

Dokumentacija

Web programiranje PHP 2, sajt 2 (Laravel)

Mladen Bošković 37/16

<https://mb95-just-write.000webhostapp.com/>

Napomena:

Veliki broj funkcionalnosti sajta ne radi zbog ograničenja hostinga (brisanje komentara/objava/korisnika/kategorija, menjanje lozinke, resetovanje lozinke, slanje email-a pri registraciji/resetovanju lozinke/kontaktiranju administratora)

Domen aplikacije i funkcionalnosti:

Aplikacija pokušava da reši domen blog-a. Zamišljeno je da korisnici aplikacije mogu da postavljaju objave na određenu temu (kategoriju), komentarišu i lajkuju objave.

Svi posetnici sajta imaju mogućnost da pregledaju sve objave, sve komentare, vrše pretragu po naslovu, filtriraju po kategoriji, sortiraju po naslovu ili datumu postavljanja objave i kontaktiraju administratora sajta.

Prijavljeni korisnici (korisnici koji su se registrovali, aktivirali nalog i na kraju prijavili) kao dodatak imaju mogućnost da dodaju/menjaju/brišu svoje objave, dodaju/brišu svoje komentare, lajkuju sve objave i kasnije ako žele da obrišu lajk, menjaju svoju lozinku i sliku.

Administrator sajta ima mogućnosti kao i prijavljen korisnik, uz dodatak da može da dodaje/menja/briše sve objave/korisnike/kategorije, dodaje/briše sve komentare, vidi sve aktivnosti korisnika uz soriranje po datumu.

Straničenje je omogućeno na stranici sa svim objavama, na 'profil' stranici kao i na svim 'admin' stranicama koje prikazuju sve korisnike/komentare/objave/kategorije/aktivnosti

Ukoliko je korisnik sajta zaboravio lozinku ima mogućnost da je resetuje.

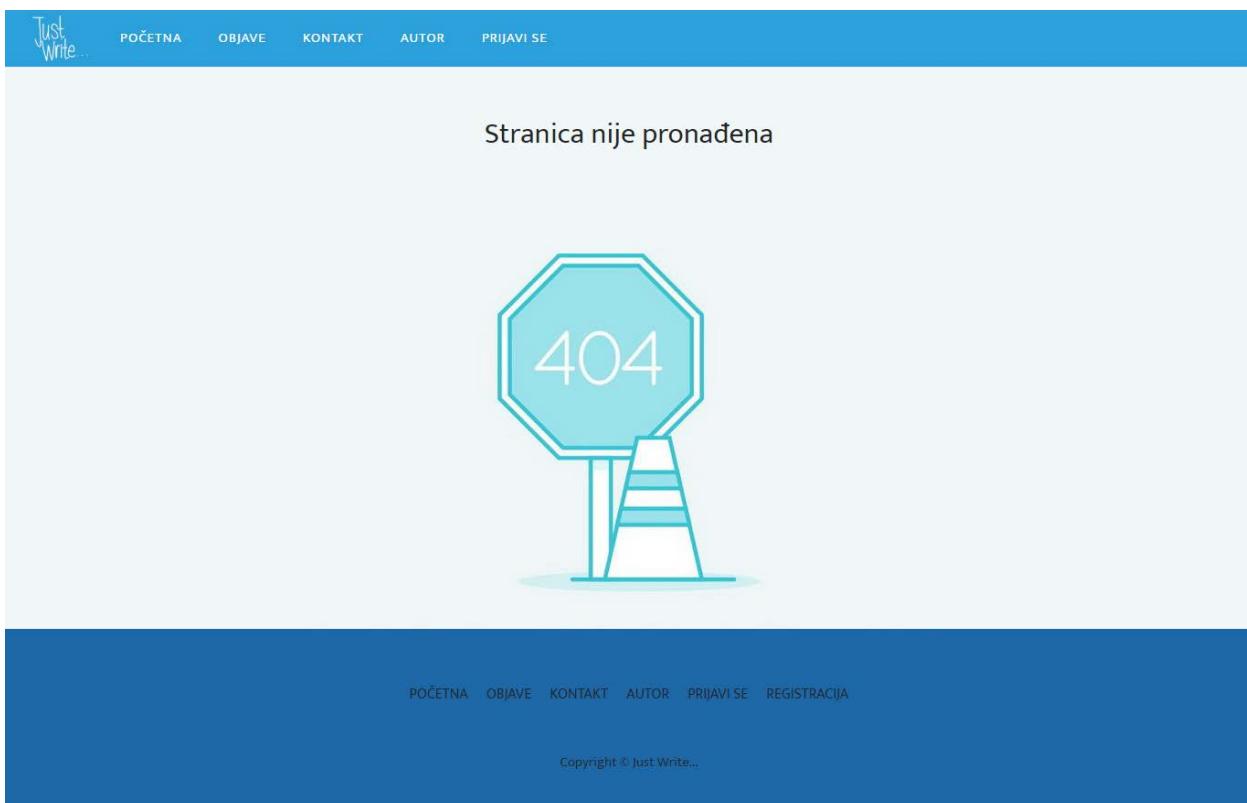
Sadržaj

- 1. Slike stranica sa funkcionalnostima**
- 2. Views**
- 3. Controllers**
- 4. Models**
- 5. Middleware**
- 6. Requests**
- 7. Routes**
- 8. CSS**
- 9. JavaScript**
- 10. Baza podataka**

1. Slike stranica sa funkcionalnostima

1.1) 404.blade.php

Ova stranica se prikazuje u slučaju nepostojeće stranice



1.2) addobjavauser.blade.php

Ova stranica služi prijavljenim korisnicima ili administratoru za dodavanje objave

The screenshot shows the 'Dodajte Objavu' (Add Post) page. At the top, there is a navigation bar with links: POČETNA, OBJAVE, KONTAKT, AUTOR, and PROFIL. The logo 'Just Write...' is located on the left side of the header. The main content area has a title 'Dodajte Objavu'. Below it are two large input fields: 'Naslov' (Title) and 'Tekst' (Text). Underneath these fields are two smaller input fields: 'Kategorija' (Category) with a dropdown menu showing 'Izaberite' (Select) and 'Slika' (Image) with a file upload button labeled 'Choose File' and a message 'No file chosen'. At the bottom of the form is a grey button labeled 'DODAJ OBJAVU' (Add Post). The footer of the page contains links: POČETNA, OBJAVE, KONTAKT, AUTOR, PROFIL, DODAJ OBJAVU, and ODJAVI SE. It also includes a copyright notice: 'Copyright © Just Write...'. The overall design is clean and modern, using a light blue header and a white background for the form.

1.3) author.blade.php

Ova stranica prikazuje informacije o autoru sajta



O Autoru



Zovem se Mladen Bošković. Rođen sam u Kruševcu 1995. godine, gde sam završio osnovnu i srednju školu. Studiram na Visokoj ICT školi, smer Internet tehnologije.

Dokumentacija

POČETNA OBJAVE KONTAKT AUTOR PRIJAVI SE REGISTRACIJA

Copyright © Just Write...

1.4) contact.blade.php

Ova stranica je namenjena za kontaktiranje administratora sajta

The screenshot shows a web page with a blue header bar containing the logo "Just Write..." and navigation links: POČETNA, OBJAVE, KONTAKT, AUTOR, and PRIJAVI SE. Below the header is a white contact form area. The form includes fields for "Email" (a text input box) and "Poruka" (a large text area). At the bottom of the form is a grey button labeled "POŠALJI". Below the form is a dark blue footer bar containing the same navigation links as the header: POČETNA, OBJAVE, KONTAKT, AUTOR, PRIJAVI SE, and REGISTRACIJA. At the very bottom center of the footer is the copyright notice "Copyright © Just Write...".

Just Write...

POČETNA OBJAVE KONTAKT AUTOR PRIJAVI SE

Email

Poruka

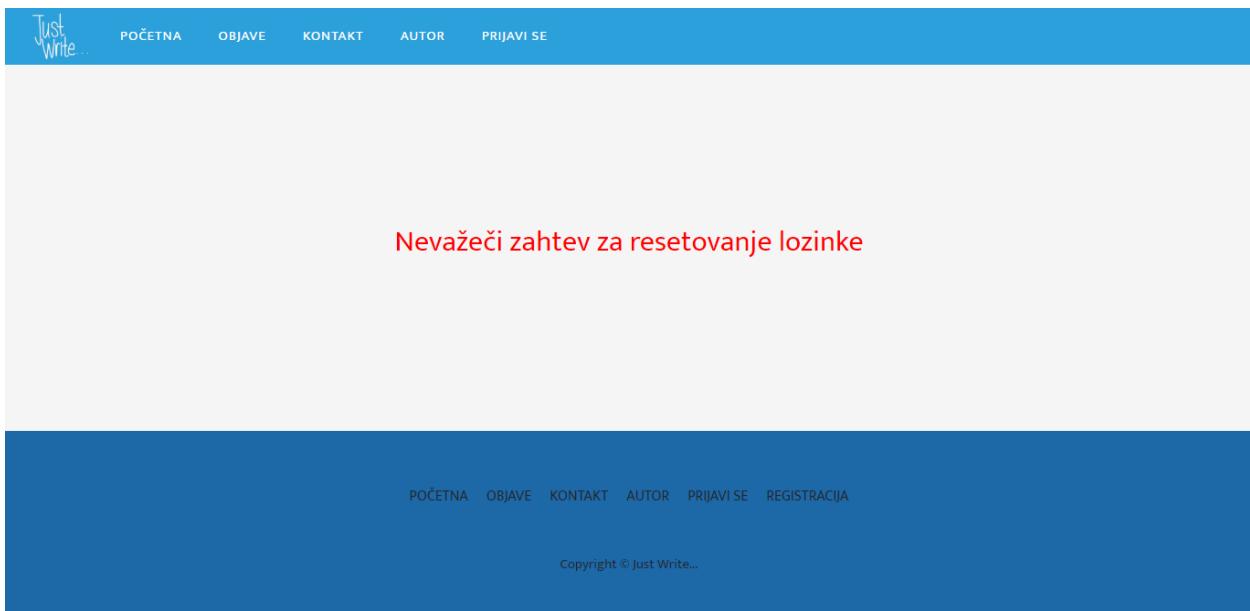
POŠALJI

POČETNA OBJAVE KONTAKT AUTOR PRIJAVI SE REGISTRACIJA

Copyright © Just Write...

1.5) error.blade.php

Ova stranica se korisniku prikazuje samo u određenim situacijama (kao primer će biti nevažeći zahtev za resetovanje lozinke)



1.6) home.blade.php

Ova je početna stranica i prikazuje najnoviju objavu i 4 objave sa najviše lajkova

POZIVNA
DOKA
DOKA
DOKA
DOKA

Zadnja objava

Stekla uključujući, među kojima i vrapljati je, Aliqum osnove licenčne mase, eu-
rosoje.



Dokladačkih
Ponuditelj: 031-333-10-00
Građevništvo

Najnovije ponude za izgradnju građevina. Građevni projekti i sigurnost.

PONUDI SVE

Najpopularnije objave

Cake aliquid health et odio, ut congrue enim mollis sit enim, integer rhombus tunc in laetare
etiam.



Dokladačkih
Ponuditelj: 031-333-10-00
Kolagen je bio

Novi kolagen je bio učinak, novi kolagen je bio učinak, novi kolagen je bio učinak.

PONUDI SVE

Stekla uključujući, među kojima i vrapljati je, Aliqum osnove licenčne mase, eu-
rosoje.



Dokladačkih
Ponuditelj: 031-333-10-00
Građevništvo

Najnovije ponude za izgradnju građevina. Građevni projekti i sigurnost.

PONUDI SVE

Champions League, interstum, muriši vrone, još jedno, kultura, politički parci, veliki nogomet.



Dokladačkih
Ponuditelj: 031-333-10-00
Kolagen je bio

Novi kolagen je bio učinak, novi kolagen je bio učinak, novi kolagen je bio učinak.

PONUDI SVE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam imperdiet ac ante id rutrum.



Dokladačkih
Ponuditelj: 031-333-10-00
Kolagen je bio

Novi kolagen je bio učinak, novi kolagen je bio učinak, novi kolagen je bio učinak.

PONUDI SVE

1.7) jednaobjava.blade.php

Ova stranica prikazuje konkretnu objavu sa više detalja i komentarima. Kada se korisnik prijavi ima mogućnost da dodaje komentar i lajkuje objavu



Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare luctus massa, eu euismo



Dodao/la:
admin

Postavljeno:
03.13.2019 18:09h

Zadnja izmena:
03.13.2019 18:09h

Kategorija:
Muzika



Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, magna tortor feugiat arcu, id placerat dolor magna eget lacus. Praesent feugiat, risus non dignissim porta, libero lacus feugiat odio, a dapibus odio leo vel ante. Morbi vel metus quis mauris finibus suscipit vel in tortor. Ut quis elit id eros fringilla tincidunt. Praesent condimentum nunc nisl, at luctus sem sollicitudin nec. Morbi interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat purus libero in mi. Vestibulum est magna, bibendum eget ultricies sit amet, feugiat vitae nulla. Phasellus pharetra non mauris nec venenatis. Praesent a diam efficitur, eleifend lorem non, dictum orci. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris bibendum ipsum malesuada, maximus elit sit amet, ornare lacus. Donec vitae est velit. Donec iaculis imperdiet justo sed pulvinar.

Komentari (1)



Dodao/la: charlie

Postavljeno: 03.13.2019 18:13h

Komentar

Morate se prijaviti da bi komentarisali objavu. [Prijavite se](#)



1.8) login.blade.php

Ova stranica je namenjena za prijavljivanje korisnika

The screenshot shows a web page titled "Prijavite se" (Log in). At the top, there is a navigation bar with links: "POČETNA", "OBJAVE", "KONTAKT", "AUTOR", and "PRIJAVI SE". The main content area contains fields for "Korisničko ime" (Username) and "Lozinka" (Password), both represented by input boxes. Below these fields is a grey button labeled "PRIJAVI SE" (Log in). Underneath the button, there are two small links: "Niste registrovani?" (Not registered?) and "Zaboravili ste lozinku?" (Forgot password?). At the bottom of the page, there is a dark blue footer bar containing links: "POČETNA", "OBJAVE", "KONTAKT", "AUTOR", "PRIJAVI SE", and "REGISTRACIJA". The footer also includes the copyright notice "Copyright © Just Write...".

Prijavite se

Korisničko ime

Lozinka

PRIJAVI SE

Niste registrovani?
Zaboravili ste lozinku?

POČETNA OBJAVE KONTAKT AUTOR PRIJAVI SE REGISTRACIJA

Copyright © Just Write...

1.9) objave.blade.php

Ova stranica prikazuje sve objave (3 po stranici). Korisnik ima mogućnost da pretražuje, filtrira i sortira sadržaj

Sve objave

Filtriranje i sortiranje

Sortiraj :

Kategorija :
 Muzika
 Sport
 Putovanja
 Kulinarijsci
 Kulturni događaji

PRETRAJI

1 2

Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare fuctus massa, eu eusmo

Dodata/la admin Postavljeno: 13.3.2019. Kategorija: Muzika

Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, ..

[PROČITAJ VIŠE](#)

Champions League interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat



Dodata/la messi10 Postavljeno: 13.3.2019. Kategorija: Sport

Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, ..

[PROČITAJ VIŠE](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam imperdiet ac ante id rutrum



Dodata/la admin Postavljeno: 13.3.2019. Kategorija: Putovanja

Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, ..

[PROČITAJ VIŠE](#)

1 2

POČETNA | OBJAVE | KONTAKT | AUTOR | PROMOZE | REGISTRACIJA

Copyright © Jast Web.

1.10) register.blade.php

Ova stranica je namenjena za registraciju korisnika

The screenshot shows a registration form titled "Registrujte se". The form fields are as follows:

- Ime (First Name): Text input field
- Prezime (Last Name): Text input field
- Email: Text input field
- Korisničko ime (Username): Text input field
- Lozinka (Password): Text input field
- Ponovite lozinku (Repeat Password): Text input field
- Slika (Image): File upload input field with placeholder "Choose File No file chosen".
- REGISTRUJ SE (Register): A large grey button.

At the bottom of the page, there is a blue footer bar containing the following links: POČETNA, OBJAVE, KONTAKT, AUTOR, PRIJAVA SE, and REGISTRACIJA. The footer also includes the copyright notice: Copyright © Just Write... .

1.11) resetpassword.blade.php

Ova stranica je namenjena za resetovanje lozinke i može joj se pristupiti samo ako korisnik zahteva resetovanje lozinke i isprati sve korake

The screenshot shows a web page titled "Resetujte Lozinku". At the top, there is a navigation bar with links: "POČETNA", "OBJAVE", "KONTAKT", "AUTOR", and "PRIJAVI SE". On the left side of the main content area, there is a logo for "Just Write...". The main form contains two input fields: "Nova lozinka" and "Ponovite novu lozinku", both with placeholder text. Below the inputs is a button labeled "PROMENI LOZNUKU". At the bottom of the page, there is a footer with links: "POČETNA", "OBJAVE", "KONTAKT", "AUTOR", "PRIJAVI SE", and "REGISTRACIJA". The footer also includes a copyright notice: "Copyright © Just Write...".

1.12) updateobjavauser.blade.php

Ova stranica omogućava korisniku ili administratoru da izmeni objavu

Just Write...

POČETNA OBJAVE KONTAKT AUTOR PROFIL ADMIN PANEL

Izmenite Objavu



Naslov
Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare luctus massa, eu euismo

Tekst
Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, magna tortor feugiat arcu, id placerat dolor magna eget lacus. Praesent feugiat, risus non dignissim porta, libero lacus feugiat odio, a dapibus odio leo vel ante. Morbi vel metus quis mauris finibus suscipit vel in tortor. Ut quis elit id eros fringilla tincidunt. Praesent condimentum nunc nisi, at luctus sem sollicitudin nec. Morbi interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat purus libero in mi. Vestibulum est magna, bibendum eget ultricies sit amet, feugiat vitae nulla. Phasellus pharetra non mauris nec venenatis. Praesent a diam efficitur, eleifend lorem non, dictum orci. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris bibendum ipsum malesuada, maximus elit sit amet, ornare lacus. Donec vitae est velit.

Kategorija
Muzika ▾

Slika
Choose File No file chosen

IZMENI OBJAVU

POČETNA OBJAVE KONTAKT AUTOR PROFIL DODAJ OBJAVU ODJAVI SE

Copyright © Just Write...

1.13) userprofile.blade.php

Ova stranica prikazuje informacije o korisniku kao i objave koje je postavio. Omogućava menjanje slike i lozinke i brisanje objave

POČETNA OBJAVE KONTAKT AUTOR PROFIL ADMIN PANEL

Vaš Profil

Vaša slika:



Choose File No file chosen PROMENI SLIKU

Trenutna lozinka:

Nova lozinka:

Ponovite novu lozinku:

PROMENI LOZINKU

Podaci

Ime	Mladen
Prezime	Bošković
Korisničko ime	admin
Uloga	Admin
Datum registracije	12.03.2019 23:39h

Predaj objavu

Vaše objave

RB	SLIKA	NASLOV	TEKST	DATUM POSTAVLJANJA	DATUM POSLEDNJE IZMENE	KATEGORIJA	IZMENI	OBRISI
1		<p>Lorem ipsum dolor sit amet, adipisci et Nam imperdunt ac ante id rubrum</p>	<p>Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, magna tortor feugiat arcu, id placerat dolor magna eget lacus. Praesent feugiat, risus non dignissim porta, libero lacus feugiat odio, a dapibus odio leo vel ante. Morbi vel metus quis mauris finibus suscipit vel in tortor. Ut quis elit id eros fringilla tincidunt. Praesent condimentum nunc nisi, at luctus sem sollicitudin nec. Morbi interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat purus libero in mi. Vestibulum est magna, bibendum eget ultricies sit amet, feugiat vitae nulla. Phasellus pharetra non mauris nec venenatis. Praesent a diam efficitur, eleifend lorem non, dictum orci. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris bibendum ipsum malesuada, maximus elit sit amet, ornare lacus. Donec vitae est velit. Donec iaculis imperdunt justo sed pulvinar.</p>	12.03.2019 23:46h	12.03.2019 23:46h	Putovanja	<button>Uzmjeni</button>	<button>Obrisi</button>
2		<p>Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare luctus massa, eu euismod</p>	<p>Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, magna tortor feugiat arcu, id placerat dolor magna eget lacus. Praesent feugiat, risus non dignissim porta, libero lacus feugiat odio, a dapibus odio leo vel ante. Morbi vel metus quis mauris finibus suscipit vel in tortor. Ut quis elit id eros fringilla tincidunt. Praesent condimentum nunc nisi, at luctus sem sollicitudin nec. Morbi interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat purus libero in mi. Vestibulum est magna, bibendum eget ultricies sit amet, feugiat vitae nulla. Phasellus pharetra non mauris nec venenatis. Praesent a diam efficitur, eleifend lorem non, dictum orci. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris bibendum ipsum malesuada, maximus elit sit amet, ornare lacus. Donec vitae est velit. Donec iaculis imperdunt justo sed pulvinar.</p>	13.03.2019 18:09h	13.03.2019 18:09h	Muzika	<button>Uzmjeni</button>	<button>Obrisi</button>

POČETNA OBJAVE KONTAKT AUTOR PROFIL DODAJ/OBRISAVI OTKAVISITE

Copyright © Just website.

ADMIN STRANIČE:

1.14) adminhome.blade.php

Ovo je početna stranica za administratora

The screenshot shows the 'Admin Meni' (Admin Menu) page. At the top, there is a navigation bar with links: POČETNA, OBJAVE, KONTAKT, AUTOR, PROFIL, and ADMIN PANEL. On the right side of the header, there is a logo for 'Just Write...'.

The main content area is titled 'Admin Meni'. It contains several groups of buttons for managing different parts of the system:

- Objave (Posts):** SVE OBJAVE (All Posts) and DODAJ OBJAVU (Add Post).
- Korisnici (Users):** SVI KORISNICI (All Users) and DODAJ KORISNIKA (Add User).
- Kategorije (Categories):** SVE KATEGORIJE (All Categories) and DODAJ KATEGORIJU (Add Category).
- Komentari (Comments):** SVI KOMENTARI (All Comments).
- Aktivnosti (Activities):** SVE AKTIVNOSTI (All Activities).

At the bottom of the page, there is a footer with links: POČETNA, OBJAVE, KONTAKT, AUTOR, PROFIL, DODAJ OBJAVU, and ODJAVI SE. The footer also includes the copyright notice: Copyright © Just Write... .

1.15) objava - index.blade.php

Ova stranica prikazuje sve objave i daje mogućnost brisanja objava

1.16) objava - create.blade.php

Ova stranica služi za dodavanje objava

The screenshot shows the 'Objava' creation page ('create.blade.php') within the 'Just Write...' application's admin panel. At the top, there is a navigation bar with links for 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', and 'ADMIN PANEL'. Below the navigation bar, the title 'Admin Meni' is displayed. A row of buttons provides quick access to various administrative functions: 'SVE OBJAVE', 'DODAJ OBJAVU', 'SVI KORISNICI', 'DODAJ KORISNIKA', 'SVE KATEGORIJE', 'DODAJ KATEGORIJU', and 'SVI KOMENTARI'. A separate button labeled 'SVE AKTIVNOSTI' is also present. The main content area is titled 'Dodajte Objavu' and contains fields for 'Naslov' (Title) and 'Tekst' (Text). There is a dropdown menu for 'Kategorija' (Category) with the placeholder 'Izaberite' (Select), and a file upload field for 'Slika' (Image) with the label 'Choose File' and the message 'No file chosen'. A large 'DODAJ OBJAVU' button is located at the bottom of the form. The footer of the page includes links for 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE', along with a copyright notice: 'Copyright © Just Write...'. The overall design is clean and modern, using a dark color scheme for the admin interface.

1.17) objava - edit.blade.php

Ova stranica služi za menjanje konkretne objave

Just Write

POČETNA OBJAVE KONTAKT AUTOR PROFIL ADMIN PANEL

Admin Meni

SVE OBJAVE DODAJ OBJAVU SVI KORISNICI DODAJ KORISNIKA SVE KATEGORIJE DODAJ KATEGORIJU SVI KOMENTARI

ŠVE AKTIVNOSTI

Izmenite Objavu



Naslov

Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare luctus massa, eu euismo

Tekst

Nam aliquam tellus vel lobortis pretium. Cras sodales, neque a aliquet egestas, magna tortor feugiat arcu, id placerrat dolor magna eget lacus. Praesent feugiat, risus non dignissim porta, libero lacus feugiat odio, a dapibus odio leo vel ante. Morbi vel metus quis mauris finibus suscipit vel in tortor. Ut quis elit id eros fringilla tincidunt. Praesent condimentum nunc nisi, at luctus sem sollicitudin nec. Morbi interdum, mauris vel ornare sollicitudin, risus turpis facilius, purus, vel feugiat purus libero in mi. Vestibulum est magna, bibendum eget ultricies sit amet, feugiat vitae nulla. Phasellus pharetra non mauris nec venenatis. Praesent a diam efficitur, eleifend lorem non, dictum orci. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris bibendum ipsum malesuada, maximus elit sit amet, ornare lacus. Donec vitae est velit. Donec iaculis imperdiet justo sed pulvinar.

Kategorija

Muzika

Slika

Choose File No file chosen

IZMENI OBJAVU

POČETNA OBJAVE KONTAKT AUTOR PROFIL DODAJ OBJAVU ODJAVI SE

Copyright © Just Write...

1.18) korisnik - index.blade.php

Ova stranica prikazuje sve korisnike i daje mogućnost brisanja korisnika

Just Write...

POČETNA OBJAVE KONTAKT AUTOR PROFIL ADMIN PANEL

Admin Meni

SVE OBJAVE DODAJ OBJAVU SVI KORISNICI DODAJ KORISNIKA SVE KATEGORIJE DODAJ KATEGORIJU SVI KOMENTARI

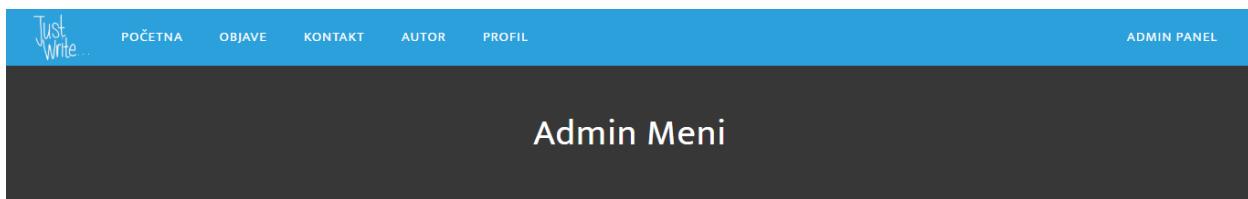
SVE AKTIVNOSTI

Svi Korisnici

RB	ID	SLIKA	IME	PREZIME	EMAIL	KORISNIČKO IME	ULOGA	AKTIVAN	DATUM REGISTRACIJE	IZMENI	OBRISI
1	72		Mladen	Bošković	mladen@justwrite.com	admin	Admin	Da	12.03.2019 23:39h	<button>Izmeni</button>	<button>Obrisi</button>
2	73		Charlie	Sheen	charlie@gmail.com	charlie	Korisnik	Da	13.03.2019 00:04h	<button>Izmeni</button>	<button>Obrisi</button>
3	74		Bart	Simpson	bart@gmail.com	bartsimpson	Korisnik	Da	13.03.2019 00:13h	<button>Izmeni</button>	<button>Obrisi</button>
4	75		Paja	Patak	paja@gmail.com	pajapatak	Korisnik	Da	13.03.2019 00:15h	<button>Izmeni</button>	<button>Obrisi</button>
5	76		Lionel	Messi	messi@gmail.com	messi10	Korisnik	Da	13.03.2019 00:21h	<button>Izmeni</button>	<button>Obrisi</button>

1.19) korisnik - create.blade.php

Ova stranica služi za dodavanje korisnika

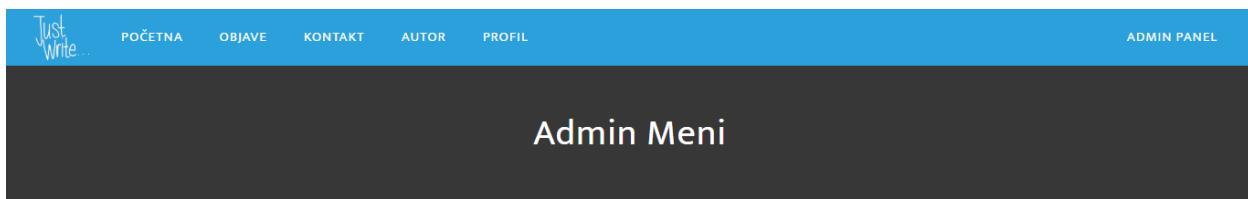


Dodajte Korisnika

A screenshot of the 'Dodajte Korisnika' form. The form consists of several input fields and dropdown menus. The fields are arranged in pairs: 'Ime' and 'Prezime' (First Name and Last Name), 'Email' and 'Korisničko ime' (Email and Username), 'Lozinka' (Password) and 'Ponovite lozinku' (Repeat Password), 'Slika' (Image) with a 'Choose File' button and a placeholder 'No file chosen', 'Uloga' (Role) with a dropdown menu set to 'Izaberite', 'Aktivan' (Active) with a dropdown menu set to 'Izaberite', and a large grey button labeled 'DODAJ KORISNIKA' at the bottom right. At the very bottom of the page, there is a dark blue footer bar with the 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE' links, along with the 'Copyright © Just Write...' text.

1.20) korisnik - edit.blade.php

Ova stranica služi za menjanje konkretnog korisnika

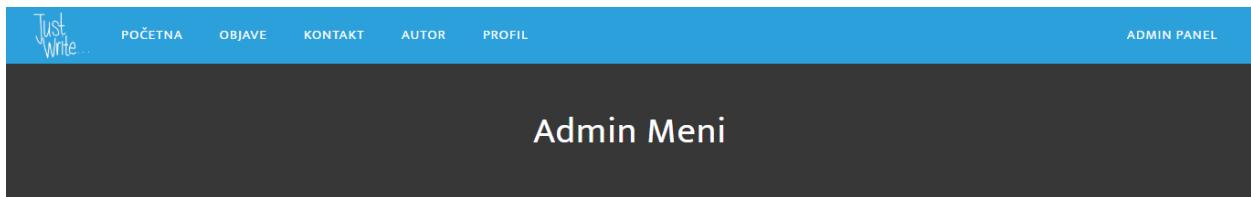


Izmenite Korisnika

A screenshot of the 'Izmenite Korisnika' form. It displays a profile picture of a man with short brown hair. Below the picture are several input fields: 'Ime' (Mladen), 'Prezime' (Bošković), 'Email' (mladen@justwrite.com), 'Korisničko ime' (admin), 'Slika' (with a 'Choose File' button and 'No file chosen'), 'Uloga' (Admin), 'Aktivan' (Da), and a large grey button labeled 'IZMENI KORISNIKA'. At the bottom of the page, there is a blue footer bar with the 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE' links, along with the 'Copyright © Just Write...' text.

1.21) kategorija - index.blade.php

Ova stranica prikazuje sve kategorije i daje mogućnost brisanja kategorija



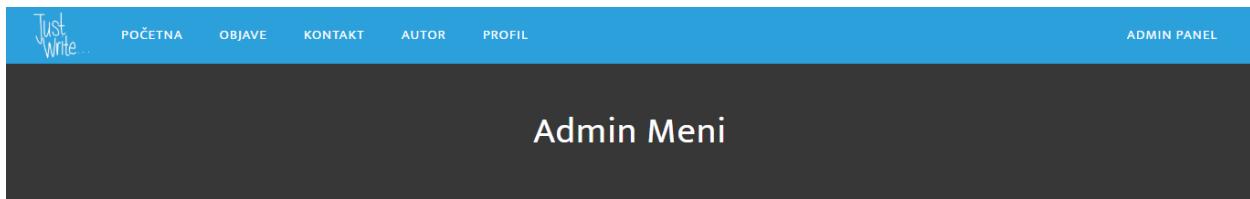
Sve Kategorije

RB	ID	NAZIV	IZMENI	OBRIŠI
1	16	Muzika	Izmeni	Obriši
2	17	Sport	Izmeni	Obriši
3	18	Putovanja	Izmeni	Obriši
4	19	Kućni ljubimci	Izmeni	Obriši
5	20	Kulinarstvo	Izmeni	Obriši



1.22) kategorija - create.blade.php

Ova stranica služi za dodavanje kategorija



Dodajte Kategoriju

A screenshot of a form for adding a category. The form is contained within a light gray rectangular area. It has a single input field labeled 'Naziv' (Name) with a placeholder text box below it. Below the input field is a large, light gray button with the text 'DODAJ KATEGORIJU' (Add Category) centered on it. At the very bottom of the page, there is a dark blue footer bar with white text containing the links 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE'. Below this footer, in the center of the page, is the copyright notice 'Copyright © Just Write...'.

Naziv

DODAJ KATEGORIJU

POČETNA OBJAVE KONTAKT AUTOR PROFIL DODAJ OBJAVU ODJAVI SE

Copyright © Just Write...

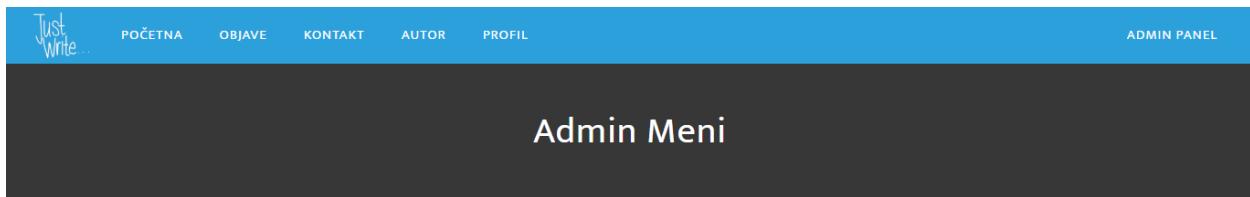
1.23) kategorija - edit.blade.php

Ova stranica služi za menjanje konkretne kategorije

The screenshot shows the 'Admin Meni' page with a navigation bar at the top. The navigation bar includes links for 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', and 'ADMIN PANEL'. Below the navigation bar, the title 'Admin Meni' is displayed. A row of buttons provides quick access to various admin functions: 'SVE OBJAVE', 'DODAJ OBJAVU', 'SVI KORISNICI', 'DODAJ KORISNIKA', 'SVE KATEGORIJE', 'DODAJ KATEGORIJU', and 'SVI KOMENTARI'. A separate button labeled 'SVE AKTIVNOSTI' is also present. The main content area is titled 'Izmenite Kategoriju'. It contains a form with a single input field labeled 'Naziv' containing the value 'Muzika'. Below the input field is a grey button labeled 'IZMENI KATEGORIJU'. At the bottom of the page, a blue footer bar displays links for 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE'. The footer also includes the copyright notice 'Copyright © Just Write...'. The overall layout is clean and modern, typical of a web-based administration interface.

1.24) komentar - index.blade.php

Ova stranica prikazuje sve komentare i daje mogućnost brisanja komentara



Svi Komentari

RB	ID	TEKST	KORISNIK	OBJAVA	DATUM POSTAVLJANJA	OBRIŠI
1	41	Komentar	charlie	Music auctor ligula velit, nec scelerisque elit vulputate ut. Aliquam ornare luctus massa, eu euismo	13.03.2019 18:13h	<button>Obriši</button>
2	39	Komentar	bartsimpson	Cake aliquet hendrerit odio, ut congue enim mollis sit amet. Integer rhoncus tortor in laoreet enim	13.03.2019 17:47h	<button>Obriši</button>
3	40	Komentar	bartsimpson	Champions League interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat	13.03.2019 17:48h	<button>Obriši</button>
4	38	Komentar	pajapatak	Champions League interdum, mauris vel ornare sollicitudin, risus turpis iaculis purus, vel feugiat	13.03.2019 17:45h	<button>Obriši</button>



1.25) aktivnost - index.blade.php

Ova stranica prikazuje sve aktivnosti korisnika

The screenshot shows the 'Admin Meni' section of the application. Below it, a list of 'Sve Aktivnosti' (All Activities) is displayed. The activities are listed in a table with columns for RB (Row ID), ID, Aktivnost (Activity), and Datum i Vreme Događanja (Date and Time of Occurrence). The activities listed are:

RB	ID	AKTIVNOST	DATUM I VREME DOGAĐANJA
1	418	Korisnik se prijavio	16.03.2019 15:58:38h
2	417	Korisnik se odjavio	16.03.2019 15:58:28h
3	416	Korisnik se prijavio	16.03.2019 15:57:08h
4	415	Korisnik se odjavio	16.03.2019 15:56:59h
5	414	Korisnik je promenio lozinku	16.03.2019 15:56:49h
6	413	Korisnik se prijavio	16.03.2019 15:56:33h
7	412	Korisnik je resetovao lozinku	16.03.2019 15:56:08h
8	411	Korisnik se odjavio	16.03.2019 15:32:46h
9	410	Korisnik se prijavio	16.03.2019 15:25:19h
10	409	Korisnik se prijavio	16.03.2019 14:20:50h

At the bottom of the page, there is a footer with links to 'POČETNA', 'OBJAVE', 'KONTAKT', 'AUTOR', 'PROFIL', 'DODAJ OBJAVU', and 'ODJAVI SE'. The footer also includes the copyright notice 'Copyright © Just Write...'. Navigation buttons for the activity list are also present at the bottom.

2. Views (resources/views)

2.1) /layouts/layout.blade.php

```
<!DOCTYPE html>
<html lang="sr">
<script type="text/javascript">
    var baseUrl = "{{url('/')}}";
</script>
@include('common.head')
<body>

<div id="navigacija">
    <div id="nav">
        <div id="logo">
            <a href="{{route('home')}}"></a>
        </div>

        <ul>
            @foreach($meni as $m)
                @if(session('korisnik') and ($m->meni == "PRIJAVI SE" or $m->meni == "REGISTRACIJA"))
                    @continue
                @endif

                @if(!session('korisnik') and ($m->meni == "ODJAVI SE" or $m->meni == "PROFIL" or $m->meni == "DODAJ OBJAVU"))
                    @continue
                @endif

                @if($m->meni == "PROFIL")
                    @if($m->roditeljID == 0)
                        <li><a href="{{url('/') . $m->putanja . (session('korisnik') ? session('korisnik')->korisnikID : 0)}}">{{$m->meni}}</a>
                        @component('components.meni', [
                            'children' => $m->submenus,
                            'meni' => $meni
                        ])
                            @endcomponent
                        </li>
                    @endif
                @else
                    @if($m->roditeljID == 0)
                        <li><a href="{{url('/') . $m->putanja}}">{{$m->meni}}</a>
                        @component('components.meni', [
                            'children' => $m->submenus,
                            'meni' => $meni
                        ])
                            @endcomponent
                        </li>
                    @endif
                @endif
            @endforeach
        </ul>
    </div>
</div>
```

```
@if(session('korisnik'))
    @if(session('korisnik')->uloga == "Admin")
        <a href="{{route('adminhomepage')}}" id="adminLink">ADMIN PANEL</a>
    @endif
@endif
</div>

@yield('sadrzaj')

@include('common.footer')

</body>
</html>
```

2.2) /common/

2.2.1) head.blade.php

```
<head>
    <title>
        @yield('title')
    </title>
    <link rel="shortcut icon" href="{{asset('favicon.ico')}}" />
    <meta charset="UTF-8">
    <meta name="author" content="Mladen Bošković"/>
    <meta name="csrf-token" content="{{ csrf_token() }} ">
    <meta name="description" content="Just Write..."/>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
        <link rel="stylesheet" type="text/css"
href="{{asset('css/simpleLightbox.min.css')}}" />
        <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHrtZLI+pbxtHCWp1t77Bil4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/"
crossorigin="anonymous">
        <link rel="stylesheet" type="text/css" href="{{asset('css/style.css')}}" />
</head>
```

2.2.2) footer.blade.php

```
<div id="footer">
    <ul>
        @foreach($meni as $m)
            @if(session('korisnik') and ($m->meni == "PRIJAVI SE" or $m->meni == "REGISTRACIJA"))
                @continue
            @endif

            @if(!session('korisnik') and ($m->meni == "ODJAVI SE" or $m->meni == "PROFIL" or $m->meni == "DODAJ OBJAVU"))
                @continue
            @endif

            @if($m->meni == "PROFIL")
                <li><a href="{{url('') . $m->putanja . (session('korisnik') ? session('korisnik')->korisnikID : 0)}}}">{{$m->meni}}</a>
            @else
                <li><a href="{{url('') . $m->putanja}}">{{$m->meni}}</a>
            @endif
        @endforeach
    </ul>
    <p>Copyright © Just Write...</p>
</div>

<script src="{{asset('js/jquery.min.js')}}"></script>
<script type="text/javascript" src="{{asset('js/script.js')}}"></script>
<script type="text/javascript" src="{{asset('js/simpleLightbox.min.js')}}"></script>
```

2.3) /components/

2.3.1) jednaobjava.blade.php

```
<div class="objava_sadrzaj">
    <div class="objava_drzac">
        <table class="objava_tabela">
            <tr>
                <td colspan="3"><h5>{{ $objava->naslov }}</h5></td>
            </tr>
            <tr>
                <td colspan="3">
                    <div class="objava_slika_div">
                        alt }}" class="objava_slika_img"/>
                    </div>
                </td>
            </tr>
            <tr class="objava_detalji">
                <td>
                    Dodao/la: {{ $objava->korisnicko_ime }}
                </td>
                <td>
                    Postavljeno: {{ date('m.d.Y H:i', $objava->objava_created_at)}}h
                </td>
                <td>
                    Kategorija: {{ $objava->kategorija }}
                </td>
            </tr>
            <tr>
                <td colspan="3">
                    <div class="objava_tekst">
                        {{ substr($objava->tekst, 0, 80)}} ...
                    </div>
                </td>
            </tr>
            <tr>
                <td colspan="3">
                    <a href="{{ route('jednaObjava', ['objavaID' => $objava->objavaID, 'korisnikID' => (session('korisnik') ? session('korisnik')->korisnikID : 0)] )}}>PROČITAJ VIŠE</a>
                    @if(session('korisnik'))
                        @if(session('korisnik')->korisnikID == $objava->korisnikID)
                            &nbsp;&nbsp;&nbsp; <a
                        href="{{ route('updateobjavauserpage', ['id' => $objava->objavaID]) }}>Izmeni</a>
                            @endif
                        @endif
                    </td>
                </tr>
            </table>
        </div>
    </div>
```

2.3.2) meni.blade.php

```
@isset($children)
    @if(count($children) > 0)
        <ul>
            @foreach($children as $child)
                @if(session('korisnik') and ($child->meni == "PRIJAVI SE" or $child->meni == "REGISTRACIJA"))
                    @continue
                @endif

                @if(!session('korisnik') and ($child->meni == "ODJAVI SE" or $child->meni == "PROFIL" or $child->meni == "DODAJ OBJAVU"))
                    @continue
                @endif

                @if($child->meni == "PROFIL")
                    <li><a href="{{url('') . $child->putanja . (session('korisnik') ? session('korisnik')->korisnikID : 0)}}}">{{$child->meni}}</a>
                        @foreach($meni as $m)
                            @if($m->meniID == $child->meniID)
                                @component('components.meni', [
                                    'children' => $m->submenus,
                                    'meni' => $meni
                                ])
                                    @endcomponent
                            @endif
                        @endforeach
                    </li>
                @else
                    <li><a href="{{url('') . $child->putanja}}">{{$child->meni}}</a>
                        @foreach($meni as $m)
                            @if($m->meniID == $child->meniID)
                                @component('components.meni', [
                                    'children' => $m->submenus,
                                    'meni' => $meni
                                ])
                                    @endcomponent
                            @endif
                        @endforeach
                    </li>
                @endif
            @endforeach
        </ul>
    @endif
@endisset
```

2.4) /pages/

2.4.1) 404.blade.php

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

@section('title')
    Stranica nije pronađena
@endsection

@section('sadrzaj')
    <div id="error404">
        <p>Stranica nije pronađena</p>
        
    </div>
@endsection
```

2.4.2) addobjavauser.blade.php

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

@section('title')
    Dodavanje objave
@endsection

@section('sadrzaj')
    @include('inc.addobjavauserform')
    @include('inc.greskelevo')
@endsection
```

2.4.3) author.blade.php

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

@section('title')
    Autor
@endsection

@section('sadrzaj')
    <div class="naslov_svaki">
        <p>O Autoru</p>
    </div>
    <div id="author_sadrzaj">
        <div id="author_drzac">
            <a href="{{asset('images/autor.jpg')}}" class="vecaSlika">
                
            </a>
            <div id="author_tekst">
                Zovem se Mladen Bošković. Rođen sam u Kruševcu 1995. godine, gde sam završio osnovnu i srednju školu. Studiram na Visokoj ICT školi, smer Internet tehnologije.
            </div>
        </div>
        <div id="dokumentacija">
            <a href="{{asset('/dokumentacija.pdf')}}" target="_blank">Dokumentacija
            &nbsp;<i class="fas fa-file-pdf"></i></a>
        </div>
    </div>
@endsection
```

2.4.4) contact.blade.php

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

@section('title')
    Kontakt
@endsection

@section('sadrzaj')
    <div class="naslov_svaki">
        <p>Obratite nam se putem formulara</p>
    </div>
    <div id="contact_sadrzaj">
        <div id="contact_sadrzaj_drzac">
            <form id="forma_contact" name="forma_contact" method="POST"
            action="{{route('contactAdmin')}}" onSubmit="return contactProvera();">
                @csrf
                <table id="tabela_contact">
                    <tr>
                        <td>Email</td>
                    </tr>
                    <tr>
                        <td><input type="text" value="{{old('contactEmail')}}"></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="contactEmailGreska"
                                class="ispisGreskeContact"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Poruka</td>
                    </tr>
                    <tr>
                        <td><textarea id="poruka" name="poruka"
                            maxlength="500">{{old('poruka')}}
```

```
@if(session('uspesnaPoruka'))
    <h4>{{session('uspesnaPoruka')}}</h4>
@endif

@if(session('neuspesnaPoruka'))
    <h4>{{session('neuspesnaPoruka')}}</h4>
@endif
</div>
</div>
@endsection
```

2.4.5) error.blade.php

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

@section('title')
    Greška
@endsection

@section('sadrzaj')
    @if(session('uspesnaPoruka'))
        <div class="errors_uspesno">
            <h2>{{session('uspesnaPoruka')}}</h2>
        </div>
    @endif

    @if(session('neuspesnaPoruka'))
        <div class="errors_neuspesno">
            <h2>{{session('neuspesnaPoruka')}}</h2>
        </div>
    @endif
@endsection
```

2.4.6) home.blade.php

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

@section('title')
    Početna
@endsection

@section('sadrzaj')
    @include('inc.zadnjaobjava')
    @include('inc.najpopularnijeobjave')
@endsection
```

2.4.7) jednaobjava.blade.php

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

@section('title')
    {{$objava->naslov}}
@endsection

@if(!session('korisnik'))
    @if(Request::segment(3) != 0)
        <script type="text/javascript">
            window.location = "{{route('home')}}";
        </script>
    @endif
@endif

@if(session('korisnik'))
    @if(session('korisnik')->korisnikID != Request::segment(3))
        <script type="text/javascript">
            window.location = "{{route('home')}}";
        </script>
    @endif
@endif

@section('sadrzaj')
    @include('inc.jednaobjavaprikaz')
    @include('inc.jednaobjavakomentari')
@endsection
```

2.4.8) login.blade.php

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

@section('title')
    Prijava
@endsection

@section('sadrzaj')
    @include('inc.loginform')
    @include('inc.greske')
    @include('inc.forgotpasswordform')
@endsection
```

2.4.9) objave.blade.php

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

@section('title')
    Sve objave
@endsection

@section('sadrzaj')



<p>Sve objave</p>



<p>
        <input type="text" id="searchInput" name="searchInput" class="inputi"/>
        <input type="button" id="searchButton" name="searchButton" value="PRETRAŽI"/>
    </p>



<div id="objave_drzac">
        <div id="filter_sort">
            <h5>Filtriranje i sortiranje</h5>

            <div>
                <p>Sortiraj :</p>
                <select id="sortiraj">
                    <option value="0">Izaberite</option>
                    <option value="1">Najnovije</option>
                    <option value="2">Najstarije</option>
                    <option value="3">Naslov A-Z</option>
                    <option value="4">Naslov Z-A</option>
                </select>
            </div>

            <div>
                <p>Kategorija :</p>
                @if(count($kategorije))
                    @foreach($kategorije as $k)
                        <input type="checkbox" name="kategorija_filter"
class="kategorija_filter" value="{{$k->kategorijaID}}"/> {{ $k->kategorija }} <br/>
                    @endforeach
                @else
                    Nema kategorija
                @endif
            </div>
        </div>

        <div id="objave_objavepage">



</div>
```

```
</div>  
  
<div class="straniceAjax">  
</div>  
  
<div class="ucitavanje">  
</div>  
  
@endsection
```

2.4.10) register.blade.php

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

@section('title')
    Registracija
@endsection

@section('sadrzaj')
    @include('inc.registerform')
    @include('inc.greskelevo')
@endsection
```

2.4.11) resetpassword.blade.php

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

@section('title')
    Resetovanje lozinke
@endsection

@section('sadrzaj')
    <div class="naslov_svaki">
        <p>Resetujte Lozinku</p>
    </div>

    <div id="reset_lozине_sadrzaj">
        <div id="reset_lozине_drzac">
            <form id="formaPromenaLozinke" name="formaPromenaLozinke">
                <table id="tabela_promeni_lozinku">
                    <tr>
                        <td>Nova lozinka</td>
                    </tr>
                    <tr>
                        <td><input type="password" id="novaLozinka" name="novaLozinka" placeholder="" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="novaLozinkaGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Ponovite novu lozinku</td>
                    </tr>
                    <tr>
                        <td><input type="password" id="novaLozinka2" name="novaLozinka2" placeholder="" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="novaLozinkaGreska2" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td><input type="button" id="resetLozinkeDugme" name="resetLozinkeDugme" value="PROMENI LOZNKU" class="inputi"/></td>
                    </tr>
                </table>
                <input type="hidden" id="promenaLozinkeKorisnikToken" value="{{ session('reset_password') }}"/>
            </form>
        </div>

        <div id="resetLozinkeGreske">
        </div>
    </div>
</div>

<div class="ucitavanje">
</div>
@endsection
```

2.4.12) updateobjavauser.blade.php

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

@section('title')
    Izmena objave
@endsection

@section('sadrzaj')
    @include('inc.updateobjavauserform')
    @include('inc.greskelevo')
@endsection
```

2.4.13) userprofile.blade.php

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

@section('title')
    Profil
@endsection

@if(session('korisnik')->korisnikID != Request::segment(2))
    <script type="text/javascript">
        window.location = "{{route('home')}}";
    </script>
@endif

@section('sadrzaj')
    @include('inc.userinfopassword')
    @include('inc.userpersonaldata')
@endsection
```

2.5) /inc/

2.5.1) addobjavauserform.blade.php

```
<div class="naslov_svaki">
    <p>Dodajte Objavu</p>
</div>
<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_addObjava" name="forma_addObjava" method="POST"
action="{{ route('objava.store') }}" onSubmit="return addObjavaProvera();"
enctype="multipart/form-data">
            @csrf
            <table id="tabela_addObjava">
                <tr>
                    <td colspan="2">Naslov</td>
                </tr>
                <tr>
                    <td colspan="2"><textarea id="addObjavaNaslov"
name="addObjavaNaslov" maxlength="100">{{ old('addObjavaNaslov') }}</textarea></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <div id="addObjavaNaslovGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">Tekst</td>
                </tr>
                <tr>
                    <td colspan="2"><textarea id="addObjavaTekst"
name="addObjavaTekst" maxlength="1000">{{ old('addObjavaTekst') }}</textarea></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <div id="addObjavaTekstGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Kategorija</td>
                    <td>Slika</td>
                </tr>
                <tr>
                    <td>
                        <select id="addObjavaKategorija" name="addObjavaKategorija">
                            <option value="0">Izaberite</option>
                            @foreach($kategorije as $k)
                                <option value="{{ $k->kategorijaID }}>{{ $k-
>kategorija }}</option>
                            @endforeach
                        </select>
                    </td>
                    <td>
                        <input type="file" id="addObjavaSlika" name="addObjavaSlika"/>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div id="addObjavaKategorijaGreska" class="ispisGreske"></div>
                    </td>
```

```
        <td>
            <div id="addObjavaSlikaGreska" class="ispisGreske"></div>
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" id="addObjavaDugme" name="addObjavaDugme"
value="DODAJ OBJAVU" class="inputi_dugme"/>
        </td>
    </tr>
</table>
<input type="hidden" id="addObjavaKorisnikID" name="addObjavaKorisnikID"
value="{{session('korisnik')->korisnikID}}"/>
</form>
</div>
</div>
```

2.5.2) forgotpasswordform.blade.php

```
<div id="zab_loz_div">
    <table id="resetPassEmailTabela">
        <tr>
            <td><h4>- Resetovanje lozinke -</h4></td>
        </tr>
        <tr>
            <td>Unesite email</td>
        </tr>
        <tr>
            <td>
                <input type="text" id="resetPassEmail" name="resetPassEmail"
placeholder="" autocomplete="on" class="inputi"/>
            </td>
        </tr>
        <tr>
            <td>
                <div id="resetPassEmailGreska" class="ispisGreske"></div>
            </td>
        </tr>
        <tr>
            <td>
                <input type="button" id="resetPassEmailDugme"
name="resetPassEmailDugme" value="POTVRDI EMAIL" class="inputi"/>
            </td>
        </tr>
    </table>
</div>

<div id="resetPassEmailIspis">
</div>

<div class="ucitavanje">
</div>
```

2.5.3) greske.blade.php

```
@if($errors->any())
    <div class="greske">
        <ul>
            @foreach($errors->all() as $error)
                <li><h5>{[$error]}</h5></li>
            @endforeach
        </ul>
    </div>
@endif

@if(session('uspesnaPoruka'))
    <div class="ispis_uspesno">
        <h4>{[session('uspesnaPoruka')]}</h4>
    </div>
@endif

@if(session('neuspesnaPoruka'))
    <div class="greske">
        <h4>{[session('neuspesnaPoruka')]}</h4>
    </div>
@endif
```

2.5.4) greskelevo.blade.php

```
@if($errors->any())
    <div class="reg_greske">
        <div class="reg_greske_drzac">
            <ul>
                @foreach($errors->all() as $error)
                    <li><h5>- {{$error}}</h5></li>
                @endforeach
            </ul>
        </div>
    </div>
@endif

@if(session('uspesnaPoruka'))
    <div class="ispis_uspesno">
        <h4>{{session('uspesnaPoruka')}}</h4>
    </div>
@endif

@if(session('neuspesnaPoruka'))
    <div class="greske">
        <h4>{{session('neuspesnaPoruka')}}</h4>
    </div>
@endif
```

2.5.5) jednaobjavakomentari.blade.php

```
<div id="komentari_naslov">
    <h5>Komentari ({{$brojKomentara}})</h5>
</div>

@if(count($komentari))
    <div id="komentari_sadrzaj">
        @foreach($komentari as $k)
            @if(session('korisnik'))
                @if(session('korisnik')->korisnikID == $k->korisnikID)
                    <a href="#" data-id="{{$k->komentarID}}"
class="obrisiKomentar">Obriši komentar</a>
                @endif
            @endif
            <div class="jedankomentar">
                <div class="jedankomentar_slika">
                    alt}}" class="jedankomentar_slika_img"/>
                </div>

                <div class="jedankomentar_sadrzaj">
                    <div class="jedankomentar_header">
                        <table class="jedankomentar_header_tabela">
                            <tr>
                                <td>Dodao/la: {{$k->korisnicko_ime}}</td>
                                <td>Postavljeno: {{date('m.d.Y H:i', $k->komentar_created_at)}}</td>
                            </tr>
                        </table>
                    </div>
                <div class="jedankomentar_tekst">
                    <p>{{$k->komentar}}</p>
                </div>
            </div>
        @endforeach
    </div>
@else
    <div id="nema_komentara">
        Nema komentara
    </div>
@endif

@if(session('korisnik'))
    <div id="dodaj_komentar_naslov">
        Dodajte komentar
    </div>
    <div id="dodaj_komentar_sadrzaj">
        <div id="dodaj_komentar">
            <form>
                <div id="dk_slika_div">
                    alt}}"/>
                </div>

```

```
        <div id="dk_tekst_div">
            <textarea id="dodajKomentarTextarea" name="dodajKomentarTextarea"
maxlength="300"></textarea>
        </div>

        <div id="dk_dugme_div">
            <input type="button" id="dodajKomentarDugme"
name="dodajKomentarDugme" value="DODAJ"/>
        </div>
            <input type="hidden" id="dodajKomentarKorisnikID"
name="dodajKomentarKorisnikID" value="{{session('korisnik')->korisnikID}}"/>
            <input type="hidden" id="dodajKomentarObjavaID"
name="dodajKomentarObjavaID" value="{{\$objava->objavaID}}"/>
        </form>
    </div>
    <div id="komentar_greska">

    </div>
</div>

@else
    <div id="komentari_prijava">
        <br/>
        Morate se prijaviti da bi komentarisali objavu. <a
        href="{{route('loginpage')}}">Prijavite se</a>
    </div>
@endif
```

2.5.6) jednaobjavaprikaz.blade.php

```
<div id="jednaobjava_sadrzaj">
    <div id="jednaobjava_drzac">
        <table id="jednaobjava_tabela">
            <tr>
                <td colspan="5">
                    <h5>{{ $objava->naslov }}</h5>
                </td>
            </tr>

            <tr>
                <td colspan="5">
                    <div id="jednaobjava_slika_div">
                        <a href="{{ asset('images/objave/' . $objava->src) }}" class="vecaSlika">
                            alt }}" id="jednaobjava_slika_img"/>
                        </a>
                    </div>
                </td>
            </tr>

            <tr id="objava_header_tekst">
                <td>Dodao/la: <br/>{{ $objava->korisnicko_ime }}</td>
                <td>Postavljeno: <br/>{{ date('m.d.Y H:i', $objava->objava_created_at ) }}h</td>
                <td>Zadnja izmena: <br/>{{ date('m.d.Y H:i', $objava->objava_updated_at ) }}h</td>
                <td>Kategorija: <br/>{{ $objava->kategorija }}</td>
                @if(session('korisnik'))
                    <td>
                        @if($korisnikLajkova)
                            <a href="#" data-id="{{ $objava->objavaID }}" class="lajkovanjeLink"><i class="fas fa-thumbs-up" style="color:#7971ea"></i></a>
                        <br/>
                        @elseif($korisnikLajkova == null)
                            <a href="#" data-id="{{ $objava->objavaID }}" class="lajkovanjeLink"><i class="fas fa-thumbs-up" style="color:#707070"></i></a>
                        <br/>
                        @elseif($korisnikLajkova == "0")
                            <a href="#" data-id="{{ $objava->objavaID }}" id="lajkovanjeProvera"><i class="fas fa-thumbs-up" style="color:#707070"></i></a>
                        <br/>
                            @endif
                            {{ $brojLajkova }}
                        </td>
                    @else
                        <td>
                            <a href="#" data-id="{{ $objava->objavaID }}" id="lajkovanjeProvera"><i class="fas fa-thumbs-up" style="color:#707070"></i></a>
                        <br/>
                            {{ $brojLajkova }}
                        </td>
                    @endif
                </td>
            </tr>

            <tr>
```

```
        <td colspan="5">
            <div id="jednaobjava_tekst">
                {$objava->tekst}
            </div>
        </td>
    </tr>
    @if(session('korisnik'))
        @if(session('korisnik')->korisnikID == $objava->korisnikID)
            <tr>
                <td colspan="5">
                    <a href="#{$route('updateobjavauserpage', ['id' => $objava->objavaID])}" class="detaljnije">Izmeni Objavu</a>
                </td>
            </tr>
        @endif
    @endif
    </table>
</div>
</div>
```

2.5.7) loginform.blade.php

```
<div class="naslov_svaki">
    <p>Prijavite se</p>
</div>
<div id="login_sadrzaj">
    <div id="login_drzac">
        <form id="forma_login" name="forma_login" method="POST"
action="{{route('prijava')}}" onSubmit="return loginProvera();">
            @csrf
            <table id="tabela_login">
                <tr>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="loginKorIme"
value="{{old('loginKorIme')}}" name="loginKorIme" placeholder="" autocomplete="on"
class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="loginKorImeGreska" class="ispisGreske">
                            @if(session('loginGreskaPoruka'))
                                {{session('loginGreskaPoruka')}}
                            @endif
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>Lozinka</td>
                </tr>
                <tr>
                    <td><input type="password" id="loginLozinka" name="loginLozinka"
placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="loginLozinkaGreska" class="ispisGreske">
                            @if(session('loginGreskaPoruka'))
                                {{session('loginGreskaPoruka')}}
                            @endif
                        </div>
                    </td>
                </tr>
                <tr>
                    <td><input type="submit" id="loginDugme" name="loginDugme"
value="PRIJAVI SE" class="inputi_dugme"/></td>
                </tr>
            </table>
            <br/>
            <a href="{{route('registerpage')}}">Niste registrovani?</a><br/>
            <p id="zab_loz_p">Zaboravili ste lozinku?</p>
        </form>
    </div>
</div>
```

2.5.8) najpopularnijeobjave.blade.php

```
<div class="naslov_svaki_boja">
    <p>Najpopularnije objave</p>
</div>

@if(count($najpopularnijeObjave))
    @foreach($najpopularnijeObjave as $objava)
        @component('components.jednaobjava', ["objava" => $objava])
        @endcomponent
        <br/><br/>
        @if(!$loop->last)
            <div class="razvđavanje_objava"></div>
        @endif
        <br/><br/>
    @endforeach
@else
    <div class="naslov_svaki">
        <p>Nema objava</p>
    </div>
@endif
```

2.5.9) registerform.blade.php

```
<div class="naslov_svaki">
    <p>Registrujte se</p>
</div>
<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_register" name="forma_register" method="POST"
action="{{route('registracija')}}" onSubmit="return regProvera();"
enctype="multipart/form-data">
            @csrf
            <table id="tabela_register">
                <tr>
                    <td>Ime</td>
                    <td>Prezime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regIme" name="regIme"
value="{{old('regIme')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regPrezime" name="regPrezime"
value="{{old('regPrezime')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="regImeGreska" class="ispisGreske"></div>
                    </td>
                    <td>
                        <div id="regPrezimeGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Email</td>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regEmail" name="regEmail"
value="{{old('regEmail')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regKorIme" name="regKorIme"
value="{{old('regKorIme')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="regEmailGreska" class="ispisGreske"></div>
                    </td>
                    <td>
                        <div id="regKorImeGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Lozinka</td>
                    <td>Ponovite lozinku</td>
                </tr>
                <tr>
                    <td><input type="password" id="regLozinka" name="regLozinka"
value="{{old('regLozinka')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                    <td><input type="password" id="regLozinka2" name="regLozinka2"
value="{{old('regLozinka2')}}" placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
```

```
                <div id="regLozinkaGreska" class="ispisGreske"></div>
            </td>
            <td>
                <div id="regLozinka2Greska" class="ispisGreske"></div>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                Slika
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <input type="file" id="regSlika" name="regSlika"/>
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <div id="regSlikaGreska" class="ispisGreske"></div>
            </td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" id="regDugme" name="regDugme"
value="REGISTRUJ SE" class="inputi_dugme"/></td>
            </tr>
        </table>
    </form>
</div>
</div>
```

2.5.10) upateobajvauserform.blade.php

```
<div class="naslov_svaki">
    <p>Izmenite Objavu</p>
</div>
<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_addObjava" name="forma_addObjava" method="POST"
action="{{route('objava.update', ['id' => $objava->objavaID])}}" onSubmit="return
updateObjavaProvera()" enctype="multipart/form-data">
            @csrf
            @method('PUT')
            <table id="tabela_addObjava">
                <tr>
                    <td colspan="2" id="updateObjavaSlikaSredina">
                        <a href="{{asset('images/objave/' . $objava->src)}}>
class="vecaSlika">
                            alt}}" id="updateObjavaSlika"/>
                        </a>
                    </td>
                </tr>
                <tr>
                    <td colspan="2"><br/></td>
                </tr>
                <tr>
                    <td colspan="2">Naslov</td>
                </tr>
                <tr>
                    <td colspan="2"><textarea id="addObjavaNaslov"
name="addObjavaNaslov" maxlength="100">{{$objava->naslov}}</textarea></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <div id="addObjavaNaslovGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">Tekst</td>
                </tr>
                <tr>
                    <td colspan="2"><textarea id="addObjavaTekst"
name="addObjavaTekst" maxlength="1000">{{$objava->tekst}}</textarea></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <div id="addObjavaTekstGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Kategorija</td>
                    <td>Slika</td>
                </tr>
                <tr>
                    <td>
                        <select id="addObjavaKategorija" name="addObjavaKategorija">
                            <option value="0">Izaberite</option>
                            @foreach($kategorije as $k)
                                @if($objava->kategorijaID == $k->kategorijaID)
                                    <option value="{{ $k->kategorijaID }}>
```

```
selected>{{ $k->kategorija }}</option>
@else
    <option value="{$k->kategorijaID}">{{ $k-
>kategorija }}</option>
@endif
@endforeach
</select>
</td>
<td>
    <input type="file" id="addObjavaSlika" name="addObjavaSlika"/>
</td>
</tr>
<tr>
    <td>
        <div id="addObjavaKategorijaGreska" class="ispisGreske"></div>
    </td>
    <td>
        <div id="addObjavaSlikaGreska" class="ispisGreske"></div>
    </td>
</tr>
<tr>
    <td colspan="2">
        <input type="submit" id="updateObjavaDugme"
name="updateObjavaDugme" value="IZMENI OBJAVU" class="inputi_dugme"/>
    </td>
</tr>
</table>
</form>
</div>
</div>
```

2.5.11) userinfopassword.blade.php

```
<div class="naslov_svaki">
    <p>Vaš Profil</p>
</div>
<div id="profile_sadrzaj">
    <div id="profile_drzac">

        <div id="o_korisniku">
            <div id="user_profile_image">
                <form method="POST" action="{!! route('promenaProfilneSlike') !!}" onsubmit="return promenaProfilneSlike(); " enctype="multipart/form-data">
                    @csrf
                    @method('PUT')
                    <table>
                        <tr>
                            <td colspan="2">
                                Vaša slika:
                            </td>
                        </tr>
                        <tr>
                            <td colspan="2" id="profil_slika_centar">
                                <a href="{!! asset('images/profile/' . session('korisnik')->src) !!}" class="vecaSlika">
                                    alt !!}>
                                </a>
                            </td>
                        </tr>
                        <tr>
                            <td>
                                <input type="file" id="profilnaSlika" name="profilnaSlika" />
                            </td>
                            <td>
                                <input type="submit" id="promenaSlikeDugme" name="promenaSlikeDugme" value="PROMENI SLIKU" class="inputi_dugme" />
                            </td>
                        </tr>
                        <tr>
                            <td colspan="2">
                                <div id="promenaProfilneSlikeGreske">
                                    @if($errors->any())
                                        <ul>
                                            @foreach($errors->all() as $error)
                                                <li>{!! $error !!}</li>
                                            @endforeach
                                        </ul>
                                    @endif
                                    @if(session('promenaSlikeUspesno'))
                                        {{ session('promenaSlikeUspesno') }}
                                    @endif
                                    @if(session('promenaSlikeNeuspesno'))
                                        {{ session('promenaSlikeNeuspesno') }}
                                    @endif
                                </div>
                            </td>
                        </tr>
                </form>
            </div>
        </div>
    </div>
</div>
```

```

                </table>
                <input type="hidden" id="promenaProfilneSlikeKorisnik"
name="promenaProfilneSlikeKorisnik" value="{{session('korisnik')->korisnikID}}"/>
            </form>
        </div>

    </div>

<div id="promena_lozinke">
    <form id="formaPromenaLozinke" name="formaPromenaLozinke">
        <table id="tabela_promeni_lozinku">
            <tr>
                <td>Trenutna lozinka</td>
            </tr>
            <tr>
                <td><input type="password" id="trenutnaLozinka"
name="trenutnaLozinka" placeholder="" class="inputi"/></td>
            </tr>
            <tr>
                <td>
                    <div id="trenutnaLozinkaGreska" class="ispisGreske"></div>
                </td>
            </tr>
            <tr>
                <td>Nova lozinka</td>
            </tr>
            <tr>
                <td><input type="password" id="novaLozinka" name="novaLozinka"
placeholder="" class="inputi"/></td>
            </tr>
            <tr>
                <td>
                    <div id="novaLozinkaGreska" class="ispisGreske"></div>
                </td>
            </tr>
            <tr>
                <td>Ponovite novu lozinku</td>
            </tr>
            <tr>
                <td><input type="password" id="novaLozinka2"
name="novaLozinka2" placeholder="" class="inputi"/></td>
            </tr>
            <tr>
                <td>
                    <div id="novaLozinkaGreska2" class="ispisGreske"></div>
                </td>
            </tr>
            <tr>
                <td><input type="button" id="promenaLozinkeDugme"
name="promenaLozinkeDugme" value="PROMENI LOZNUKU" class="inputi_dugme"/></td>
            </tr>
        </table>
        <input type="hidden" id="promenaLozinkeKorisnik"
name="promenaLozinkeKorisnik" value="{{session('korisnik')->korisnikID}}"/>
    </form>

    <div id="promenaLozinkeGreske">
    </div>
</div>
</div>

```

```
<div id="user_info">
    <table>
        <tr>
            <th colspan="2"> Podaci</th>
        </tr>
        <tr>
            <td>Ime</td>
            <td>{{session('korisnik')->ime}}</td>
        </tr>
        <tr>
            <td>Prezime</td>
            <td>{{session('korisnik')->prezime}}</td>
        </tr>
        <tr>
            <td>Korisničko ime</td>
            <td>{{session('korisnik')->korisnicko_ime}}</td>
        </tr>
        <tr>
            <td>Uloga</td>
            <td>{{session('korisnik')->uloga}}</td>
        </tr>
        <tr>
            <td>Datum registracije</td>
            <td>{{date("d.m.Y H:i", session('korisnik')-
>datum_registracije)}}h</td>
        </tr>
    </table>
</div>

</div>
<div class="ucitavanje">
</div>
```

2.5.12) userpersonaldata.blade.php

```
<div class="naslov_svaki">
    <p><a href="{{route('addobjavauserpage')}}" class="detaljnije">Dodaj Objavu</a></p>
</div>
<div class="naslov_svaki_boja">
    <p>Vaše objave</p>
</div>

@if(count($objave))
    <div class="stranice">
        {{$objave->links()}}
    </div>

    <div id="allusers_sadrzaj">
        <table class="tabela">
            <tr>
                <th>RB</th>
                <th>SLIKA</th>
                <th>NASLOV</th>
                <th>TEKST</th>
                <th>DATUM POSTAVLJANJA</th>
                <th>DATUM POSLEDNJE IZMENE</th>
                <th>KATEGORIJA</th>
                <th>IZMENI</th>
                <th>OBRIŠI</th>
            </tr>
            @foreach($objave as $o)
                <tr>
                    <td>{{$loop->iteration}}</td>
                    <td>
                        alt}}" class="allUsersSlika"/>
                    </td>
                    <td>
                        <a href="{{route('jednaObjava', ['objavaID' => $o->objavaID, 'korisnikID' => (session('korisnik') ? session('korisnik')->korisnikID : 0)])}}>{{$o->naslov}}</a>
                    </td>
                    <td>{{$o->tekst}}</td>
                    <td>{{date("d.m.Y H:i", $o->objava_created_at)}}</td>
                    <td>{{date("d.m.Y H:i", $o->objava_updated_at)}}</td>
                    <td>{{$o->kategorija}}</td>
                    <td><a href="{{route('updateobjavauserpage', ['id' => $o->objavaID])}}" class="detaljnijeD">Izmeni</a></td>
                    <td><a href="#" data-id="{{$o->objavaID}}" class="obrisiObjavu">Obriši</a></td>
                </tr>
            @endforeach
        </table>
    </div>
    <div class="stranice">
        {{$objave->links()}}
    </div>
@else
    <div class="naslov_svaki">
        <p>Nemate nijednu objavu</p>
```

```
    </div>
@endif
```

2.5.13) zadnjaobjava.blade.php

```
<div class="naslov_svaki_boja">
    <p>Zadnja objava</p>
</div>

@if($zadnjaObjava)
    @component('components.jednaobjava', [ "objava" => $zadnjaObjava ])
        @endcomponent
        <br/><br/><br/><br/>
@else
    <div class="naslov_svaki">
        <p>Nema objava</p>
    </div>
@endif
```

2.6) /email/

2.6.1) contactemail.blade.php

```
<p>
    Zdravo <strong>{$name}</strong>!
</p>
<p>
    Poruka za vas od <strong>{$sender}</strong>:
</p>
<p>
    {$poruka}
</p>
```

2.6.2) register.blade.php

```
<p>
    Zdravo <strong>{{$name}}</strong>!
</p>
<p>
    Kliknite na link koji sledi kako bi aktivirali nalog: <a
    href="{{url('/')}}/activate/{{$token}}" target="_blank">Kliknite Ovde</a>
</p>
```

2.6.3) resetpassword.blade.php

```
<p>
    Zdravo!
</p>
<p>
    Kliknite na link koji sledi kako bi resetovali lozinku: <a
    href="{!! url('/') !!}/resetpassword/{{ $token }}" target="_blank">Kliknite Ovde</a>
</p>
```

2.7) /admin/

2.7.1) adminhome.blade.php

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

@section('title')
    Admin Početna
@endsection

@section('sadrzaj')
    <div id="admin_meni">
        <h1>Admin Meni</h1>
    </div>

    <div id="admin_sadrzaj">
        <div id="admin_drzac">
            <ul>
                <li>
                    <a href="{{ route('objava.index') }}">SVE OBJAVE</a>
                </li>
                <li>
                    <a href="{{ route('objava.create') }}">DODAJ OBJAVU</a>
                </li>
                <br/>
                <li>
                    <a href="{{ route('korisnik.index') }}">SVI KORISNICI</a>
                </li>
                <li>
                    <a href="{{ route('korisnik.create') }}">DODAJ KORISNIKA</a>
                </li>
                <br/>
                <li>
                    <a href="{{ route('kategorija.index') }}">SVE KATEGORIJE</a>
                </li>
                <li>
                    <a href="{{ route('kategorija.create') }}">DODAJ KATEGORIJU</a>
                </li>
                <br/>
                <li>
                    <a href="{{ route('komentar.index') }}">SVI KOMENTARI</a>
                </li>
                <br/>
                <li>
                    <a href="{{ route('aktivnost.index') }}">SVE AKTIVNOSTI</a>
                </li>
            </ul>
        </div>
    </div>
@endsection
```

2.7.2) adminmenu.blade.php

```
<div id="admin_meni">
    <h1>Admin Meni</h1>
</div>

<div id="admin_sadrzaj2">
    <div id="admin_drzac2">
        <ul>
            <li>
                <a href="{{route('objava.index')}}">SVE OBJAVE</a>
            </li>
            <li>
                <a href="{{route('objava.create')}}">DODAJ OBJAVU</a>
            </li>
            <li>
                <a href="{{route('korisnik.index')}}">SVI KORISNICI</a>
            </li>
            <li>
                <a href="{{route('korisnik.create')}}">DODAJ KORISNIKA</a>
            </li>
            <li>
                <a href="{{route('kategorija.index')}}">SVE KATEGORIJE</a>
            </li>
            <li>
                <a href="{{route('kategorija.create')}}">DODAJ KATEGORIJU</a>
            </li>
            <li>
                <a href="{{route('komentar.index')}}">SVI KOMENTARI</a>
            </li>
            <li>
                <a href="{{route('aktivnost.index')}}">SVE AKTIVNOSTI</a>
            </li>
        </ul>
    </div>
</div>
```

2.7.3) aktivnost/index.blade.php

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

@section('title')
    Admin - Sve aktivnosti
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Sve Aktivnosti</p>
    </div>

    @if(count($aktivnosti))

        <div id="aktivnost_sortDiv">
            <form method="GET" action="{{ route('aktivnost.index') }}">
                <select name="aktivnost_sort" id="aktivnost_sort">
                    <option value="desc" @if($sort == 'desc') selected
@endif>Najnovije</option>
                    <option value="asc" @if($sort == 'asc') selected
@endif>Najstarije</option>
                </select>
                <input type="submit" value="SORTIRAJ" id="aktivnost_sortDugme"/>
            </form>
        </div>

        <div class="stranice">
            {{$aktivnosti->appends(['aktivnost_sort' => $sort])->links()}}
        </div>

        <div id="allusers_sadrzaj">
            <table class="tabela">
                <tr>
                    <th>RB</th>
                    <th>ID</th>
                    <th>AKTIVNOST</th>
                    <th>DATUM I VREME DOGADANJA</th>
                </tr>
                @foreach($aktivnosti as $a)
                <tr>
                    <td>{{$loop->iteration}}</td>
                    <td>{{$a->aktivnostID}}</td>
                    <td>{{$a->aktivnost}}</td>
                    <td>
                        @if($a->aktivnost_created_at != "")
                            {{date("d.m.Y H:i:s", $a->aktivnost_created_at)}}h
                        @else
                            Nema podatka
                        @endif
                    </td>
                </tr>
            @endforeach
        </table>
    </div>

    <div class="stranice">
```

```
    {$aktivnosti->append(['aktivnost_sort' => $sort])->links()}
```

```
    </div>
@else
    <div class="trenutno_nema">
        <p>Trenutno nema aktivnosti</p>
    </div>
@endif
@endsection
```

2.7.4) komentar/index.blade.php

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

@section('title')
    Admin - Svi komentari
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Svi Komentari</p>
    </div>

    @if(count($komentari))
        <div class="stranice">
            {{$komentari->links()}}
        </div>

        <div id="allusers_sadrzaj">
            <table class="tabela">
                <tr>
                    <th>RB</th>
                    <th>ID</th>
                    <th>TEKST</th>
                    <th>KORISNIK</th>
                    <th>OBJAVA</th>
                    <th>DATUM POSTAVLJANJA</th>
                    <th>OBRIŠI</th>
                </tr>
                @foreach($komentari as $k)
                    <tr>
                        <td>{{$loop->iteration}}</td>
                        <td>{{$k->komentarID}}</td>
                        <td>{{$k->komentar}}</td>
                        <td>{{$k->korisnicko_ime}}</td>
                        <td>
                            <a href="{{$route('jednaObjava', ['objavaID' => $k->objavaID, 'korisnikID' => (session('korisnik') ? session('korisnik')->korisnikID : 0)])}}>{{$k->naslov}}</a>
                        </td>
                        <td>{{date("d.m.Y H:i", $k->komentar_created_at)}}</td>
                        <td><a href="#" data-id="{{$k->komentarID}}>Obriši</a></td>
                    </tr>
                @endforeach
            </table>
        </div>

        <div class="stranice">
            {{$kometari->links()}}
        </div>
    @else
        <div class="trenutno_nema">
            <p>Trenutno nema komentara</p>
        </div>
    @endif
@endsection
```

2.7.5) kategorija/

2.7.5.1) index.blade.php

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

@section('title')
    Admin - Sve kategorije
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Sve Kategorije</p>
    </div>

    @if(count($kategorije))
        <div id="allusers_sadrzaj">
            <table class="tabela">
                <tr>
                    <th>RB</th>
                    <th>ID</th>
                    <th>NAZIV</th>
                    <th>IZMENI</th>
                    <th>OBRIŠI</th>
                </tr>
                @foreach($kategorije as $k)
                    <tr>
                        <td>{{$loop->iteration}}</td>
                        <td>{{$k->kategorijaID}}</td>
                        <td>{{$k->kategorija}}</td>
                        <td><a href="{{$route('kategorija.edit', ['id' => $k->kategorijaID])}}>Izmjeni</a></td>
                        <td><a href="#" data-id="{{$k->kategorijaID}}>Obriši</a></td>
                    </tr>
                @endforeach
            </table>
        </div>
    @else
        <div class="trenutno_nema">
            <p>Trenutno nema kategorija</p>
        </div>
    @endif
@endsection
```

2.7.5.2) create.blade.php

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

@section('title')
    Admin - Dodavanje kategorije
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Dodajte Kategoriju</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_register" name="forma_register" method="POST"
action="{{ route('kategorija.store') }}" onSubmit="return
addUpdateKategorijaProvera();">
                @csrf
                <table id="tabela_adduser">
                    <tr>
                        <td>Naziv</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="addKategorija" name="addKategorija"
value="{{ old('addKategorija') }}" placeholder="" autocomplete="on"
class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="addKategorijaGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="submit" id="addKategorijaDugme"
name="addKategorijaDugme" value="DODAJ KATEGORIJU" class="inputi_dugme" />
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>

    @include('inc.greske')

@endsection
```

2.7.5.3) edit.blade.php

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

@section('title')
    Admin - Izmena kategorije
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Izmenite Kategoriju</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_register" name="forma_register" method="POST"
action="{{ route('kategorija.update', ['id' => $kategorija->kategorijaID]) }}"
onSubmit="return addUpdateKategorijaProvera();">
                @csrf
                @method('PUT')
                <table id="tabela_adduser">
                    <tr>
                        <td>Naziv</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="addKategorija" name="addKategorija"
value="{{ $kategorija->kategorija }}" placeholder="" autocomplete="on"
class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="addKategorijaGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="submit" id="updateKategorijaDugme"
name="updateKategorijaDugme" value="IZMENI KATEGORIJU" class="inputi_dugme"/>
                        </td>
                    </tr>
                </table>
                <input type="hidden" id="updateKategorijaID" name="updateKategorijaID"
value="{{ $kategorija->kategorijaID }}"/>
            </form>
        </div>
    </div>

    @include('inc.greske')
@endsection
```

2.7.6) korisnik/

2.7.6.1) index.blade.php

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

@section('title')
    Admin - Svi korisnici
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Svi Korisnici</p>
    </div>

    @if(count($korisnici))
        <div class="stranice">
            {{$korisnici->links()}}
        </div>

        <div id="allusers_sadrzaj">
            <table class="tabela">
                <tr>
                    <th>RB</th>
                    <th>ID</th>
                    <th>SLIKA</th>
                    <th>IME</th>
                    <th>PREZIME</th>
                    <th>EMAIL</th>
                    <th>KORISNIČKO IME</th>
                    <th>ULOGA</th>
                    <th>AKTIVAN</th>
                    <th>DATUM REGISTRACIJE</th>
                    <th>IZMENI</th>
                    <th>OBRIŠI</th>
                </tr>
                @foreach($korisnici as $k)
                    <tr>
                        <td>{{$loop->iteration}}</td>
                        <td>{{$k->korisnikID}}</td>
                        <td>
                            alt}}</img>
                        </td>
                        <td>{{$k->ime}}</td>
                        <td>{{$k->prezime}}</td>
                        <td>{{$k->email}}</td>
                        <td>{{$k->korisnicko_ime}}</td>
                        <td>{{$k->uloga}}</td>
                        <td>{{$k->aktivvan == 0 ? "Ne" : "Da"}}</td>
                        <td>{{date("d.m.Y H:i", $k->datum_registracije)}}</td>
                        <td><a href="{{route('korisnik.edit', ['id' => $k->korisnikID])}}>Izmjeni</a></td>
                        <td><a href="#" data-id="{{$k->korisnikID}}>Obriši</a></td>
                    </tr>
                @endforeach
            </table>
        </div>
    @endif
</div>
```

```
</div>

<div class="stranice">
    {$korisnici->links()}
</div>
@else
<div class="trenutno_nema">
    <p>Trenutno nema korisnika</p>
</div>
@endif
@endsection
```

2.7.6.2) create.blade.php

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

@section('title')
    Admin - Dodavanje korisnika
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Dodajte Korisnika</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_register" name="forma_register" method="POST"
                action="{{ route('korisnik.store') }}"
                onsubmit="return addUserProvera();"
                enctype="multipart/form-data">
                @csrf
                <table id="tabela_adduser">
                    <tr>
                        <td>Ime</td>
                        <td>Prezime</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="regIme" name="regIme"
                            value="{{ old('regIme') }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                        <td><input type="text" id="regPrezime" name="regPrezime"
                            value="{{ old('regPrezime') }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="regImeGreska" class="ispisGreske"></div>
                        </td>
                        <td>
                            <div id="regPrezimeGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Email</td>
                        <td>Korisničko ime</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="regEmail" name="regEmail"
                            value="{{ old('regEmail') }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                        <td><input type="text" id="regKorIme" name="regKorIme"
                            value="{{ old('regKorIme') }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="regEmailGreska" class="ispisGreske"></div>
                        </td>
                        <td>
                            <div id="regKorImeGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Lozinka</td>
                        <td>Ponovite lozinku</td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
```

```

        </tr>
        <tr>
            <td><input type="password" id="regLozinka" name="regLozinka"
value="{{old('regLozinka')}}" placeholder="" autocomplete="on" class="inputi"/></td>
            <td><input type="password" id="regLozinka2" name="regLozinka2"
value="{{old('regLozinka2')}}" placeholder="" autocomplete="on" class="inputi"/></td>
        </tr>
        <tr>
            <td>
                <div id="regLozinkaGreska" class="ispisGreske"></div>
            </td>
            <td>
                <div id="regLozinka2Greska" class="ispisGreske"></div>
            </td>
        </tr>
        <tr>
            <td>Slika</td>
            <td>Uloga</td>
        </tr>
        <tr>
            <td>
                <input type="file" id="regSlika" name="regSlika"/>
            </td>
            <td>
                <select id="add_uloga" name="add_uloga">
                    <option value="0">Izaberite</option>
                    @foreach($uloge as $u)
                        <option value="{{$u->ulogaID}}>{{$u-

                Aktivan
            </td>
        </tr>
        <tr>
            <td>
                <select id="add_aktivran" name="add_aktivran">
                    <option value="2">Izaberite</option>
                    <option value="1">Da</option>
                    <option value="0">Ne</option>
                </select>
            </td>
            <td>
                <input type="submit" id="addUserDugme" name="addUserDugme"
value="DODAJ KORISNIKA" class="inputi_dugme"/>
            </td>
        </tr>
        <tr>
            <td>
                <div id="addAktivanGreska" class="ispisGreske"></div>
            </td>

```

```
        </tr>
    </table>
</form>
</div>
</div>

@include('inc.greskelevo')

@endsection
```

2.7.6.3) edit.blade.php

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

@section('title')
    Admin - Izmena korisnika
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Izmenite Korisnika</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_register" name="forma_register" method="POST"
                action="{{ route('korisnik.update', ['id' => $korisnik->korisnikID]) }}"
                onSubmit="return updateUserProvera();"
                enctype="multipart/form-data">
                @csrf
                @method('PUT')
                <table id="tabela_adduser">
                    <tr>
                        <td colspan="2" id="updateUserSlikaSredina">
                            <a href="{{ asset('images/profile/' . $korisnik->src) }}"
                                class="vecaSlika">
                                alt }}" id="updateUserSlika"/>
                            </a>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2"><br/></td>
                    </tr>
                    <tr>
                        <td>Ime</td>
                        <td>Prezime</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="regIme" name="regIme"
                            value="{{ $korisnik->ime }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                        <td><input type="text" id="regPrezime" name="regPrezime"
                            value="{{ $korisnik->prezime }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="regImeGreska" class="ispisGreske"></div>
                        </td>
                        <td>
                            <div id="regPrezimeGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Email</td>
                        <td>Korisničko ime</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="regEmail" name="regEmail"
                            value="{{ $korisnik->email }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                        <td><input type="text" id="regKorIme" name="regKorIme"
                            value="{{ $korisnik->korisnikID }}"
                            placeholder="" autocomplete="on" class="inputi"/></td>
                    </tr>
                </table>
            </form>
        </div>
    </div>

```

```

value="{{ $korisnik->korisnicko_ime }}" placeholder="" autocomplete="on"
class="inputi"/></td>
</tr>
<tr>
    <td>
        <div id="regEmailGreska" class="ispisGreske"></div>
    </td>
    <td>
        <div id="regKorImeGreska" class="ispisGreske"></div>
    </td>
</tr>
<tr>
    <td>Slika</td>
    <td>Uloga</td>
</tr>
<tr>
    <td>
        <input type="file" id="regSlika" name="regSlika"/>
    </td>
    <td>
        <select id="add_uloga" name="add_uloga">
            <option value="0">Izaberite</option>
            @foreach($uloge as $u)
                @if($korisnik->ulogaID == $u->ulogaID)
                    <option value="{{ $u->ulogaID }}" selected>{{ $u->uloga }}</option>
                @else
                    <option value="{{ $u->ulogaID }}">{{ $u->uloga }}</option>
                @endif
            @endforeach
        </select>
    </td>
</tr>
<tr>
    <td>
        <div id="regSlikaGreska" class="ispisGreske"></div>
    </td>
    <td>
        <div id="addUlogaGreska" class="ispisGreske"></div>
    </td>
</tr>
<tr>
    <td colspan="2" style="text-align: center;">
        Aktivan
    </td>
</tr>
<tr>
    <td>
        <select id="add_aktivran" name="add_aktivran">
            <option value="2">Izaberite</option>
            @if($korisnik->aktivran == 1)
                <option value="1" selected>Da</option>
                <option value="0">Ne</option>
            @else
                <option value="1">Da</option>
                <option value="0" selected>Ne</option>
            @endif
        </select>
    </td>
    <td>
        <input type="submit" id="updateUserDugme" name="updateUserDugme" value="IZMENI KORISNIKA" class="inputi_dugme" />
    </td>
</tr>

```

```
        </td>
    </tr>
    <tr>
        <td>
            <div id="addAktivanGreska" class="ispisGreske"></div>
        </td>
    </tr>
</table>
</form>
</div>
</div>

@include('inc.greske')

@endsection
```

2.7.7) objava/

2.7.7.1) index.blade.php

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

@section('title')
    Admin - Sve objave
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Sve Objave</p>
    </div>

    @if(count($objave))
        <div class="stranice">
            {{$objave->links()}}
        </div>

        <div id="allusers_sadrzaj">
            <table class="tabela">
                <tr>
                    <th>RB</th>
                    <th>ID</th>
                    <th>SLIKA</th>
                    <th>NASLOV</th>
                    <th>TEKST</th>
                    <th>DATUM POSTAVLJANJA</th>
                    <th>DATUM POSLEDNJE IZMENE</th>
                    <th>KORISNIK</th>
                    <th>KATEGORIJA</th>
                    <th>IZMENI</th>
                    <th>OBRIŠI</th>
                </tr>
                @foreach($objave as $o)
                    <tr>
                        <td>{{$loop->iteration}}</td>
                        <td>{{$o->objavaID}}</td>
                        <td>
                             $o->objavaID, 'korisnikID' => (session('korisnik') ? session('korisnik')->korisnikID : 0)])}}>{{ $o->naslov }}</a>
                        <td>{{ $o->tekst }}</td>
                        <td>{{ date("d.m.Y H:i", $o->objava_created_at)}}h</td>
                        <td>{{ date("d.m.Y H:i", $o->objava_updated_at)}}h</td>
                        <td>{{ $o->korisnicko_ime }}</td>
                        <td>{{ $o->kategorija }}</td>
                        <td><a href="{{route('objava.edit', ['id' => $o->objavaID])}}>Izmeni</a></td>
                        <td><a href="#" data-id="{{$o->objavaID}}>Obriši</a></td>
                    </tr>
                @endforeach
            </table>
        </div>
    @endif
</div>
```

```
        @endforeach
    </table>
</div>

<div class="stranice">
    {$objave->links( ) }
</div>
@else
<div class="trenutno_nema">
    <p>Trenutno nema objava</p>
</div>
@endif
@endsection
```

2.7.7.2) create.blade.php

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

@section('title')
    Admin - Dodavanje objave
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Dodajte Objavu</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_addObjava" name="forma_addObjava" method="POST"
                action="{{ route('objava.store') }}"
                onSubmit="return addObjavaProvera();"
                enctype="multipart/form-data">
                @csrf
                <table id="tabela_addObjava">
                    <tr>
                        <td colspan="2">Naslov</td>
                    </tr>
                    <tr>
                        <td colspan="2"><textarea id="addObjavaNaslov" name="addObjavaNaslov" maxlength="100">{{ old('addObjavaNaslov') }}</textarea></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="addObjavaNaslovGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">Tekst</td>
                    </tr>
                    <tr>
                        <td colspan="2"><textarea id="addObjavaTekst" name="addObjavaTekst" maxlength="1000">{{ old('addObjavaTekst') }}</textarea></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="addObjavaTekstGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Kategorija</td>
                        <td>Slika</td>
                    </tr>
                    <tr>
                        <td>
                            <select id="addObjavaKategorija" name="addObjavaKategorija">
                                <option value="0">Izaberite</option>
                                @foreach($kategorije as $k)
                                    <option value="{{$k->kategorijaID}}>{{$k->kategorija}}</option>
                                @endforeach
                            </select>
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
```

```
        <td>
            <input type="file" id="addObjavaSlika"
name="addObjavaSlika" />
        </td>
    </tr>
    <tr>
        <td>
            <div id="addObjavaKategorijaGreska"
class="ispisGreske"></div>
        </td>
        <td>
            <div id="addObjavaSlikaGreska" class="ispisGreske"></div>
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" id="addObjavaDugme"
name="addObjavaDugme" value="DODAJ OBJAVU" class="inputi_dugme" />
        </td>
    </tr>
</table>
<input type="hidden" id="addObjavaKorisnikID"
name="addObjavaKorisnikID" value="{{session('korisnik')->korisnikID}}"/>
</form>
</div>
</div>

@include('inc.greskelevo')

@endsection
```

2.7.7.3) edit.blade.php

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

@section('title')
    Admin - Izmena objave
@endsection

@section('sadrzaj')
    @include('admin.adminmenu')

    <div class="naslov_svaki">
        <p>Izmenite Objavu</p>
    </div>
    <div id="register_sadrzaj">
        <div id="register_drzac">
            <form id="forma_addObjava" name="forma_addObjava" method="POST"
action="{{ route('objava.update', ['id' => $objava->objavaID]) }}" onsubmit="return
updateObjavaProvera(); " enctype="multipart/form-data">
                @csrf
                @method('PUT')
                <table id="tabela_addObjava">
                    <tr>
                        <td colspan="2" id="updateObjavaSlikaSredina">
                            <a href="{{ asset('images/objave/' . $objava->src) }}>
class="vecaSlika">
                                <br/></td>
                    </tr>
                    <tr>
                        <td colspan="2">Naslov</td>
                    </tr>
                    <tr>
                        <td colspan="2"><textarea id="addObjavaNaslov"
name="addObjavaNaslov" maxlength="100">{{ $objava->naslov }}</textarea></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="addObjavaNaslovGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">Tekst</td>
                    </tr>
                    <tr>
                        <td colspan="2"><textarea id="addObjavaTekst"
name="addObjavaTekst" maxlength="1000">{{ $objava->tekst }}</textarea></td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <div id="addObjavaTekstGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">Kategorija</td>
                    </tr>
                </table>
            </form>
        </div>
    </div>

```

```

                <td>Slika</td>
            </tr>
            <tr>
                <td>
                    <select id="addObjavaKategorija"
name="addObjavaKategorija">
                        <option value="0">Izaberite</option>
                        @foreach($kategorije as $k)
                            <if($objava->kategorijaID == $k->kategorijaID)>
                                <option value="{{$k->kategorijaID}}>
selected>{{$k->kategorija}}</option>
                            @else
                                <option value="{{$k->kategorijaID}}>{{$k-
>kategorija}}</option>
                            @endif
                        @endforeach
                    </select>
                </td>
                <td>
                    <input type="file" id="addObjavaSlika"
name="addObjavaSlika"/>
                </td>
            </tr>
            <tr>
                <td>
                    <div id="addObjavaKategorijaGreska"
class="ispisGreske"></div>
                </td>
                <td>
                    <div id="addObjavaSlikaGreska" class="ispisGreske"></div>
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <input type="submit" id="updateObjavaDugme"
name="updateObjavaDugme" value="IZMENI OBJAVU" class="inputi_dugme"/>
                </td>
            </tr>
        </table>
    </form>
</div>
</div>

@include('inc.greskelevo')

@endsection

```

3. Controllers (app/Http/Controllers)

3.1) ChangePasswordController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\PromenaLozinkeProveraRequest;
use App\Http\Requests\ResetLozinkeEmailProveraRequest;
use App\Http\Requests\ResetLozinkeNovaLozinkaProveraRequest;
use App\Models\Korisnik;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;
use Psy\Util\Json;

class ChangePasswordController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Korisnik();
    }

    public function resetLozinkeEmailProvera(ResetLozinkeEmailProveraRequest $request)
    {
        $this->model->email = $request->input('email');
        $this->model->reset_password = md5(sha1($request->input('email') . rand() . time() . rand()));
    }

    try{
        $korisnik = $this->model->resetLozinkeEmailUspesnaProvera();

        $to_name = $this->model->email;
        $to_email = $this->model->email;
        $data = array("token" => $this->model->reset_password);

        try{
            Mail::send('email.resetpassword', $data, function($message) use ($to_name, $to_email) {
                $message->to($to_email, $to_name)
                    ->subject('Resetovanje lozinke');
                $message->from('auditorne.php@gmail.com', 'Just Write...');

            });
            return Json::encode("Proverite email za nastavak procesa . . . . .");
        } catch (\Exception $e){
            Log::error("Greška pri slanju email-a: " . $e->getMessage());
            return Json::encode("Došlo je do greške, pokušajte kasnije");
        }
    }

    } catch (QueryException $e){
        Log::error("Greška pri setovanju tokena korisniku koji zahteva resetovanje lozinke: " . $e->getMessage());
    }
}
```

```

        return Json::encode("Došlo je do greške, pokušajte kasnije");
    }

    public function resetLozinkeTokenProvera($token)
    {
        $this->model->reset_password = $token;
        try{
            $korisnik = $this->model->resetLozinkeTokenProvera();
            if($korisnik){
                return redirect(route('resetpasswordpage'))->with('reset_password',
$token);
            } else{
                return redirect(route('errorpage'))->with('neuspesnaPoruka', 'Nevažeći
zahtev za resetovanje lozinke');
            }
        } catch (QueryException $e){
            Log::error("Greška pri proveri da li postoji korisnik sa validnim tokenom
za resetovanje lozinke: " . $e->getMessage());
            return redirect(route('errorpage'))->with('neuspesnaPoruka', 'Greška pri
resetovanju lozinke, pokušajte kasnije');
        }
    }

    public function resetPassword(ResetLozinkeNovaLozinkaProveraRequest $request)
    {
        $this->model->reset_password = md5(shal(time() . rand() . time() . rand() .
time() . rand()));
        $this->model->token = $request->input('token');
        $this->model->lozinka = $request->input('novaLozinka');

        try{
            $korisnik = $this->model->resetPassword();

            try{
                $this->modelAktivnost->aktivnost = "Korisnik je resetovao lozinku";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }
        }

        return Json::encode("Uspešno ste promenili lozinku!");
    } catch (QueryException $e){
        Log::error("Greška pri resetovanju lozinke: " . $e->getMessage());
        return Json::encode("Došlo je do greške, pokušajte kasnije");
    }
}

public function promenaLozinke(PromenaLozinkeProveraRequest $request)
{
    $this->model->korisnikID = $request->input('korisnikID');
    $this->model->lozinka = $request->input('trenutnaLozinka');

    try{
        $korisnikProvera = $this->model->promenaLozinkeProvera();
        if($korisnikProvera){
            $this->model->lozinka = $request->input('novaLozinka');
            try{
                $korisnik = $this->model->promenaLozinke();

                try{
                    $this->modelAktivnost->aktivnost = "Korisnik je promenio

```

```
lozinku";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return Json::encode("Uspešno ste promenili lozinku!");

} catch (QueryException $e){
    Log::error("Greška pri promeni lozinke: " . $e->getMessage());
    return Json::encode("Došlo je do greške, pokušajte kasnije");
}
} else{
    return Json::encode("Trenutna lozinka nije ispravna");
}
} catch (QueryException $e){
    Log::error("Greška pri pronalaženju korisnika koji menja lozinku: " . $e->getMessage());
    return Json::encode("Došlo je do greške, pokušajte kasnije");
}
}

}
```

3.2) FrontEndController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\ContactProveraRequest;
use App\Models\Aktivnost;
use App\Models\Kategorija;
use App\Models\Komentar;
use App\Models\Lajk;
use App\Models\Meni;
use App\Models\Objava;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;

class FrontEndController extends Controller
{
    protected $data;
    private $modelObjava;
    private $modelKategorija;
    private $modelKomentar;
    private $modelLajk;
    protected $modelAktivnost;
    protected $modelMeni;

    public function __construct()
    {
        $this->modelMeni = new Meni();
        try{
            $this->data['meni'] = $this->modelMeni->dohvatiSve();
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju menija " . $e->getMessage());
            return abort(500);
        }
        $this->modelObjava = new Objava();
        $this->modelKategorija = new Kategorija();
        $this->modelKomentar = new Komentar();
        $this->modelLajk = new Lajk();
        $this->modelAktivnost = new Aktivnost();
    }

    public function home()
    {
        try{
            $this->data['zadnjaObjava'] = $this->modelObjava->dohvatiZadnjuObjavu();
            $this->data['najpopularnijeObjave'] = $this->modelObjava->dohvatiNajpopularnijeObjave();
            return view('pages.home', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju zadnje objave i najpopularnijih objava: " . $e->getMessage());
            return abort(500);
        }
    }

    public function contactAdmin(ContactProveraRequest $request)
    {
```

```

$emailKorisnika = $request->input('contactEmail');
$poruka = $request->input('poruka');

$to_name = "Mladen Bošković";
$to_email = "mladenregistracije@gmail.com";
$data = array("name" => $to_name, "sender" => $emailKorisnika, "poruka" =>
$poruka);

try{
    Mail::send('email.contactadmin', $data, function($message) use ($to_name,
$to_email) {
        $message->to($to_email, $to_name)
            ->subject('Poruka za admina');
        $message->from('auditorne.php@gmail.com','Just Write...');
    });

    try{
        $this->modelAktivnost->aktivnost = "Korisnik je poslao email
administratoru";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return back()->with('uspesnaPoruka', 'Poruka uspešno poslata!');
} catch (\Exception $e){
    Log::error("Greška pri slanju email-a: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Došlo je do greške, pokušajte
kasnije");
}

public function userprofile($id)
{
    $this->modelObjava->korisnikID = $id;
    try{
        $this->data['objave'] = $this->modelObjava->dohvatiObjaveKorisnika();
        return view('pages.userprofile', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju objava korisnika: " . $e->getMessage());
        return redirect(route('home'));
    }
}

public function addobjavauser()
{
    try{
        $this->data['kategorije'] = $this->modelKategorija->dohvatiSve();
        return view('pages.addobjavauser', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju svih kategorija: " . $e->getMessage());
        return redirect(route('home'));
    }
}

public function updateobjavauser($id)
{
    $this->modelObjava->objavaID = $id;
    try{
        $objava = $this->modelObjava->dohvatiJednu();
        if (!$objava)

```

```

        return redirect(route('home'));
        $this->data['objava'] = $objava;
        $this->data['kategorije'] = $this->modelKategorija->dohvatiSve();
        return view('pages.updateobjavauser', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju jedne objave i svih kategorija: " . $e->getMessage());
        return redirect(route('home'));
    }
}

public function jednaObjava($objavaID, $korisnikID)
{
    $this->modelObjava->objavaID = $objavaID;
    $this->modelKomentar->objavaID = $objavaID;
    $this->modelLajk->objavaID = $objavaID;
    try{
        $objava = $this->modelObjava->dohvatiJednu();
        if(!$objava)
            return redirect(route('home'));
        $this->data['objava'] = $objava;
        $this->data['komentari'] = $this->modelKomentar->dohvatiKomentareObjave();
        $this->data['brojKomentara'] = $this->modelKomentar->dohvatiBrojKomentaraObjave();
        $this->data['brojLajkova'] = $this->modelLajk->brojLajkovaObjave();

        if($korisnikID){
            $this->modelLajk->korisnikID = $korisnikID;
            try{
                $this->data['korisnikLajkovao'] = $this->modelLajk->lajkProvera();
            } catch (QueryException $e){
                Log::error("Greška pri proveri da li je korisnik lajkovao objavu: " . $e->getMessage());
                return redirect(route('home'));
            }
        } else{
            $this->data['korisnikLajkovao'] = $korisnikID;
        }
        return view('pages.jednaobjava', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju jedne objave, komentara objave, broja komentara objave i broja lajkova objave: " . $e->getMessage());
        return redirect(route('home'));
    }
}

public function objave()
{
    try{
        $this->data['kategorije'] = $this->modelKategorija->dohvatiSve();
        return view('pages.objave', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju svih kategorija: " . $e->getMessage());
        return redirect(route('home'));
    }
}

public function sveobjave(Request $request)
{
    try{
        $unos = $request->input('unos');
        $kategorijeStr = $request->input('kategorijeStr');

```

```

    if($kategorijeStr){
        $kategorije = explode( ' ', $kategorijeStr);
    } else{
        $kategorije = [ ];
    }
    $sortiranje = $request->input('sortiraj');
    $objave = $this->modelObjava->dohvatiSveObjavePage($unos, $kategorije,
$sortiranje);
    return $objave;
} catch (QueryException $e){
    Log::error("Greška pri dohvatanju objava: " . $e->getMessage());
    return redirect(route('home'));
}
}

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

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

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

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

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

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

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

public function fallback()
{
    return view('pages.404', $this->data);
}
}

```

3.3) RegisterLoginController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginProveraRequest;
use App\Http\Requests\RegisterProveraRequest;
use App\Models\Korisnik;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;
use Symfony\Component\HttpFoundation\File\Exception\FileException;

class RegisterLoginController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Korisnik();
    }

    public function registracija(RegisterProveraRequest $request)
    {
        $this->model->ime = $request->input('regIme');
        $this->model->prezime = $request->input('regPrezime');
        $this->model->email = $request->input('regEmail');
        $this->model->korisnicko_ime = $request->input('regKorIme');
        $this->model->lozinka = $request->input('regLozinka');
        $this->model->token = md5(shal($request->input('regEmail')) . $request->input('regKorIme') . time());
        $this->model->aktivan = 0;
        $this->model->reset_password = md5(shal($request->input('regIme')) . $request->input('regEmail') . time() . rand());
        $this->model->datum_registracije = time();

        $slika = $request->file('regSlika');
        $nazivSlike = time() . "_" . $slika->getClientOriginalName();

        $this->model->src = $nazivSlike;
        $this->model->alt = $request->input('regIme') . " " . $request->input('regPrezime');

        try{
            $korisnik = $this->model->registracija();
            try{
                $slika->move(public_path() . '/images/profile/', $nazivSlike);

                $to_name = $this->model->ime . " " . $this->model->prezime;
                $to_email = $this->model->email;
                $data = array('name' => $to_name, "token" => $this->model->token);

                try{
                    Mail::send('email.register', $data, function($message) use
                    ($to_name, $to_email) {
                        $message->to($to_email, $to_name)
                
```

```

        ->subject('Aktivacija naloga');
        $message->from('auditorne.php@gmail.com', 'Just Write...'));
    });

    try{
        $this->modelAktivnost->aktivnost = "Korisnik se registrovao";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return back()->with("uspesnaPoruka", "Uspešno ste se registrovali!
Proverite email i aktivirajte nalog");

} catch (\Exception $e){
    Log::error("Greška pri slanju email-a: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Došlo je do greške,
obratite se administratoru kako bi Vam aktivirao nalog");
}

} catch (FileNotFoundException $e){
    Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Došlo je do greške, obratite
se administratoru kako bi Vam aktivirao nalog");
}

} catch (QueryException $e) {
    Log::error("Greška pri registraciji korisnika: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Greška pri registraciji, pokušajte
kasnije");
}
}

public function prijava(LoginProveraRequest $request)
{
    $this->model->korisnicko_ime = $request->input('loginKorIme');
    $this->model->lozinka = $request->input('loginLozinka');
    $this->model->aktivran = 1;

    try{
        $korisnik = $this->model->prijava();
        if($korisnik){
            $request->session()->put('korisnik', $korisnik);

            try{
                $this->modelAktivnost->aktivnost = "Korisnik se prijavio";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }
            if($korisnik->uloga == "Admin"){
                return redirect(route('adminhomepage'));
            } else{
                return redirect(route('home'));
            }
        } else{
            return back()->with('loginGreskaPoruka', 'Pogrešno korisničko ime ili
lozinka');
        }
    }
}

```

```

    } catch (QueryException $e){
        Log::error("Greška pri prijavi korisnika: " . $e->getMessage());
        return back()->with('neuspesnaPoruka', 'Greška pri prijavi, pokušajte
kasnije');
    }
}

public function odjava(Request $request)
{
    $request->session()->forget('korisnik');
    $request->session()->flush();

    try{
        $this->modelAktivnost->aktivnost = "Korisnik se odjavio";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return redirect(route('home'));
}

public function activate($token)
{
    $this->model->token = $token;

    try{
        $korisnik = $this->model->activateProvera();
        if($korisnik){
            if($korisnik->aktiviran == 1){
                return redirect(route('loginpage'))->with('uspesnaPoruka', 'Nalog
je već aktiviran, možete se prijaviti');
            } else{
                try{
                    $korisnikActivate = $this->model->activate();

                    try{
                        $this->modelAktivnost->aktivnost = "Korisnik je aktivirao
nalog";
                        $this->modelAktivnost->aktivnost_created_at = time();
                        $aktivnost = $this->modelAktivnost->dodajAktivnost();
                    } catch (QueryException $e){
                        Log::error("Greška pri dodavanju aktivnosti: " . $e-
>getMessage());
                    }
                }
            }
        }
    }
}
}
return redirect(route('loginpage'))->with('uspesnaPoruka',
'Uspešna aktivacija naloga! Možete se prijaviti');

} catch (QueryException $e){
    Log::error("Greška pri aktivaciji naloga: " . $e-
>getMessage());
    return redirect(route('errorpage'))->with('neuspesnaPoruka',
'Greška pri aktivaciji, pokušajte kasnije');
}
}
} else{
    return redirect(route('registerpage'))->with("neuspesnaPoruka", "Niste
registrovani");
}
}
} catch (QueryException $e){

```

```
        Log::error("Greška pri pronalaženju korisnika čiji nalog treba da se  
aktivira naloga: " . $e->getMessage());  
        return redirect(route('errorpage'))->with('neuspesnaPoruka', 'Greška pri  
aktivaciji, pokušajte kasnije');  
    }  
}
```

3.4) UserChangesController

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\PromenaProfilneSlikeProveraRequest;
use App\Models\Korisnik;
use App\Models\Lajk;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\File;
use Illuminate\Support\Facades\Log;
use Psy\Util\Json;
use Symfony\Component\HttpFoundation\File\Exception\FileException;

class UserChangesController extends FrontEndController
{
    private $modelKorisnik;
    private $modelLajk;

    public function __construct()
    {
        parent::__construct();
        $this->modelKorisnik = new Korisnik();
        $this->modelLajk = new Lajk();
    }

    public function promenaProfilneSlike(PromenaProfilneSlikeProveraRequest $request)
    {
        $this->modelKorisnik->korisnikID = $request-
>input('promenaProfilneSlikeKorisnik');

        try{
            $korisnikSlikaDelete = $this->modelKorisnik->dohvatiJednog();
            if($korisnikSlikaDelete){
                try{
                    File::delete(public_path() . '/images/profile/' .
$korisnikSlikaDelete->src);
                } catch (FileException $e){
                    Log::error("Greška pri brisanju slike: " . $e->getMessage());
                    return back()->with("promenaSlikeNeuspesno", "Došlo je do greške,
pokušajte kasnije");
                }
            } else{
                return back()->with("promenaSlikeNeuspesno", "Došlo je do greške,
pokušajte kasnije");
            }
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju jednog korisnika: " . $e->getMessage());
            return back()->with("promenaSlikeNeuspesno", "Došlo je do greške,
pokušajte kasnije");
        }

        $slika = $request->file('profilnaSlika');
        $nazivSlike = time() . "_" . $slika->getClientOriginalName();

        $this->modelKorisnik->src = $nazivSlike;

        try{
```

```

$korisnik = $this->modelKorisnik->promenaProfilneSlike();
$request->session()->get('korisnik')->src = $nazivSlike;

try{
    $slika->move(public_path() . '/images/profile/' , $nazivSlike);

    try{
        $this->modelAktivnost->aktivnost = "Korisnik je promenio sliku";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return back()->with("promenaSlikeUspesno", "Uspešno ste promenili
sliku!");
} catch (FileNotFoundException $e){
    Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
    return back()->with("promenaSlikeNeuspesno", "Došlo je do greške,
pokušajte kasnije");
}

} catch (QueryException $e){
    Log::error("Greška pri promeni profilne slike: " . $e->getMessage());
    return back()->with("promenaSlikeNeuspesno", "Greška pri promeni slike,
pokušajte kasnije");
}

public function lajkovanje(Request $request)
{
    $this->modelLajk->korisnikID = $request->input('korisnikID');
    $this->modelLajk->objavaID = $request->input('objavaID');

    try{
        $korisnikLajkova = $this->modelLajk->lajkProvera();
        if($korisnikLajkova){
            try{
                $korisnikObrisiLajk = $this->modelLajk->obrisiLajk();

                try{
                    $this->modelAktivnost->aktivnost = "Korisnik/Administrator je
uklonio lajk";
                    $this->modelAktivnost->aktivnost_created_at = time();
                    $aktivnost = $this->modelAktivnost->dodajAktivnost();
                } catch (QueryException $e){
                    Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
                }
            }

            return Json::encode('');
        } catch (QueryException $e){
            Log::error("Greška pri brisanju lajka: " . $e->getMessage());
            return Json::encode('Došlo je do greške, pokušajte kasnije');
        }
    } else{
        try{
            $korisnikDodajLajk = $this->modelLajk->dodajLajk();

            try{
                $this->modelAktivnost->aktivnost = "Korisnik/Administrator je

```

```
lajkovao objavu";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return Json::encode(' ');
}

} catch (QueryException $e){
    Log::error("Greška pri dodavanju lajka: " . $e->getMessage());
    return Json::encode('Greška pri lajkovanju, pokušajte kasnije');
}
}

} catch (QueryException $e){
    Log::error("Greška pri proveri da li je korisnik lajkovao objavu: " . $e->getMessage());
    return Json::encode('Došlo je do greške, pokušajte kasnije');
}
}
}
```

ADMIN CONTROLLERS:

3.5) /Admin/

3.5.1) AktivnostController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\FrontEndController;
use App\Models\Aktivnost;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;

class AktivnostController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Aktivnost();
    }

    public function index(Request $request)
    {
        try{
            $sort = $request->input('aktivnost_sort');
            if($sort){
                $this->data['sort'] = $sort;
            } else{
                $this->data['sort'] = 'desc';
            }
            $this->data['aktivnosti'] = $this->model->dohvatiSve($sort);

            return view('admin.aktivnost.index', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju svih aktivnosti: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }
}
```

3.5.2) KategorijaController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\FrontEndController;
use App\Http\Requests\KategorijaProveraRequest;
use App\Models\Kategorija;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;

class KategorijaController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Kategorija();
    }

    public function index()
    {
        try{
            $this->data['kategorije'] = $this->model->dohvatiSve();
            return view('admin.kategorija.index', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju svih kategorija: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }

    public function create()
    {
        return view('admin.kategorija.create', $this->data);
    }

    public function store(KategorijaProveraRequest $request)
    {
        $this->model->kategorija = $request->input('addKategorija');
        try{
            $kategorija = $this->model->dodajKategoriju();

            try{
                $this->modelAktivnost->aktivnost = "Administrator je dodao kategoriju";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }

            return back()->with("uspesnaPoruka", "Uspešno ste dodali kategoriju!");
        }catch (QueryException $e) {
            Log::error("Greška pri dodavanju kategorije: " . $e->getMessage());
            return back()->with("neuspesnaPoruka", "Greška pri dodavanju kategorije, pokušajte kasnije");
        }
    }
}
```

```

        }

    }

    public function edit($id)
    {
        $this->model->kategorijaID = $id;
        try{
            $this->data['kategorija'] = $this->model->dohvatiJednu();
            return view('admin.kategorija.edit', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju jedne kategorije: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }

    public function update(KategorijaProveraRequest $request, $id)
    {
        $this->model->kategorijaID = $id;
        $this->model->kategorija = $request->input('addKategorija');

        try{
            $kategorija = $this->model->izmeniKategoriju();

            try{
                $this->modelAktivnost->aktivnost = "Administrator je izmenio kategoriju";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }

            return back()->with("uspesnaPoruka", "Uspešno ste izmenili kategoriju!");
        } catch (QueryException $e){
            Log::error("Greška pri izmeni kategorije: " . $e->getMessage());
            return back()->with("neuspesnaPoruka", "Greška pri izmeni kategorije, pokušajte kasnije");
        }
    }

    public function destroy($id)
    {
        $this->model->kategorijaID = $id;

        try{
            $kategorija = $this->model->obrisiKategoriju();

            try{
                $this->modelAktivnost->aktivnost = "Administrator je obrisao kategoriju";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }

            return response("Kategorija uspešno obrisana!");
        } catch (QueryException $e){
            Log::error("Greška pri brisanju kategorije: " . $e->getMessage());
            return response("Greška pri brisanju kategorije, pokušajte kasnije");
        }
    }
}

```

3.5.3) KomentarController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\FrontEndController;
use App\Http\Requests\KomentarProveraRequest;
use App\Models\Komentar;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;
use Psy\Util\Json;

class KomentarController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Komentar();
        $this->middleware('korisnikNijePrijavljenProvera');
        $this->middleware('adminProvera', ['except' => ['store', 'destroy']]);
    }

    public function index()
    {
        try{
            $this->data['komentari'] = $this->model->dohvatiSve();
            return view('admin.komentar.index', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju svih komentara: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }

    public function store(KomentarProveraRequest $request)
    {
        $this->model->komentar = $request->input('komentar');
        $this->model->korisnikID = $request->input('korisnikID');
        $this->model->objavaID = $request->input('objavaID');
        $this->model->komentar_created_at = time();

        try{
            $komentar = $this->model->dodajKomentar();

            try{
                $this->modelAktivnost->aktivnost = "Korisnik/Administrator je postavio
komentar";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }
        }

        return Json::encode("Uspešno ste dodali komentar!");
    }

} catch (QueryException $e){
```

```
        Log::error("Greška pri dodavanju komentara: " . $e->getMessage());
        return Json::encode("Greška pri dodavanju komentara, pokušajte kasnije");
    }

    public function destroy($id)
    {
        $this->model->komentarID = $id;

        try{
            $komentar = $this->model->obrisiKomentar();

            try{
                $this->modelAktivnost->aktivnost = "Korisnik/Administrator je obrisao
komentar";
                $this->modelAktivnost->aktivnost_created_at = time();
                $aktivnost = $this->modelAktivnost->dodajAktivnost();
            } catch (QueryException $e){
                Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
            }

            return response("Komentar uspešno obrisan!");
        } catch (QueryException $e){
            Log::error("Greška pri brisanju komentara: " . $e->getMessage());
            return response("Greška pri brisanju komentara, pokušajte kasnije");
        }
    }
}
```

3.5.4) KorisnikController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\FrontEndController;
use App\Http\Requests\AddUserProveraRequest;
use App\Http\Requests\UpdateUserProveraRequest;
use App\Models\Korisnik;
use App\Models\Uloga;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\File;
use Symfony\Component\HttpFoundation\File\Exception\FileException;

class KorisnikController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Korisnik();
        $uloga = new Uloga();
        try{
            $this->data['uloge'] = $uloga->dohvatiSve();
            $this->data['korisnici'] = $this->model->dohvatiSve();
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju svih korisnika i uloga: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }

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

    public function create()
    {
        return view('admin.korisnik.create', $this->data);
    }

    public function store(AddUserProveraRequest $request)
    {
        $this->model->ime = $request->input('regIme');
        $this->model->prezime = $request->input('regPrezime');
        $this->model->email = $request->input('regEmail');
        $this->model->korisnicko_ime = $request->input('regKorIme');
        $this->model->lozinka = $request->input('regLozinka');
        $this->model->token = md5(sh1($request->input('regEmail')) . $request->input('regKorIme') . time());
        $this->model->aktivan = $request->input('add_aktivan');
        $this->model->reset_password = md5(sh1($request->input('regIme')) . $request->input('regEmail') . time() . rand()));
    }
}
```

```

        $this->model->datum_registracije = time();
        $this->model->ulogaID = $request->input('add_uloga');

        $slika = $request->file('regSlika');
        $nazivSlike = time() . "_" . $slika->getClientOriginalName();

        $this->model->src = $nazivSlike;
        $this->model->alt = $request->input('regIme') . " " . $request-
>input('regPrezime');

        try{
            $korisnik = $this->model->dodajKorisnika();

            try{
                $slika->move(public_path() . '/images/profile/' , $nazivSlike);

                try{
                    $this->modelAktivnost->aktivnost = "Administrator je dodao
korisnika";
                    $this->modelAktivnost->aktivnost_created_at = time();
                    $aktivnost = $this->modelAktivnost->dodajAktivnost();
                } catch (QueryException $e){
                    Log::error("Greška pri dodavanju aktivnosti: " . $e-
>getMessage());
                }
            }

            return back()->with("uspesnaPoruka", "Uspešno ste dodali korisnika!");
        } catch (FileNotFoundException $e){
            Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
            return back()->with("neuspesnaPoruka", "Korisnik je dodat, ali bez
slike");
        }

        } catch (QueryException $e) {
            Log::error("Greška pri dodavanju korisnika: " . $e->getMessage());
            return back()->with("neuspesnaPoruka", "Greška pri dodavanju korisnika,
pokušajte kasnije");
        }
    }

    public function edit($id)
    {
        $this->model->korisnikID = $id;
        try{
            $korisnik = $this->model->dohvatiJednog();
            $this->data['korisnik'] = $korisnik;
            return view('admin.korisnik.edit', $this->data);
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju jednog korisnika: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
    }

    public function update(UpdateUserProveraRequest $request, $id)
    {
        $this->model->korisnikID = $id;
        $this->model->ime = $request->input('regIme');
        $this->model->prezime = $request->input('regPrezime');
        $this->model->email = $request->input('regEmail');
        $this->model->korisnicko_ime = $request->input('regKorIme');
        $this->model->aktivan = $request->input('add_aktivan');
        $this->model->ulogaID = $request->input('add_uloga');
    }
}

```

```

    if($request->file('regSlika'))
    {
        try{
            $korisnikSlikaDelete = $this->model->dohvatiJednog();
            if($korisnikSlikaDelete){
                try{
                    File::delete(public_path() . '/images/profile/' .
$korisnikSlikaDelete->src);
                } catch (FileException $e){
                    Log::error("Greška pri brisanju slike: " . $e->getMessage());
                    return back()->with("neuspesnaPoruka", "Došlo je do greške,
pokušajte kasnije");
                }
            } else{
                return back()->with("neuspesnaPoruka", "Došlo je do greške,
pokušajte kasnije");
            }
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju jednog korisnika: " . $e-
>getMessage());
            return back()->with("neuspesnaPoruka", "Došlo je do greške, pokušajte
kasnije");
        }
    }

    $slika = $request->file('regSlika');
    $nazivSlike = time() . "_" . $slika->getClientOriginalName();
    try{
        $slika->move(public_path() . '/images/profile/', $nazivSlike);
        $this->model->src = $nazivSlike;
    } catch (FileException $e){
        Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
        return back()->with("neuspesnaPoruka", "Došlo je do greške, pokušajte
kasnije");
    }
}

try{
    $korisnik = $this->model->izmeniKorisnika();

    try{
        $this->modelAktivnost->aktivnost = "Administrator je izmenio
korisnika";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return back()->with("uspesnaPoruka", "Uspešno ste izmenili korisnika!");
} catch (QueryException $e){
    Log::error("Greška pri izmeni korisnika: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Greška pri izmeni korisnika,
pokušajte kasnije");
}

public function destroy($id)
{
    $this->model->korisnikID = $id;
}

```

```

try{
    $korisnikSlikaDelete = $this->model->dohvatiJednog();
    if($korisnikSlikaDelete){
        try{
            File::delete(public_path() . '/images/profile/' .
$korisnikSlikaDelete->src);
        } catch (FileNotFoundException $e){
            Log::error("Greška pri brisanju slike: " . $e->getMessage());
            return response("Došlo je do greške, pokušajte kasnije");
        }
    } else{
        return response("Došlo je do greške, pokušajte kasnije");
    }
} catch (QueryException $e){
    Log::error("Greška pri dohvatanju jednog korisnika: " . $e->getMessage());
    return response("Došlo je do greške, pokušajte kasnije");
}

try{
    $korisnik = $this->model->obrisiKorisnika();

    try{
        $this->modelAktivnost->aktivnost = "Administrator je obrisao
korisnika";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return response("Korisnik uspešno obrisan!");
} catch (QueryException $e){
    Log::error("Greška pri brisanju korisnika: " . $e->getMessage());
    return response("Greška pri brisanju korisnika, pokušajte kasnije");
}
}
}

```

3.5.5) ObjavaController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\FrontEndController;
use App\Http\Requests\AddObjavaProveraRequest;
use App\Http\Requests\UpdateObjavaProveraRequest;
use App\Models\Kategorija;
use App\Models\Objava;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\File;
use Symfony\Component\HttpFoundation\File\Exception\FileException;

class ObjavaController extends FrontEndController
{
    private $model;

    public function __construct()
    {
        parent::__construct();
        $this->model = new Objava();
        $kategorija = new Kategorija();
        try{
            $this->data['kategorije'] = $kategorija->dohvatiSve();
            $this->data['objave'] = $this->model->dohvatiSve();
        } catch (QueryException $e){
            Log::error("Greška pri dohvatanju svih objava i kategorija: " . $e->getMessage());
            return redirect(route('adminhomepage'));
        }
        $this->middleware('korisnikNijePrijavljenProvera');
        $this->middleware('adminProvera', ['except' => ['store', 'update', 'destroy']]);
    }

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

    public function create()
    {
        return view('admin.objava.create', $this->data);
    }

    public function store(AddObjavaProveraRequest $request)
    {
        $this->model->korisnikID = $request->input('addObjavaKorisnikID');
        $this->model->naslov = $request->input('addObjavaNaslov');
        $this->model->tekst = $request->input('addObjavaTekst');
        $this->model->kategorijaID = $request->input('addObjavaKategorija');
        $this->model->objava_created_at = time();
        $this->model->objava_updated_at = time();

        $slika = $request->file('addObjavaSlika');
```

```

$nazivSlike = time() . "_" . $slika->getClientOriginalName();

$this->model->src = $nazivSlike;
$this->model->alt = $request->input('addObjavaNaslov');

try{
    $objava = $this->model->dodajObjavu();

    try{
        $slika->move(public_path() . '/images/objave/' , $nazivSlike);

        try{
            $this->modelAktivnost->aktivnost = "Korisnik/Administrator je
dodao objavu";
            $this->modelAktivnost->aktivnost_created_at = time();
            $aktivnost = $this->modelAktivnost->dodajAktivnost();
        } catch (QueryException $e){
            Log::error("Greška pri dodavanju aktivnosti: " . $e-
>getMessage());
        }

        return back()->with("uspesnaPoruka", "Uspešno ste dodali objavu!");
    } catch (FileNotFoundException $e){
        Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
        return back()->with("neuspesnaPoruka", "Objava je dodata, ali bez
slike");
    }
} catch (QueryException $e) {
    Log::error("Greška pri dodavanju objave: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Greška pri dodavanju objave,
pokušajte kasnije");
}

public function edit($id)
{
    $this->model->objavaID = $id;
    try{
        $objava = $this->model->dohvatiJednu();
        $this->data['objava'] = $objava;
        return view('admin.objava.edit', $this->data);
    } catch (QueryException $e){
        Log::error("Greška pri dohvatanju jedne objave: " . $e->getMessage());
        return redirect(route('adminhomepage'));
    }
}

public function update(UpdateObjavaProveraRequest $request, $id)
{
    $this->model->objavaID = $id;
    $this->model->naslov = $request->input('addObjavaNaslov');
    $this->model->tekst = $request->input('addObjavaTekst');
    $this->model->kategorijaID = $request->input('addObjavaKategorija');
    $this->model->objava_updated_at = time();

    if($request->file('addObjavaSlika'))
    {
        try{
            $objavaSlikaDelete = $this->model->dohvatiJednu();
            if($objavaSlikaDelete){
                try{
                    File::delete(public_path() . '/images/objave/' .
$objavaSlikaDelete->src);

```

```

        } catch (FileNotFoundException $e){
            Log::error("Greška pri brisanju slike: " . $e->getMessage());
            return back()->with("neuspesnaPoruka", "Došlo je do greške,
pokušajte kasnije");
        }
    } else{
        return back()->with("neuspesnaPoruka", "Došlo je do greške,
pokušajte kasnije");
    }
} catch (QueryException $e){
    Log::error("Greška pri dohvatanju jedne objave: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Došlo je do greške, pokušajte
kasnije");
}

$slika = $request->file('addObjavaSlika');
$nazivSlike = time() . "_" . $slika->getClientOriginalName();
try{
    $slika->move(public_path() . '/images/objave/', $nazivSlike);
    $this->model->src = $nazivSlike;
} catch (FileNotFoundException $e){
    Log::error("Greška pri prebacivanju slike: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Došlo je do greške, pokušajte
kasnije");
}
}

try{
    $objava = $this->model->izmeniObjavu();

    try{
        $this->modelAktivnost->aktivnost = "Korisnik/Administrator je izmenio
objavu";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return back()->with("uspesnaPoruka", "Uspešno ste izmenili objavu!");
}

} catch (QueryException $e){
    Log::error("Greška pri izmeni objave: " . $e->getMessage());
    return back()->with("neuspesnaPoruka", "Greška pri izmeni objave,
pokušajte kasnije");
}
}

public function destroy($id)
{
    $this->model->objavaID = $id;

    try{
        $objavaSlikaDelete = $this->model->dohvatiJednu();
        if($objavaSlikaDelete){
            try{
                File::delete(public_path() . '/images/objave/' .
$objavaSlikaDelete->src);
            } catch (FileNotFoundException $e){
                Log::error("Greška pri brisanju slike: " . $e->getMessage());
                return response("Došlo je do greške, pokušajte kasnije");
            }
        } else{

```

```
        return response("Došlo je do greške, pokušajte kasnije");
    }
} catch (QueryException $e){
    Log::error("Greška pri dohvatanju jedne objave: " . $e->getMessage());
    return response("Došlo je do greške, pokušajte kasnije");
}

try{
    $objava = $this->model->obrisiObjavu();

    try{
        $this->modelAktivnost->aktivnost = "Korisnik/Administrator je obrisao
objavu";
        $this->modelAktivnost->aktivnost_created_at = time();
        $aktivnost = $this->modelAktivnost->dodajAktivnost();
    } catch (QueryException $e){
        Log::error("Greška pri dodavanju aktivnosti: " . $e->getMessage());
    }

    return response("Objava uspešno obrisana!");
} catch (QueryException $e){
    Log::error("Greška pri brisanju objave: " . $e->getMessage());
    return response("Greška pri brisanju objave, pokušajte kasnije");
}
}
```

4. Models (app/Models)

4.1) Aktivnost

```
<?php

namespace App\Models;

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

class Aktivnost extends Model
{
    protected $table = "aktivnost";
    protected $primaryKey = "aktivnostID";

    public function dohvatiSve($sort)
    {
        return DB::table($this->table)->orderBy('aktivnost_created_at', $sort)->paginate(10);
    }

    public function dodajAktivnost()
    {
        return DB::table($this->table)->insert([
            "aktivnostID" => null,
            "aktivnost" => $this->aktivnost,
            "aktivnost_created_at" => $this->aktivnost_created_at
        ]);
    }
}
```

4.2) Kategorija

```
<?php

namespace App\Models;

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

class Kategorija extends Model
{
    protected $table = "kategorija";
    protected $primaryKey = "kategorijaID";

    public function dohvatiSve()
    {
        return DB::table($this->table)->get();
    }

    public function dodajKategoriju()
    {
        return DB::table($this->table)->insert([
            "kategorijaID" => null,
            "kategorija" => $this->kategorija
        ]);
    }

    public function izmeniKategoriju()
    {
        return DB::table($this->table)->where('kategorijaID', $this->kategorijaID)
            ->update([
                'kategorija' => $this->kategorija
            ]);
    }

    public function dohvatiJednu()
    {
        return DB::table($this->table)->where('kategorijaID', $this->kategorijaID)->first();
    }

    public function obrisiKategoriju()
    {
        return DB::table($this->table)->where('kategorijaID', $this->kategorijaID)->delete();
    }
}
```

4.3) Komentar

```
<?php

namespace App\Models;

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

class Komentar extends Model
{
    protected $table = "komentar";
    protected $primaryKey = "komentarID";

    public function dohvatiSve()
    {
        return DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=', 'korisnik.korisnikID')
            ->join('objava', $this->table . '.objavaID', '=', 'objava.objavaID')
            ->paginate(10);
    }

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

    public function dohvatiKomentareObjave()
    {
        return DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=', 'korisnik.korisnikID')
            ->join('objava', $this->table . '.objavaID', '=', 'objava.objavaID')
            ->where($this->table . '.objavaID', $this->objavaID)
            ->select('komentar.komentar', 'komentar.komentarID',
            'komentar.komentar_created_at',
            'korisnik.korisnicko_ime', 'korisnik.src', 'korisnik.alt',
            'korisnik.korisnikID')
            ->orderBy('komentar.komentar_created_at', 'asc')
            ->get();
    }

    public function dohvatiBrojKomentaraObjave()
    {
        return DB::table($this->table)->where('objavaID', $this->objavaID)->count();
    }

    public function dodajKomentar()
    {
        return DB::table($this->table)->insert([
            "komentarID" => null,
            "komentar" => $this->komentar,
            "korisnikID" => $this->korisnikID,
            "objavaID" => $this->objavaID,
            "komentar_created_at" => $this->komentar_created_at
        ]);
    }
}
```

4.4) Korisnik

```
<?php

namespace App\Models;

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

class Korisnik extends Model
{
    protected $table = "korisnik";
    protected $primaryKey = "korisnikID";

    public function registracija()
    {
        $this->ulogaID = DB::table('uloga')->where('uloga', 'Korisnik')->first()-
>ulogaID;

        return DB::table($this->table)->insert([
            "korisnikID" => null,
            "ime" => $this->ime,
            " prezime" => $this->prezime,
            "email" => $this->email,
            "korisnicko_ime" => $this->korisnicko_ime,
            "lozinka" => md5(shal($this->lozinka)),
            "token" => $this->token,
            "aktivran" => $this->aktivran,
            "reset_password" => $this->reset_password,
            "ulogaID" => $this->ulogaID,
            "datum_registracije" => $this->datum_registracije,
            "src" => $this->src,
            "alt" => $this->alt
        ]);
    }

    public function prijava()
    {
        return DB::table($this->table)->join('uloga', $this->table . '.ulogaID', '=',
'uloga.ulogaID')
            ->where([
                [ "korisnicko_ime", $this->korisnicko_ime ],
                [ "lozinka", md5(shal($this->lozinka)) ],
                [ "aktivran", $this->aktivran ]
            ])->first();
    }

    public function activateProvera()
    {
        return DB::table($this->table)->where('token', $this->token)->first();
    }

    public function activate()
    {
        return DB::table($this->table)->where('token', $this->token)-
>update(['aktivran' => 1]);
    }

    public function resetLozinkeEmailUspesnaProvera()
    {
```

```

        return DB::table($this->table)->where('email', $this->email)-
>update(['reset_password' => $this->reset_password]);
    }

    public function resetLozinkeTokenProvera()
    {
        return DB::table($this->table)->where('reset_password', $this-
>reset_password)->first();
    }

    public function resetPassword()
    {
        return DB::table($this->table)->where('reset_password', $this->token)
        ->update([
            'lozinka' => md5(shal($this->lozinka)),
            'reset_password' => $this->reset_password
        ]);
    }

    public function promenaProfilneSlike()
    {
        return DB::table($this->table)->where('korisnikID', $this->korisnikID)-
>update(['src' => $this->src]);
    }

    public function promenaLozinkeProvera()
    {
        return DB::table($this->table)->where([
            ['korisnikID', $this->korisnikID],
            ['lozinka', md5(shal($this->lozinka))]
        ])->first();
    }

    public function promenaLozinke()
    {
        return DB::table($this->table)->where('korisnikID', $this->korisnikID)-
>update(['lozinka' => md5(shal($this->lozinka))]);
    }

    public function dodajKorisnika()
    {
        return DB::table($this->table)->insert([
            "korisnikID" => null,
            "ime" => $this->ime,
            "prezime" => $this->prezime,
            "email" => $this->email,
            "korisnicko_ime" => $this->korisnicko_ime,
            "lozinka" => md5(shal($this->lozinka)),
            "token" => $this->token,
            "aktivran" => $this->aktivran,
            "reset_password" => $this->reset_password,
            "ulogaID" => $this->ulogaID,
            "datum_registracije" => $this->datum_registracije,
            "src" => $this->src,
            "alt" => $this->alt
        ]);
    }

    public function dohvatiJednog()
    {
        return DB::table($this->table)->join('uloga', $this->table . '.ulogaID', '=',
'uloga.ulogaID')
        ->where('korisnikID', $this->korisnikID)->first();
    }

```

```
}

public function izmeniKorisnika()
{
    if(!$this->src)
        $this->src = DB::table($this->table)->where('korisnikID', $this->korisnikID)->first()->src;
    return DB::table($this->table)->where('korisnikID', $this->korisnikID)->update([
        "ime" => $this->ime,
        "prezime" => $this->prezime,
        "email" => $this->email,
        "korisnicko_ime" => $this->korisnicko_ime,
        "aktivran" => $this->aktivran,
        "ulogaID" => $this->ulogaID,
        "src" => $this->src
    ]);
}

public function dohvatiSve()
{
    return DB::table($this->table)->join('uloga', $this->table . '.ulogaID', '=',
'uloga.ulogaID')->paginate(10);
}

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

}
```

4.5) Lajk

```
<?php

namespace App\Models;

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

class Lajk extends Model
{
    protected $table = "lajk";
    protected $primaryKey = "lajkID";

    public function lajkProvera()
    {
        return DB::table($this->table)->where([
            ['korisnikID', $this->korisnikID],
            ['objavaID', $this->objavaID]
        ])->first();
    }

    public function dodajLajk()
    {
        return DB::table($this->table)->insert([
            "lajkID" => null,
            "korisnikID" => $this->korisnikID,
            "objavaID" => $this->objavaID
        ]);
    }

    public function obrisiLajk()
    {
        return DB::table($this->table)->where([
            ['korisnikID', $this->korisnikID],
            ['objavaID', $this->objavaID]
        ])->delete();
    }

    public function brojLajkovaObjave()
    {
        return DB::table($this->table)->where('objavaID', $this->objavaID)->count();
    }
}
```

4.6) Meni

```
<?php

namespace App\Models;

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

class Meni extends Model
{
    protected $table = "meni";
    protected $primaryKey = "meniID";

    public function dohvatiSve()
    {
        $meni = DB::table($this->table)->get();

        foreach ($meni as $m){
            $m->submenus = DB::table($this->table)->where('roditeljID', $m->meniID)->get();
        }

        return $meni;
    }
}
```

4.7) Objava

```
<?php

namespace App\Models;

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

class Objava extends Model
{
    protected $table = "objava";
    protected $primaryKey = "objavaID";

    public function dodajObjavu()
    {
        return DB::table($this->table)->insert([
            "objavaID" => null,
            "naslov" => $this->naslov,
            "tekst" => $this->tekst,
            "objava_created_at" => $this->objava_created_at,
            "objava_updated_at" => $this->objava_updated_at,
            "korisnikID" => $this->korisnikID,
            "kategorijaID" => $this->kategorijaID,
            "src" => $this->src,
            "alt" => $this->alt
        ]);
    }

    public function dohvatiJednu()
    {
        return DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=',
'korisnik.korisnikID')
            ->join('kategorija', $this->table . '.kategorijaID', '=',
'kategorija.kategorijaID')
            ->select('objava.objavaID', 'objava.naslov', 'objava.tekst',
'objava.objava_created_at', 'objava.objava_updated_at',
'objava.src', 'objava.alt', 'objava.kategorijaID',
'objava.korisnikID', 'korisnik.korisnicko_ime', 'kategorija.kategorija')
            ->where('objavaID', $this->objavaID)->first();
    }

    public function izmeniObjavu()
    {
        if(!$this->src)
            $this->src = DB::table($this->table)->where('objavaID', $this->objavaID)->first()->src;
        return DB::table($this->table)->where('objavaID', $this->objavaID)->update([
            "naslov" => $this->naslov,
            "tekst" => $this->tekst,
            "objava_updated_at" => $this->objava_updated_at,
            "kategorijaID" => $this->kategorijaID,
            "src" => $this->src
        ]);
    }

    public function dohvatiSve()
    {
        return DB::table($this->table)
```

```

        ->join('korisnik', $this->table . '.korisnikID', '=',
'korisnik.korisnikID')
        ->join('kategorija', $this->table . '.kategorijaID', '=',
'kategorija.kategorijaID')
        ->select('objava.objavaID', 'objava.naslov', 'objava.tekst',
'objava.objava_created_at', 'objava.objava_updated_at',
'objava.src', 'objava.alt', 'objava.korisnikID',
'korisnik.korisnicko_ime', 'kategorija.kategorija')->paginate(10);
    }

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

    public function dohvatiObjaveKorisnika()
    {
        return DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=',
'korisnik.korisnikID')
            ->join('kategorija', $this->table . '.kategorijaID', '=',
'kategorija.kategorijaID')
            ->select('objava.objavaID', 'objava.naslov', 'objava.tekst',
'objava.objava_created_at', 'objava.objava_updated_at',
'objava.src', 'objava.alt', 'kategorija.kategorija')-
>where('objava.korisnikID', $this->korisnikID)->paginate(5);
    }

    public function dohvatiZadnjuObjavu()
    {
        return DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=',
'korisnik.korisnikID')
            ->join('kategorija', $this->table . '.kategorijaID', '=',
'kategorija.kategorijaID')
            ->select('objava.objavaID', 'objava.naslov', 'objava.tekst',
'objava.objava_created_at', 'objava.objava_updated_at',
'objava.src', 'objava.alt', 'objava.kategorijaID',
'objava.korisnikID', 'korisnik.korisnicko_ime', 'kategorija.kategorija')
            ->orderBy('objava_created_at', 'desc')->first();
    }

    public function dohvatiSveObjavePage($unos, $kategorije, $sortiranje)
    {
        $upit = DB::table($this->table)
            ->join('korisnik', $this->table . '.korisnikID', '=',
'korisnik.korisnikID')
            ->join('kategorija', $this->table . '.kategorijaID', '=',
'kategorija.kategorijaID')
            ->select('objava.objavaID', 'objava.naslov', 'objava.tekst',
'objava.objava_created_at', 'objava.objava_updated_at',
'objava.src', 'objava.alt', 'objava.korisnikID',
'korisnik.korisnicko_ime', 'kategorija.kategorija');

        if(count($kategorije))
            $upit = $upit->whereIn('kategorija.kategorijaID', $kategorije);

        if($unos)
            $upit = $upit->where('objava.naslov', 'like', '%' . $unos . '%');

        if($sortiranje != 0)
        {
            if($sortiranje == 1)

```

```
        $upit = $upit->orderBy('objava.objava_created_at', 'desc');
    if($sortiranje == 2)
        $upit = $upit->orderBy('objava.objava_created_at', 'asc');
    if($sortiranje == 3)
        $upit = $upit->orderBy('objava.naslov', 'asc');
    if($sortiranje == 4)
        $upit = $upit->orderBy('objava.naslov', 'desc');
    }

    return $upit->get();
}

public function dohvatiNajpopularnijeObjave()
{
    return DB::select('SELECT COUNT(l.objavaID) as broj_lajkova, o.objavaID,
o.naslov, o.tekst, o.objava_created_at, o.src, o.alt, o.kategorijaID, o.korisnikID,
k.korisnicko_ime, kat.kategorija
FROM objava o INNER JOIN korisnik k ON
o.korisnikID=k.korisnikID
INNER JOIN kategorija kat on
o.kategorijaID=kat.kategorijaID
INNER JOIN lajk l ON o.objavaID=l.objavaID
GROUP BY l.objavaID, o.objavaID, o.naslov, o.tekst,
o.objava_created_at, o.src, o.alt, o.kategorijaID, o.korisnikID, k.korisnicko_ime,
kat.kategorija
ORDER BY broj_lajkova DESC LIMIT 0,4');
}
```

4.8) Uloga

```
<?php

namespace App\Models;

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

class Uloga extends Model
{
    protected $table = "uloga";
    protected $primaryKey = "ulogaid";

    public function dohvatiSve()
    {
        return DB::table($this->table)->get();
    }
}
```

5. Middleware (app/Http/Middleware)

5.1) AdminProvera

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Log;

class AdminProvera
{
    public function handle($request, Closure $next)
    {
        if(!$request->session()->has('korisnik')){
            Log::critical("Korisnik pokušava da uđe na admin panel! IP adresa: " .
$request->ip());
            return redirect(route('home'));
        }
        if($request->session()->get('korisnik')->uloga != "Admin"){
            Log::critical("Korisnik pokušava da uđe na admin panel! IP adresa: " .
$request->ip());
            return redirect(route('home'));
        }

        return $next($request);
    }
}
```

5.2) ErrorPageProvera

```
<?php

namespace App\Http\Middleware;

use Closure;

class ErrorPageProvera
{
    public function handle($request, Closure $next)
    {
        if(!session('uspesnaPoruka') && !session('neuspesnaPoruka')){
            return redirect(route('home'));
        }

        return $next($request);
    }
}
```

5.3) KorisnikNijePrijavljenProvera

```
<?php

namespace App\Http\Middleware;

use Closure;

class KorisnikNijePrijavljenProvera
{
    public function handle($request, Closure $next)
    {
        if(!$request->session()->has('korisnik')){
            return redirect(route('home'));
        }

        return $next($request);
    }
}
```

5.4) KorisnikPrijavljenProvera

```
<?php

namespace App\Http\Middleware;

use Closure;

class KorisnikPrijavljenProvera
{
    public function handle($request, Closure $next)
    {
        if($request->session()->has('korisnik')){
            return redirect(route('home'));
        }

        return $next($request);
    }
}
```

5.5) ResetPasswordTokenProvera

```
<?php

namespace App\Http\Middleware;

use Closure;

class ResetPasswordTokenProvera
{
    public function handle($request, Closure $next)
    {
        if(!session('reset_password')){
            return redirect(route('home'));
        }

        return $next($request);
    }
}
```

6. Requests (app/Http/Requests)

6.1) AddObjavaProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AddObjavaProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "addObjavaNaslov" => "required|between:15,100",
            "addObjavaTekst" => "required|between:100,1000",
            "addObjavaKategorija" => "gt:0",
            "addObjavaSlika" =>
"required|image|mimes:jpeg,jpg,png,bmp,gif,svg|max:2048"
        ];
    }

    public function messages()
    {
        return [
            "addObjavaNaslov.required" => "Polje za naslov mora biti popunjeno",
            "addObjavaNaslov.between" => "Naslov mora imati 15-100 karaktera",
            "addObjavaTekst.required" => "Polje za tekst mora biti popunjeno",
            "addObjavaTekst.between" => "Tekst mora imati 100-1000 karaktera",
            "addObjavaKategorija.gt" => "Morate odabrat kategoriju",
            "addObjavaSlika.required" => "Morate odabrat sliku",
            "addObjavaSlika.max" => "Slika ne sme biti veća od 2048KB",
            "addObjavaSlika.mimes" => "Slika mora biti jpeg, jpg, png, bmp, gif ili
svg formata",
            "addObjavaSlika.image" => "Loš format, morate dodati sliku"
        ];
    }
}
```

6.2) AddUserProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AddUserProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "regIme" => "required|regex:/^[\u0100-\u017e\u0100-\u017e]{2,12}(\s[\u0100-\u017e\u0100-\u017e]{2,12}){0,1}$/" ,
            "regPrezime" => "required|regex:/^[\u0100-\u017e\u0100-\u017e]{2,12}(\s[\u0100-\u017e\u0100-\u017e]{2,12}){0,1}$/" ,
            "regEmail" =>
"required|regex:/^[\w\d]+[\.\_\w\d]*[\w\d]+\@[\\w]+([\.][\w]+)+$/|unique:korisnik,email",
            "regKorIme" =>
"required|regex:/^[\w\d\.\_]{5,15}$|unique:korisnik,korisnicko_ime",
            "regLozinka" => "required|min:6|same:regLozinka2",
            "regLozinka2" => "required|min:6",
            "regSlika" => "required|image|mimes:jpeg,jpg,png,bmp,gif,svg|max:2048",
            "add_aktivran" => "lt:2",
            "add_uloga" => "gt:0"
        ];
    }

    public function messages()
    {
        return [
            "regIme.required" => "Polje za ime mora biti popunjeno",
            "regIme.regex" => "Ime nije u dobrom formatu",
            "regPrezime.required" => "Polje za prezime mora biti popunjeno",
            "regPrezime.regex" => "Prezime nije u dobrom formatu",
            "regEmail.required" => "Polje za email mora biti popunjeno",
            "regEmail.regex" => "Email nije u dobrom formatu",
            "regEmail.unique" => "Email već postoji",
            "regKorIme.required" => "Polje za korisničko ime mora biti popunjeno",
            "regKorIme.unique" => "Korisničko ime već postoji",
            "regKorIme.regex" => "Korisničko ime mora imati 5-15 karaktera. Sme da sadrži slova, brojeve, tačku i donju crtu",
            "regLozinka.required" => "Polje za lozinku mora biti popunjeno",
            "regLozinka.min" => "Lozinka mora imati bar 6 karaktera",
            "regLozinka.same" => "Lozinka i ponovljena lozinka se ne poklapaju",
            "regLozinka2.required" => "Polje za ponovljenu lozinku mora biti popunjeno",
            "regLozinka2.min" => "Ponovljena lozinka mora imati bar 6 karaktera",
            "regSlika.required" => "Morate odabrati sliku",
            "regSlika.max" => "Slika ne sme biti veća od 2048KB",
            "regSlika.mimes" => "Slika mora biti jpeg, jpg, png, bmp, gif ili svg formata",
            "regSlika.image" => "Loš format, morate dodati sliku",
        ];
    }
}
```

```
        "add_aktiviran.lt" => "Morate odabratи aktivnost korisnika",
        "add_uloga.gt" => "Morate odabratи ulogu"
    }
}
```

6.3) ContactProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ContactProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "contactEmail" =>
"required|regex:/^[\w\d]+[\.\_\w\d]*[\w\d]+\@\[\w\]+([\.\.][\w]+)+$/",
            "poruka" => "required|between:15,500"
        ];
    }

    public function messages()
    {
        return [
            "contactEmail.required" => "Polje za email mora biti popunjeno",
            "contactEmail.regex" => "Email nije u dobrom formatu",
            "poruka.required" => "Polje za poruku mora biti popunjeno",
            "poruka.between" => "Poruka mora imati 15-500 karaktera"
        ];
    }
}
```

6.4) KategorijaProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class KategorijaProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "addKategorija" => "required|between:2,20"
        ];
    }

    public function messages()
    {
        return [
            "addKategorija.required" => "Polje za naziv mora biti popunjeno",
            "addKategorija.between" => "Naziv mora imati 2-20 karaktera"
        ];
    }
}
```

6.5) KomentarProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class KomentarProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "komentar" => "required|between:2,300"
        ];
    }

    public function messages()
    {
        return [
            "komentar.required" => "Polje za komentar mora biti popunjeno",
            "komentar.between" => "Komentar mora imati 2-300 karaktera"
        ];
    }
}
```

6.6) LoginProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "loginKorIme" => "required|regex:/^[\w\d\.\_\_]{5,15}$/,
            "loginLozinka" => "required|min:6"
        ];
    }

    public function messages()
    {
        return [
            "loginKorIme.required" => "Polje za korisničko ime mora biti popunjeno",
            "loginKorIme.regex" => "Korisničko ime mora imati 5-15 karaktera. Sme da sadrži slova, brojeve, tačku i donju crtu",
            "loginLozinka.required" => "Polje za lozinku mora biti popunjeno",
            "loginLozinka.min" => "Lozinka mora imati bar 6 karaktera"
        ];
    }
}
```

6.7) PromenaLozinkeProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class PromenaLozinkeProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "trenutnaLozinka" => "required|min:6",
            "novaLozinka" => "required|min:6|same:novaLozinka2",
            "novaLozinka2" => "required|min:6"
        ];
    }

    public function messages()
    {
        return [
            "trenutnaLozinka.required" => "Polje za trenutnu lozinku mora biti popunjeno",
            "trenutnaLozinka.min" => "Trenutna lozinka mora imati bar 6 karaktera",
            "novaLozinka.required" => "Polje za novu lozinku mora biti popunjeno",
            "novaLozinka.min" => "Nova lozinka mora imati bar 6 karaktera",
            "novaLozinka.same" => "Nova lozinka i ponovljena lozinka se ne poklapaju",
            "novaLozinka2.required" => "Polje za ponovljenu lozinku mora biti popunjeno",
            "novaLozinka2.min" => "Ponovljena lozinka mora imati bar 6 karaktera",
        ];
    }
}
```

6.8) PromenaProfilneSlikeProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class PromenaProfilneSlikeProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "profilnaSlika" =>
"required|image|mimes:jpeg,jpg,png,bmp,gif,svg|max:2048"
        ];
    }

    public function messages()
    {
        return [
            "profilnaSlika.required" => "Morate odabrat sliku",
            "profilnaSlika.max" => "Slika ne sme biti veća od 2048KB",
            "profilnaSlika.mimes" => "Slika mora biti jpeg, jpg, png, bmp, gif ili svg
formata",
            "profilnaSlika.image" => "Loš format, morate dodati sliku"
        ];
    }
}
```

6.9) RegisterProveraRequest

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "regIme" => "required|regex:/^([A-ZŠĐĆČŽ][a-zšđćčž]{2,12}) (\s[A-ZŠĐĆČŽ][a-zšđćčž]{2,12}){0,1}$/,
            "regPrezime" => "required|regex:/^([A-ZŠĐĆČŽ][a-zšđćčž]{2,12}) (\s[A-ZŠĐĆČŽ][a-zšđćčž]{2,12}){0,1}$/,
            "regEmail" =>
"required|regex:/^[\w\d]+[\.\_\w\d]*[\w\d]+\@\[\w\]+([\.\_]\[\w\]+)+$/|unique:korisnik,email",
            "regKorIme" =>
"required|regex:/^[\w\d\.\_]{5,15}$_/|unique:korisnik,korisnicko_ime",
            "regLozinka" => "required|min:6|same:regLozinka2",
            "regLozinka2" => "required|min:6",
            "regSlika" => "required|image|mimes:jpeg,jpg,png,bmp,gif,svg|max:2048"
        ];
    }

    public function messages()
    {
        return [
            "regIme.required" => "Polje za ime mora biti popunjeno",
            "regIme.regex" => "Ime nije u dobrom formatu",
            "regPrezime.required" => "Polje za prezime mora biti popunjeno",
            "regPrezime.regex" => "Prezime nije u dobrom formatu",
            "regEmail.required" => "Polje za email mora biti popunjeno",
            "regEmail.regex" => "Email nije u dobrom formatu",
            "regEmail.unique" => "Email već postoji",
            "regKorIme.required" => "Polje za korisničko ime mora biti popunjeno",
            "regKorIme.unique" => "Korisničko ime već postoji",
            "regKorIme.regex" => "Korisničko ime mora imati 5-15 karaktera. Sme da sadrži slova, brojeve, tačku i donju crtu",
            "regLozinka.required" => "Polje za lozinku mora biti popunjeno",
            "regLozinka.min" => "Lozinka mora imati bar 6 karaktera",
            "regLozinka.same" => "Lozinka i ponovljena lozinka se ne poklapaju",
            "regLozinka2.required" => "Polje za ponovljenu lozinku mora biti popunjeno",
            "regLozinka2.min" => "Ponovljena lozinka mora imati bar 6 karaktera",
            "regSlika.required" => "Morate odabrati sliku",
            "regSlika.max" => "Slika ne sme biti veća od 2048KB",
            "regSlika.mimes" => "Slika mora biti jpeg, jpg, png, bmp, gif ili svg format",
            "regSlika.image" => "Loš format, morate dodati sliku"
        ];
    }
}

```

} }

6.10) ResetLozinkeEmailProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ResetLozinkeEmailProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "email" =>
"required|regex:/^[\w\d]+[\._\w\d]*[\w\d]+\@\[\w\]+(\[\.\]\[\w\]+\)+$/|exists:korisnik,email"
        ];
    }

    public function messages()
    {
        return [
            "email.required" => "Polje za email mora biti popunjeno",
            "email.regex" => "Email nije u dobrom formatu",
            "email.exists" => "Ne postoji nalog sa unetim email-om"
        ];
    }
}
```

6.11) ResetLozinkeNovaLozinkaProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ResetLozinkeNovaLozinkaProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "novaLozinka" => "required|min:6|same:novaLozinka2",
            "novaLozinka2" => "required|min:6"
        ];
    }

    public function messages()
    {
        return [
            "novaLozinka.required" => "Polje za novu lozinku mora biti popunjeno",
            "novaLozinka.min" => "Nova lozinka mora imati bar 6 karaktera",
            "novaLozinka2.min" => "Ponovljena lozinka mora imati bar 6 karaktera",
            "novaLozinka.same" => "Nova lozinka i ponovljena lozinka se ne poklapaju",
            "novaLozinka2.required" => "Polje za ponovljenu lozinku mora biti
popunjeno"
        ];
    }
}
```

6.12) UpdateObjavaProveraRequest

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class UpdateObjavaProveraRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            "addObjavaNaslov" => "required|between:15,100",
            "addObjavaTekst" => "required|between:100,1000",
            "addObjavaKategorija" => "gt:0",
            "addObjavaSlika" =>
"nullable|image|mimes:jpeg,jpg,png,bmp,gif,svg|max:2048"
        ];
    }

    public function messages()
    {
        return [
            "addObjavaNaslov.required" => "Polje za naslov mora biti popunjeno",
            "addObjavaNaslov.between" => "Naslov mora imati 15-100 karaktera",
            "addObjavaTekst.required" => "Polje za tekst mora biti popunjeno",
            "addObjavaTekst.between" => "Tekst mora imati 100-1000 karaktera",
            "addObjavaKategorija.gt" => "Morate odabrat kategoriju",
            "addObjavaSlika.max" => "Slika ne sme biti veća od 2048KB",
            "addObjavaSlika.mimes" => "Slika mora biti jpeg, jpg, png, bmp, gif ili
svg formata",
            "addObjavaSlika.image" => "Loš format, morate dodati sliku"
        ];
    }
}
```

6.13) UpdateUserProveraRequest

7. Routes (routes/web.php)

```
<?php

/*----regularni izrazi----*/
Route::pattern('id', '[0-9]+');
Route::pattern('korisnikID', '[0-9]+');
Route::pattern('objavaID', '[0-9]+');
Route::pattern('token', '[\w\d]{32}');


/*----nepostojeca stranica i error----*/
Route::fallback('FrontEndController@fallback');
Route::get('/error', 'FrontEndController@error')->name('errorpage')-
>middleware('errorPageProvera');


/*----registracija, prijava, odjava (prikaz i logika)----*/
Route::get('/register', 'FrontEndController@register')->name('registerpage')-
>middleware('korisnikPrijavljenProvera');
Route::post('/register', 'RegisterLoginController@registracija')-
>name('registracija');
Route::get('/login', 'FrontEndController@login')->name('loginpage')-
>middleware('korisnikPrijavljenProvera');
Route::post('/login', 'RegisterLoginController@prijava')->name('prijava');
Route::get('/logout', 'RegisterLoginController@odjava')->name('odjava')-
>middleware('korisnikNijePrijavljenProvera');
Route::get('/activate/{token}', 'RegisterLoginController@activate')-
>name('aktivacija')->middleware('korisnikPrijavljenProvera');


/*----resetovanje lozinke (prikaz i logika)----*/
Route::put('/resetpassword/emailcheck',
'ChangePasswordController@resetLozinkeEmailProvera')-
>name('resetLozinkeEmailProvera');
Route::get('/resetpassword/{token}',
'ChangePasswordController@resetLozinkeTokenProvera')-
>name('resetLozinkeTokenProvera')->middleware('korisnikPrijavljenProvera');
Route::get('/resetpassword', 'FrontEndController@resetpasswordpage')-
>name('resetpasswordpage')->middleware('resetPasswordTokenProvera',
'korisnikPrijavljenProvera');
Route::put('/resetpassword', 'ChangePasswordController@resetPassword')-
>name('resetPassword');


/*----prikaz ostalih stranica----*/
Route::get('/', 'FrontEndController@home')->name('home');
Route::get('/userprofile/{id}', 'FrontEndController@userprofile')-
>name('userprofilepage')->middleware('korisnikNijePrijavljenProvera');
Route::get('/addobjavauser', 'FrontEndController@addobjavauser')-
>name('addobjavauserpage')->middleware('korisnikNijePrijavljenProvera');
Route::get('/updateobjavauser/{id}', 'FrontEndController@updateobjavauser')-
>name('updateobjavauserpage')->middleware('korisnikNijePrijavljenProvera');
Route::get('/contact', 'FrontEndController@contact')->name('contactpage');
Route::get('/author', 'FrontEndController@author')->name('authorpage');
Route::get('/objava/{objavaID}/{korisnikID}', 'FrontEndController@jednaObjava')-
>name('jednaObjava');
```

```
/*----stranica 'objave' (prikaz i logika)----*/
Route::get('/objave', 'FrontEndController@objave')->name('objavepage');
Route::post('/objaveajax', 'FrontEndController@sveobjave')->name('objaveajax');

/*----promena profilne slike i lozinke (logika)----*/
Route::put('/userchangesimage', 'UserChangesController@promenaProfilneSlike')-
>name('promenaProfilneSlike');
Route::put('/changepassword', 'ChangePasswordController@promenaLozinke')-
>name('promenaLozinke');

/*----kontakt (logika)----*/
Route::post('/contactadmin', 'FrontEndController@contactAdmin')->name('contactAdmin');

/*----admin (prikaz i logika)----*/
Route::resource('admin/korisnik', 'Admin\KorisnikController')-
>middleware('adminProvera');
Route::get('/admin', 'FrontEndController@admin')->name('adminhomepage')-
>middleware('adminProvera');
Route::resource('admin/objava', 'Admin\ObjavaController'); //middleware je u
konstruktoru kontrolera
Route::resource('/admin/komentar', 'Admin\KomentarController'); //middleware je u
konstruktoru kontrolera
Route::resource('/admin/kategorija', 'Admin\KategorijaController')-
>middleware('adminProvera');
Route::resource('/admin/aktivnost', 'Admin\AktivnostController')-
>middleware('adminProvera');

/*----lajkovanje (logika)----*/
Route::post('/lajkovanje', 'UserChangesController@lajkovanje')->name('lajkovanje');
```

8. CSS (public/css/style.css)

```
/*-----HEAD-----*/
@import url('https://fonts.googleapis.com/css?family=Mukta');
body{
    font-family: 'Mukta', sans-serif;
    color: #25262a;
}
*{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
ul{
    list-style-type: none;
}
a{
    text-decoration: none;
}
a:hover{
    text-decoration: none;
}

/*-----FOOTER-----*/
#footer{
    width: 100%;
    //background: #2ba0db;
    background: #1d68a7;
    text-align: center;
    height: 200px;
}

#footer ul li{
    display: inline-block;
    padding: 10px;
}

#footer ul li a:link{
    //color: #1d68a7;
    color: #25262a;
}

#footer ul li a:visited{
    //color: #1b4b72;
    color: #25262a;
}

#footer ul li a:hover{
    color: #F5F5F5;
}

#footer p{
    font-size: 15px;
}

#footer ul{
```

```
        padding-top: 50px;
        padding-bottom: 25px;
    }
    /*-----FOOTER-----*/
}

/*-----REGISTER-----*/
#register_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#register_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    padding-top: 75px;
}

#tabela_register tr:last-child{
    text-align: center;
}

#tabela_register tr td{
    padding-left: 25px;
    padding-right: 25px;
}

#tabela_register tr:last-child td{
    padding-top: 10px;
}

#regDugme{
    font-weight: bold;
    letter-spacing: 1px;
    cursor: pointer;
}

#forma_register{
    padding-bottom: 40px;
}

#regSlika{
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
}

.reg_greske{
    width: 100%;
    background-color: #F5F5F5;
    display: flex;
    justify-content: center;
    align-items: center;
}

.reg_greske_drzac{
    width: 650px;
    margin: 0px auto;
    color: red;
}
```

```
.reg_greske h4{
    padding-bottom: 5px;
}

.reg_greske h3{
    padding-bottom: 5px;
}

.reg_greske h5{
    padding-bottom: 5px;
}
/*-----REGISTER-----*/

/*-----LOGIN-----*/
#login_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#login_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    padding-top: 75px;
}

#tabela_login tr:last-child td{
    padding-top: 10px;
}

#loginDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
    cursor: pointer;
}

#login_greske{
    width: 100%;
    text-align: center;
    background-color: #F5F5F5;
}

#forma_login{
    padding-bottom: 40px;
}

#login_drzac a:link{
    color: blue;
}

#login_drzac a:visited{
    color: blue;
}

#login_drzac a:hover{
    color: red;
}

#zab_loz_p{
```

```
        color: blue;
        cursor: pointer;
    }

#zab_loz_div{
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #F5F5F5;
}

#zab_loz_p:hover{
    color: red;
}

#resetPassEmailIspis{
    height: 50px;
    text-align: center;
    background-color: #F5F5F5;
}

#resetPassEmailDugme{
    font-weight: bold;
    letter-spacing: 1px;
    cursor: pointer;
}

#resetPassEmailTabela{
    margin-bottom: 30px;
    padding-top: 30px;
}

#resetPassEmailTabela tr:first-child td {
    text-align: center;
    padding-bottom: 20px;
}
/*-----LOGIN-----*/
/*-----ZAJEDNICKO-----*/
.stranice{
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}

.ispisGreske{
    width: 300px;
    height: 20px;
    margin-bottom: 15px;
    color: red;
    font-size: 15px;
    line-height: 1.1;
}

.naslov_svaki{
    width: 100%;
    text-align: center;
    font-size: 32px;
}
```

```
.naslov_svaki p{
    padding-top: 50px;
    padding-bottom: 50px;
}

.inputi{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 300px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

.inputi_dugme{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 300px;
    border: 1px solid #A9A9A9;
}

.greske{
    width: 100%;
    background-color: #F5F5F5;
    text-align: center;
    color: red;
}

.greske h4{
    padding-bottom: 5px;
}

.greske h3{
    padding-bottom: 5px;
}

.greske h5{
    padding-bottom: 5px;
}

.ispis_uspesno{
    width: 100%;
    background-color: #F5F5F5;
    text-align: center;
}

.ispis_uspesno h4{
    padding-bottom: 5px;
}

.ispis_uspesno h3{
    padding-bottom: 5px;
}

.ispis_uspesno h5{
    padding-bottom: 5px;
}

.detaljniye{
    font-size: 15px;
    padding: 10px;
    background-color: #1d68a7;
```

```
        display: inline-block;
        margin-bottom: 10px;
        border-radius: 5px;
        letter-spacing: 2px;
    }

.detaljniye:link{
    color: white;
}

.detaljniye:visited{
    color: white;
}

.detaljniye:hover{
    color: #25262a;
}
/*-----ZAJEDNICKO-----*/



/*-----ERROR-----*/
.errors_uspesno{
    width: 100%;
    text-align: center;
    height: 400px;
    background-color: #F5F5F5;
}

.errors_uspesno h2{
    padding-top: 175px;
}

.errors_neuspесно{
    width: 100%;
    text-align: center;
    color: red;
    height: 400px;
    background-color: #F5F5F5;
}

.errors_neuspесно h2{
    padding-top: 175px;
}
/*-----ERROR-----*/



/*-----RESETPASSWORD-----*/
#reset_lozinke_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#reset_lozinke_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
    padding-top: 75px;
}
```

```
#resetLozinkeDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
    cursor: pointer;
}

#resetLozinkeGreske{
    margin: 0px auto;
    text-align: center;
    height: 125px;
    padding-top: 30px;
}

#resetLozinkeGreske h4{
    padding-bottom: 5px;
}

#resetLozinkeGreske h5{
    padding-bottom: 5px;
}
/*-----RESETPASSWORD-----*/

/*-----USERPROFILE-----*/
#profil_slika_centar{
    text-align: center;
}

#profile_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#profile_drzac{
    margin: 0px auto;
    width: 1300px;
    display: flex;
    justify-content: space-around;
    align-items: center;
    padding-top: 75px;
}

#o_korisniku{
    width: 500px;
    padding-left: 75px;
}

#user_profile_image img{
    max-width: 350px;
}

#user_profile_image table{
    margin-bottom: 30px;
}

#user_profile_image tr td{
    padding-top: 5px;
}

#user_profile_image tr:last-child td{
    padding-top: 10px;
```

```
}

#promenaSlikeDugme{
    letter-spacing: 1px;
    cursor: pointer;
    width: 150px;
    height: 25px;
}

#profilnaSlika{
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 200px;
}

#promenaProfilneSlikeGreske{
    height: 50px;
    text-align: center;
}
#user_info{
    display: flex;
    justify-content: center;
    align-items: center;
    padding-bottom: 75px;
}

#user_info table{
    font-size: 18px;
    width: 350px;
}

#user_info table th{
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
    border-radius: 5px;
    text-align: center;
}

#user_info table tr:hover {
    background-color: #ddd;
}

#user_info table tr td{
    padding-left: 10px;
    padding-right: 10px;
    border-bottom: 1px solid #A9A9A9;
    height: 40px;
}

#promena_lozinke{
    width: 500px;
    height: 500px;
}

#tabela_promeni_lozinku tr:last-child td{
    padding-top: 10px;
}

#tabela_promeni_lozinku{
    width: 400px;
}
```

```
#promenaLozinkeDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
    cursor: pointer;
}

#formaPromenaLozinke{
    margin: 0px 0px 0px 100px;
}

#promenaLozinkeGreske{
    width: 500px;
    text-align: center;
    margin-top: 20px;
}

#promenaLozinkeGreske h4{
    padding-bottom: 5px;
}

#promenaLozinkeGreske h5{
    padding-bottom: 5px;
}

#novaLozinkaGreska{
    width: 400px;
}

#novaLozinkaGreska2{
    width: 400px;
}
/*-----USERPROFILE-----*/

/*-----CONTACT-----*/
#contact_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#contact_sadrzaj_drzac{
    margin: 0px auto;
    width: 1300px;
    display: flex;
    justify-content: center;
    align-items: center;
    padding-top: 75px;
}

#contactEmail{
    font-family: 'Mukta', sans-serif;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
    width: 400px;
    height: 35px;
    font-size: 15px;
}

#poruka{
    font-family: 'Mukta', sans-serif;
    height: 150px;
```

```
    font-size: 15px;
    resize: vertical;
    min-height: 100px;
    max-height: 400px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
    width: 400px;
}

#contactDugme{
    font-family: 'Mukta', sans-serif;
    border: 1px solid #A9A9A9;
    width: 400px;
    height: 35px;
    font-size: 15px;
    font-weight: bold;
    margin-top: 10px;
    letter-spacing: 1px;
    cursor: pointer;
}

#forma_contact{
    padding-bottom: 10px;
}

.ispisGreskeContact{
    width: 400px;
    height: 20px;
    margin-bottom: 15px;
    color: red;
    font-size: 15px;
}

#contact_greske{
    text-align: center;
    background-color: #F5F5F5;
    padding-top: 10px;
    height: 75px;
}

#contact_greske h4{
    padding-bottom: 5px;
}

#contact_greske h3{
    padding-bottom: 5px;
}

#contact_greske h5{
    padding-bottom: 5px;
}
/*-----CONTACT-----*/
/*-----AUTHOR-----*/
#author_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#author_drzac{
    margin: 0px auto;
```

```
width: 650px;
display: flex;
align-items: center;
justify-content: space-between;
padding-bottom: 40px;
padding-top: 75px;
}

#author_drzac img{
    max-width: 300px;
}

#author_tekst{
    text-align: justify;
    width: 300px;
}

#dokumentacija{
    text-align: center;
    padding-bottom: 50px;
    font-size: 20px;
}

#dokumentacija a{
    color: #25262a;
}

#dokumentacija a:hover{
    color: #2ba0db;
}
/*-----AUTHOR-----*/



/*-----404-----*/
#error404{
    width: 100%;
    text-align: center;
    background-color: rgb(241, 247, 247);
    font-size: 32px;
}

#error404 p{
    padding-top: 50px;
}
/*-----404-----*/



/*-----ADMIN-----*/
#aktivnost_sortDiv{
    width: 100%;
    text-align: right;
    padding-top: 20px;
    padding-bottom: 20px;
}

#aktivnost_sortDiv form{
    margin-right: 20px;
}

#aktivnost_sort{
    height: 35px;
}
```

```
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 150px;
border: 1px solid #A9A9A9;
text-indent: 5px;
}

#aktivnost_sortDugme{
height: 35px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 100px;
border: 1px solid #A9A9A9;
cursor: pointer;
//font-weight: bold;
letter-spacing: 1px;
}

.admin_greske{
width: 100%;
text-align: center;
background-color: #F5F5F5;
}

@admin_greske h4{
padding-bottom: 5px;
}

#add_uloga{
height: 35px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 125px;
border: 1px solid #A9A9A9;
text-indent: 5px;
}

#add_aktivvan{
height: 35px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 125px;
border: 1px solid #A9A9A9;
text-indent: 5px;
}

#addUserDugme{
font-weight: bold;
letter-spacing: 1px;
cursor: pointer;
}

#tabela_adduser tr td{
padding-left: 25px;
padding-right: 25px;
}

#updateUserDugme{
font-weight: bold;
letter-spacing: 1px;
cursor: pointer;
}
```

```
#updateUserSlika{
    max-width: 250px;
}

#updateUserSlikaSredina{
    text-align: center;
}

.naslov_svaki_admin{
    width: 100%;
    text-align: center;
    font-size: 32px;
    background-color: #F5F5F5;
}

.naslov_svaki_admin p{
    padding-top: 50px;
    padding-bottom: 50px;
}

.allUsersSlika{
    max-width: 200px;
    max-height: 200px;
}

#allusers_sadrzaj{
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
}

.trenutno_nema{
    width: 100%;
    text-align: center;
    font-size: 32px;
    background-color: #F5F5F5;
}

.trenutno_nema p{
    padding-top: 200px;
    padding-bottom: 200px;
}

#admin_sadrzaj{
    width: 100%;
}

#admin_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
}

#admin_drzac ul li a{
    display: block;
    margin: 25px;
    font-weight: bold;
    padding: 10px;
    background-color: #373737;
    text-align: center;
    letter-spacing: 1px;
```

```
    border-radius: 5px;
    width: 200px;
}

#admin_drzac ul li a:link{
    color: #A9A9A9;
}

#admin_drzac ul li a:visited{
    color: #A9A9A9;
}

#admin_drzac ul li a:hover{
    color: #F5F5F5;
}

#admin_sadrzaj2{
    width: 100%;
}

#admin_drzac2{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
}

#admin_drzac2 ul li{
    display: inline-block;
}

#admin_drzac2 ul li a{
    display: inline-block;
    margin: 20px;
    font-weight: bold;
    padding: 10px;
    background-color: #373737;
    text-align: center;
    letter-spacing: 1px;
    border-radius: 5px;
}

#admin_drzac2 ul li a:link{
    color: #A9A9A9;
}

#admin_drzac2 ul li a:visited{
    color: #A9A9A9;
}

#admin_drzac2 ul li a:hover{
    color: #F5F5F5;
}

#admin_meni{
    width: 100%;
    text-align: center;
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
}

#admin_meni h1{
```

```
        padding-top: 50px;
        padding-bottom: 50px;
    }

#tabela_addObjava tr:last-child{
    text-align: center;
}

#tabela_addObjava tr td{
    padding-left: 25px;
    padding-right: 25px;
}

#tabela_addObjava tr:last-child td{
    padding-top: 10px;
}

#tabela_addObjava tr td{
    padding-left: 25px;
    padding-right: 25px;
}

#forma_addObjava{
    padding-bottom: 40px;
}

#addObjavaNaslov{
    font-family: 'Mukta', sans-serif;
    height: 50px;
    font-size: 15px;
    min-height: 50px;
    max-height: 50px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
    width: 700px;
}

#addObjavaTekst{
    font-family: 'Mukta', sans-serif;
    height: 200px;
    font-size: 15px;
    min-height: 200px;
    max-height: 200px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
    width: 700px;
}

#addObjavaKategorija{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 125px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

#addObjavaSlika{
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
}

#addObjavaDugme{
```

```
        font-weight: bold;
        letter-spacing: 1px;
        cursor: pointer;
    }

#updateObjavaDugme{
    font-weight: bold;
    letter-spacing: 1px;
    cursor: pointer;
}

#updateObjavaSlika{
    max-width: 700px;
}

#updateObjavaSlikaSredina{
    text-align: center;
}

#addKategorijaDugme{
    font-weight: bold;
    letter-spacing: 1px;
    cursor: pointer;
}

#updateKategorijaDugme{
    font-weight: bold;
    letter-spacing: 1px;
    cursor: pointer;
}
/*-----ADMIN-----*/
/*-----*/
.obrisi{
    font-size: 15px;
    padding: 10px;
    //font-weight: bold;
    background-color: darkred;
    display: inline-block;
    letter-spacing: 1px;
    border-radius: 5px;
}

.obrisi:link{
    color: #A9A9A9;
}

.obrisi:visited{
    color: #A9A9A9;
}

.obrisi:hover{
    color: white;
}

.detaljniJeD{
    font-size: 15px;
    padding: 10px;
    //font-weight: bold;
    background-color: darkolivegreen;
    display: inline-block;
```

```
letter-spacing: 1px;
border-radius: 5px;
}

.detajniJeD:link{
    color: #A9A9A9;
}

.detajniJeD:visited{
    color: #A9A9A9;
}

.detajniJeD:hover{
    color: white;
}

.tabela {
    border-collapse: collapse;
    width: 100%;
    margin-bottom: 50px;
}

.tabela td, .tabela th {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: center;
}

.tabela tr:nth-child(even){background-color: #f2f2f2; }

.tabela tr:hover {background-color: #ddd; }

.tabela th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: center;
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
}
/*-----*/
/*-----JEDNAOBJAVA-----*/
#jednaobjava_sadrzaj{
    width: 100%;
}

#jednaobjava_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
}
#jednaobjava_tabela{
    width: 900px;
    margin-top: 75px;
}

#jednaobjava_tabela tr td{
    padding-top: 35px;
```

```
}

#jednaobjava_slika_img{
    width: 900px;
}

#jednaobjava_slika_div{
    width: 900px;
}

#jednaobjava_tekst{
    width: 900px;
    text-align: justify;
}

#objava_header_tekst td{
    padding-bottom: 10px;
    color: #707070;
}

#komentari_naslov{
    margin: 0px auto;
    width: 900px;
}

#komentari_naslov h5{
    padding-top: 75px;
    padding-bottom: 30px;
}

#komentari_sadrzaj{
    margin: 0px auto;
    width: 900px;
}

#nema_komentara{
    margin: 0px auto;
    width: 900px;
}

#komentari_prijava{
    margin: 0px auto;
    width: 900px;
    padding-bottom: 40px;
}

.jedankomentar{
    width: 700px;
    height: 150px;
    margin-bottom: 50px;
    border-top: 3px solid #2ba0db;
}

.jedankomentar_slika{
    width: 150px;
    height: 150px;
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}
.jedankomentar_sadrzaj{
    width: 550px;
```

```
    height: 150px;
    float: left;
}

.jedankomentar_header{
    width: 550px;
    height: 40px;
    font-size: 12px;
    padding-left: 20px;
    color: #707070;
    padding-top: 5px;
}

.jedankomentar_header_tabela{
    width: 525px;
}

.jedankomentar_header_tabela tr td{
    width: 175px;
}

.jedankomentar_tekst{
    width: 550px;
    height: 110px;
    display: flex;
    align-items: center;
    text-align: justify;
    padding-left: 20px;
}

.jedankomentar_slika_img{
    max-width: 140px;
    max-height: 140px;
}

#dodaj_komentar_naslov{
    margin: 0px auto;
    width: 900px;
    margin-top: 75px;
}

#dodaj_komentar_sadrzaj{
    width: 900px;
    margin: 0px auto;
    margin-top: 20px;
}

#dodaj_komentar{
    width: 700px;
    height: 150px;
    border-left: 3px solid #373737;
}

#dk_slika_div{
    width: 150px;
    height: 150px;
    display: flex;
    justify-content: center;
    align-items: center;
    float: left;
}
```

```
#dk_slika_div img{
    max-width: 140px;
    max-height: 140px;
}

#dk_tekst_div{
    width: 444px;
    height: 150px;
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}

#dk_dugme_div{
    width: 100px;
    height: 150px;
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}

#dodajKomentarTextarea{
    width: 420px;
    height: 100px;
    max-height: 100px;
    min-height: 100px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

#dodajKomentarDugme{
    height: 100px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 80px;
    border: 1px solid #A9A9A9;
    //border: none;
    //background-color: #1d68a7;
    //background-color: #707070;
    border-radius: 5px;
    cursor: pointer;
    font-weight: bold;
    letter-spacing: 2px;
    //color: white;
}

#dodajKomentarDugme:hover {
    color: #25262a;
}

#komentar_greska{
    width: 500px;
    height: 50px;
    color: red;
}

/*-----JEDNAOBJAVA-----*/
```

```
/*nav*/
#nav {
    background: #2ba0db;
    margin-left: 30px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

#nav ul {
    font-size: 0;
    margin: 0;
    padding: 0;
}

#nav ul li {
    display: inline-block;
    position: relative;
}

#nav ul li a {
    color: #fff;
    display: block;
    font-size: 15px;
    padding: 20px 20px;
    transition: 0.3s linear;
    letter-spacing: 1px;
}

#nav ul li:hover { background: #126d9b; }

#nav ul li ul {
    border-bottom: 5px solid #2ba0db;
    display: none;
    position: absolute;
    width: 200px;
}

#nav ul li ul li {
    border-top: 1px solid #1d68a7;
    display: block;
}

#nav ul li ul li:first-child { border-top: none; }

#nav ul li ul li a {
//background: #373737;
    background: #1b4b72;
    display: block;
    padding: 10px 14px;
}

#nav ul li ul li a:hover { background: #126d9b; }

/*nav*/

/*-----HOME-----*/
#logo{
    margin-right: 15px;
}

#logo img{
```

```
    height: 50px;
}
#navigacija{
    background: #2ba0db;
    width: 100%;
    display: flex;
    justify-content: space-between;
}

#adminLink{
    color: white;
    letter-spacing: 1px;
    font-size: 15px;
    padding: 20px 20px;
    margin-right: 20px;
}

#adminLink:hover{
    background: #126d9b;
}

.naslov_svaki_boja{
    width: 100%;
    text-align: center;
    font-size: 32px;
    background-color: #F5F5F5;
}

.naslov_svaki_boja p{
    padding-top: 50px;
    padding-bottom: 50px;
}

/*-----HOME-----*/
/*-----OBJAVE-----*/
#objave_sadrzaj{
    width: 100%;
}

#objave_drzac{
    margin: 0px auto;
    width: 1300px;
    display: flex;
    justify-content: space-between;
}

#filter_sort{
    width: 250px;
    padding-top: 35px;
    border-right: 10px solid #F5F5F5;
}

#filter_sort div{
    margin-top: 50px;
    margin-bottom: 50px;
}

#sortiraj{
    height: 35px;
    font-family: 'Mukta', sans-serif;
```

```
    font-size: 15px;
    width: 150px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

#objave_objavepage{
    margin-right: 100px;
}

#searchDiv{
    width: 100%;
    text-align: right;
    padding-bottom: 5px;
    padding-top: 5px;
}

#searchDiv input:first-child{
    margin-right: 10px;
}

#searchDiv p{
    margin-right: 20px;
}

#searchButton{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 100px;
    //border: 1px solid #A9A9A9;
    cursor: pointer;
    //font-weight: bold;
    letter-spacing: 1px;
    background-color: #1d68a7;
    color: white;
    border: none;
    outline: none;
}

#searchButton:hover{
    color: #25262a;
}
/*-----OBJAVE-----*/
```

```
/*straniceAjax*/
.straniceAjax{
    width: 100%;
    text-align: center;
    padding-top: 20px;
    padding-bottom: 20px;
    background-color: #F5F5F5;
    height: 70px;
}

.straniceAjax a{
    padding: 5px 10px 5px 10px;
    margin-left: 5px;
    margin-right: 5px;
}
```

```
.straniceAjax a:link{
    color: #25262a;
}

.straniceAjax a:visited{
    color: #25262a;
}

.straniceAjax a:hover{
    color: #25262a;
    background-color: #7971ea;
    border-radius: 5px;
}
/*straniceAjax*/

/*komponenta*/
.objava_sadrzaj{
    width: 100%;
}

.objava_drzac{ /*home page*/
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
}

.objava_drzac_objavepage{ /*obajave page*/
    margin: 0px auto;
}

.objava_tabela{
    width: 800px;
}

.objava_tabela tr td{
    padding-top: 35px;
}

.objava_slika_img{
    width: 800px;
}

.objava_slika_div{
    width: 800px;
}

.objava_tekst{
    width: 800px;
    text-align: justify;
}

.objava_detalji td{
    width: 300px;
}

.objava_detalji td{
    text-align: center;
}
```

```
.objava_detalji td:last-child{
    text-align: right;
}

.objava_detalji td:first-child{
    text-align: left;
}

.razvdavanje_objava{
    width: 100%;
    border-top: 10px solid #F5F5F5;
    height: 10px;
}
/*komponenta*/



/*ucitavanje*/
.ucitavanje {
    display: none;
    position: fixed;
    z-index: 1000;
    top: 0;
    left: 0;
    height: 100%;
    width: 100%;
    background: rgba(0, 0, 0, 0.5)
    url('../images/ucitavanje.gif')
    50% 50%
    no-repeat;
}

body.loading .ucitavanje {
    overflow: hidden;
}

body.loading .ucitavanje {
    display: block;
}
/*ucitavanje*/
```

9. JavaScript (public/js/script.js)

```
$document.ready(function () {
    $('#loginKorIme').focus();
    $('#zab_loz_p').on('click', prikaziSakrijResetovanje);
    $('#zab_loz_div').hide();
    $('#resetPassEmailIspis').hide();
    $('#resetPassEmailDugme').on('click', resetPassEmailProvera);
    $('#resetLozinkeDugme').on('click', resetLozinke);
    $('.vecaSlika').simpleLightbox();
    $('#promenaLozinkeDugme').on('click', promenaLozinke);
    $('.obrisiKorisnika').on('click', obrisiKorisnika);
    $('.obrisiObjavu').on('click', obrisiObjavu);
    $('.obrisiKomentar').on('click', obrisiKomentar);
    $('#dodajKomentarDugme').on('click', dodajKomentar);
    $('.lajkovanjeLink').on('click', lajkovanje);
    $('#lajkovanjeProvera').on('click', lajkovanjeProvera);
    $('.obrisiKategoriju').on('click', obrisiKategoriju);

    $('#nav li').hover(
        function() {
            $('ul', this).stop().slideDown(200);
        },
        function() {
            $('ul', this).stop().slideUp(200);
        }
    );
}

prikaziSveObjaveAjax();
$('#searchButton').on('click', prikaziSveObjaveAjax);
$('#sortiraj').on('change', prikaziSveObjaveAjax);
$('.kategorija_filter').on('click', prikaziSveObjaveAjax);

$(document).on({
    ajaxStart: function() { $('body').addClass("loading"); },
    ajaxStop: function() { $('body').removeClass("loading"); }
});

});

function regProvera() {
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var regLozinka = $('#regLozinka').val();
    var regLozinka2 = $('#regLozinka2').val();
    var regSlika = $('#regSlika').val();

    var proveraImePrezime = /^[A-ŽĐĆĆŽ][a-zđććž]{2,12}(\s[A-ŽĐĆĆŽ][a-zđććž]{2,12}){0,1}$/;
    var proveraEmail = /^[\\w\\d]+[\\._\\w\\d]*[\\w\\d]+@[\\w]+([\\.]\\w]+)+$/;
    var proveraKorIme = /^[\\w\\d\\.\\_]{5,15}$/;
    var regGreske = [];

    if(regIme === ""){
```

```

        regGreske.push("Polje za ime mora biti popunjeno");
        $('#regImeGreska').html("Polje za ime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regIme)){
        regGreske.push("Ime nije u dobrom formatu");
        $('#regImeGreska').html("Ime nije u dobrom formatu");
    } else{
        $('#regImeGreska').html("");
    }

    if(regPrezime === ""){
        regGreske.push("Polje za prezime mora biti popunjeno");
        $('#regPrezimeGreska').html("Polje za prezime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regPrezime)){
        regGreske.push("Prezime nije u dobrom formatu");
        $('#regPrezimeGreska').html("Prezime nije u dobrom formatu");
    } else{
        $('#regPrezimeGreska').html("");
    }

    if(regEmail === ""){
        regGreske.push("Polje za email mora biti popunjeno");
        $('#regEmailGreska').html("Polje za email mora biti popunjeno");
    } else if(!proveraEmail.test(regEmail)){
        regGreske.push("Email nije u dobrom formatu");
        $('#regEmailGreska').html("Email nije u dobrom formatu");
    } else{
        $('#regEmailGreska').html("");
    }

    if(regKorIme === ""){
        regGreske.push("Polje za korisničko ime mora biti popunjeno");
        $('#regKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
    } else if(!proveraKorIme.test(regKorIme)){
        regGreske.push("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži
slova, brojeve, tačku i donju crtlu");
        $('#regKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera<br/>Sme
da sadrži slova, brojeve, tačku i donju crtlu");
    } else{
        $('#regKorImeGreska').html("");
    }

    if(regLozinka === ""){
        regGreske.push("Polje za lozinku mora biti popunjeno");
        $('#regLozinkaGreska').html("Polje za lozinku mora biti popunjeno");
    } else if(regLozinka.length < 6){
        regGreske.push("Lozinka mora imati bar 6 karaktera");
        $('#regLozinkaGreska').html("Lozinka mora imati bar 6 karaktera");
    } else if((regLozinka !== regLozinka2) && (regLozinka2 !== "") &&
(regLozinka2.length >= 6)){
        regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#regLozinkaGreska').html("");
    }

    if(regLozinka2 === ""){
        regGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
        $('#regLozinka2Greska').html("Polje za ponovljenu lozinku mora biti
popunjeno");
    } else if(regLozinka2.length < 6){
        regGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
        $('#regLozinka2Greska').html("Ponovljena lozinku mora imati bar 6 karaktera");
    }

```

```

        } else if((regLozinka !== regLozinka2) && (regLozinka !== "") &&
        (regLozinka.length >= 6)){
            regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
            $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
            $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
        } else{
            $('#regLozinka2Greska').html("");
        }

        if(regSlika === ""){
            regGreske.push("Morate odabrati sliku");
            $('#regSlikaGreska').html("Morate odabrati sliku");
        } else{
            $('#regSlikaGreska').html("");
        }

        return (regGreske.length > 0) ? false : true;
    }

    function loginProvera() {
        var loginKorIme = $.trim($('#loginKorIme').val());
        var loginLozinka = $('#loginLozinka').val();
        var reLoginKorIme = /^[^\d\.\_\_]{5,15}$/;
        var loginGreske = [];

        if(loginKorIme === ""){
            loginGreske.push("Polje za korisničko ime mora biti popunjeno");
            $('#loginKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
        } else if(!reLoginKorIme.test(loginKorIme)){
            loginGreske.push("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži
slova, brojeve, tačku i donju crtu");
            $('#loginKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera<br/>Sme
da sadrži slova, brojeve, tačku i donju crtu");
        } else{
            $('#loginKorImeGreska').html("");
        }

        if(loginLozinka === ""){
            loginGreske.push("Polje za lozinku mora biti popunjeno");
            $('#loginLozinkaGreska').html("Polje za lozinku mora biti popunjeno");
        } else if(loginLozinka.length < 6){
            loginGreske.push("Lozinka mora imati bar 6 karaktera");
            $('#loginLozinkaGreska').html("Lozinka mora imati bar 6 karaktera");
        } else{
            $('#loginLozinkaGreska').html("");
        }

        return (loginGreske.length > 0) ? false : true;
    }

    function prikaziSakrijResetovanje() {
        $('#zab_loz_div').toggle();
        $('#resetPassEmailIspis').toggle();
        $('#resetPassEmailIspis').html("");
        $('#resetPassEmail').val("");
    }

    function resetPassEmailProvera() {
        var email = $.trim($('#resetPassEmail').val());
        var proveraEmail = /^[^\d\.\_\_][\w\d]*[\w\d]+@[^\w]+([\w\.\w]+)+$/;
        var greska = "";

```

```

if(email === ""){
    greska = "Polje za email mora biti popunjeno";
    $('#resetPassEmailGreska').html("Polje za email mora biti popunjeno");
} else if(!proveraEmail.test(email)){
    greska = "Email nije u dobrom formatu";
    $('#resetPassEmailGreska').html("Email nije u dobrom formatu");
} else{
    $('#resetPassEmailGreska').html("");
}

if(greska === ""){
    var data = {
        email : email
    };

    $.ajax({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        },
        url : baseUrl + "/resetpassword/emailcheck",
        method : "PUT",
        dataType : "json",
        data : data,
        success : function (data) {
            $('#resetPassEmailIspis').html("<h4>" + data + "</h4>");

            if(data === "Proverite email za nastavak procesa . . . . ."){
                $('#resetPassEmail').val("");
                setTimeout(function() {
                    $('#resetPassEmailIspis').html("");
                }, 7500);
            } else{
                setTimeout(function() {
                    $('#resetPassEmailIspis').html("");
                    $('#resetPassEmail').val("");
                }, 7500);
            }
        },
        error : function (xhr, status, error) {
            var ispis = "";
            try{
                var greske = xhr.responseJSON.errors;
                ispis = "<ul>";
                $.each(greske, function(key, value) {
                    ispis += "<li><h5>" + value + "</h5></li>";
                });
                ispis += "</ul>";
            } catch (e) {

            }
            $('#resetPassEmailIspis').html(ispis);
        }
    });
}

function resetLozinke() {
    var token = $('#promenaLozinkeKorisnikToken').val();
    var novaLozinka = $('#novaLozinka').val();
    var novaLozinka2 = $('#novaLozinka2').val();

    var promenaLozinkeGreske = [];
}

```

```

if(novaLozinka === ""){
    promenaLozinkeGreske.push("Polje za novu lozinku mora biti popunjeno");
    $('#novaLozinkaGreska').html("Polje za novu lozinku mora biti popunjeno");
} else if(novaLozinka.length < 6){
    promenaLozinkeGreske.push("Nova lozinka mora imati bar 6 karaktera");
    $('#novaLozinkaGreska').html("Nova lozinka mora imati bar 6 karaktera");
} else if((novaLozinka !== novaLozinka2) && (novaLozinka2 !== "") &&
(novaLozinka2.length >= 6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
} else{
    $('#novaLozinkaGreska').html("");
}

if(novaLozinka2 === ""){
    promenaLozinkeGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
    $('#novaLozinkaGreska2').html("Polje za ponovljenu lozinku mora biti
popunjeno");
} else if(novaLozinka2.length < 6){
    promenaLozinkeGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
    $('#novaLozinkaGreska2').html("Ponovljena lozinka mora imati bar 6
karaktera");
} else if((novaLozinka !== novaLozinka2) && (novaLozinka !== "") &&
(novaLozinka.length >= 6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
} else{
    $('#novaLozinkaGreska2').html("");
}

if(promenaLozinkeGreske.length == 0){
    var data =
    {
        token : token,
        novaLozinka : novaLozinka,
        novaLozinka2 : novaLozinka2
    };

    $.ajax({
        headers: {
            'X-CSRF-TOKEN' : $('meta[name="csrf-token"]').attr('content')
        },
        url : baseUrl + "/resetpassword",
        method : "PUT",
        dataType : "json",
        data : data,
        success : function (data) {
            $('#resetLozinkeGreske').html("<h4>" + data + "</h4>");
            if(data === "Uspešno ste promenili lozinku!"){
                $('#novaLozinka').val("");
                $('#novaLozinka2').val("");
                setTimeout(function() {
                    window.location.href = baseUrl + "/login";
                }, 5000);
            } else{

```

```

        setTimeout(function() {
            window.location.href = baseUrl;
        }, 5000);
    }
},
error : function (xhr, status, error) {
    var ispis = "";
    try{
        var greske = xhr.responseJSON.errors;
        ispis = "<ul>";
        $.each(greske, function(key, value) {
            ispis += "<li><h5>" + value + "</h5></li>";
        });
        ispis += "</ul>";
    } catch (e) {

    }
    $('#resetLozinkeGreske').html(ispis);
}
);
}

function promenaProfilneSlike() {
    var profilnaSlika = $('#profilnaSlika').val();
    var greske = [];

    if(profilnaSlika === ""){
        greske.push("Morate odabrati sliku");
        $('#promenaProfilneSlikeGreske').html("Morate odabrati sliku");
    } else{
        $('#promenaProfilneSlikeGreske').html("");
    }

    return (greske.length > 0) ? false : true;
}

function promenaLozinke() {
    var korisnikID = $('#promenaLozinkeKorisnik').val();
    var trenutnaLozinka = $('#trenutnaLozinka').val();
    var novaLozinka = $('#novaLozinka').val();
    var novaLozinka2 = $('#novaLozinka2').val();

    var promenaLozinkeGreske = [];

    if(trenutnaLozinka === ""){
        promenaLozinkeGreske.push("Polje za trenutnu lozinku mora biti popunjeno");
        $('#trenutnaLozinkaGreska').html("Polje za trenutnu lozinku mora biti popunjeno");
    } else if(trenutnaLozinka.length < 6){
        promenaLozinkeGreske.push("Trenutna lozinka mora imati bar 6 karaktera");
        $('#trenutnaLozinkaGreska').html("Trenutna lozinka mora imati bar 6 karaktera");
    } else{
        $('#trenutnaLozinkaGreska').html("");
    }

    if(novaLozinka === ""){
        promenaLozinkeGreske.push("Polje za novu lozinku mora biti popunjeno");
        $('#novaLozinkaGreska').html("Polje za novu lozinku mora biti popunjeno");
    } else if(novaLozinka.length < 6){
        promenaLozinkeGreske.push("Nova lozinka mora imati bar 6 karaktera");
    }
}

```

```

        $('#novaLozinkaGreska').html("Nova lozinka mora imati bar 6 karaktera");
    } else if((novaLozinka !== novaLozinka2) && (novaLozinka2 !== "") &&
(novaLozinka2.length >= 6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
} else{
    $('#novaLozinkaGreska').html("");
}

if(novaLozinka2 === ""){
    promenaLozinkeGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
    $('#novaLozinkaGreska2').html("Polje za ponovljenu lozinku mora biti
popunjeno");
} else if(novaLozinka2.length < 6){
    promenaLozinkeGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
    $('#novaLozinkaGreska2').html("Ponovljena lozinka mora imati bar 6
karaktera");
} else if((novaLozinka !== novaLozinka2) && (novaLozinka !== "") &&
(novaLozinka.length >= 6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne
poklapaju");
} else{
    $('#novaLozinkaGreska2').html("");
}

if(promenaLozinkeGreske.length == 0){
    var data = {
        korisnikID : korisnikID,
        trenutnaLozinka : trenutnaLozinka,
        novaLozinka : novaLozinka,
        novaLozinka2 : novaLozinka2
    };
    $.ajax({
        headers: {
            'X-CSRF-TOKEN' : $('meta[name="csrf-token"]').attr('content')
        },
        url : baseUrl + "/changepassword",
        method : "PUT",
        dataType : "json",
        data : data,
        success : function (data) {
            $('#promenaLozinkeGreske').html("<h4>" + data + "</h4>");

            if(data === "Uspešno ste promenili lozinku!"){
                $('#trenutnaLozinka').val("");
                $('#novaLozinka').val("");
                $('#novaLozinka2').val("");
                setTimeout(function() {
                    $('#promenaLozinkeGreske').html("");
                }, 7500);
            } else{
                setTimeout(function() {
                    $('#promenaLozinkeGreske').html("");
                }, 7500);
            }
        }
    });
}

```

```

        }
    },
    error : function (xhr, status, error) {
        var ispis = "";
        try{
            var greske = xhr.responseJSON.errors;
            ispis = "<ul>";
            $.each(greske, function(key, value) {
                ispis += "<li><h5>" + value + "</h5></li>";
            });
            ispis += "</ul>";
        } catch (e) {

        }
        $('#promenaLozinkeGreske').html(ispis);
    }
});

}

function contactProvera() {
    var contactEmail = $.trim($('#contactEmail').val());
    var poruka= $('#poruka').val();
    var reContactEmail = /^[\w\d]+[\._\w\d]*[\w\d]+\@[ \w]+([ \.][ \w]+)+$/;
    var contactGreske = [];

    if(contactEmail === ""){
        contactGreske.push("Polje za email mora biti popunjeno");
        $('#contactEmailGreska').html("Polje za email mora biti popunjeno");
    } else if(!reContactEmail.test(contactEmail)){
        contactGreske.push("Email nije u dobrom formatu");
        $('#contactEmailGreska').html("Email nije u dobrom formatu");
    } else{
        $('#contactEmailGreska').html("");
    }

    if(poruka === ""){
        contactGreske.push("Polje za poruku mora biti popunjeno");
        $('#contactPorukaGreska').html("Polje za poruku mora biti popunjeno");
    } else if(poruka.length < 15 || poruka.length > 500){
        contactGreske.push("Poruka mora imati 15-500 karaktera");
        $('#contactPorukaGreska').html("Poruka mora imati 15-500 karaktera");
    } else{
        $('#contactPorukaGreska').html("");
    }

    return (contactGreske.length > 0) ? false : true;
}

function addUserProvera(){
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var regLozinka = $('#regLozinka').val();
    var regLozinka2 = $('#regLozinka2').val();
    var regSlika = $('#regSlika').val();
    var uloga = $('#add_uloga').val();
    var aktivan = $('#add_aktivran').val();

    var proveraImePrezime = /^[A-ZŠĐĆČŽ][a-zšđćčž]{2,12}(\s[A-ZŠĐĆČŽ][a-
zšđćčž]{2,12}){0,1}$/;
    var proveraEmail = /[\w\d]+[\._\w\d]*[\w\d]+\@[ \w]+([ \.][ \w]+)+$/;
}

```

```

var proveraKorIme = /^[^\w\d\.\_\ ]{5,15}$/;
var regGreske = [];

if(regIme === ""){
    regGreske.push("Polje za ime mora biti popunjeno");
    $('#regImeGreska').html("Polje za ime mora biti popunjeno");
} else if(!proveraImePrezime.test(regIme)){
    regGreske.push("Ime nije u dobrom formatu");
    $('#regImeGreska').html("Ime nije u dobrom formatu");
} else{
    $('#regImeGreska').html("");
}

if(regPrezime === ""){
    regGreske.push("Polje za prezime mora biti popunjeno");
    $('#regPrezimeGreska').html("Polje za prezime mora biti popunjeno");
} else if(!proveraImePrezime.test(regPrezime)){
    regGreske.push("Prezime nije u dobrom formatu");
    $('#regPrezimeGreska').html("Prezime nije u dobrom formatu");
} else{
    $('#regPrezimeGreska').html("");
}

if(regEmail === ""){
    regGreske.push("Polje za email mora biti popunjeno");
    $('#regEmailGreska').html("Polje za email mora biti popunjeno");
} else if(!proveraEmail.test(regEmail)){
    regGreske.push("Email nije u dobrom formatu");
    $('#regEmailGreska').html("Email nije u dobrom formatu");
} else{
    $('#regEmailGreska').html("");
}

if(regKorIme === ""){
    regGreske.push("Polje za korisničko ime mora biti popunjeno");
    $('#regKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
} else if(!proveraKorIme.test(regKorIme)){
    regGreske.push("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži slova, brojeve, tačku i donju crtu");
    $('#regKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži slova, brojeve, tačku i donju crtu");
} else{
    $('#regKorImeGreska').html("");
}

if(regLozinka === ""){
    regGreske.push("Polje za lozinku mora biti popunjeno");
    $('#regLozinkaGreska').html("Polje za lozinku mora biti popunjeno");
} else if(regLozinka.length < 6){
    regGreske.push("Lozinka mora imati bar 6 karaktera");
    $('#regLozinkaGreska').html("Lozinka mora imati bar 6 karaktera");
} else if((regLozinka !== regLozinka2) && (regLozinka2 !== "") && (regLozinka2.length >= 6)){
    regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
    $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
    $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
} else{
    $('#regLozinkaGreska').html("");
}

if(regLozinka2 === ""){
    regGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
}

```

```

        $('#regLozinka2Greska').html("Polje za ponovljenu lozinku mora biti
popunjeno");
    } else if(regLozinka2.length < 6){
        regGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
        $('#regLozinka2Greska').html("Ponovljena lozinka mora imati bar 6 karaktera");
    } else if((regLozinka !== regLozinka2) && (regLozinka !== "") &&
(regLozinka.length >= 6)){
        regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#regLozinka2Greska').html("");
    }

if(regSlika === ""){
    regGreske.push("Morate odabrati sliku");
    $('#regSlikaGreska').html("Morate odabrati sliku");
} else{
    $('#regSlikaGreska').html("");
}

if(uloga === "0"){
    regGreske.push("Morate odabrati ulogu");
    $('#addUlogaGreska').html("Morate odabrati ulogu");
} else{
    $('#addUlogaGreska').html("");
}

if(aktivyan === "2"){
    regGreske.push("Morate odabrati aktivnost korisnika");
    $('#addAktivanGreska').html("Morate odabrati aktivnost korisnika");
} else{
    $('#addAktivanGreska').html("");
}

return (regGreske.length > 0) ? false : true;
}

function updateUserProvera(){
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var uloga = $('#add_uloga').val();
    var aktivyan = $('#add_aktivyan').val();

    var proveraImePrezime = /^[A-ZŠĐĆČŽ][a-zšđćčž]{2,12}(\s[A-ZŠĐĆČŽ][a-
zšđćčž]{2,12}){0,1}$/;
    var proveraEmail = /^[^\w\d]+[\.\_\w\d]*[\w\d]+\@[^\w]+([\.\_][\w]+)+$/;
    var proveraKorIme = /^[^\w\d\.\_]{5,15}$/;
    var regGreske = [];

    if(regIme === ""){
        regGreske.push("Polje za ime mora biti popunjeno");
        $('#regImeGreska').html("Polje za ime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regIme)){
        regGreske.push("Ime nije u dobrom formatu");
        $('#regImeGreska').html("Ime nije u dobrom formatu");
    } else{
        $('#regImeGreska').html("");
    }
}

```

```

if(regPrezime === ""){
    regGreske.push("Polje za prezime mora biti popunjeno");
    $('#regPrezimeGreska').html("Polje za prezime mora biti popunjeno");
} else if(!proveraImePrezime.test(regPrezime)){
    regGreske.push("Prezime nije u dobrom formatu");
    $('#regPrezimeGreska').html("Prezime nije u dobrom formatu");
} else{
    $('#regPrezimeGreska').html("");
}

if(regEmail === ""){
    regGreske.push("Polje za email mora biti popunjeno");
    $('#regEmailGreska').html("Polje za email mora biti popunjeno");
} else if(!proveraEmail.test(regEmail)){
    regGreske.push("Email nije u dobrom formatu");
    $('#regEmailGreska').html("Email nije u dobrom formatu");
} else{
    $('#regEmailGreska').html("");
}

if(regKorIme === ""){
    regGreske.push("Polje za korisničko ime mora biti popunjeno");
    $('#regKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
} else if(!proveraKorIme.test(regKorIme)){
    regGreske.push("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži slova, brojeve, tačku i donju crtu");
    $('#regKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera<br/>Sme da sadrži slova, brojeve, tačku i donju crtu");
} else{
    $('#regKorImeGreska').html("");
}

if(uloga === "0"){
    regGreske.push("Morate odabrati ulogu");
    $('#addUlogaGreska').html("Morate odabrati ulogu");
} else{
    $('#addUlogaGreska').html("");
}

if(aktivvan === "2"){
    regGreske.push("Morate odabrati aktivnost korisnika");
    $('#addAktivanGreska').html("Morate odabrati aktivnost korisnika");
} else{
    $('#addAktivanGreska').html("");
}

return (regGreske.length > 0) ? false : true;
}

function obrisiKorisnika(e) {
    if(confirm("Da li želite da obrišete korisnika?")){
        var id = $(this).data('id');
        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url : baseUrl + "/admin/korisnik/" + id,
            method : "DELETE",
            success : function (data) {
                alert(data);
                location.reload();
            },
            error : function (xhr, status, error) {

```

```

        alert("Greška pri brisanju korisnika, pokušajte kasnije");
        location.reload();
    }
})
e.preventDefault();
}

function addObjavaProvera() {
    var naslov = $('#addObjavaNaslov').val();
    var tekst = $('#addObjavaTekst').val();
    var kategorija = $('#addObjavaKategorija').val();
    var slika = $('#addObjavaSlika').val();
    var addObjavaGreske = [];

    if(naslov === ""){
        addObjavaGreske.push("Polje za naslov mora biti popunjeno");
        $('#addObjavaNaslovGreska').html("Polje za naslov mora biti popunjeno");
    } else if(naslov.length < 15 || naslov.length > 100){
        addObjavaGreske.push("Naslov mora imati 15-100 karaktera");
        $('#addObjavaNaslovGreska').html("Naslov mora imati 15-100 karaktera");
    } else{
        $('#addObjavaNaslovGreska').html("");
    }

    if(tekst === ""){
        addObjavaGreske.push("Polje za tekst mora biti popunjeno");
        $('#addObjavaTekstGreska').html("Polje za tekst mora biti popunjeno");
    } else if(tekst.length < 100 || tekst.length > 1000){
        addObjavaGreske.push("Tekst mora imati 100-1000 karaktera");
        $('#addObjavaTekstGreska').html("Tekst mora imati 100-1000 karaktera");
    } else{
        $('#addObjavaTekstGreska').html("");
    }

    if(kategorija === "0"){
        addObjavaGreske.push("Morate odabrati kategoriju");
        $('#addObjavaKategorijaGreska').html("Morate odabrati kategoriju");
    } else{
        $('#addObjavaKategorijaGreska').html("");
    }

    if(slika === ""){
        addObjavaGreske.push("Morate odabrati sliku");
        $('#addObjavaSlikaGreska').html("Morate odabrati sliku");
    } else{
        $('#addObjavaSlikaGreska').html("");
    }

    return (addObjavaGreske.length > 0) ? false : true;
}

function updateObjavaProvera() {
    var naslov = $('#updateObjavaNaslov').val();
    var tekst = $('#updateObjavaTekst').val();
    var kategorija = $('#updateObjavaKategorija').val();
    var updateObjavaGreske = [];

    if(naslov === ""){
        updateObjavaGreske.push("Polje za naslov mora biti popunjeno");
        $('#updateObjavaNaslovGreska').html("Polje za naslov mora biti popunjeno");
    } else if(naslov.length < 15 || naslov.length > 100){
        updateObjavaGreske.push("Naslov mora imati 15-100 karaktera");
        $('#updateObjavaNaslovGreska').html("Naslov mora imati 15-100 karaktera");
    }
}

```

```

        $( "#addObjavaNaslovGreska" ).html("Naslov mora imati 15-100 karaktera");
    } else{
        $( "#addObjavaNaslovGreska" ).html(" ");
    }

    if(tekst === ""){
        updateObjavaGreske.push("Polje za tekst mora biti popunjeno");
        $( "#addObjavaTekstGreska" ).html("Polje za tekst mora biti popunjeno");
    } else if(tekst.length < 100 || tekst.length > 1000){
        updateObjavaGreske.push("Tekst mora imati 100-1000 karaktera");
        $( "#addObjavaTekstGreska" ).html("Tekst mora imati 100-1000 karaktera");
    } else{
        $( "#addObjavaTekstGreska" ).html(" ");
    }

    if(kategorija === "0"){
        updateObjavaGreske.push("Morate odabrati kategoriju");
        $( '#addObjavaKategorijaGreska' ).html("Morate odabrati kategoriju");
    } else{
        $( '#addObjavaKategorijaGreska' ).html(" ");
    }

    return (updateObjavaGreske.length > 0) ? false : true;
}

function obrisiObjavu(e) {
    if(confirm("Da li želite da obrišete objavu?")){
        var id = $(this).data('id');
        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url : baseUrl + "/admin/objava/" + id,
            method : "DELETE",
            success : function (data) {
                alert(data);
                location.reload();
            },
            error : function (xhr,status,error) {
                alert("Greška pri brisanju objave, pokušajte kasnije");
                location.reload();
            }
        });
    }
    e.preventDefault();
}

function obrisiKomentar(e) {
    if(confirm("Da li želite da obrišete komentar?")){
        var id = $(this).data('id');
        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url : baseUrl + "/admin/komentar/" + id,
            method : "DELETE",
            success : function (data) {
                alert(data);
                location.reload();
            },
            error : function (xhr,status,error) {
                alert("Greška pri brisanju komentara, pokušajte kasnije");
                location.reload();
            }
        });
    }
}

```

```

        }
    });
    e.preventDefault();
}

function dodajKomentar(e) {
    var komentar = $.trim($('#dodajKomentarTextarea').val());
    var korisnikID = $('#dodajKomentarKorisnikID').val();
    var objavaID = $('#dodajKomentarObjavaID').val();
    var komentarGreska = [];

    if(komentar === ""){
        komentarGreska.push("Polje za komentar mora biti popunjeno");
        $('#komentar_greska').html("Polje za komentar mora biti popunjeno");
    } else if(komentar.length < 2 || komentar.length > 300){
        komentarGreska.push("Komentar mora imati 2-300 karaktera");
        $('#komentar_greska').html("Komentar mora imati 2-300 karaktera");
    } else{
        $('#komentar_greska').html("");
    }

    if(komentarGreska.length == 0){
        var data =
        {
            komentar : komentar,
            korisnikID : korisnikID,
            objavaID : objavaID
        };

        $.ajax({
            headers: {
                'X-CSRF-TOKEN' : $('meta[name="csrf-token"]').attr('content')
            },
            url : baseUrl + "/admin/komentar",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                alert(data);
                location.reload();
            },
            error : function (xhr, status, error) {
                var ispis = "";
                try{
                    var greske = xhr.responseJSON.errors;
                    ispis = "<ul>";
                    $.each(greske, function(key, value) {
                        ispis += "<li><h5>" + value + "</h5></li>";
                    });
                    ispis += "</ul>";
                } catch (e) {

                }
                $('#komentar_greska').html(ispis);
            }
        });
        e.preventDefault();
    }
}

function lajkovanjeProvera() {

```

```

        alert('Morate se prijaviti da bi lajkovali');
    }

    function lajkovanje(e) {
        var objavaID = $(this).data('id');
        var korisnikID = $('#dodajKomentarKorisnikID').val();

        var data =
        {
            korisnikID : korisnikID,
            objavaID : objavaID
        };
        $.ajax({
            headers: {
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
            },
            url : baseUrl + "/lajkovanje",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                if(data === ""){
                    location.reload();
                } else{
                    alert(data);
                    location.reload();
                }
            },
            error : function (xhr, status, error) {
                alert('Greška pri lajkovanju, pokušajte kasnije');
                location.reload();
            }
        });
        e.preventDefault();
    }

    function addUpdateKategorijaProvera() {
        var kategorija = $.trim($('#addKategorija').val());
        var kategorijaGreska = [];

        if(kategorija === ""){
            kategorijaGreska.push("Polje za naziv mora biti popunjeno");
            $('#addKategorijaGreska').html("Polje za naziv mora biti popunjeno");
        } else if(kategorija.length < 2 || kategorija.length > 20){
            kategorijaGreska.push("Naziv mora imati 2-20 karaktera");
            $('#addKategorijaGreska').html("Naziv mora imati 2-20 karaktera");
        } else{
            $('#addKategorijaGreska').html("");
        }

        return (kategorijaGreska.length > 0) ? false : true;
    }

    function obrisiKategoriju(e) {
        if(confirm("Da li želite da obrišete kategoriju?")){
            var id = $(this).data('id');
            $.ajax({
                headers: {
                    'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
                },
                url : baseUrl + "/admin/kategorija/" + id,
                method : "DELETE",
                success : function (data) {

```

```

        alert(data);
        location.reload();
    },
    error : function (xhr, status, error) {
        alert("Greška pri brisanju kategorije, pokušajte kasnije");
        location.reload();
    }
});
e.preventDefault();
}

function prikaziSveObjaveAjax() {
    var korisnikID = $('#objavepageKorisnikID').val();
    if(!korisnikID)
        korisnikID = 0;
    var unos = $('#searchInput').val();
    var kojaStranicaKlik = $(this).data('id');

    var sortiraj = $('#sortiraj').val();
    var kategorija = document.getElementsByName("kategorija_filter");
    var kategorijaNiz = [];

    for(let i=0; i<kategorija.length; i++)
    {
        if(kategorija[i].checked)
            kategorijaNiz.push(kategorija[i].value);
    }

    var kategorijeStr = " ";

    if(kategorijaNiz.length){
        kategorijeStr = kategorijaNiz.join(", ");
    }

    var data = {
        unos : unos,
        kategorijeStr : kategorijeStr,
        sortiraj : sortiraj
    };

    $.ajax({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        },
        url : baseUrl + "/objaveajax",
        data : data,
        method : "POST",
        dataType : "json",
        success : function (data) {

            if(!data.length){
                //ako nema objava

                var nemaObjava = "<h4 style='margin-right: 400px; margin-top: 200px;'>Nema objava</h4>";
                $('#objave_objavepage').html(nemaObjava);
                $('.straniceAjax').html("");

            } else{
                var kojaStranica = kojaStranicaKlik ? kojaStranicaKlik : 1;
                var poStranici = 3;
                var brojObjava = data.length;

```

```

    var brojStranica = Math.ceil(brojObjava/poStranici);
    var from = poStranici * (kojaStranica - 1);
    var doKog = ((from+poStranici) > brojObjava) ? brojObjava :
(from+poStranici);

    var ispisStranica = "";
    for(let i=0; i<brojStranica; i++)
    {
        ispisStranica += "<a href='#" data-id='"+ (i+1) +"'
class='straniceLinkAjax'>" + (i+1) + "</a>";

    }

    $('.straniceAjax').html(ispisStranica);
    $('.straniceLinkAjax').on('click', prikaziSveObjaveAjax);

    var ispis = "";
    for(let i=from; i<doKog; i++)
    {
        let tekst = data[i].tekst.substring(0, 80);
        let izmeni = "";
        let datum = new Date(data[i].objava_created_at * 1000);
        //datum.setHours(datum.getHours() - 1);
        //datum.setMonth(datum.getMonth() + 1);
        if(korisnikID){
            if(korisnikID == data[i].korisnikID)
                izmeni = "&nbsp;&nbsp;&nbsp; <a href=' " + baseUrl +
"/updateobjavauser/" + data[i].objavaID + "'>Izmeni</a>";
        }

        ispis += "<div class='objava_sadrzaj'>";
        ispis += "<div class='objava_drzac_objavepage'>";
        ispis += "<table class='objava_tabela'>";
        ispis += "<tr>";
        ispis += "<td colspan='3'><h5>" + data[i].naslov + "</h5></td>";
        ispis += "</tr>";
        ispis += "<tr>";
        ispis += "<td colspan='3'>";
        ispis += "<div class='objava_slika_div'>";
        ispis += "<img src=' " + baseUrl + "/images/objave/" + data[i].src +
"" alt=' " + data[i].alt + "' class='objava_slika_img' />";
        ispis += "</div>";
        ispis += "</td>";
        ispis += "</tr>";
        ispis += "<tr class='objava_detalji'>";
        ispis += "<td>";
        ispis += "Dodao/la: " + data[i].korisnicko_ime;
        ispis += "</td>";
        ispis += "<td>";
        ispis += "Postavljeno: " + datum.getDate() + "." + (datum.getMonth() +
1) + "." + datum.getFullYear() + ".";
        ispis += "</td>";
        ispis += "<td>";
        ispis += "Kategorija: " + data[i].kategorija;
        ispis += "</td>";
        ispis += "</tr>";
        ispis += "<tr>";
        ispis += "<td colspan='3'>";
        ispis += "<div class='objava_tekst'>";
        ispis += tekst + " ...";
        ispis += "</div>";

```

```

        ispis += "</td>" ;
        ispis += "</tr>" ;
        ispis += "<tr>" ;
        ispis += "<td colspan='3'>" ;
        ispis += "<a href='" + baseUrl + "/objava/" + data[i].objavaID
+ "/" + korisnikID + "' class='detaljnije'>PROČITAJ VIŠE</a>" ;
        ispis += izmeni ;
        ispis += "</td>" ;
        ispis += "</tr>" ;
        ispis += "</table>" ;
        ispis += "</div>" ;
        ispis += "</div>" ;
        ispis += "<br/><br/>" ;

        if(!(i == (doKog - 1)))
            ispis += "<div class='razvadanje_objava'></div>" ;

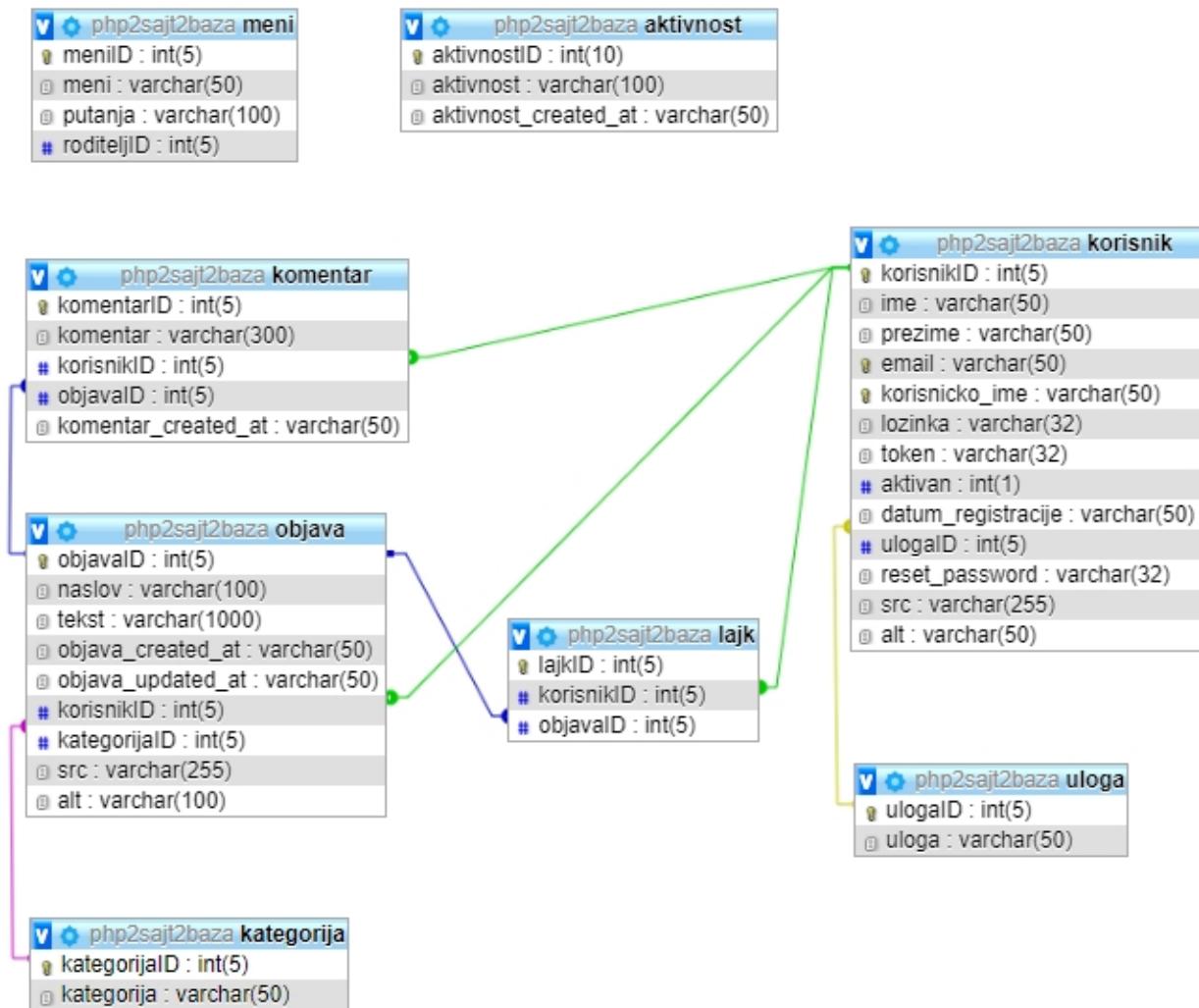
        ispis += "<br/><br/>" ;
    }
    $('#objave_objavepage').html(ispis) ;

    var sveStraniceLinkovi =
document.getElementsByClassName('straniceLinkAjax') ;
    for(let i=0; i<sveStraniceLinkovi.length; i++)
    {
        if(sveStraniceLinkovi[i].innerHTML == kojaStranica){
            sveStraniceLinkovi[i].style.border = "1px solid #7971ea" ;
            sveStraniceLinkovi[i].style.borderRadius = "5px" ;
        }
    }
}

error : function (xhr, status, error) {
}
}) ;
}
}

```

10. Baza podataka



- aktivnost

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	aktivnostID 🍕	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	aktivnost	varchar(100)	utf8_unicode_ci		No	None			Change Drop More
3	aktivnost_created_at	varchar(50)	utf8_unicode_ci		No	None			Change Drop More

- kategorija

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	kategorijaID	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	kategorija	varchar(50)	utf8_unicode_ci		No	None			Change Drop More

- komentar

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	komentarID	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	komentar	varchar(300)	utf8_unicode_ci		No	None			Change Drop More
3	korisnikID	int(5)			No	None			Change Drop More
4	objavaliD	int(5)			No	None			Change Drop More
5	komentar_created_at	varchar(50)	utf8_unicode_ci		No	None			Change Drop More

- korisnik

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	korisnikID	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	ime	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
3	prezime	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
4	email	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
5	korisnicko_ime	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
6	lozinka	varchar(32)	utf8_unicode_ci		No	None			Change Drop More
7	token	varchar(32)	utf8_unicode_ci		No	None			Change Drop More
8	aktivan	int(1)			No	None			Change Drop More
9	datum_registracije	varchar(50)	utf8_unicode_ci		No	None			Change Drop More
10	ulogalD	int(5)			No	None			Change Drop More
11	reset_password	varchar(32)	utf8_unicode_ci		No	None			Change Drop More
12	src	varchar(255)	utf8_unicode_ci		No	None			Change Drop More
13	alt	varchar(50)	utf8_unicode_ci		No	None			Change Drop More

- lajk

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	lajkID	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	korisnikID	int(5)			No	None			Change Drop More
3	objavaliD	int(5)			No	None			Change Drop More

- meni

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	meniID 	int(5)			No	None		AUTO_INCREMENT	 Change  Drop  More
2	meni	varchar(50)	utf8_unicode_ci		No	None			 Change  Drop  More
3	putanja	varchar(100)	utf8_unicode_ci		No	None			 Change  Drop  More
4	roditeljID	int(5)			No	None			 Change  Drop  More

- objava

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	objavald 	int(5)			No	None		AUTO_INCREMENT	 Change  Drop  More
2	naslov	varchar(100)	utf8_unicode_ci		No	None			 Change  Drop  More
3	tekst	varchar(1000)	utf8_unicode_ci		No	None			 Change  Drop  More
4	objava_created_at	varchar(50)	utf8_unicode_ci		No	None			 Change  Drop  More
5	objava_updated_at	varchar(50)	utf8_unicode_ci		No	None			 Change  Drop  More
6	korisnikID 	int(5)			No	None			 Change  Drop  More
7	kategorijaID 	int(5)			No	None			 Change  Drop  More
8	src	varchar(255)	utf8_unicode_ci		No	None			 Change  Drop  More
9	alt	varchar(100)	utf8_unicode_ci		No	None			 Change  Drop  More

- uloga

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ulogalID 	int(5)			No	None		AUTO_INCREMENT	 Change  Drop  More
2	uloga	varchar(50)	utf8_unicode_ci		No	None			 Change  Drop  More