        let html;
                        if(res.success){
                            html = `<h3 class="roboto-slab-
bold">Uspešno ste dodali raspravu.</h3>` ;
                            clearForm([
                                $("#subjectNewDiscusion"),
                                $("#messageNewDiscusion"),
                                $("#categoryNewDiscusion")
                            ]);
                            $("#categoryNewDiscusion").val("-
1");
                        }else if(res.error){
                            html = `<h3 class="roboto-slab-
bold">Došlo je do greške prilikom obrade zahteva.</h3>` ;
                        }else if(res.errorSever){
                            html = `<h3 class="roboto-slab-
bold">Došlo je do greške na serveru.</h3>` ;
                        }
                        modal.set(html);
                        modal.show();
                    });
            }
        });
    }
if(document.updateCommentForm != undefined){
    const updateCommentControls = [
        {
            "element" : $("#messageUpdateComment"),
            "regex" : messageRegEx,
            "messageSpan" : $("#message")
        }
    ];
    addBlurEventToControls(updateCommentControls);

document.getElementById("updateCommentForm").addEventListener("s
ubmit", function(e){
        e.preventDefault();
        if(formValuesAcceptable(updateCommentControls)){
            $.ajax({
                "url" : "/comments/update",
                "dataType" : "application/json",
                "method" : "PUT",
                "data" : {
                    "content" :
$("#messageUpdateComment").val(),
                    "id" : $("#messageUpdateComment").data("id")
                },
                "statusCode": {
                    200: function(res){
                        json = JSON.parse(res.responseText);
                        displayResponseMessage(json, "Uspešno
ste izmenili komentar.");
                    }
                }
            });
        }
    });
}
function displayResponseMessage(res, message){
    let html = "";
    if(res.success) {
        html = "<h3 class='roboto-slab-bold'>" + message +
"</h3>"
    }else if(res.nothingUpdated){
        html = "<h3 class='roboto-slab-bold'>Niste ništa
izmenili.</h3>"
    }
    else if(res.error) {
        html = "<h3 class='roboto-slab-bold'>Došlo je do greške
na serveru.</h3>"
    }
    modal.set(html);
    modal.show();
}
    if(document.updateDiscusionForm != undefined) {
        const updateDiscsusionControls = [
            {
                "element": $("#subjectUpdateDiscusion"),
                "regex": subjectRegEx,
                "messageSpan": $("#subject")
            },
            {
                "element": $("#messageUpdateDiscusion"),
                "regex": messageRegEx,
                "messageSpan": $("#message")
            },
            {
                "element": $("#categoryUpdateDiscusion"),
                "regex": categoryRegEx,
                "messageSpan": $("#category")
            },
        ];
        addBlurEventToControls(updateDiscsusionControls);
        document.updateDiscusionForm.onsubmit = function (e) {
            e.preventDefault();
            if (formValuesAcceptable(updateDiscsusionControls))
{
                $.ajax({
                    "url": "/discusions/update",
                    "dataType": "json",
                    "method": "PUT",
                    "data": {
                        "subject":
$("#subjectUpdateDiscusion").val(),
                        "content":
$("#messageUpdateDiscusion").val(),
                        "category":
$("#categoryUpdateDiscusion").val(),
                        "id":
$("#subjectUpdateDiscusion").data("id")
                    },
                    "success": function (res) {
                        displayResponseMessage(res, "Uspešno ste
izmenili raspravu.");
                    }
                });
            }
        }
    }
```