

Visoka škola strukovnih studija za  
informacione i komunikacione  
tehnologije

## **POSTMOON**

<https://postmoon.000webhostapp.com/index.php>

Smer: Internet tehnologije  
Modul: Web programiranje  
Predmet: Web programiranje PHP 1

Nemanja Jezdic 30/19

# Sadržaj

## 1. Uvod

### 1.1 Korisćene tehnologije

### 1.2 Opis funkcionalnosti

### 1.3 Template

## 2. Organizacija

### 2.1 Organizacija sajta

### 2.2 Opis strane sajta

## 3. Kodovi

### 3.1 PHP

### 3.2 CSS

### 3.3 JavaScript

# **1 . Uvod**

## **1.1 Korisćene tehnologije**

- HTML 5
- PHP & MySQL
- CSS 3
- Bootstrap
- JavaScript
- jQuery
- Xampp
- Visual Studio code

## **1.2 Opis funkcionalnosti**

- Dinaičko ispisivanje navigacije,opcija za izbor drzave,kategorija I svih ostalih neophodnih stavki,sve iz baze.
- Kompletna registracija I logovanje sa klijentskom I serverskom validacijom.Svaka validacija uredjena regexima.Radjeno AJAX-om.
- Da bi korisnik video sadrzaj,mora biti autorizovan,tacnije mora da se uloguje.Svakim pokusajem neovlascenog prisutpa vrsi se redirekcija.
- Kreiranje posta.
- Dinamicki prikazani postovi koji su krejirali korisnici povuceno iz baze.
- Svaki post ima opis,sliku,kategoriju kojoj pripada,ime I prezime korisnika koji je napravio taj post,broj korisnika koji su lajkovali post kao I broj komentara na tom postu.
- Mogucnost lajkovanja u vidu ankete.Svaki korisnik moze samo jednom da lajkuje jedan post.
- Svaki post ima link ka comment sekciji gde se vrsi dodavanje komentara na taj post I pregled komentara drugih korisnika.
- Svaki korisnik moze da izbriše svoj komentar ili svoj post.

- Kompletna funkcionalnost filtriranja, pretrage i paginacije postova, sve uskladjeno, sa dinamickim prikazom iz baze.
- Editovanje profila korisnika
- Kompletan Admin Panel kome moze pristupiti samo admin, i kompletna funkcionalnost editovanja i brisanja korisnika, postova i kategorija. Kao i dodavanje novih kategorija.

### **1.3 Template**

Nije koriscen gotov template.

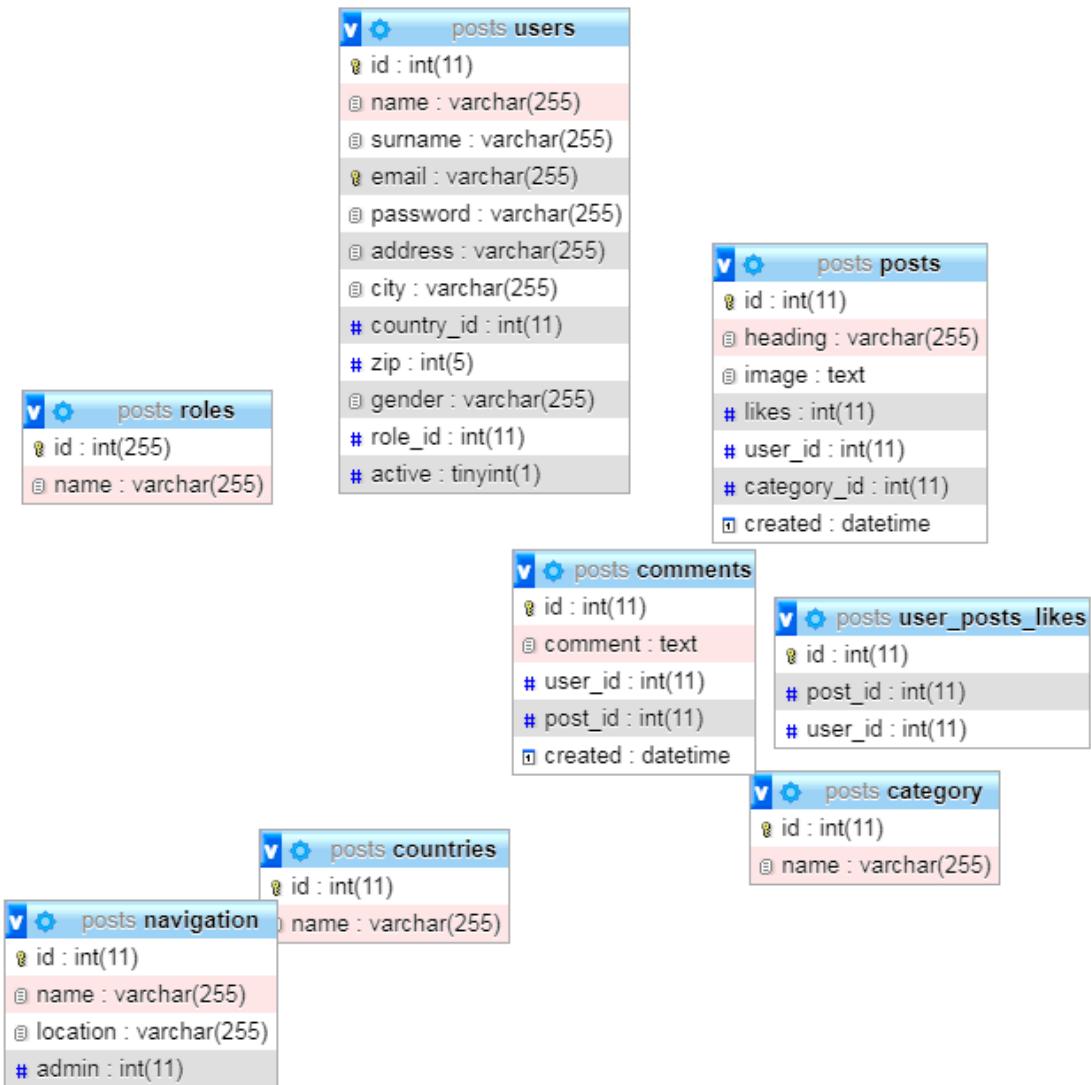
## **2 . Organizacija**

### **2.1 Organizacija sajta**

Skoro svim stranicama moze da se pristupi od svake.

Kada to nije slucaj:

- 1.Na index.php mogu pristupiti samo neautorizovani korisnici. Korisnik postaje autorizovan kad se uloguje i tad se vrsti redirekcija na posts.php.
- 2.Da bi pristupio ka ostalim stranicama korisnik mora biti autorizovan.
- 3.Stranici adminpanel.php moze pristupiti samo admin sa bilo koje stranice koja trazi autorizaciju. Kao i na stranicama gde admin vrsti izmene.



Baza:Nisu upisane veze u mysql,ali je ispostovana sva normalizacija upitima u kodu(spajanje tabela i brisanje iz tabela).

## 2.2 Mapa sajta

```
sitemap.xml X

sitemap.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <urlset
3  |   xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
4  |   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5  |   xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
6  |   http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
7
8
9  <url>
10 |   <loc>https://postmoon.000webhostapp.com/</loc>
11 |   <lastmod>2021-04-23</lastmod>
12 |   <priority>1.00</priority>
13 </url>
14 <url>
15 |   <loc>https://postmoon.000webhostapp.com/index.php</loc>
16 |   <lastmod>2021-04-23</lastmod>
17 |   <priority>0.80</priority>
18 </url>
19 <url>
20 |   <loc>https://postmoon.000webhostapp.com/author.php</loc>
21 |   <lastmod>2021-04-23</lastmod>
22 |   <priority>0.80</priority>
23 </url>
24 <url>
25 |   <loc>https://postmoon.000webhostapp.com/Documentation.pdf</loc>
26 |   <lastmod>2021-04-22T19:12:51+00:00</lastmod>
27 |   <priority>0.80</priority>
28 </url>
29
30
31 </urlset>
```

## 2.3 Opis strane sajta

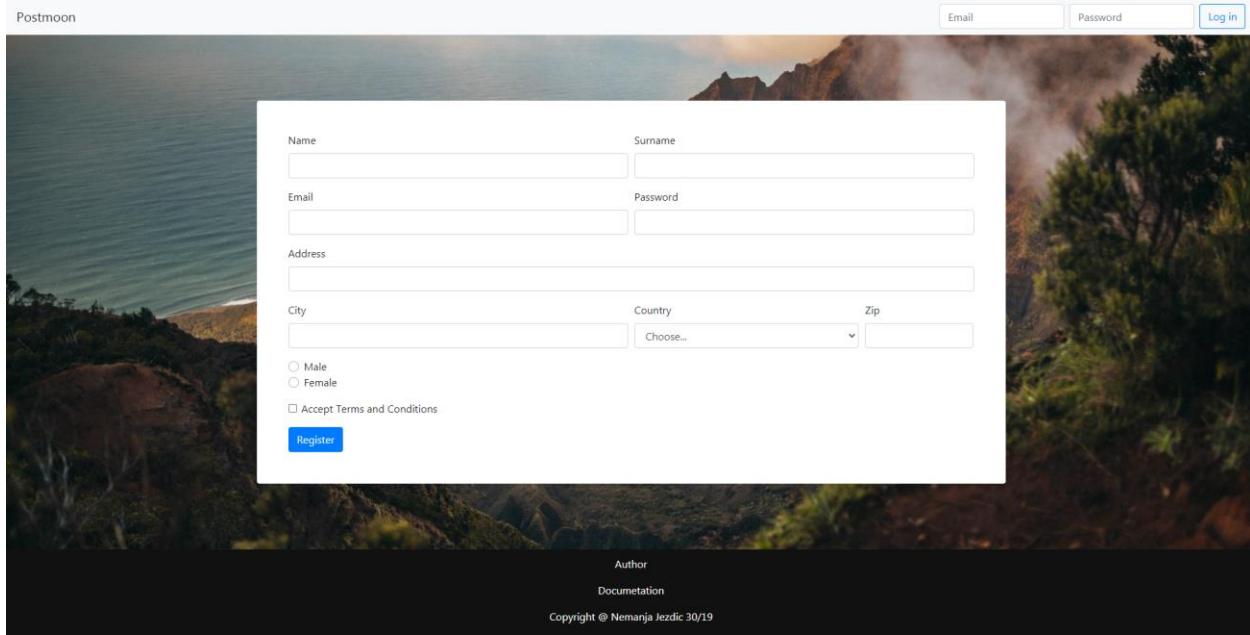
Na svim mestima gde ima bilo kakav vid forme je uradjena I serverska I klijentska validacija sa regexima.  
Index.php

Na ovoj stranici se vrši registracija i logovanje.Ukoliko korisnik nije registrovan ne može da se uloguje,ukoliko nije ulogovan ne može da posecuje ostale stranice.Kada se korisnik uloguje,vise ne može da posecuje index.php vec je automatski redirektovan na posts.php,index.php može da poseti tek kad se izloguje.

Registracija i logovanje je radjeno AJAX-om.

Registracija:Sva polja moraju biti popunjena ispravnim sadržajem.Ukoliko se submituje forma i neko polje nije ispravno popunjeno ispod tog polja pise kako treba da se popuni (kao i za svaku narednu formu sa klijentske strane).Ako su sva polja ispravno popunjeno i submituje se forma,vrsi se upis u bazu novog korisnika i sa servera se dobija odgovor da je korisnik uspesno registrovan.Ukoliko korisnik unese email koji vec postoji u bazi,korisnik nece biti upisan i dobija se poruka od servera da email vec postoji u bazi.

Logovanje:Korisnik unosi podatke za email i sifru.Pritiskom na Log in vrsi se provera dali postoji zadati email u bazi,kao i da li zadata sifra odgovara tom emailu.Ukoliko ne vazi neki od ovih slučajeva dobija se poruka da treba ispravno popuniti email i password.Ukoliko su ispunjeni slučajevi,korisnik je ulogovan i pocinje njegova sesija.



## posts.php i category.php

(ove dve stranice sam ovako stavio jer su slicne po izgledu i imaju povezane funkcionalnosti)

posts.php se odnosi na sve postove iz baze,dok se category.php odnosi za sve postove iz baze koje pripadaju izabranoj kategoriji.

Navigacija: navigacija je ispisana dinamicki iz baze podataka.Link ka Admin Panel u navigaciji ce biti dinamicki isписан samo ukoliko je ulogovani korisnik admin.Na link Log out vrsi se prekidanje sesije

ulogovanog korisnika I redirekcija na index.php gde se trazi ponovno logovanje.Polje Search ce biti vidljivo samo na stranicama posts.php I category.php gde se vrsti pretraga postova.

Lista kategorija je ispisana dinamicki iz baze.

Na posts.php:Dinamicki je ispisani svaki post iz baze.

Na category.php:Dinamicki je ispisani svaki post iz baze koji pripada kategoriji koja je izabrana.  
(Prvo se prikazuju postovi sa najskorijim datumom kreiranja).

Paginacija: menjaju se dinamicki broj stranica u zavisnosti od broja postova.Na svakoj stranici su dinamicki ucitana po 2 posta po pravilnom redosledu iz baze.

Pretraga: vrsti se pretraga postova po naslovu posta I njihov dinamicki ispis.Ukoliko je polje u search-u praznu nece se vrsiti pretraga.

Na posts.php:Paginacija I Pretraga su uskladjeni.

Na category.php:Paginacija I Pretraga,kao I Filtriranje po kategoriji su uskladjeni.

Svaki post sadrzi:

(svi podaci dinamicki ispisani,sa nekim podacima koji su ucitani ili upisani u drugim tabalema u bazi)

1.sliku posta

2.naslov posta

3.kategoriju u kojoj dati post pripada

4.Ime I prezime korisnika koji je napravio post

5.broj lajkova na postu I opciju za lajk(korisnik moze da lajkuje samo jednom,broj lajkova se cita iz baze kao sto se I odgovor korisnika cuva u bazi)

6.broj komentara vezan za taj post I link ka comment sekciji(post.php).

7.link za delete posta ukoliko je ulogovani korisnik napravio taj post.

[Funny](#)  
[Music](#)  
[Awesome](#)  
[TV](#)**Norway**

Awesome

(Created by: Petronije Petronijevic)

[Delete your post](#)

Likes: 1 Comments:(1)

**Any show recomendetion**

TV

(Created by: Nemanja Jokic)

Likes: 0 Comments:(0)

[1](#) [2](#)

Author

Documentation

Copyright © Nemanja Jezdic 30/19

## post.php

Na stranicu se dolazi preko linka za comment na svakom postu i dinamicki se filtrira taj post i komentari koji su vezani za taj post(sve iz baze). Post sadrzi sve iste podatke kao u prethodnoj stavci.

Korisnik moze da upise vise komentara koji se upisuju u bazu.

Svaki komentar sadrzi:

- 1.Datum kad je kreiran
  - 2.Ime i prezime user-a koji je napisao taj komentar
  - 3.tekst koji je user napisao
  - 4.x za brisanje komentara(ovo vidi samo onaj user koji je napisao taj komentar).
- Komentari su ispisani iz baze po najskorijem datumu.

The screenshot shows a social media interface. At the top, there's a navigation bar with links: Postmoon, New Post, Profile, Admin Panel, and Log out. Below the navigation, there are category links: Funny, Music, Awesome, and TV. The main content area features a post with the title "Where are our dessssks". The post includes a photograph of a man in a flight suit standing at a whiteboard, with two long snakes crawling across the floor in front of him. Below the photo, the user "tiger-in-the-flightdeck" has commented: "The lack of context here is thrilling". Another user, "mark-watney-spacepirate", has responded with "introductory python programming course". This response is part of a larger thread under the heading "introductory python programming course". The thread includes a comment from "Class" and another from "Funny". A note indicates "(Created by: Petronije Petronijevic)". The post has received 265 likes. There is a comment input field with placeholder "Comment here..." and a "Send" button. Below the comment section, there are two older comments: one from "Nemanja Jokic" dated 22 Apr 2021 17:18:35, and another from "Petronije Petronijevic" dated 13 Mar 2021 17:47:35. The bottom of the page contains footer text: "Author", "Documentation", and "Copyright © Nemanja Jezdic 30/19".

## newpost.php

Stranica za kreiranje novog posta.Post se kreira tako sto se unese naslov,I izabere kategorija u kojoj ce se post nalaziti.U dropdown listi su dinamicki upisane kategorije iz baze.Odradjeni je upload fajla koji cuva podatke u bazi i cuva slike u poseban fajlu.Slika mora biti u odredjenim formatima.Ako je prosla

validacija post je kreiran I dinamicki ispisani u posts.php gde se vrsti redirekcija I izbacuje se poruka o uspesnom kreiranju I post jedinamicki ispisani I u category.php u onoj kategoriji koja je izabrana.

Postmoon   New Post   Profile   Admin Panel   Log out

Heading

Category  
 Choose...

Upload post image  
 Choose File   No file chosen

Author  
Documentation  
Copyright © Nemanja Jezdic 30/19

## profile.php

Stranica sa podacima korisnika koji je trenutno ulogovan kao I forma za njihovo menjanje.  
Korisnikovi podaci su dinamicki upisani iz baze,ti podaci su ujedno I upisani u formu za editovanje profila kako bi korisnik promenio jednu stavku ako nece sve,da ne bi popunjavao celu formu.  
Cim korisnik submituje formu,ti podaci su promenjeni u bazi.

Postmoon   New Post   Profile   Admin Panel   Log out

Successfully edited profile

Edit your profile:

Name  
 Petronije

Surname  
 Petronijevic

City  
 Baku

Zip  
 11000

Address  
 Petrova 2

Country  
 Azerbaijan

Author  
Documentation  
Copyright © Nemanja Jezdic 30/19

## adminpanel.php

Ovoj stranici moze da pristupi samo admin,svako drugi bice redirektovan.Admin na ovoj stranici vrsi brisanje I izmenu korisnika,postova I kategorija,kao I dodavanje kategorija.Dinamicki su ispisani podaci iz baze u vidu tabele za korisnike,postove I kategorije.Vrsi se filtriranje tabele u zavisnosti od izbora.Svaki

red tabele sadrzi podatke za svakog korisnika,posta ili kategoriju,na kraju svakog reda se nalaze linkovi za izmenu i brisanje.Kod kategorija se nalazi i link za dodavanje nove kategorije.

Brisanje : Vrsi se brisanje izabrane stavke iz baze

Izmene:vodi na zasebne stranice gde su dinamicki vec upisani podaci iz baze u formi,I promenom nekom od ovih stavki vrsi se izmena tih stavki u bazi na submit.

Dodavanje nove kategorije:upisuje se nova kategorija u bazu.

	<a href="#">Postmoon</a>	<a href="#">New Post</a>	<a href="#">Profile</a>	<a href="#">Admin Panel</a>	<a href="#">Log out</a>			
	<a href="#">Users</a>	<a href="#">Posts</a>	<a href="#">Category</a>					
	ID	name	surname	email	address	city	country	zip
	26	Petronije	Petronijevic	per@gmail.com	Petrova 2	Baku	Azerbaijan	11000
	1	Nemanja	Jezdic	nemanja@gmail.com	bulevar 23	Belgrade	Afghanistan	11000
	2	Aleksa	Jovanovic	malijova@gmail.com	ispod pecurke	Mirijevo	Bahamas	11000
	3	Pera	Peric	pera@gmail.com	asddw	Belgrade	Austria	11000
	5	Vucko	Bakic	vuk@gmail.com	Bastovanska 13b	Surcin	Afghanistan	12000
	8	Nikola	Nikic	dzoni@gmail.com	Dzonijeva	Belgrade	Afghanistan	11000
	13	Nemanja	Jokic	nemke@gmail.com	asdasd	Beasdsa	Afghanistan	11000
	14	Nemke	Jokic	nemked@gmail.com	asdasd	Beasdsa	Afghanistan	11000
	15	Nemke	Jokic	neca@gmail.com	asdasd	Beasdsa	Afghanistan	11000
	17	Nevena	Jokic	nevena@gmail.com	asdasd	Beasdsa	Afghanistan	11000
	23	Nemanja	Jezdic	nemanja.jezdic@yahoo.com	asdasd	Beasdsa	Afghanistan	11000
	25	Nemanja	Jezdic	nemanja.jezdic2@gmail.com	Bulevar 3	Belgrade	Afghanistan	11000
	25	Nemanja	Jezdic	nemanja.jezdic2@gmail.com	Bulevar 3	Belgrade	Afghanistan	11000
	27	Pera	Petronije	petronije@gmail.com	sdsad	Paris	Afghanistan	15000
	29	Nemanja	Jokic	nem@gmail.com	Bulevar 2	Belgrade	Serbia	11000
	30	Mitar	Miric	mitrivoje@gmail.com	Mitrina 23	Chicago	Bahrain	12345

Author  
Documentation  
Copyright © Nemanja Jezdic 30/19

	<a href="#">Users</a>	<a href="#">Posts</a>	<a href="#">Category</a>							
	ID	heading	image	likes	name	surname	email	category	created	
	4	Class		265	Petronije	Petronijevic	per@gmail.com	Funny	0000-00-00 00:00:00	
	10	Love this vibe		0	Nemanja	Jokic	nem@gmail.com	Music	2021-03-14 17:46:37	
	11	Any show recomendation		0	Nemanja	Jokic	nem@gmail.com	TV	2021-03-14 17:48:14	
	13	Norway		1	Petronije	Petronijevic	per@gmail.com	Awesome	2021-03-14 18:17:51	

ading	image	likes	name	surname	email	category	created	actions
Class		265	Petronije	Petronijevic	per@gmail.com	Funny	0000-00-00 00:00:00	<a href="#">Edit</a> <a href="#">Delete</a>
this vibe		0	Nemanja	Jokic	nem@gmail.com	Music	2021-03-14 17:46:37	<a href="#">Edit</a> <a href="#">Delete</a>
/ show iendition		0	Nemanja	Jokic	nem@gmail.com	TV	2021-03-14 17:48:14	<a href="#">Edit</a> <a href="#">Delete</a>
orway		1	Petronije	Petronijevic	per@gmail.com	Awesome	2021-03-14 18:17:51	<a href="#">Edit</a> <a href="#">Delete</a>

Users	ID	name	actions
Posts	1	Funny	<a href="#">Edit</a> <a href="#">Delete</a>
Category	3	Music	<a href="#">Edit</a> <a href="#">Delete</a>
	4	Awesome	<a href="#">Edit</a> <a href="#">Delete</a>
	5	TV	<a href="#">Edit</a> <a href="#">Delete</a>

[Add new Category](#)

## author.php

Nikakva funkcionalnost se ne nalazi ovde.

Postmoon



I would describe myself as positive hard working person that's ready for new challenges. So far i'm skilled in HTML,CSS,Bootstrap,C,Java,C#,JavaScript,Jquery,AJAX and i'm also trying to improve by learning new technologies.

Author

Documentation

Copyright © Nemanja Jezdic 30/19

## 3 . Kodovi

### 3.1 PHP

#### connection.php

```
connection.php X
assets > config > connection.php > ...
1  <?php
2
3  define("SERVER", "localhost");
4  define("DATABASE", "posts");
5  define("USERNAME", "root");
6  define("PASSWORD", "");
7
8  try {
9      $conn = new PDO("mysql:host=".SERVER.";dbname=".DATABASE.";charset=utf8", USERNAME, PASSWORD);
10
11     $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
12
13     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14 }
15 catch(PDOException $ex){
16     echo "Couldn't connect to database";
17 }
```

## error.php

```
error.php X  
assets > error > error.php > ...  
1  <?php if (session_status() === PHP_SESSION_NONE) {  
2      session_start();  
3  }  
4  ?>  
5  <?php if(!empty($_SESSION["error"])):?>  
6  
7  <div class="alert alert-danger">  
8      <p><?php  
9          echo $_SESSION["error"];  
10         unset($_SESSION["error"]);  
11     ?></p>  
12 </div>  
13  
14  
15 <?php endif;?>  
16
```

## success.php

```
success.php X  
assets > success > success.php > div.alert.alert-success > p  
1  <?php if (session_status() === PHP_SESSION_NONE) {  
2      session_start();  
3  }  
4  ?>  
5  <?php if(!empty($_SESSION["success"])):?>  
6  
7  <div class="alert alert-success">  
8      <p><?php  
9          echo $_SESSION["success"];  
10         unset($_SESSION["success"]);  
11     ?></p>  
12 </div>  
13  
14  
15 <?php endif;?>
```

## header.php

```
header.php x
header.php > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <meta name="viewport" content="initial-scale=1, maximum-scale=1">
8      <title>Postmoon</title>
9      <meta name="keywords" content="Postmoon,Post,Category,User">
10     <meta name="description" content="Postmoon is a website where all loged Users can post their funny posts,see posts from other users,see them in different category and so on.">
11     <meta name="author" content="Nemanja Jezdic">
12     <link rel="shortcut icon" type="image/x-icon" href="/assets/img/post.png"/>
13     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDP" />
14     <link rel="stylesheet" href="/assets/css/style.css" type="text/css"/>
15  </head>
16 </body>
```

## footer.php

```
footer.php x
footer.php > footer.footerbg > p.textfooter > a.text-light
1  <footer class="footerbg">
2      <p class="textfooter"><a href="/author.php" class="text-light">Author</a></p>
3      <p class="textfooter"><a href="/documentation.png" class="text-light">Documentation</a></p>
4      <p class="textfooter">Copyright © Nemanja Jezdic 30/19</p>
5  </footer>
6
7
8
9
10
11     <script src="https://kit.fontawesome.com/4a4a3bed48.js" crossorigin="anonymous"></script>
12     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
13     <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7ywK8fhQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrth/tE3MoK7ZeDy" />
14     <script src="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.js"></script>
15     <script src="/assets/js/main.js"></script>
16     <script src="/assets/js/post.js"></script>
17 </body>
18 </html>
```

## index.php

```
index.php x
index.php > nav.navbar.navbar-expand-lg.navbar-light.bg-light > a.navbar-brand
1  <?php session_start();
2  if(empty($_SESSION["user"])){
3      header("Location:/posts.php");
4      exit();
5  }
6
7  include "assets/config/connection.php";
8
9  $drzave=$conn->query("SELECT * FROM countries")->fetchAll();
10
11 ?>
<?php include "header.php";?>
12
13
14     <nav class="navbar navbar-expand-lg navbar-light bg-light">
15         <a class="navbar-brand" href="index.php">Postmoon</a>
16         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
17             <span class="navbar-toggler-icon"></span>
18         </button>
19
20         <div class="collapse navbar-collapse" id="navbarSupportedContent">
21             <ul class="navbar-nav mr-auto" id="navigation">
22                 </ul>
23                 <form action="" method="POST" class="form-inline my-2 my-lg-0">
24                     <input class="form-control mr-sm-2" type="text" id="logemail" placeholder="Email" aria-label="Search">
25                     <input class="form-control mr-sm-2" type="password" id="logpass" placeholder="Password" aria-label="Search">
26                     <button class="btn btn-outline-primary my-2 my-sm-0" type="submit" id="btnLogin">Log in</button>
27                 </form>
28             </div>
29         </div>
30     </nav>
31
32     <div class="backgroundPhoto">
33         <div class="container p-5 rounded regholder">
34             <form action="" method="POST" class="needs-validation">
35                 <div class="form-row">
36                     <div class="form-group col-md-6">
37                         <label for="inputName">Name</label>
38                         <input type="text" class="form-control" id="inputName" value="Nemanja Jezdic" required="required" aria-describedby="inputNameHelp" aria-label="Name">
39                         <p id="registerName" class="text-danger">Minimum 3 letters with first capital letter</p>
40                     </div>
41                     <div class="form-group col-md-6">
42                         <label for="inputSurname">Surname</label>
43                         <input type="text" class="form-control" id="inputSurname" value="Jezdic" required="required" aria-describedby="inputSurnameHelp" aria-label="Surname">
44                         <p id="registerSurname" class="text-danger">Minimum 3 letters with first capital letter</p>
45                     </div>
46                 </div>
47                 <div class="form-row">
48                     <div class="form-group col-md-6">
49                         <label for="inputEmail4" class="form-label">Email</label>
```

```
index.php ×
index.php > nav.navbar.navbar-expand-lg.navbar-light.bg-light > a.navbar-brand
48   <div class="form-group col-md-6">
49     <label for="inputEmail4">Email</label>
50     <input type="email" class="form-control" id="inputEmail4">
51     <p id="registerEmail" class="text-danger">Example:pera@gmail.com</p>
52   </div>
53   <div class="form-group col-md-6">
54     <label for="inputPassword4">Password</label>
55     <input type="password" class="form-control" id="inputPassword4">
56     <p id="registerPassword" class="text-danger">Minimum 2 characters without speacial characters</p>
57   </div>
58 </div>
59 <div class="form-group">
60   <label for="inputAddress">Address</label>
61   <input type="text" class="form-control" id="inputAddress">
62   <p id="registerAddress" class="text-danger">Can contain numbers and letters</p>
63 </div>
64 <div class="form-row">
65   <div class="form-group col-md-6">
66     <label for="inputCity">City</label>
67     <input type="text" class="form-control" id="inputCity">
68     <p id="registerCity" class="text-danger">Minimum 3 letters with first capital letter</p>
69   </div>
70   <div class="form-group col-md-4">
71     <label for="inputCountry">Country</label>
72     <select id="inputCountry" class="form-control">
73       <option selected value="0">Choose...</option>
74       <?php foreach($drzave as $dr):?>
75         <option value="<?php echo $dr->id;?><?php echo $dr->name;?>"><?php endforeach;?>
76       <?php endforeach;?>
77     </select>
78     <p id="registerCountry" class="text-danger">Please choose country</p>
79   </div>
80   <div class="form-group col-md-2">
81     <label for="inputZip">Zip</label>
82     <input type="text" class="form-control" id="inputZip">
83     <p id="registerZip" class="text-danger">Exactly 5 numbers</p>
84   </div>
85 </div>
86 <div class="custom-control custom-radio">
87   <input type="radio" id="customRadio1" name="customRadio" class="custom-control-input" value="Male">
88   <label class="custom-control-label" for="customRadio1">Male</label>
89 </div>
90 <div class="custom-control custom-radio">
91   <input type="radio" id="customRadio2" name="customRadio" class="custom-control-input" value="Female">
92   <label class="custom-control-label" for="customRadio2">Female</label>
93 </div>
94   <div class="alert alert-danger valradio">Please pick your gender </div>
95   <div class="form-group mt-3">
96     <div class="form-check">
```

```

96   <div class="form-check">
97     <input class="form-check-input" name="customCheck" type="checkbox" id="gridCheck" value="accept">
98     <label class="form-check-label" for="gridCheck">
99       | Accept Terms and Conditions
100    </label>
101  </div>
102 </div>
103 <div class="alert alert-danger valcheck">Please agree with our terms</div>
104 <button type="submit" id="btnRegister" class="btn btn-primary">Register</button>
105 <div id="odgovor">
106
107 </div>
108 </form>
109 </div>
110 </div>
111 <?php include "footer.php";?>
113

```

## nav.php

```

nav.php ×
nav.php > nav.navbar.navbar-expand-lg.navbar-light.bg-light > div#navbarSupportedContent.collapse.navbar-collapse
1  <?php
2  $navigation = $conn->query("SELECT * FROM navigation")->fetchAll();
3  ?>
4
5  <nav class="navbar navbar-expand-lg navbar-light bg-light">
6    <a class="navbar-brand" href="/posts.php">Postmoon</a>
7    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
8      <span class="navbar-toggler-icon"></span>
9    </button>
10
11   <div class="collapse navbar-collapse" id="navbarSupportedContent">
12     <ul class="navbar-nav mr-auto" id="navigation">
13       <?php foreach ($navigation as $nav) : ?>
14         <?php
15           if (!empty($_SESSION["user"]) && $_SESSION["user"]->role_id == "1" && $nav->admin == 1) :
16             ?>
17             <li class="nav-item">
18               <a class="nav-link" href="/<?php echo $nav->location?>"><?php echo $nav->name?></a>
19             </li>
20           <?php
21           endif;
22         ?>
23         <?php
24           if ($nav->admin == 0) :
25             ?>
26             <li class="nav-item">
27               <a class="nav-link" href="/<?php echo $nav->location?>"><?php echo $nav->name?></a>
28             </li>
29           <?php
30           endif;
31         ?>
32       <?php endforeach; ?>
33     </ul>
34   <?php
35   if($_SERVER["PHP_SELF"]=="/posts.php" || $_SERVER["PHP_SELF"]=="/category.php"):
36   ?>
37   <form action="" method="GET" class="form-inline my-2 my-lg-0">
38     <?php if(!empty($kategorijaid)):>
39     <input type="hidden" id="id" name="id" value="<?php echo $kategorijaid ?>"/>
40     <?php endif;?>
41     <input class="form-control mr-sm-2" type="search" id="search" name="search" placeholder="Search" aria-label="Search" value="<?php
42       echo isset($_GET["search"]) ? $_GET["search"] : "";
43     ?>">
44     <button type="submit" class="btn btn-primary fas fa-search p-2"></button>
45   </form>
46   <?php
47   endif;
48   ?>
49 </div>

```

```

value="<?php
echo isset($_GET["search"]) ? $_GET["search"] : "";
?>">

```

## logout.php

```
logout.php X
logout.php
1 <?php session_start();
2
3     if(!empty($_SESSION["user"])){
4         unset($_SESSION["user"]);
5     }
6     header("Location:/index.php");
7     exit();
8 ?>
```

## posts.php

```
posts.php X
posts.php > ...
1 <?php session_start();
2 if (empty($_SESSION["user"])) {
3     header("Location:/index.php");
4     exit();
5 }
6
7 include "assets/config/connection.php";
8 $number_of_posts = 0;
9 $posts_per_page = 2;
10
11 if (empty($_GET["page"])) {
12     $page = 1;
13 } else {
14     $page = $_GET["page"];
15 }
16 $kategorie_id="";
17 if (!empty($_GET["search"])) {
18     $number_of_post_search = $conn->prepare("SELECT COUNT(*) FROM posts
19                                         WHERE heading LIKE :search");
20     $search="%".$_GET["search"]."%";
21     $number_of_post_search->bindParam(":search",$search);
22     $number_of_post_search->execute();
23     $number_of_posts = $number_of_post_search->fetchColumn();
24
25     $this_page = ($page - 1) * $posts_per_page;
26     $number_of_pages = ceil($number_of_posts / $posts_per_page);
27     if (!empty($_GET["page"])) && ($_GET["page"] > $number_of_pages || $_GET["page"] < 0)) {
28         header("Location:/posts.php?page=1");
29         exit();
30     }
31
32     $resul = $conn->prepare("SELECT p.*,u.id as user_id,u.name as user_name,u.surname as user_surname,c.name as category_name,upl.id as user_post_like_id,COUNT(com.id) as commentnumber
33                             FROM posts as p
34                             JOIN users as u ON p.user_id=u.id
35                             JOIN category as c ON p.category_id=c.id
36                             LEFT JOIN user_posts_likes as upl ON upl.post_id=p.id AND upl.user_id=u.id
37                             LEFT JOIN comments as com ON com.post_id=p.id
38                             WHERE p.heading LIKE :search
39                             GROUP BY p.id
40                             ORDER BY p.created DESC
41                             LIMIT $this_page,$posts_per_page");
42
43     $resul->bindParam(":userid", $_SESSION["user"]->id);
44     $resul->bindParam(":search",$search);
45     $resul->execute();
46 } else {
47     $number_of_posts = $conn->query("SELECT COUNT(*) FROM posts")->fetchColumn();
48
49     $this_page = ($page - 1) * $posts_per_page;
50 }
```

COUNT(com.id) as commentnumber

```

posts.php > ...
posts.php > ...
48
49     $this_page = ($page - 1) * $posts_per_page;
50     $number_of_pages = ceil($number_of_posts / $posts_per_page);
51     if (!empty($_GET["page"]) && ($_GET["page"] > $number_of_pages || $_GET["page"] < 0)) {
52         header("Location:/posts.php?page=1");
53         exit();
54     }
55
56     $resul = $conn->prepare("SELECT p.*,u.id as user_id,u.name as user_name,u.surname as user_surname,c.name as category_name,upl.id as user_post_like_id,COUNT(com.id) as commentnu
57     FROM posts as p
58         JOIN users as u ON p.user_id=u.id
59         JOIN category as c ON p.category_id=c.id
60         LEFT JOIN user_posts_likes as upl ON upl.post_id=p.id AND upl.user_id=:userid
61         LEFT JOIN comments as com ON com.post_id=p.id
62         GROUP BY p.id
63         ORDER BY p.created DESC
64         LIMIT $this_page,$posts_per_page");
65
66     $resul->bindParam(":userid", $_SESSION["user"]->id);
67     $resul->execute();
68 }
69
70
71
72
73
74
75
76
77     $kategorije = $conn->query("SELECT * FROM category")->fetchAll();
78
79     $result = $resul->fetchAll();
80 ?>
81 <?php include "header.php"; ?>
82
83 <?php include "nav.php"; ?>
84
85
86
87     <div class="container p-3">
88         <div class="row">
89             <div class="col-3 border-right">
90                 <?php foreach ($kategorije as $kt) : ?>
91                     <p><a class="text-secondary" href="/category.php?id=<?php echo $kt->id ?>"><?php echo $kt->name; ?></a></p>
92                 </?php endforeach; ?>
93             </div>
94             <div class="col-9 col-lg-7 text-center border-right">
95                 <?php include "assets/error/error.php" ?>
96                 <?php include "assets/success/success.php" ?>

```

---

```

posts.php > ...
posts.php > ...
96     <?php include "assets/success/success.php" ?>
97     <?php foreach ($result as $res) : ?>
98         <div class="card shadow-lg bg-white rounded mt-5">
99             
100            <div class="card-body">
101                <h5 class="card-title"><?php echo $res->heading; ?></h5>
102                <p class="card-text"><?php echo $res->category_name; ?></p>
103                <p class="card-text text-muted">(Created by: <?php echo $res->user_name . " " . $res->user_surname; ?>)</p>
104                <small><?php echo $res->user_id == $_SESSION["user"]->id ? "<a class='text-secondary' href='/assets/models/deletepost.php?id=$res->id'>Delete your post</a>" : ""></small>
105            </div>
106            <div class="card-footer">
107                <span id="like_<?php echo $res->id; ?>" class="text-muted fas <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?> like-button" data-post-id="<?php echo $res->id ?>">
108                    <span><?php echo $res->commentnumber ?></span>Comments:<?php echo $res->commentnumber ?></span>
109                </div>
110            </?php endforeach; ?>
111        <ul class="pagination pagination-lg justify-content-center mt-3">
112            <?php
113                for ($page = 1; $page <= $number_of_pages; $page++) {
114                    if(!empty($_GET["search"])){
115                        echo '<li class="page-item"><a class="page-link" href="posts.php?page=' . $page . '&search=' . $_GET["search"] . '">' . $page . '</a></li>';
116                    }else{
117                        echo '<li class="page-item"><a class="page-link" href="posts.php?page=' . $page . '">' . $page . '</a></li>';
118                    }
119                }
120            ?>
121        </ul>
122    </div>
123    </div>
124    </div>
125    <?php include "footer.php"; ?>

```

---

```

<?php echo $res->id ?>"Delete your post</a>" : ""); ?></small>

arrow-up : "fa-arrow-down"; ?> like-button" data-post-id="<?php echo $res->id; ?>"> Likes: <?php echo $res->likes; ?></span>
number ?></a></span>

```

## category.php

```

category.php X
category.php > ...
48     LIMIT $this_page,$posts_per_page );
49
50     $post->bindParam(":id", $kategorijaid);
51     $post->bindParam(":userid", $_SESSION["user"]->id);
52     $post->bindParam(":search", $search);
53     $post->execute();
54 } else {
55     $number_of_post_search = $conn->prepare("SELECT COUNT(*) FROM posts
56                                         WHERE category_id=:id");
57     $number_of_post_search->bindParam(":id", $kategorijaid);
58     $number_of_post_search->execute();
59     $number_of_posts = $number_of_post_search->fetchColumn();
60
61     $this_page = ($page - 1) * $posts_per_page;
62     $number_of_pages = ceil($number_of_posts / $posts_per_page);
63     if (!empty($_GET["page"]) && ($_GET["page"] > $number_of_pages || $_GET["page"] < 0)) {
64         header("Location:/posts.php?page=1");
65         exit();
66     }
67
68     $post = $conn->prepare("SELECT p.* ,u.id as user_id,u.name as user_name,u.surname as user_surname,c.name as category_name,upl.id as user_post_like_id,COUNT(com.id) as com_count
69     FROM posts as p
70     JOIN users as u ON p.user_id=u.id
71     JOIN category as c ON p.category_id=c.id
72     LEFT JOIN user_posts_likes as upl ON upl.post_id=p.id AND upl.user_id=:userid
73     LEFT JOIN comments as com ON com.post_id=p.id
74     WHERE p.category_id=:id
75     GROUP BY p.id
76     ORDER BY p.created DESC
77     LIMIT $this_page,$posts_per_page");
78
79     $post->bindParam(":id", $kategorijaid);
80     $post->bindParam(":userid", $_SESSION["user"]->id);
81     $post->execute();
82 }
83
84     $result = $post->fetchAll();
85
86 } else {
87     header("Location:/posts.php");
88     exit();
89 }
90
91
92 ?>
93
94     <?php include "header.php"; ?>
95
96
category.php X
category.php > ...
97     <?php include "nav.php"; ?>
98
99
100    <div class="container p-3">
101        <div class="row">
102            <div class="col-3 border-right">
103                <?php foreach ($kategorije as $kt) : ?>
104                    |   <p><a class="text-secondary" href="/category.php?id=<?php echo $kt->id ?>"><?php echo $kt->name; ?></a></p>
105                <?php endforeach; ?>
106            </div>
107            <div class="col-9 col-lg-7 text-center border-right">
108                <?php include "assets/error/error.php" ?>
109                <?php include "assets/success/success.php" ?>
110                <?php foreach ($res as $res) : ?>
111                    <div class="card shadow-lg bg-white rounded mt-5">
112                        
113                        <div class="card-body">
114                            <h5 class="card-title"><?php echo $res->heading; ?></h5>
115                            <p class="card-text"><?php echo $res->category_name; ?></p>
116                            <p class="card-text text-muted">(Created by: <?php echo $res->user_name . " " . $res->user_surname; ?>)</p>
117                            <small><?php echo ($res->user_id == $_SESSION["user"]->id ? "<a class='text-secondary' href='/assets/models/deletepost.php?id=$res->id'>Delete your post</a>" : "<a class='text-secondary' href='/post.php?id=<?php echo $res->id ?>'>Comments:(<?php echo $res->commentnumber ?>)</a>")</small>
118                        </div>
119                        <div class="card-footer">
120                            <span id="like"><?php echo $res->id; ?>" class="text-muted fas <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?> like-button"
121                            <span><?php echo ($res->user_id == $_SESSION["user"]->id ? "<a class='text-secondary' href='/assets/models/deletepost.php?id=$res->id'>Delete your post</a>" : "<a class='text-secondary' href='/post.php?id=<?php echo $res->id ?>'>Comments:(<?php echo $res->commentnumber ?>)</a>")</span>
122                        </div>
123                    </div>
124                <?php endforeach; ?>
125                <ul class="pagination pagination-lg justify-content-center mt-3">
126                    <?php
127                        for ($page = 1; $page <= $number_of_pages; $page++) {
128                            if (!empty($_GET["search"])) {
129                                echo '<li class="page-item"><a class="page-link" href="category.php?id=' . $_GET["id"] . '&page=' . $page . '&search=' . $_GET["search"] . '">' . $page . '</a></li>';
130                            } else {
131                                echo '<li class="page-item"><a class="page-link" href="category.php?id=' . $_GET["id"] . '&page=' . $page . '">' . $page . '</a></li>';
132                            }
133                        }
134                    </?php
135                </ul>
136            </div>
137        </div>
138
139     <?php include "footer.php"; ?>
140
141

```

```

117 ->user_name . " " . $res->user_surname; >></p>
118 '<a class="text-secondary" href="/assets/models/deletepost.php?id='.$res->id'>Delete your post</a>" : ""); ?></small>
119
120 : <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?> like-button" data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
121 > $res->id ?>Comments:(<?php echo $res->commentnumber >>)</a></span>
122
123
124
125
126
127
128
129
130 >my.php?id='$_GET["id"]'&page=' . $page . '&search=' . $_GET["search"] . '">' . $page . '</a></li>';
131
132 >my.php?id='$_GET["id"]'&page=' . $page . "'>' . $page . '</a></li>';

```

## post.php

```

post.php ✘
post.php > ...
post.php > ...
1 <?php session_start();
2 if (empty($_SESSION["user"])) {
3   header("Location:/index.php");
4   exit();
5 }
6
7 include "assets/config/connection.php";
8
9 $kategorije = $conn->query("SELECT * FROM category")->fetchAll();
10
11 if ($_SERVER['REQUEST_METHOD'] == 'GET') {
12
13   $postid = $_GET["id"];
14
15   if (!empty($postid)) {
16     $post = $conn->prepare("SELECT p.*,u.name as user_name,u.surname as user_surname,c.name as category_name,upl.id as user_post_like_id
17                           FROM posts as p
18                           JOIN users as u ON p.user_id=u.id
19                           JOIN category as c ON p.category_id=c.id
20                           LEFT JOIN user_posts_likes as upl ON upl.post_id=p.id AND upl.user_id=:userid
21                           WHERE p.id=:id");
22
23     $post->bindParam(":id", $postid);
24     $post->bindParam(":userid", $_SESSION["user"]->id);
25     $post->execute();
26
27     $res = $post->fetch();
28
29     $komnt=$conn->prepare("SELECT c.*,u.id as user_id,u.name as user_name,u.surname as user_surname
30                           FROM comments as c
31                           JOIN users as u ON c.user_id=u.id
32                           WHERE post_id=:postid
33                           ORDER BY c.created DESC");
34     $komnt->bindParam(":postid",$postid);
35     $komnt->execute();
36
37     $reskomnt=$komnt->fetchAll();
38   } else {
39     header("Location:/posts.php");
40     exit();
41   }
42 }
43
44
45 ?>
46 <?php include "header.php"; ?>
47
48 <?php include "nav.php"; ?>

```

```
post.php < ...>
48    <?php include "nav.php"; ?>
49
50
51
52
53    <div class="container p-3">
54        <div class="row">
55            <div class="col-3 border-right">
56                <?php foreach ($kategorije as $kt) : ?>
57                | <p><a class="text-secondary" href="/category.php?id=<?php echo $kt->id ?>"><?php echo $kt->name; ?></a></p>
58                | <?php endforeach; ?>
59            </div>
60            <div class="col-9 col-lg-7 text-center border-right">
61                <?php include "assets/error/error.php"?>
62                <?php include "assets/success/success.php"?>
63                <div class="card shadow-lg bg-white rounded mt-5">
64                    
65                    <div class="card-body">
66                        <h5 class="card-title"><?php echo $res->heading; ?></h5>
67                        <p class="card-text"><?php echo $res->category_name . " " . $res->user_surname; ?></p>
68                    </div>
69                    <div class="card-footer">
70                        <span id="like_<?php echo $res->id; ?>" class="text-muted fas <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>">
71                    </span>
72                </div>
73            </div>
74        <div class="">
75            <form id="comment-form" action="/assets/models/insercomment.php" method="POST" class="needs-validation">
76                <div class="form-group">
77                    <label for="ExampleFormControlTextareal">Comments:</label>
78                    <textarea class="form-control" id="ExampleFormControlTextareal" name="textarea" placeholder="Comment here..." cols="150" rows="3"></textarea>
79                    <p class="commentter text-danger">Must be minimum 5 characters without special characters</p>
80                    <input type="hidden" name="postid" value="<?php echo $res->id?>">
81                </div>
82                <button type="submit" class="btn btn-primary">Send</button>
83            </form>
84        </div>
85        <div class="list-group">
86            <?php foreach ($reskomnt as $kom) :?>
87            <div class="list-group-item list-group-item-action" aria-current="true">
88                <div class="d-flex w-100 justify-content-between">
89                    <small><?php echo date("d M Y H:i:s", strtotime($kom->created))?></small>
90                    <small><?php echo ($kom->user_id == $_SESSION["user"]->id ? "a class='text-secondary' href='/assets/models/deletecomment.php?id=$kom->id'"><span class='fas fa-times'><?php echo $res->user_name . " " . $kom->user_surname; ?></span></a>": ""?></small>
91                </div>
92                <p class="mb-1"><?php echo $kom->comment?></p>
93                <small><?php echo ($kom->user_name . " " . $kom->user_surname); ?></small>
94            </div>
95            <?php endforeach; ?>
96        <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
97    <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
98    <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
99    <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
100   <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
101   <?php endforeach; ?>
102   <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
103   <?php echo empty($res->user_post_like_id) ? "fa-arrow-up" : "fa-arrow-down"; ?>" like-button data-post-id="<?php echo $res->id; ?>" Likes: <?php echo $res->likes; ?></span>
104   <?php include "footer.php"; ?>
```

## newpost.php

```
newpost.php X

newpost.php > div.container.p-3 > form#newpostform > div.form-group > select#category_id.form-control
1  <?php session_start();
2  |  if(empty($_SESSION["user"])){
3  |  |  header("Location:/index.php");
4  |  |  exit();
5  |
6  |
7  |  include "assets/config/connection.php";
8  |
9  |  $kategorije=$conn->query("SELECT * FROM category")->fetchAll();
10 |
11 ?>
12 <?php include "header.php";?>
13 |
14 <?php include "nav.php";?>
15 |
16 |
17 |
18 |
19 <div class="container p-3">
20 <?php include "assets/error/error.php"?>
21 <?php include "assets/success/success.php"?>
22 <form id="newpostform" action="/assets/models/insertpost.php" method="POST" enctype="multipart/form-data">
23   <div class="form-group">
24     <label for="heading">Heading</label>
25     <input type="text" class="form-control" name="heading" id="heading">
26     <p id="newpostheading" class="text-danger">Must contain minimum 5 characters without special characters</p>
27   </div>
28   <div class="form-group">
29     <label for="categoryid">Category</label>
30     <select class="form-control" name="category_id" id="category_id">
31       <option value="0">Choose....</option>
32       <?php foreach($kategorije as $kt):?>
33         <option value="<?php echo $kt->id?><?php echo $kt->name ?>"></option>
34       <?php endforeach;?>
35     </select>
36     <p id="newpostcategory" class="text-danger">Please select category</p>
37   </div>
38   <div class="form-group">
39     <label for="image">Upload post image</label>
40     <input type="file" class="form-control-file" name="image" id="image">
41   </div>
42   <button type="submit" class="btn btn-primary mb-2">Create new post</button>
43 </form>
44 </div>
45 |
46 <?php include "footer.php";?>
```

profile.php

```
profile.php x

48 <div>.container.p-3> <div>.row.align-items-center> <div>.col-12.col-lg-8> <form>#profileform> <div>.form-group> <input>#name.form-control
49   <div>.text-center>Edit your profile:</div>
50   <form id="profileform" action="/assets/models/userupdate.php" method="POST">
51     <div class="form-group">
52       <label for="name">Name</label>
53       <input type="text" class="form-control" name="name" id="name" value="<?php echo $result->name; ?>">
54       <p id="profileName" class="text-danger">Minimum 3 letters with first capital letter</p>
55     </div>
56     <div class="form-group">
57       <label for="name">Surname</label>
58       <input type="text" class="form-control" name="surname" id="surname" value="<?php echo $result->surname; ?>">
59       <p id="profileSurname" class="text-danger">Minimum 3 letters with first capital letter</p>
60     </div>
61     <div class="form-group">
62       <label for="name">City</label>
63       <input type="text" class="form-control" name="city" id="city" value="<?php echo $result->city; ?>">
64       <p id="profileCity" class="text-danger">Minimum 3 letters with first capital letter</p>
65     </div>
66     <div class="form-group">
67       <label for="name">Zip</label>
68       <input type="text" class="form-control" name="zip" id="zip" value="<?php echo $result->zip; ?>">
69       <p id="profileZip" class="text-danger">Exactly 5 numbers</p>
70     </div>
71     <div class="form-group">
72       <label for="name">Address</label>
73       <input type="text" class="form-control" name="address" id="address" value="<?php echo $result->address; ?>">
74       <p id="profileAddress" class="text-danger">Can contain numbers and letters</p>
75     </div>
76     <div class="form-group">
77       <label for="country">Country</label>
78       <select name="country" class="form-control" id="country">
79         <option value="0">Choose...</option>
80         <?php foreach ($drzave as $dr) : ?>
81           <?php if ($dr->id==$result->country_id) : ?>
82             <option selected value="<?php echo $dr->id; ?>><?php echo $dr->name; ?>></option>
83           <?php else :?>
84             <option value="<?php echo $dr->id; ?>><?php echo $dr->name; ?>></option>
85           <?php endif; ?>
86         <?php endforeach; ?>
87       </select>
88       <p id="profileCountry" class="text-danger">Please choose country</p>
89     </div>
90     <button type="submit" class="btn btn-primary mb-2">Confirm changes</button>
91   </form>
92 </div>
93 </div>
94
95
96 <?php include "footer.php"; ?>
```

## adminpanel.php

```
adminpanel.php X
adminpanel.php > div.container,p-3 > div.row > div.col-9.text-center
1  <?php session_start();
2  if(empty($_SESSION["user"])){
3      header("Location:/index.php");
4      exit();
5  }
6  if($_SESSION["user"]->role_id!=1){
7      header("Location:/posts.php");
8      exit();
9  }
10
11 include "assets/config/connection.php";
12
13 ?>
14 <?php include "header.php";?>
15
16 <?php include "nav.php";?>
17
18
19
20 <div class="container p-3">
21 <div class="row">
22 <div class="col-3">
23     <p><a class="text-secondary" href="/adminpanel.php?id=1">Users</a></p>
24     <p><a class="text-secondary" href="/adminpanel.php?id=2">Posts</a></p>
25     <p><a class="text-secondary" href="/adminpanel.php?id=3">Category</a></p>
26 </div>
27 <div class="col-9 text-center">
28     <?php include "assets/error/error.php"?>
29     <?php include "assets/success/success.php"?>
30     <?php
31         if(empty($_GET["id"])){
32             $id=1;
33         }else{
34             $id=$_GET["id"];
35         }
36         switch($id){
37             case 1: {include "adminusers.php";break;}
38             case 2: {include "adminposts.php";break;}
39             case 3: {include "admincategory.php";break;}
40             case 4: {include "edituseradmin.php";break;}
41             case 5: {include "editpostadmin.php";break;}
42             case 6: {[include "editcategoryadmin.php";break;]}
43             case 7: {include "addcategoryadmin.php";break;}
44             default: {include "adminusers.php";break;}
45         };
46     ?>
47     </div>
48 </div>
49 </div>
```

```
49 |     </div>
50 |
51 |     <?php include "footer.php";?>
```

## adminusers.php

```
�能 adminusers.php X
adminusers.php > table.table.table-responsive > tbody > tr > td
1   <?php
2
3   $user= $conn->prepare("SELECT u.*,c.name as country_name,c.id as country_id,r.id as roleid,r.name as role_name
4   |           FROM users as u JOIN countries as c ON u.country_id=c.id
5   |           JOIN roles as r ON u.role_id=r.id");
6
7   $user->execute();
8
9   $result = $user->fetchAll();
10 ?>
11
12 <table class="table table-responsive">
13   <thead class="thead-dark">
14     <tr>
15       <th scope="col">ID</th>
16       <th scope="col">name</th>
17       <th scope="col">surname</th>
18       <th scope="col">email</th>
19       <th scope="col">address</th>
20       <th scope="col">city</th>
21       <th scope="col">country</th>
22       <th scope="col">zip</th>
23       <th scope="col">gender</th>
24       <th scope="col">role</th>
25       <th scope="col">actions</th>
26     </tr>
27   </thead>
28   <tbody>
29     <?php foreach($result as $res):?>
30     <tr>
31       <th scope="row"><?php echo $res->id?></th>
32       <td><?php echo $res->name;?></td>
33       <td><?php echo $res->surname;?></td>
34       <td><?php echo $res->email;?></td>
35       <td><?php echo $res->address;?></td>
36       <td><?php echo $res->city;?></td>
37       <td><?php echo $res->country_name;?></td>
38       <td><?php echo $res->zip;?></td>
39       <td><?php echo $res->gender;?></td>
40       <td><?php echo $res->role_name;?></td>
41       <td><a href="/adminpanel.php?id=4&userid=<?php echo $res->id?>">Edit</a>
42       | <br/>
43       | <a href="/assets/models/deleteuser.php?userid=<?php echo $res->id?>">Delete</a></td>
44     </tr>
45     <?php endforeach;?>
46   </tbody>
47 </table>
```

## adminposts.php

```
adminposts.php X
adminposts.php > table.table-responsive > tbody > tr > td
1  <?php
2
3  $user= $conn->prepare("SELECT p.*,u.id as userid,u.name as user_name,u.surname as user_surname,u.email as user_email,c.id as categoryid,c.name as category_name
4  |           FROM posts as p JOIN users as u ON p.user_id=u.id
5  |           JOIN category as c ON p.category_id=c.id");
6
7  $user->execute();
8
9  $result = $user->fetchAll();
10 ?>
11
12 <table class="table table-responsive">
13   <thead class="thead-dark">
14     <tr>
15       <th scope="col">ID</th>
16       <th scope="col">heading</th>
17       <th scope="col">image</th>
18       <th scope="col">likes</th>
19       <th scope="col">name</th>
20       <th scope="col">surname</th>
21       <th scope="col">email</th>
22       <th scope="col">category</th>
23       <th scope="col">created</th>
24       <th scope="col">actions</th>
25     </tr>
26   </thead>
27   <tbody>
28     <?php foreach($result as $res):?>
29     <tr>
30       <th scope="row"><?php echo $res->id?></th>
31       <td><?php echo $res->heading;?></td>
32       <td></td>
33       <td><?php echo $res->likes;?></td>
34       <td><?php echo $res->user_name;?></td>
35       <td><?php echo $res->user_surname;?></td>
36       <td><?php echo $res->user_email;?></td>
37       <td><?php echo $res->category_name;?></td>
38       <td><?php echo $res->created;?></td>
39       <td><a href="/adminpanel.php?id=$res->id">Edit</a>
40       <br/>
41       <a href="/assets/models/deletepostadmin.php?postid=<?php echo $res->id?>">Delete</a></td>
42     </tr>
43     <?php endforeach;?>
44   </tbody>
45 </table>
```

## admincategory.php

## edituseradmin.php

```
edituseradmin.php X
edituseradmin.php > form#edituserform > div.form-group > p#profileCountry.text-danger
48     <label for="name">City</label>
49     <input type="text" class="form-control" name="city" id="city" value=<?php echo $result->city; ?>>
50     <p id="profileCity" class="text-danger">Minimum 3 letters with first capital letter</p>
51   </div>
52   <div class="form-group">
53     <label for="name">Zip</label>
54     <input type="text" class="form-control" name="zip" id="zip" value=<?php echo $result->zip; ?>>
55     <p id="profileZip" class="text-danger">Exactly 5 numbers</p>
56   </div>
57   <div class="form-group">
58     <label for="name">Address</label>
59     <input type="text" class="form-control" name="address" id="address" value=<?php echo $result->address; ?>>
60     <p id="profileAddress" class="text-danger">Can contain numbers and letters</p>
61   </div>
62   <div class="form-group">
63     <label for="country">Country</label>
64     <select name="country" class="form-control" id="country">
65       <option value="0">Choose...</option>
66       <?php foreach ($drzave as $dr) : ?>
67         <?php if ($dr->id == $result->country_id) : ?>
68           <option selected value=<?php echo $dr->id; ?>><?php echo $dr->name; ?></option>
69         <?php else : ?>
70           <option value=<?php echo $dr->id; ?>><?php echo $dr->name; ?></option>
71         <?php endif; ?>
72       <?php endforeach; ?>
73     </select>
74     <p id="profileCountry" class="text-danger">Please choose country</p>
75   </div>
76   <button type="submit" class="btn btn-primary mb-2">Confirm changes</button>
77 </form>
```

## editpostadmin.php

```
editpostadmin.php X

editpostadmin.php > div.container.p-3 > form#editpostform > div.form-group > p#newpostcategory.text-danger
1  <?php
2
3  $kategorije = $conn->query("SELECT * FROM category")->fetchAll();
4  $post = $conn->prepare("SELECT *
5                            FROM posts
6                            WHERE id=:id");
7  $post->bindParam(":id", $_GET["postid"]);
8
9  $post->execute();
10
11 $result = $post->fetch();
12
13 if (empty($result)) {
14     header("Location:adminpanel.php?id=2");
15     $_SESSION["error"] = "post not found";
16     exit();
17 }
18 ?>
19
20
21
22
23
24
25
26 <div class="container p-3">
27     <?php include "assets/error/error.php" ?>
28     <?php include "assets/success/success.php" ?>
29     <form id="editpostform" action="/assets/models/editpost.php" method="POST" enctype="multipart/form-data">
30         <div class="form-group">
31             <label for="heading">Heading</label>
32             <input type="text" class="form-control" name="heading" id="heading" value="<?php echo $result->heading ?>">
33             <p id="newpostheading" class="text-danger">Must contain minimum 5 characters without special characters</p>
34             <input type="hidden" class="form-control" name="postid" id="postid" value="<?php echo $_GET["postid"] ?>">
35         </div>
36         <div class="form-group">
37             <label for="categoryid">Category</label>
38             <select class="form-control" name="category_id" id="category_id">
39                 <option value="0">Choose...</option>
40                 <?php foreach ($kategorije as $kt) : ?>
41                     <?php if ($kt->id == $result->category_id) : ?>
42                         <option selected value="<?php echo $kt->id; ?>"><?php echo $kt->name; ?></option>
43                     <?php else : ?>
44                         <option value="<?php echo $kt->id; ?>"><?php echo $kt->name; ?></option>
45                     <?php endif; ?>
46                 <?php endforeach; ?>
47             </select>
48             <p id="newpostcategory" class="text-danger">Please select category</p>
49         </div>
50             </select>
51             <p id="newpostcategory" class="text-danger">Please select category</p>
52         </div>
53         <div class="form-group">
54             <label for="image">Upload post image</label>
55             <input type="file" class="form-control-file" name="image" id="image">
56         </div>
57         <button type="submit" class="btn btn-primary mb-2">Update</button>
58     </form>
59 </div>
```

## editcategoryadmin.php

```
editcategoryadmin.php X
editcategoryadmin.php > div.container.p-3 > form#editcategoryform > div.form-group > p#editcategoryerr.text-danger
1  <?php
2
3  $cat = $conn->prepare("SELECT *
4      |   FROM category
5      |   WHERE id=:id");
6  $cat->bindParam(":id", $_GET["categoryid"]);
7
8  $cat->execute();
9
10 $result = $cat->fetch();
11
12 if (empty($result)) {
13     header("Location:adminpanel.php?id=3");
14     $_SESSION["error"] = "category not found";
15     exit();
16 }
17
18 ?>
19
20
21
22
23
24
25 <div class="container p-3">
26     <?php include "assets/error/error.php" ?
27     <?php include "assets/success/success.php" ?
28     <form id="editcategoryform" action="/assets/models/editcategory.php" method="POST">
29         <h1>Change category name</h1>
30         <div class="form-group">
31             <label for="oldcategoryname">Old category name</label>
32             <input type="text" class="form-control" name="oldcategoryname" id="oldcategoryname" value="<?php echo $result->name?>" disabled>
33         </div>
34         <div class="form-group">
35             <label for="categoryname">New category name</label>
36             <input type="text" class="form-control" name="categoryname" id="categoryname"/>
37             <p id="editcategoryerr" class="text-danger">Minimum 3 letters first capital letter</p>
38             <input type="hidden" class="form-control" name="categoryid" id="categoryid" value="<?php echo $_GET["categoryid"] ?>">
39         </div>
40         <button type="submit" class="btn btn-primary mb-2">Update</button>
41     </form>
42 </div>
```

## addcategoryadmin.php

```
addcategoryadmin.php X
addcategoryadmin.php > div.container.p-3 > form#addcatform > div.form-group > p#addcategoryerr.text-danger
1
2  <div class="container p-3">
3      <?php include "assets/error/error.php" ?
4      <?php include "assets/success/success.php" ?
5      <form id="addcatform" action="/assets/models/addcategory.php" method="POST">
6          <h1>Add new Category</h1>
7          <div class="form-group">
8              <label for="catname">Name new Category</label>
9              <input type="text" class="form-control" name="catname" id="catname">
10             <p id="addcategoryerr" class="text-danger">Minimum 3 letters first capital letter</p>
11         </div>
12         <button type="submit" class="btn btn-primary mb-2">Add</button>
13     </form>
14 </div>
```

models...

## registration.php

```
registration.php ×
assets > models > registration.php > ...
1  <?php session_start();
2  header("Content-type: application/json");
3  $response=[ "success" => false,"message" =>"Server internal error","data" =>[]];
4  if($_SERVER['REQUEST_METHOD'] == 'POST'){
5      include "../config/connection.php";
6
7
8      try{
9          $ime= $_POST['imep'];
10         $prezime= $_POST['prezimep'];
11         $email = $_POST['emailp'];
12         $lozinka = $_POST['lozinkap'];
13         $adresa = $_POST['adresap'];
14         $grad = $_POST['gradp'];
15         $zip = $_POST['zipp'];
16         $drzava = $_POST['drzavap'];
17         $pol=$_POST['polp'];
18         $check=$_POST['checkp'];
19
20
21         $regName="/^A-Z[a-z]{3,}/";
22         $regSurname="/^A-Z[a-z]{3,}/";
23         $regEmail="/[a-z](\.(|-)?[a-z0-9])\{2,\}@[a-z]((\.|-)?[a-z0-9]+\{2,\}\.[a-z]\{2,6\})/";
24         $regPassword="/[a-z A-Z 0-9]\{2,}/";
25         $regAdresa="/[0-9 a-z A-Z \s,]+$/";
26         $regCity="/[A-Z][a-z]\{3,}/";
27         $regZip="/([0-9]\{5\})/";
28
29
30         $greske=[];
31
32         if(!preg_match($regName,$ime)){
33             $greske["ime"]="ime nije validno";
34         }
35         if(!preg_match($regSurname,$prezime)){
36             $greske["prezime"]="Prezime nije validno";
37         }
38         if(!preg_match($regEmail,$email)){
39             $greske["email"]="Email nije validan";
40         }
41         if(!preg_match($regPassword,$lozinka)){
42             $greske["lozinka"]="Lozinka nije validna";
43         }
44         if(!preg_match($regAdresa,$adresa)){
45             $greske["adresa"]="Adresa nije validna";
46         }
47         if(!preg_match($regCity,$grad)){
48             $greske["grad"]="Grad nije validan";
49         }
50     }
51
52     $response["success"] = true;
53     $response["message"] = "Success";
54     $response["data"] = [
55         "ime" => $ime,
56         "prezime" => $prezime,
57         "email" => $email,
58         "lozinka" => $lozinka,
59         "adresa" => $adresa,
60         "grad" => $grad,
61         "zip" => $zip,
62         "drzava" => $drzava,
63         "pol" => $pol,
64         "check" => $check
65     ];
66
67     echo json_encode($response);
68 }
```

```
registration.php ×
assets > models > registration.php > ...
48     $greske[ "grad" ]= Grad nije validan ;
49 }
50 if( !preg_match( $regZip, $zip ) ){
51     $greske[ "zip" ]= "zip nije validan";
52 }
53
54 if($drzava==0){
55     $greske[ "drzava" ]= "Izaberite drzavu";
56 }
57 if($check!="accept"){
58     $greske[ "check" ]= "Prihvati uslove";
59 }
60
61 if($pol!="Male" && $pol!="Female"){
62     $greske[ "pol" ]= "Prihvati uslove";
63 }
64
65 $sifralozinka=md5($lozinka);
66 $idrole=2;
67 $active=0;
68
69 if(count($greske)){
70     $response[ "message" ]= "Validation error";
71     $response[ "data" ]= $greske;
72     echo json_encode($response);
73     http_response_code(400);
74 }else{
75
76     $upit1="SELECT id FROM users WHERE email=:email";
77     $prepare=$conn->prepare($upit1);
78     $prepare->bindParam(":email",$email);
79     $prepare->execute();
80
81     $getemail=$prepare->fetch();
82
83     if($getemail!=false){
84         $response[ "message" ]= "Email must be unique";
85         echo json_encode($response);
86         http_response_code(400);
87         exit();
88     }
89
90
91
92     $upit = "INSERT INTO users(name,surname,email,password,address,city,country_id,zip,gender,role_id,active)
93     VALUES (:ime,: prezime,:email,:lozinka,:adresa,:grad,:drzava,:zip,:pol,:idrole,:active)";
94
95     $prep = $conn->prepare($upit);
96     $prep->bindParam(':ime', $ime);
97     $prep->bindParam(': prezime', $prezime);
98     $prep->bindParam(':email', $email);
99     $prep->bindParam(':lozinka', $sifralozinka);
100    $prep->bindParam(':adresa', $adresa);
101    $prep->bindParam(':grad', $grad);
102    $prep->bindParam(':drzava', $drzava);
103    $prep->bindParam(':zip', $zip);
104    $prep->bindParam(':pol', $pol);
105    $prep->bindParam(':idrole', $idrole);
106    $prep->bindParam(':active', $active);
```

```
registration.php ×
assets > models > registration.php > ...
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157



```

$prep->bindParam(':ime', $ime);
$prep->bindParam(':prezime', $prezime);
$prep->bindParam(':email', $email);
$prep->bindParam(':lozinka', $sifralozinka);
$prep->bindParam(':adresa', $adresa);
$prep->bindParam(':grad', $grad);
$prep->bindParam(':drzava', $drzava);
$prep->bindParam(':zip', $zip);
$prep->bindParam(':pol', $pol);
$prep->bindParam(':idrole', $idrole);
$prep->bindParam(':active', $active);

$result = $prep->execute();

if($result)[
    $lastid=$conn->lastInsertId();
    $upit3="SELECT * FROM users WHERE id=:lastid";
    $pr=$conn->prepare($upit3);
    $pr->bindParam(":lastid",$lastid);
    $pr->execute();
    $emailsend=$pr->fetch();

    $emailresult=mail(
        $emailsend->email,
        "Email activation",
        "Hello". $emailsend->name."To activate this mail pleas click next link <a href='http://dev.php1-sajt.com/activate_account.php?id=".$emailsend->id."'>Link</a>";
    );
}

$response["success"]=true;
$response["message"]="You successfully registered";
http_response_code(201);
}
else{
    $response["message"]="Failed to register";
    http_response_code(500);
}
echo json_encode($response);
die();
}

} catch(PDOException $exception){
    echo json_encode($response);
    http_response_code(500);
}
else{
    echo json_encode($response);
    http_response_code(404);
}
?>
```


```

## logovanje.php

```
logovanje.php X
assets > models > logovanje.php > ...
1  <?php
2      session_start();
3      header("Content-type: application/json");
4      $response=[ "success" => false, "message" =>"Server internal error", "data" =>[]];
5      if($_SERVER['REQUEST_METHOD'] == 'POST'){
6          include "../config/connection.php";
7
8
9      try{
10         $email = $_POST[ 'email'];
11         $lozinka = $_POST[ 'lozinka'];
12
13         $regEmail="/^([a-z]((\.|-)?[a-z0-9])\{2,\})@[a-z]((\.|-)?[a-z0-9]+\{2,\}).[a-z]\{2,6\}$/";
14         $regPassword="/^[a-z A-Z 0-9]\{2,\}$/";
15
16         $greske=[];
17
18         if(!preg_match($regEmail,$email)){
19             $greske["email"]="Email nije validan";
20         }
21         if(!preg_match($regPassword,$lozinka)){
22             $greske["lozinka"]="Lozinka nije validna";
23         }
24
25         $sifralozinka=md5($lozinka);
26
27         if(count($greske)){
28             $response[ "message"]="Validation error";
29             $response[ "data"]=$greske;
30             echo json_encode($response);
31             http_response_code(400);
32         }else{
33             $upit1="SELECT * FROM users WHERE email=:email AND password=:lozinka";
34             $prepare=$conn->prepare($upit1);
35             $prepare->bindParam(":email",$email);
36             $prepare->bindParam(":lozinka",$sifralozinka);
37             $prepare->execute();
38
39             $getuser=$prepare->fetch();
40
41             if($getuser==false){
42                 $response[ "message"]="Please enter valid user and password";
43                 echo json_encode($response);
44                 http_response_code(400);
45                 exit();
46             }
47
48             $_SESSION[ "user"]=$getuser;
49         }
    }
```

 logovanje.php X

```
assets > models > logovanje.php > ...
48     $SESSION["user"] = $getUser;
49
50
51     $response["success"] = true;
52     $response["message"] = "You successfully loged in";
53     echo json_encode($response);
54     http_response_code(200);
55 }
56 }
57 }
58 catch(PDOException $exception){
59     echo json_encode($response);
60     http_response_code(500);
61 }
62 }
63 else{
64     echo json_encode($response);
65     http_response_code(404);
66 }
67 ?>
```

## addcateogry.php

```
addcategory.php X
assets > models > addcategory.php > ...
1  <?php
2  session_start();
3
4  if (empty($_SESSION["user"])) {
5      header("Location:/index.php");
6      exit();
7  }
8  if ($_SESSION["user"]->role_id != 1) {
9      header("Location:/posts.php");
10     exit();
11 }
12
13 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
14
15     if (empty($_POST["catname"])) {
16         header("Location:/adminpanel.php?id=7");
17         exit();
18     }
19
20     include "../config/connection.php";
21
22
23     try {
24
25
26         $name = $_POST["catname"];
27
28         $regex = "/^([A-Z][a-z]{3,})$/";
29
30
31         if (!preg_match($regex, $name)) {
32             $_SESSION["error"] = "Name is invalid";
33             header("Location:/adminpanel.php?id=7");
34             exit();
35         }
36
37         $doesexist = $conn->prepare("SELECT *
38             FROM category
39             WHERE name=:name");
40         $doesexist->bindParam(":name", $name);
41
42         if (!$doesexist->execute()) {
43             $_SESSION["error"] = "Server internal error";
44             header("Location:/adminpanel.php?id=7");
45             exit();
46         }
47
48         $result = $doesexist->fetch();
49 }
```

## addcategory.php X

```
assets > models > addcategory.php > ...
48     $result = $doeexist->fetch();
49
50     if (!empty($result)) {
51         $_SESSION["error"] = "Category already exists";
52         header("Location:/adminpanel.php?id=7");
53         exit();
54     }
55
56     $insertcat = $conn->prepare("INSERT INTO category (name)
57     VALUES(:name)");
58     $insertcat->bindParam(":name", $name);
59
60     if (!$insertcat->execute()) {
61         $_SESSION["error"] = "Server internal error";
62         header("Location:/adminpanel.php?id=7");
63         exit();
64     } else {
65         $_SESSION["success"] = "Successfully added category";
66         header("Location:/adminpanel.php?id=7");
67         exit();
68     }
69 } catch (PDOException $exception) {
70     $_SESSION["error"] = "Server internal error";
71     header("Location:/adminpanel.php?id=7");
72     exit();
73 }
74 } else {
75     $_SESSION["error"] = "Invalid request";
76     header("Location:/adminpanel.php?id=7");
77     exit();
78 }
79 }
```

## deletecategory.php

```
deletecategory.php X
assets > models > deletecategory.php > ...
48     $result = $doeexist->return();
49
50     if (empty($result)) {
51         $_SESSION["error"] = "Category not found";
52         header("Location:/adminpanel.php?id=3");
53         exit();
54     }
55
56     $conn->beginTransaction();
57     $deletedcomm = $conn->prepare("DELETE c
58                                     | FROM comments AS c JOIN posts AS p ON c.post_id=p.id
59                                     | WHERE p.category_id=:id");
60     $deletedcomm->bindParam(":id", $id);
61
62     if (!$deletedcomm->execute()) {
63         $conn->rollBack();
64         $_SESSION["error"] = "Server internal error";
65         header("Location:/adminpanel.php?id=3");
66         exit();
67     }
68
69     $deletedlikes= $conn->prepare("DELETE upi
70                                     | FROM user_posts_likes AS upi JOIN posts AS p ON upi.post_id=p.id
71                                     | WHERE p.category_id=:id");
72     $deletedlikes->bindParam(":id", $id);
73
74     if (!$deletedlikes->execute()) {
75         $conn->rollBack();
76         $_SESSION["error"] = "Server internal error";
77         header("Location:/adminpanel.php?id=3");
78         exit();
79     }
80     $deletedpost= $conn->prepare("DELETE
81                                     | FROM posts
82                                     | WHERE category_id=:id");
83     $deletedpost->bindParam(":id", $id);
84
85     if (!$deletedpost->execute()) {
86         $conn->rollBack();
87         $_SESSION["error"] = "Server internal error";
88         header("Location:/adminpanel.php?id=3");
89         exit();
90     }
91
92     $deletecat = $conn->prepare("DELETE
93                                     | FROM category
94                                     | WHERE id=:id");
95     $deletecat->bindParam(":id", $id);
96 --
```

```
96     $deletecat->bindParam(":id", $id);
97
98     if (!$deletecat->execute()) {
99         $conn->rollBack();
100        $_SESSION["error"] = "Server internal error";
101        header("Location:/adminpanel.php?id=3");
102        exit();
103    } else {
104        $conn->commit();
105        $_SESSION["success"] = "Successfully deleted category";
106        header("Location:/adminpanel.php?id=3");
107        exit();
108    }
109
110 } catch (PDOException $exception) {
111     $_SESSION["error"] = "Server internal error";
112     header("Location:/adminpanel.php?id=3");
113     exit();
114 }
115 } else {
116     $_SESSION["error"] = "Invalid request";
117     header("Location:/adminpanel.php?id=3");
118     exit();
119 }
120 }
```

## deletecomment.php

```
deletecomment.php ×
assets > models > deletecomment.php > ...
1  <?php
2  session_start();
3
4
5  if($_SERVER['REQUEST_METHOD'] == 'GET'){
6
7      if(empty($_SESSION["user"])){
8          header("Location:/index.php");
9          exit();
10     }
11
12     include "../config/connection.php";
13
14
15    try{
16
17        $id=$_GET["id"];
18
19
20        $regexid="/^[\d]+/";
21
22
23
24        if(!preg_match($regexid,$id)){
25            $_SESSION["error"]="Invalid post";
26            header("Location:/posts.php");
27            exit();
28        }
29
30        $doesexist=$conn->prepare("SELECT com.*,post.id as idpost
31                                FROM comments as com
32                                JOIN posts as post ON com.post_id=post.id
33                                WHERE com.id=:id");
34        $doesexist->bindParam(":id",$id);
35
36        if(!$doesexist->execute()){
37            $_SESSION["error"]="Server internal error";
38            header("Location:/posts.php");
39            exit();
40        }
41        $result=$doesexist->fetch();
42
43        if(empty($result)){
44            $_SESSION["error"]="Post not found";
45            header("Location:/posts.php");
46            exit();
47        }
48
49        if($_SESSION["user"]->id!=$result->user_id){
```

## deletecomment.php ×

assets > models > deletecomment.php > ...

```
48
49         if($_SESSION["user"]->id!=$result->user_id){
50             $_SESSION["error"]="Premission denied";
51             header("Location:/post.php?id={$result->idpost}");
52             exit();
53         }
54         $deletedcomm=$conn->prepare("DELETE
55                                         |           |           |
56                                         |           |           |   FROM comments
57                                         |           |           | WHERE id=:id");
58         $deletedcomm->bindParam(":id",$id);
59
60         if(!$deletedcomm->execute()){
61             $_SESSION["error"]="Server internal error";
62             header("Location:/post.php?id={$result->idpost}");
63             exit();
64         }else{
65             $_SESSION["success"]="Successfully deleted comment";
66             header("Location:/post.php?id={$result->idpost}");
67             exit();
68
69
70
71
72     }
73
74     catch(PDOException $exception){
75         $_SESSION["error"]="Server internal error";
76         header("Location:/posts.php");
77         exit();
78     }
79 }
80 else{
81     $_SESSION["error"]="Invalid request";
82     header("Location:/posts.php");
83     exit();
84 }
85 }
```

## deletepost.php

```
deletepost.php ×
assets > models > deletepost.php > ...
1  <?php
2  session_start();
3
4
5  if ($_SERVER['REQUEST_METHOD'] == 'GET') {
6
7      if (empty($_SESSION["user"])) {
8          header("Location:/index.php");
9          exit();
10     }
11
12     include "../config/connection.php";
13
14
15    try {
16
17
18        $id = $_GET["id"];
19
20
21        $regexid = "/^[0-9]+/";
22
23
24        if (!preg_match($regexid, $id)) {
25            $_SESSION["error"] = "Invalid post";
26            header("Location:/posts.php");
27            exit();
28        }
29
30        $doesexist = $conn->prepare("SELECT *
31                                     |   |
32                                     |   FROM posts
33                                     |   WHERE id=:id");
34        $doesexist->bindParam(":id", $id);
35
36        if (!$doesexist->execute()) {
37            $_SESSION["error"] = "Server internal error";
38            header("Location:/posts.php");
39            exit();
40        }
41        $result = $doesexist->fetch();
42
43        if (empty($result)) {
44            $_SESSION["error"] = "Post not found";
45            header("Location:/posts.php");
46            exit();
47        }
48        if ($_SESSION["user"]->id != $result->user_id) {
49            $_SESSION["error"] = "Premission denied";
```

```
deletepost.php ×
assets > models > deletepost.php > ...
48     if ($_SESSION['user_id'] != $result['user_id']) {
49         $_SESSION['error'] = "Premission denied";
50         header("Location:/posts.php");
51         exit();
52     }
53     $conn->beginTransaction();
54     $deletedcomm = $conn->prepare("DELETE
55                                     | FROM comments
56                                     | WHERE post_id=:id");
57     $deletedcomm->bindParam(":id", $id);
58
59     if (!$deletedcomm->execute()) {
60         $conn->rollBack();
61         $_SESSION['error'] = "Server internal error";
62         header("Location:/posts.php");
63         exit();
64     }
65
66     $deletedlikes= $conn->prepare("DELETE
67                                     | FROM user_posts_likes
68                                     | WHERE post_id=:id");
69     $deletedlikes->bindParam(":id", $id);
70
71     if (!$deletedlikes->execute()) {
72         $conn->rollBack();
73         $_SESSION['error'] = "Server internal error";
74         header("Location:/posts.php");
75         exit();
76     }
77
78
79     $deletedpost = $conn->prepare("DELETE
80                                     | FROM posts
81                                     | WHERE id=:id");
82     $deletedpost->bindParam(":id", $id);
83
84     if (!$deletedpost->execute()) {
85         $conn->rollBack();
86         $_SESSION['error'] = "Server internal error";
87         header("Location:/posts.php");
88         exit();
89     } else {
90         $conn->commit();
91         $_SESSION["success"] = "Successfully deleted post";
92         header("Location:/posts.php");
93         exit();
94     }
95 } catch (PDOException $exception) {
96     $_SESSION['error'] = "Server internal error";
97
98 } catch (PDOException $exception) {
99     $_SESSION['error'] = "Server internal error";
100    header("Location:/posts.php");
101    exit();
102}
103}
104}
105}
```

## deletepostadmin.php

```
deletepostadmin.php X
assets > models > deletepostadmin.php > ...
1  <?php
2  session_start();
3
4  if(empty($_SESSION["user"])){
5      header("Location:/index.php");
6      exit();
7  }
8  if($_SESSION["user"]->role_id!=1){
9      header("Location:/posts.php");
10     exit();
11 }
12
13 if ($_SERVER['REQUEST_METHOD'] == 'GET') {
14
15     if (empty($_GET["postid"])) {
16         header("Location:/adminpanel.php?id=2");
17         exit();
18     }
19
20     include "../config/connection.php";
21
22
23     try {
24
25         $id = $_GET["postid"];
26
27
28         $regexid = "/^[0-9]+/";
29
30
31         if (!preg_match($regexid, $id)) {
32             $_SESSION["error"] = "Invalid post";
33             header("Location:/adminpanel.php?id=2");
34             exit();
35         }
36
37         $doesexist = $conn->prepare("SELECT *
38                                     FROM posts
39                                     WHERE id=:id");
40         $doesexist->bindParam(":id", $id);
41
42         if (!$doesexist->execute()) {
43             $_SESSION["error"] = "Server internal error";
44             header("Location:/adminpanel.php?id=2");
45             exit();
46         }
47         $result = $doesexist->fetch();
```

```
deletepostadmin.php x
assets > models > deletepostadmin.php > ...
48     $result = $doeexist->fetch();
49
50     if (empty($result)) {
51         $_SESSION["error"] = "Post not found";
52         header("Location:/adminpanel.php?id=2");
53         exit();
54     }
55
56     $conn->beginTransaction();
57     $deletedcomm = $conn->prepare("DELETE
58                                     |           |           |
59                                     |           |           | FROM comments
60                                     |           |           | WHERE post_id=:id");
61     $deletedcomm->bindParam(":id", $id);
62
63     if (!$deletedcomm->execute()) {
64         $conn->rollBack();
65         $_SESSION["error"] = "Server internal error";
66         header("Location:/adminpanel.php?id=2");
67         exit();
68     }
69
70     $deletedlikes= $conn->prepare("DELETE
71                                     |           |
72                                     |           |           |
73                                     |           |           | FROM user_posts_likes
74                                     |           |           | WHERE post_id=:id");
75     $deletedlikes->bindParam(":id", $id);
76
77     if (!$deletedlikes->execute()) {
78         $conn->rollBack();
79         $_SESSION["error"] = "Server internal error";
80         header("Location:/adminpanel.php?id=2");
81         exit();
82
83     $deletedpost = $conn->prepare("DELETE
84                                     |           |           |
85                                     |           |           | FROM posts
86                                     |           |           | WHERE id=:id");
87     $deletedpost->bindParam(":id", $id);
88
89     if (!$deletedpost->execute()) {
90         $conn->rollBack();
91         $_SESSION["error"] = "Server internal error";
92         header("Location:/adminpanel.php?id=2");
93         exit();
94     } else {
95         $conn->commit();
96         $_SESSION["success"] = "Successfully deleted post";
97         header("Location:/adminpanel.php?id=2");
98         exit();
99     }
100 } catch (PDOException $exception) {
101     $_SESSION["error"] = "Server internal error";
102     header("Location:/adminpanel.php?id=2");
103     exit();
104 } else {
105     $_SESSION["error"] = "Invalid request";
106     header("Location:/adminpanel.php?id=2");
107     exit();
108 }
```

## deleteuser.php

```
deleteuser.php X
assets > models > deleteuser.php > ...
1  <?php
2  session_start();
3
4  if(empty($_SESSION["user"])){
5      header("Location:/index.php");
6      exit();
7  }
8  if($_SESSION["user"]->role_id!=1){
9      header("Location:/posts.php");
10     exit();
11 }
12
13 if ($_SERVER['REQUEST_METHOD'] == 'GET') {
14
15     if (empty($_GET["userid"])) {
16         header("Location:/adminpanel.php");
17         exit();
18     }
19
20     include "../config/connection.php";
21
22
23     try {
24
25
26         $id = $_GET["userid"];
27
28
29         $regexid = "/^[0-9]+/";
30
31
32         if (!preg_match($regexid, $id)) {
33             $_SESSION["error"] = "Invalid user";
34             header("Location:/adminpanel.php");
35             exit();
36         }
37
38         $doesexist = $conn->prepare("SELECT *
39                                     |   |   |   |   |
40                                     |   |   |   |   |   FROM users
41                                     |   |   |   |   |   WHERE id=:id");
42         $doesexist->bindParam(":id", $id);
43
44         if (!$doesexist->execute()) {
45             $_SESSION["error"] = "Server internal error";
46             header("Location:/adminpanel.php");
47             exit();
48         }
49         $result = $doesexist->fetch();
```

```
deleteuser.php X
assets > models > deleteuser.php > ...
48     $result = $aoesexist->retcn();
49
50     if (empty($result)) {
51         $_SESSION["error"] = "User not found";
52         header("Location:/adminpanel.php");
53         exit();
54     }
55
56     $conn->beginTransaction();
57     $deletedcomm = $conn->prepare("DELETE
58             |           |           |
59             |           |           |   FROM comments
60             |           |           |       WHERE user_id=:id");
61     $deletedcomm->bindParam(":id", $id);
62
63     if (!$deletedcomm->execute()) {
64         $conn->rollBack();
65         $_SESSION["error"] = "Server internal error";
66         header("Location:/adminpanel.php");
67         exit();
68     }
69
70     $deletedlikes= $conn->prepare("DELETE
71             |           |           |
72             |           |           |   FROM user_posts_likes
73             |           |           |       WHERE user_id=:id");
74     $deletedlikes->bindParam(":id", $id);
75
76     if (!$deletedlikes->execute()) {
77         $conn->rollBack();
78         $_SESSION["error"] = "Server internal error";
79         header("Location:/adminpanel.php");
80         exit();
81     }
82
83     $deletedpost = $conn->prepare("DELETE
84             |           |           |
85             |           |           |   FROM posts
86             |           |           |       WHERE user_id=:id");
87     $deletedpost->bindParam(":id", $id);
88
89     if (!$deletedpost->execute()) {
90         $conn->rollBack();
91         $_SESSION["error"] = "Server internal error";
92         header("Location:/adminpanel.php");
93         exit();
94     }
95
96     $deleteduser = $conn->prepare("DELETE
97             |           |           |
98             |           |           |   FROM users
99             |           |           |       WHERE id=:id");
100
```

```
94 |         $deleteduser = $conn->prepare("DELETE
95 |         |         |         |         |         |         |         |         |
96 |         |         |         |         |         |         |         |         |
97 |         |         |         |         |         |         |         |         |
98 |         |         |         |         |         |         |         |         |
99 |         |         |         |         |         |         |         |         |
100|         |         |         |         |         |         |         |         |
101|         |         |         |         |         |         |         |         |
102|         |         |         |         |         |         |         |         |
103|         |         |         |         |         |         |         |         |
104|         |         |         |         |         |         |         |         |
105|         |         |         |         |         |         |         |         |
106|         |         |         |         |         |         |         |         |
107|         |         |         |         |         |         |         |         |
108|         |         |         |         |         |         |         |         |
109|         |         |         |         |         |         |         |         |
110|
111|
112|
113|
114} catch (PDOException $exception) {
115|         $_SESSION["error"] = "Server internal error";
116|         header("Location:/adminpanel.php");
117|         exit();
118}
119} else {
120|         $_SESSION["error"] = "Invalid request";
121|         header("Location:/adminpanel.php");
122|         exit();
123}
124
```

## editcategory.php

editcategory.php X

assets > models > editcategory.php > ...

```
48     $result = $checkcategory->fetch();
49
50
51     if (empty($result)) {
52         $_SESSION["error"] = "Category not found";
53         header("Location:/adminpanel.php?id=3");
54         exit();
55     }
56
57     $categoryupdate=$conn->prepare("UPDATE category
58                                         |           |           |           |
59                                         |           |           |           SET name=:name
60                                         |           |           |           WHERE id=:id");
61     $categoryupdate->bindParam(":name",$ime);
62     $categoryupdate->bindParam(":id",$categoryid);
63
64     if (!$categoryupdate->execute()) {
65         $_SESSION["error"] = "Server internal error";
66         header("Location:/adminpanel.php?id=3");
67         exit();
68     } else {
69         $_SESSION["success"] = "Successfully edited category";
70         header("Location:/adminpanel.php?id=3");
71         exit();
72     }
73
74
75
76
77     }catch(PDOException $exception){
78         $_SESSION["error"] = "Server internal error";
79         header("Location:/adminpanel.php?id=3");
80         exit();
81     }
82 }
83 else{
84     $_SESSION["error"] = "Invalid request";
85     header("Location:/adminpanel.php?id=3");
86     exit();
87 }
88
89 ?>
90
```

## editpost.php

```
editpost.php X
assets > models > editpost.php > ...
1  <?php
2  session_start();
3
4  if (empty($_SESSION["user"])) {
5      header("Location:/index.php");
6      exit();
7  }
8  if ($_SESSION["user"]->role_id != 1) {
9      header("Location:/posts.php");
10     exit();
11 }
12
13
14 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
15
16     if (empty($_POST["postid"])) {
17         header("Location:/adminpanel.php?id=2");
18         exit();
19     }
20
21     include "../config/connection.php";
22
23
24     try {
25
26
27         $heading = $_POST["heading"];
28         $category_id = $_POST["category_id"];
29         $image = $_FILES["image"];
30         $supportedfiletypes = ["image/jpeg" => ".jpg", "image/jpg" => ".jpg", "image/png" => ".png", "image/gif" => ".gif"];
31
32     }
33
34     if (!empty($image) && $image["error"] != 4) {
35         if (!(in_array($image["type"], array_keys($supportedfiletypes)))) {
36             $_SESSION["error"] = "Image type is invalid(supported:jpg,jpeg,png,gif)";
37             header("Location:/adminpanel.php?id=2");
38             exit();
39         }
40         if ($image["error"] != 0) {
41             $_SESSION["error"] = "Image error";
42             header("Location:/adminpanel.php?id=2");
43             exit();
44         }
45
46         if ($image["size"] > 5000000) {
47             $_SESSION["error"] = "Image size is over 5mb";
48             header("Location:/adminpanel.php?id=2");
49             exit();
50     }
51
52 }
```

```
editpost.php X
assets > models > editpost.php > ...
48     header("Location:/adminpanel.php?id=2");
49     exit();
50 }
51
52 $imagename = time() . $supportedfiletypes[$image["type"]];
53 $imagepath = dirname(__FILE__) . "\..\img\\" . $imagename;
54 move_uploaded_file($image["tmp_name"], $imagepath);
55 }
56
57
58
59 $regex = "/^a-z A-Z 0-9 \s . ? !]{5,200}$/";
60 $regexid = "/^[0-9]+/";
61
62 if (!preg_match($regex, $heading)) {
63     $_SESSION["error"] = "Heading is invalid";
64     header("Location:/adminpanel.php?id=2");
65     exit();
66 }
67
68 if (!preg_match($regexid, $category_id)) {
69     $_SESSION["error"] = "Category is invalid";
70     header("Location:/adminpanel.php?id=2");
71     exit();
72 };
73
74
75 $doesexist = $conn->prepare("SELECT *
76                             |          |          |          |
77                             |          |          |          |          FROM category
78                             |          |          |          |          WHERE id=:id");
79 $doesexist->bindParam(":id", $category_id);
80
81 if (!$doesexist->execute()) {
82     $_SESSION["error"] = "Server internal error";
83     header("Location:/adminpanel.php?id=2");
84     exit();
85 }
86
87 $result = $doesexist->fetch();
88
89 if (empty($result)) {
90     $_SESSION["error"] = "Category not found";
91     header("Location:/adminpanel.php?id=2");
92     exit();
93 }
94
95 $updatepost = null;
96
97 if (!empty($image) && $image["error"] != 4) {
```

```
editpost.php X
assets > models > editpost.php > ...
96     if (!empty($image) && $image["error"] != 4) {
97
98         $updatepost = $conn->prepare("UPDATE posts
99                                     | SET heading=:heading,image=:image,category_id=:categoryid
100                                    | WHERE id=:id");
101        $updatepost->bindParam(":heading", $heading);
102        $updatepost->bindParam(":image", $imagename);
103        $updatepost->bindParam(":categoryid", $category_id);
104        $updatepost->bindParam(":id", $_POST["postid"]);
105    } else {
106        $updatepost = $conn->prepare("UPDATE posts
107                                     | SET heading=:heading,category_id=:categoryid
108                                     | WHERE id=:id");
109        $updatepost->bindParam(":heading", $heading);
110        $updatepost->bindParam(":categoryid", $category_id);
111        $updatepost->bindParam(":id", $_POST["postid"]);
112    }
113
114
115
116
117    if (!$updatepost->execute()) {
118        $_SESSION["error"] = "Server internal error";
119        header("Location:/adminpanel.php?id=2");
120        exit();
121    } else {
122        $_SESSION["success"] = "Successfully updated post";
123        header("Location:/adminpanel.php?id=2");
124        exit();
125    }
126 } catch (PDOException $exception) {
127     $_SESSION["error"] = "Server internal error";
128     header("Location:/adminpanel.php?id=2");
129     exit();
130 }
131 } else {
132     $_SESSION["error"] = "Invalid request";
133     header("Location:/adminpanel.php?id=2");
134     exit();
135 }
136 }
```

## edituser.php

```
edituser.php X
assets > models > edituser.php > ...
1 <?php session_start();
2
3 if(empty($_SESSION["user"])){
4     header("Location:/index.php");
5     exit();
6 }
7 if($_SESSION["user"]->role_id!=1){
8     header("Location:/posts.php");
9     exit();
10}
11
12
13 if($_SERVER['REQUEST_METHOD'] == 'POST'){
14     include "../config/connection.php";
15
16     try{
17         $ime= $_POST['name'];
18         $prezime= $_POST['surname'];
19         $adresa = $_POST['address'];
20         $grad = $_POST['city'];
21         $zip = $_POST['zip'];
22         $drzava = $_POST['country'];
23         $userid=$_POST['userid'];
24
25
26         $regName="/^A-Z[a-z]{3,}$/";
27         $regSurname="/^A-Z[a-z]{3,}$/";
28         $regAdresa="/[0-9 a-z A-Z \s, ]+$/";
29         $regCity="/^A-Z[a-z]{3,}$/";
30         $regZip="/^([0-9]{5})$/";
31
32
33
34
35         if(!preg_match($regName,$ime)){
36             $_SESSION["error"] = "Invalid name";
37             header("Location:/adminpanel.php");
38             exit();
39         }
40         if(!preg_match($regSurname,$prezime)){
41             $_SESSION["error"] = "Invalid surname";
42             header("Location:/adminpanel.php");
43             exit();
44         }
45         if(!preg_match($regAdresa,$adresa)){
46             $_SESSION["error"] = "Invalid address";
47             header("Location:/adminpanel.php");
48             exit();
49     }
```

```
assets > models > edituser.php > ...
48     } | exit();
49 }
50 if(!preg_match($regCity,$grad)){
51     $_SESSION["error"] = "Invalid city";
52     header("Location:/adminpanel.php");
53     exit();
54 }
55 if(!preg_match($regZip,$zip)){
56     $_SESSION["error"] = "Invalid zip";
57     header("Location:/adminpanel.php");
58     exit();
59 }
60 if($drzava==0){
61     $_SESSION["error"] = "Invalid country";
62     header("Location:/adminpanel.php");
63     exit();
64 }
65
66 if(!is_numeric($userid)){
67     $_SESSION["error"] = "Invalid user";
68     header("Location:/adminpanel.php");
69     exit();
70 }
71
72 $checkcountry=$conn->prepare("SELECT *
73                             | FROM countries
74                             | WHERE id=:id");
75 $checkcountry->bindParam(":id",$drzava);
76
77 if (!$checkcountry->execute()) {
78     $_SESSION["error"] = "Server internal error";
79     header("Location:/adminpanel.php");
80     exit();
81 }
82
83 $result = $checkcountry->fetch();
84
85 if (empty($result)) {
86     $_SESSION["error"] = "Country not found";
87     header("Location:/adminpanel.php");
88     exit();
89 }
90
91 $userupdate=$conn->prepare("UPDATE users
92                             | SET name=:name,surname=:surname,address=:address,city=:city,country_id=:drzava,zip=:zip
93                             | WHERE id=:id");
94 $userupdate->bindParam(":name",$ime);
95 $userupdate->bindParam(":surname",$prezime);
96 $userupdate->bindParam(":address",$adresa);
97 $userupdate->bindParam(":city",$grad);
98 $userupdate->bindParam(":drzava",$drzava);
99 $userupdate->bindParam(":zip",$zip);
```

```
edituser.php ×
assets > models > edituser.php > ...
90
91
92     $userupdate=$conn->prepare("UPDATE users
93                             | SET name=:name,surname=:surname,address=:address,city=:city,country_id=:drzava,zip=:zip
94                             | WHERE id=:id");
95     $userupdate->bindParam(":name",$ime);
96     $userupdate->bindParam(":surname",$prezime);
97     $userupdate->bindParam(":address",$adresa);
98     $userupdate->bindParam(":city",$grad);
99     $userupdate->bindParam(":drzava",$drzava);
100    $userupdate->bindParam(":zip",$zip);
101    $userupdate->bindParam(":id",$userid);
102
103   if (!$userupdate->execute()) {
104       $_SESSION["error"] = "Server internal error";
105       header("Location:/adminpanel.php");
106       exit();
107   } else {
108       $_SESSION["success"] = "Successfully edited profile";
109       header("Location:/adminpanel.php");
110       exit();
111   }
112
113
114
115
116
117 }catch(PDOException $exception){
118     $_SESSION["error"] = "Server internal error";
119     header("Location:/adminpanel.php");
120     exit();
121 }
122 }
123 else{
124     $_SESSION["error"] = "Invalid request";
125     header("Location:/adminpanel.php");
126     exit();
127 }
128 ?>
129
130
```

## Insertcomment.php

```
insercomment.php X
assets > models > insercomment.php > ...
1  <?php
2      session_start();
3
4
5      if($_SERVER['REQUEST_METHOD'] == 'POST'){
6
7          if(empty($_SESSION["user"])){
8              header("Location:/index.php");
9              exit();
10         }
11
12         include "../config/connection.php";
13
14
15         try{
16
17             $komentar=$_POST["textarea"];
18             $postid=$_POST["postid"];
19
20             $regex="/[a-z A-Z 0-9 \s . ? !]{5,200}/";
21             $regexid="/[0-9]+/";
22
23             if(!preg_match($regex,$komentar)){
24                 $_SESSION["error"]="Form is invalid";
25                 header("Location:/post.php?id=$postid");
26                 exit();
27             }
28
29
30
31             if(!preg_match($regexid,$postid)){
32                 $_SESSION["error"]="Invalid post";
33                 header("Location:/posts.php");
34                 exit();
35             }
36             $doesexist=$conn->prepare("SELECT *
37                                         | FROM posts
38                                         | WHERE id=:id");
39             $doesexist->bindParam(":id",$postid);
40
41             if(!$doesexist->execute()){
42                 $_SESSION["error"]="Server internal error";
43                 header("Location:/post.php?id=$postid");
44                 exit();
45             }
46
47             $result=$doesexist->fetch();
48
49             if(empty($result)){


```

```
insercomment.php ×
assets > models > insercomment.php > ...
46
47         $result=$doesexist->fetch();
48
49         if(empty($result)){
50             $_SESSION["error"]="Post not found";
51             header("Location:/posts.php");
52             exit();
53         }
54
55         $insertcomm=$conn->prepare("INSERT INTO comments (comment,post_id,user_id,created)
56                                     | VALUES(:komentar,:postid,:userid,:date)");
57         $insertcomm->bindParam(":komentar",$komentar);
58         $insertcomm->bindParam(":postid",$postid);
59         $insertcomm->bindParam(":userid",$_SESSION["user"]->id);
60         $insertcomm->bindParam(":date",date('Y-m-d H:i:s'));
61
62         if(!$insertcomm->execute()){
63             $_SESSION["error"]="Server internal error";
64             header("Location:/post.php?id=$postid");
65             exit();
66         }else{
67             $_SESSION["success"]="Successfully commented";
68             header("Location:/post.php?id=$postid");
69             exit();
70         }
71
72
73
74
75
76     }
77     catch(PDOException $exception){
78         $_SESSION["error"]="Server internal error";
79         header("Location:/post.php?id=$postid");
80         exit();
81     }
82 }
83 else{
84     $_SESSION["error"]="Invalid request";
85     header("Location:/post.php?id=$postid");
86     exit();
87 }
88 ?>
89
```

## insertpost.php

```
insertpost.php x
assets > models > insertpost.php > ...
1  <?php
2  session_start();
3
4
5  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
6
7      if (empty($_SESSION["user"])) {
8          header("Location:/index.php");
9          exit();
10     }
11
12     include "../config/connection.php";
13
14
15     try {
16
17         $heading = $_POST["heading"];
18         $category_id = $_POST["category_id"];
19         $image = $_FILES["image"];
20         $supportedfiletypes=["image/jpeg" => ".jpg","image/jpg" => ".jpg","image/png" => ".png","image/gif" => ".gif"];
21
22
23
24         if(!(in_array($image["type"],array_keys($supportedfiletypes)))){
25             $_SESSION["error"] = "Image type is invalid(supported:jpg,jpeg,png,gif)";
26             header("Location:/newpost.php");
27             exit();
28         }
29         if($image["error"]!=0){
30             $_SESSION["error"] = "Image error";
31             header("Location:/newpost.php");
32             exit();
33         }
34
35         if($image["size"]>5000000){
36             $_SESSION["error"] = "Image size is over 5mb";
37             header("Location:/newpost.php");
38             exit();
39         }
40
41         $regex = "/^[_a-zA-Z0-9\s\.\?\!]{5,200}$/";
42         $regexid = "/^[_0-9]+/";
43
44         if (!preg_match($regex, $heading)) {
45             $_SESSION["error"] = "Heading is invalid";
46             header("Location:/newpost.php");
47             exit();
48         }
49     }
```

```
insertpost.php ×
assets > models > insertpost.php > ...
48     j
49
50     if (!preg_match($regexid, $category_id)) {
51         $_SESSION["error"] = "Category is invalid";
52         header("Location:/newpost.php");
53         exit();
54     };
55
56
57     $doesexist = $conn->prepare("SELECT *
58                                 |           FROM category
59                                 |           WHERE id=:id");
60     $doesexist->bindParam(":id", $category_id);
61
62     if (!$doesexist->execute()) {
63         $_SESSION["error"] = "Server internal error";
64         header("Location:/newpost.php");
65         exit();
66     }
67
68     $result = $doesexist->fetch();
69
70     if (empty($result)) {
71         $_SESSION["error"] = "Category not found";
72         header("Location:/newpost.php");
73         exit();
74     }
75
76     $imagename=time().$supportedfiletypes[$image["type"]];
77     $imagepath=dirname(__FILE__)."\..\img\".$imagename;
78     move_uploaded_file($image["tmp_name"],$imagepath);
79
80     $date=date('Y-m-d H:i:s');
81
82
83
84     $insertpost = $conn->prepare("INSERT INTO posts (heading,image,likes,user_id,category_id,created)
85                                 |           VALUES(:heading,:image,0,:userid,:categoryid,:date)");
86     $insertpost->bindParam(":heading", $heading);
87     $insertpost->bindParam(":image", $imagename);
88     $insertpost->bindParam(":userid", $_SESSION["user"]->id);
89     $insertpost->bindParam(":categoryid", $category_id);
90     $insertpost->bindParam(":date",$date);
91
92     if (!$insertpost->execute()) {
93         $_SESSION["error"] = "Server internal error";
94         header("Location:/newpost.php");
95         exit();
96     } else {
97         $_SESSION["success"] = "Successfully created post";
98         header("Location:/posts.php");
99         exit();
100    }
101 } catch (PDOException $exception) {
102     $_SESSION["error"] = "Server internal error";
103     header("Location:/newpost.php");
104     exit();
105 }
106 } else {
107     $_SESSION["error"] = "Invalid request";
108     header("Location:/newpost.php");
109     exit();
110 }
111
```

## like.php

```
like.php  X
assets > models > like.php > ...
1  <?php
2      session_start();
3      header("Content-type: application/json");
4      $response=[ "success" => false,"message" =>"Server internal error","data" =>[]];
5      if($_SERVER['REQUEST_METHOD'] == 'POST'){
6
7          if(empty($_SESSION["user"])){
8              header("Location:/index.php");
9          }
10
11         include "../config/connection.php";
12
13
14     try[ 15         if(!empty($_POST["id"])){
16
17             $id=$_POST["id"];
18
19             $post=$conn->prepare("SELECT *
20                                 |   FROM posts
21                                 | WHERE id=:id
22                                 |   ");
23             $post->bindParam(":id",$id);
24             $post->execute();
25
26             $result=$post->fetch();
27
28             if(empty($result)){
29                 $response["message"]="Post not found";
30                 echo json_encode($response);
31                 http_response_code(400);
32                 exit();
33             }
34
35             $iduserlike=$conn->prepare("SELECT *
36                                 |   FROM user_posts_likes
37                                 | WHERE post_id=:postid AND user_id=:userid");
38             $iduserlike->bindParam(":postid",$id);
39             $iduserlike->bindParam(":userid",$_SESSION["user"]->id);
40             $iduserlike->execute();
41
42             $iduserlikeresult=$iduserlike->fetch();
43
44             if(!empty($iduserlikeresult)){
45                 $response["message"]="User already liked";
46                 echo json_encode($response);
47                 http_response_code(400);
48                 exit();
49             }
}
```

```
like.php  X
assets > models > like.php > ...
48 |     }
49 | }
50 |
51 |     $likes=$result->likes+1;
52 |
53 |     $upd=$conn->prepare("UPDATE posts
54 |                           SET likes=:likes
55 |                           WHERE id=:id
56 |                           ");
57 |     $upd->bindParam(":likes",$likes);
58 |     $upd->bindParam(":id",$id);
59 |     $update_result=$upd->execute();
60 |
61 | if($update_result){
62 |     $rememberlikes=$conn->prepare("INSERT INTO user_posts_likes(post_id,user_id)
63 |                                       VALUES(:postid,:userid)");
64 |     $rememberlikes->bindParam(":postid",$id);
65 |     $rememberlikes->bindParam(":userid",$_SESSION["user"]->id);
66 |
67 |
68 |     if(!$rememberlikes->execute()){
69 |         $response["message"]="SERVER INTERNAL ERROR";
70 |         echo json_encode($response);
71 |         http_response_code(500);
72 |         exit();
73 |     }
74 |
75 |     $response["data"]=$likes;
76 |     $response["success"]=true;
77 |     $response["message"]="Successfully liked";
78 |     echo json_encode($response);
79 |     http_response_code(200);
80 | }else{
81 |     $response["message"]="Could not save like";
82 |     echo json_encode($response);
83 |     http_response_code(500);
84 | }
85 |
86 |
87 | }else{
88 |     $response["message"]="Id has not been send";
89 |     echo json_encode($response);
90 |     http_response_code(400);
91 | }
92 |
93 | catch(PDOException $exception){
94 |     echo json_encode($response);
95 |     http_response_code(500);
96 | }
97 |
98 | else{
99 |     echo json_encode($response);
100 |     http_response_code(404);
101 | }
102 ?>
```

## unlike.php

```
uni unlike.php ×  
assets > models > unlike.php > ...  
1  <?php  
2      session_start();  
3      header("Content-type: application/json");  
4      $response=["success" => false,"message" =>"Server internal error","data" =>[]];  
5      if($_SERVER['REQUEST_METHOD'] == 'POST'){  
6  
7          if(empty($_SESSION["user"])){  
8              header("Location:/index.php");  
9          }  
10  
11         include "../config/connection.php";  
12  
13  
14         try{  
15             if(!empty($_POST["id"])){  
16  
17                 $id=$_POST["id"];  
18  
19                 $post=$conn->prepare("SELECT *  
20                                         FROM posts  
21                                         WHERE id=:id  
22                                         ");  
23                 $post->bindParam(":id",$id);  
24                 $post->execute();  
25  
26                 $result=$post->fetch();  
27  
28                 if(empty($result)){  
29                     $response["message"]="Post not found";  
30                     echo json_encode($response);  
31                     http_response_code(400);  
32                     exit();  
33                 }  
34  
35                 $iduserlike=$conn->prepare("SELECT *  
36                                         FROM user_posts_likes  
37                                         WHERE post_id=:postid AND user_id=:userid");  
38                 $iduserlike->bindParam(":postid",$id);  
39                 $iduserlike->bindParam(":userid",$_SESSION["user"]->id);  
40                 $iduserlike->execute();  
41  
42                 $iduserlikeresult=$iduserlike->fetch();  
43  
44                 if(empty($iduserlikeresult)){  
45                     $response["message"]="User already liked";  
46                     echo json_encode($response);  
47                     http_response_code(400);  
48                     exit();  
49                 }  
50             }  
51         }  
52     }  
53 }
```

不像 unlike.php X

```
assets > models >不像 unlike.php > ...
48 | | | | | exit();
49 | | | | }
50 |
51 | | | $likes=$result->likes-1;
52 |
53 | | | $upd=$conn->prepare("UPDATE posts
54 | | | | | SET likes=:likes
55 | | | | WHERE id=:id
56 | | | | ");
57 | | | $upd->bindParam(":likes",$likes);
58 | | | $upd->bindParam(":id",$id);
59 | | | $update_result=$upd->execute();

60 |
61 | if($update_result){
62 | | | $rememberlikes=$conn->prepare("DELETE FROM user_posts_likes
63 | | | | | WHERE id=:id");
64 | | | $rememberlikes->bindParam(":id",$iduserlikeresult->id);

65 |
66 |
67 |
68 |
69 | if(!$rememberlikes->execute()){
70 | | | $response["message"]="SERVER INTERNAL ERROR";
71 | | | echo json_encode($response);
72 | | | http_response_code(500);
73 | | | exit();
74 | }
75 |
76 | $response["data"]=$likes;
77 | $response["success"]=true;
78 | $response["message"]="Successfully unliked";
79 | echo json_encode($response);
80 | http_response_code(200);
81 | }else{
82 | | | $response["message"]="Could not save unlike";
83 | | | echo json_encode($response);
84 | | | http_response_code(500);
85 | }
86 |
87 |
88 |
89 |
90 |
91 | }else{
92 | | | $response["message"]="Id has not been send";
93 | | | echo json_encode($response);
94 | | | http_response_code(400);
95 | }
96 }
```

```
96
97
98
99
100
101    }
102    catch(PDOException $exception){
103        echo json_encode($response);
104        http_response_code(500);
105    }
106}
107else{
108    echo json_encode($response);
109    http_response_code(404);
110}
111?>
```

## userupdate.php

```
userupdate.php X
assets > models > userupdate.php > ...
1 <?php session_start();
2
3 if (empty($_SESSION["user"])){
4     header("Location:/index.php");
5     exit();
6 }
7
8
9 if($_SERVER['REQUEST_METHOD'] == 'POST'){
10    include "../config/connection.php";
11
12    try{
13        $ime= $_POST['name'];
14        $prezime= $_POST['surname'];
15        $adresa = $_POST['address'];
16        $grad = $_POST['city'];
17        $zip = $_POST['zip'];
18        $drzava = $_POST['country'];
19
20
21        $regName="/[A-Z][a-z]{3,}/";
22        $regSurname="/[A-Z][a-z]{3,}/";
23        $regAdresa="/[0-9 a-z A-Z \s,]+/";
24        $regCity="/[A-Z][a-z]{3,}/";
25        $regZip="/([0-9]{5})/";
26
27
28
29
30        if(!preg_match($regName,$ime)){
31            $_SESSION["error"] = "Invalid name";
32            header("Location:/profile.php");
33            exit();
34        }
35        if(!preg_match($regSurname,$prezime)){
36            $_SESSION["error"] = "Invalid surname";
37            header("Location:/profile.php");
38            exit();
39        }
40        if(!preg_match($regAdresa,$adresa)){
41            $_SESSION["error"] = "Invalid address";
42            header("Location:/profile.php");
43            exit();
44        }
45        if(!preg_match($regCity,$grad)){
46            $_SESSION["error"] = "Invalid city";
47            header("Location:/profile.php");
48            exit();
49    }
```

```
userupdate.php ×
assets > models > userupdate.php > ...
48         }
49     }
50     if(!preg_match($regZip,$zip)){
51         $_SESSION["error"] = "Invalid zip";
52         header("Location:/profile.php");
53         exit();
54     }
55
56     if($drzava==0){
57         $_SESSION["error"] = "Invalid country";
58         header("Location:/profile.php");
59         exit();
60     }
61
62     $checkcountry=$conn->prepare("SELECT *
63                                     |           |
64                                     |           |
65                                     |           |
66                                     |           |
67                                     |           |
68                                     |           |
69                                     |           |
70                                     |           |
71                                     |           |
72                                     |           |
73                                     |           |
74                                     |           |
75                                     |           |
76                                     |           |
77                                     |           |
78                                     |           |
79                                     |           |
80                                     |           |
81                                     |           |
82                                     |           |
83                                     |           |
84                                     |           |
85                                     |           |
86                                     |           |
87                                     |           |
88                                     |           |
89                                     |           |
90                                     |           |
91                                     |           |
92                                     |           |
93                                     |           |
94                                     |           |
95                                     |           |
96                                     |           |
```

```

96     } else {
97         $_SESSION["success"] = "Successfully edited profile";
98         header("Location:/profile.php");
99         exit();
100    }
101
102
103
104
105
106    }catch(PDOException $exception){
107        $_SESSION["error"] = "Server internal error";
108        header("Location:/newpost.php");
109        exit();
110    }
111}
112else{
113    $_SESSION["error"] = "Invalid request";
114    header("Location:/profile.php");
115    exit();
116}
117
118?>
119

```

## author.php

```

author.php ×
author.php > div.container.p-3 > div.text-center
1  <?php include "header.php";?>
2
3
4  <nav class="navbar navbar-expand-lg navbar-light bg-light">
5      <a class="navbar-brand" href="/index.php">Postmoon</a>
6  </nav>
7
8  <div class="container p-3">
9      <div class="text-center">
10         
11         </div>
12         <div class="text-center mt-3">
13             <p>I would describe myself as positive hard working person that's ready for new challenges.
14             So far i'm skilled in HTML,CSS,Bootstrap,C,Java,C#,JavaScript,Jquery,AJAX and i'm also trying to improve by learning new technologies.</p>
15         </div>
16     </div>
17
18
19
20 <?php include "footer.php";?>

```

## 3.2 CSS

(CSS korisnčen bootstrapov)

## 3.3 JavaScript

main.js

```
JS main.js  ×
assets > js > JS main.js > ⚡ ready() callback > ⚡ on('click') callback
1  $(document).ready(function(){
2      $(".valcheck").hide();
3      $(".valradio").hide();
4      $("#registerName").hide();
5      $("#registerSurname").hide();
6      $("#registerEmail").hide();
7      $("#registerPassword").hide();
8      $("#registerAddress").hide();
9      $("#registerCity").hide();
10     $("#registerCountry").hide();
11     $("#registerZip").hide();
12     $("#btnRegister").on('click',e => {
13         e.preventDefault();
14         let ime,prezime, email, lozinka, adresa,grad,drzava,zip,pol,check,brojGresaka;
15         ime = $('#inputName');
16         prezime= $('#inputSurname');
17         email = $('#inputEmail4');
18         lozinka = $('#inputPassword4');
19         adresa = $('#inputAddress');
20         grad = $('#inputCity');
21         drzava = $('#inputCountry');
22         zip = $('#inputZip');
23         pol = $('input[name=customRadio]:checked');
24         check = $('input[name=customCheck]:checked');
25
26         brojGresaka = 0;
27
28         var regName=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{3,}$/g
29         var regSurname=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{3,}$/g
30         var regEmail=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{2,}@[a-z]{1,}((\.\|-)?[a-z0-9]+){2,}[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{2,6}$/g
31         var regPassword=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{8,}$/g
32         var regAdresa=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{3,}[\u041f\u0435\u0440\u0435\u0434\u043d\u0430\u043f\u0430]$/g
33         var regCity=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430][\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{3,}[\u041f\u0435\u0440\u0435\u0434\u043d\u0430\u043f\u0430]$/g
34         var regZip=/^[\u0410-\u042f\u0420\u0435\u0434\u043d\u0430\u043f\u0430]{5}$/g
35
36         if(!regName.test($("#ime").val())){
37             brojGresaka++;
38             $("#ime").addClass("is-invalid");
39             $("#registerName").show();
40         }
41         else{
42             $("#ime").removeClass("is-invalid");
43             $("#registerName").hide();
44         }
45
46         if(!regSurname.test($("#prezime").val())){
47             brojGresaka++;
48             $("#prezime").addClass("is-invalid");
49             $("#registerSurname").show();
50         }
51
52         if(!regEmail.test($("#email").val())){
53             brojGresaka++;
54             $("#email").addClass("is-invalid");
55             $("#registerEmail").show();
56         }
57
58         if(!regPassword.test($("#password").val())){
59             brojGresaka++;
60             $("#password").addClass("is-invalid");
61             $("#registerPassword").show();
62         }
63
64         if(!regAdresa.test($("#address").val())){
65             brojGresaka++;
66             $("#address").addClass("is-invalid");
67             $("#registerAddress").show();
68         }
69
70         if(!regCity.test($("#city").val())){
71             brojGresaka++;
72             $("#city").addClass("is-invalid");
73             $("#registerCity").show();
74         }
75
76         if(!regZip.test($("#zip").val())){
77             brojGresaka++;
78             $("#zip").addClass("is-invalid");
79             $("#registerZip").show();
80         }
81
82         if(brojGresaka>0){
83             alert("Broj gresaka je: " + brojGresaka);
84         }
85     }
86 
```

**JS main.js**

```
assets > js > JS main.js > ⚡ ready() callback > ⚡ on('click') callback
48     $(prezime).addClass( "is-invalid" );
49     $("#registerSurname").show();
50 }
51 else{
52     $(prezime).removeClass("is-invalid");
53     $("#registerSurname").hide();
54 }
55
56 if(!regEmail.test($("#email").val())){
57     brojGresaka++;
58     $(email).addClass("is-invalid");
59     $("#registerEmail").show();
60 }
61 else{
62     $(email).removeClass("is-invalid");
63     $("#registerEmail").hide();
64 }
65
66 if(!regPassword.test($("#lozinka").val())){
67     brojGresaka++;
68     $(lozinka).addClass("is-invalid");
69     $("#registerPassword").show();
70 }
71 else{
72     $(lozinka).removeClass("is-invalid");
73     $("#registerPassword").hide();
74 }
75 if(!regAdresa.test($("#adresa").val())){
76     brojGresaka++;
77     $(adresa).addClass("is-invalid");
78     $("#registerAddress").show();
79 }
80 else{
81     $(adresa).removeClass("is-invalid");
82     $("#registerAddress").hide();
83 }
84 if(!regCity.test($("#grad").val())){
85     brojGresaka++;
86     $(grad).addClass("is-invalid");
87     $("#registerCity").show();
88 }
89 else{
90     $(grad).removeClass("is-invalid");
91     $("#registerCity").hide();
92 }
93 if(!regZip.test($("#zip").val())){
94     brojGresaka++;
95     $(zip).addClass("is-invalid");
96     $("#registerZip").show();
97 }
```

JS main.js X

assets > js > JS main.js > ready() callback > on('click') callback

```
96     $("#" + registerZip).show();
97 }
98 else{
99     $(zip).removeClass("is-invalid");
100    $("#" + registerZip).hide();
101 }
102
103 if($("#drzava").val() == ""){
104     brojGresaka++;
105     $("#drzava").addClass("is-invalid");
106     $("#" + registerCountry).show();
107 }
108 else{
109     $("#drzava").removeClass("is-invalid");
110     $("#" + registerCountry).hide();
111 }
112
113 if($("#pol").length == 0){
114     brojGresaka++;
115     $(".valradio").show();
116 }
117 else{
118     $(".valradio").hide();
119 }
120 if($("#check").length == 0){
121     brojGresaka++;
122     $(".valcheck").show();
123 }
124 else{
125     $(".valcheck").hide();
126 }
127
128
129
130 if(brojGresaka == 0){
131     let podaci = {
132         imep : $(ime).val(),
133         prezimep : $(prezime).val(),
134         emailp : $(email).val(),
135         lozinkap: $(lozinka).val(),
136         adresap : $(adresa).val(),
137         gradp : $(grad).val(),
138         drzavap : $(drzava).val(),
139         zipp : $(zip).val(),
140         polp : $(pol).val(),
141         checkp: $(check).val()
142     }
143
144 }
```

```
is main.js  ×
assets > js > JS main.js > ⚡ ready() callback > ⚡ on('click') callback
143
144
145     ajaxCallBack("/assets/models/registration.php", "post", podaci, function(result){
146         $('#odgovor').html('<p class="alert alert-success my-3">${result.message}</p>');
147     },function(err){
148         $('#odgovor').html('<p class="alert alert-danger my-3">${err.responseJSON.message}</p>');
149     });
150 }
151
152 })
153
154
155
156
157 $('#btnLogin').on('click',e=>{
158     e.preventDefault();
159     let email, lozinka, brojGresaka;
160     email = $('#logemail');
161     lozinka = $('#logpass');
162     brojGresaka = 0;
163
164     var regEmail=/^[a-z](\.(|-)?[a-z0-9])\{2,}\@[a-z](\.(|-)?[a-z0-9]+\{2,}\.\[a-z]\{2,6}\$/
165     var regPassword=/^[\w A-Z 0-9]\{2,}\$/g
166
167     if(!regEmail.test($('#email').val())){
168         brojGresaka++;
169         $('#email').addClass("is-invalid");
170     }
171     else{
172         $('#email').removeClass("is-invalid");
173     }
174
175     if(!regPassword.test($('#lozinka').val())){
176         brojGresaka++;
177         $('#lozinka').addClass("is-invalid");
178     }
179     else{
180         $('#lozinka').removeClass("is-invalid");
181     }
182
183
184     if(brojGresaka == 0){
185         let podaci = {
186             email: $('#email').val(),
187             lozinka: $('#lozinka').val()
188         }
189
190         ajaxCallBack("/assets/models/logovanje.php", "post", podaci,function(result){
191             window.location.replace("/posts.php");
```

```
JS main.js ×

assets > js > JS main.js > ⚡ ready() callback > ⚡ on('click') callback
190     ajaxCallBack( "/assets/models/togovanje.php" , "post" , podaci,function(result){
191         window.location.replace("/posts.php");
192     },function(err){
193         alert(err.responseJSON.message);
194     });
195 }
196 })
197
198 $(".like-button").on("click",function(){
199     var podaci={
200         id:$this.attr("data-post-id")
201     };
202     var _this=$this;
203     //arrow function neutralise $this
204     if($this.hasClass("fas fa-arrow-up")){
205         ajaxCallBack("/assets/models/like.php","post",podaci,function(result){
206             $('#like_${podaci.id}`).html("Likes:" + result.data);
207             $_this.removeClass("fas fa-arrow-up").addClass("fas fa-arrow-down");
208         },function(err){
209             alert(err.responseJSON.message);
210         });
211     }
212     if($this.hasClass("fas fa-arrow-down")){
213         ajaxCallBack("/assets/models/unlike.php","post",podaci,function(result){
214             $('#like_${podaci.id}`).html("Likes:" + result.data);
215             $_this.removeClass("fas fa-arrow-down").addClass("fas fa-arrow-up");
216         },function(err){
217             alert(err.responseJSON.message);
218         });
219     }
220
221 })
222 })
223
224 })
225
226 function ajaxCallBack(url, method, data, result,err){
227     $.ajax({
228         url: url,
229         method: method,
230         data: data,
231         dataType: "json",
232         success: result,
233         error: err
234     });
235 }
```

## post.js

```
js post.js  X  
assets > js > JS post.js > ⌚ ready() callback > ⌚ on("submit") callback  
1  $(document).ready(function () {  
2      $(".commenterr").hide();  
3      $("#comment-form").on("submit", function (e) {  
4          var text = $("#exampleFormControlTextarea1");  
5  
6          var regex = /^[a-z A-Z 0-9 \s . ? !]{5,200}$/g  
7  
8          brojGresaka = 0;  
9          if (!regex.test($(text).val())) {  
10             brojGresaka++;  
11             $(text).addClass("is-invalid");  
12             $(".commenterr").show();  
13         }  
14         else {  
15             $(text).removeClass("is-invalid");  
16             $(".commenterr").hide();  
17         }  
18  
19         if (brojGresaka != 0) {  
20             e.preventDefault();  
21         }  
22     })  
23     $("#newpostheading").hide();  
24     $("#newpostcategory").hide();  
25     $("#newpostform").on("submit", function (e) {  
26         var text = $("#heading");  
27         var country=$("#category_id");  
28         var regex = /^[a-z A-Z 0-9 \s . ? !]{5,200}$/g  
29  
30         brojGresaka = 0;  
31         if (!regex.test($(text).val())) {  
32             brojGresaka++;  
33             $(text).addClass("is-invalid");  
34             $("#newpostheading").show();  
35         }  
36         else {  
37             $(text).removeClass("is-invalid");  
38             $("#newpostheading").hide();  
39         }  
40         if($("#country").val()==0){  
41             brojGresaka++;  
42             $(country).addClass("is-invalid");  
43             $("#newpostcategory").show();  
44         }else{  
45             $(country).removeClass("is-invalid");  
46             $("#newpostcategory").hide();  
47         }  
48     })  
49 }
```

```
js post.js  x

assets > js > JS post.js > ready() callback > on("submit") callback
48     $(country).removeClass( "is-invalid" );
49     $("#newpostcategory").hide();
50 }
51
52     if (brojGresaka != 0) {
53         e.preventDefault();
54     }
55 })
56     $("#profileName").hide();
57     $("#profileSurname").hide();
58     $("#profileAddress").hide();
59     $("#profileCity").hide();
60     $("#profileCountry").hide();
61     $("#profileZip").hide();
62     $("#profileform").on("submit", function (e) {
63         var ime = $('#name');
64         var prezime= $('#surname');
65         var adresa = $('#address');
66         var grad = $('#city');
67         var drzava = $('#country');
68         var zip = $('#zip');

69         brojGresaka=0;
70
71         var regName=/^([A-Z][a-z]{3,})$/g
72         var regSurname=/^([A-Z][a-z]{3,})$/g
73         var regAdresa=/^([0-9 a-z A-Z \s,]+)$/g
74         var regCity=/^([A-Z][a-z]{3,})$/g
75         var regZip=/^([0-9]{5})$/g
76
77         if(!regName.test($('#name').val())){
78             brojGresaka++;
79             $('#name').addClass("is-invalid");
80             $('#profileName').show();
81         }
82         else{
83             $('#name').removeClass("is-invalid");
84             $('#profileName').hide();
85         }
86
87         if(!regSurname.test($('#surname').val())){
88             brojGresaka++;
89             $('#surname').addClass("is-invalid");
90             $('#profileSurname').show();
91         }
92         else{
93             $('#surname').removeClass("is-invalid");
94             $('#profileSurname').hide();
95         }
96     }
97 
```

**js post.js** ×

assets > js > **js post.js** > ready() callback > on("submit") callback

```
96      }
97      if(!regAdresa.test($("#adresa").val())){
98          brojGresaka++;
99          $("#adresa").addClass("is-invalid");
100         $("#profileAddress").show();
101     }
102     else{
103         $("#adresa").removeClass("is-invalid");
104         $("#profileAddress").hide();
105     }
106     if(!regCity.test($("#grad").val())){
107         brojGresaka++;
108         $("#grad").addClass("is-invalid");
109         $("#profileCity").show();
110     }
111     else{
112         $("#grad").removeClass("is-invalid");
113         $("#profileCity").hide();
114     }
115     if(!regZip.test($("#zip").val())){
116         brojGresaka++;
117         $("#zip").addClass("is-invalid");
118         $("#profileZip").show();
119     }
120     else{
121         $("#zip").removeClass("is-invalid");
122         $("#profileZip").hide();
123     }
124
125     if($("#drzava").val()==0){
126         brojGresaka++;
127         $("#drzava").addClass("is-invalid");
128         $("#profileCountry").show();
129     }
130     else{
131         $("#drzava").removeClass("is-invalid");
132         $("#profileCountry").hide();
133     }
134
135
136     if (brojGresaka != 0) {
137         e.preventDefault();
138     }
139 }
140 $("#edituserform").on("submit", function (e) {
141     var ime = $('#name');
142     var prezime= $('#surname');
143     var adresa = $('#address');
144     var grad = $('#city');
```

**JS post.js**

X

```
assets > js > JS post.js > ⚡ ready() callback > ⚡ on("submit") callback
143     var adresa = $('#address');
144     var grad = $('#city');
145     var drzava = $('#country');
146     var zip = $('#zip');
147
148     brojGresaka=0;
149
150     var regName=/^([A-Z][a-z]{3,})$/g
151     var regSurname=/^([A-Z][a-z]{3,})$/g
152     var regAdresa=/^([0-9 a-zA-Z \s,]+)$$/g
153     var regCity=/^([A-Z][a-z]{3,})$/g
154     var regZip=/^([0-9]{5})$/g
155
156     if(!regName.test($('#ime').val())){
157         brojGresaka++;
158         $('#ime').addClass("is-invalid");
159         $('#profileName').show();
160     }
161     else{
162         $('#ime').removeClass("is-invalid");
163         $('#profileName').hide();
164     }
165
166     if(!regSurname.test($('#prezime').val())){
167         brojGresaka++;
168         $('#prezime').addClass("is-invalid");
169         $('#profileSurname').show();
170     }
171     else{
172         $('#prezime').removeClass("is-invalid");
173         $('#profileSurname').hide();
174     }
175     if(!regAdresa.test($('#adresa').val())){
176         brojGresaka++;
177         $('#adresa').addClass("is-invalid");
178         $('#profileAddress').show();
179     }
180     else{
181         $('#adresa').removeClass("is-invalid");
182         $('#profileAddress').hide();
183     }
184     if(!regCity.test($('#grad').val())){
185         brojGresaka++;
186         $('#grad').addClass("is-invalid");
187         $('#profileCity').show();
188     }
189     else{
190         $('#grad').removeClass("is-invalid");
191         $('#profileCity').hide();
```

**JS postjs** X

```
assets > js > JS post.js > ⚡ ready() callback > ⚡ on("submit") callback
190     $(grau).removeClass("is-invalid"),
191     $("#profileCity").hide();
192     }
193     if(!regZip.test($("#zip").val())){
194         brojGresaka++;
195         $("#zip").addClass("is-invalid");
196         $("#profileZip").show();
197     }
198     else{
199         $("#zip").removeClass("is-invalid");
200         $("#profileZip").hide();
201     }
202
203     if($("#drzava").val()==0){
204         brojGresaka++;
205         $("#drzava").addClass("is-invalid");
206         $("#profileCountry").show();
207     }
208     else{
209         $("#drzava").removeClass("is-invalid");
210         $("#profileCountry").hide();
211     }
212
213
214     if (brojGresaka != 0) {
215         e.preventDefault();
216     }
217 })
218 $("#editpostform").on("submit", function (e) {
219
220     var text = $("#heading");
221     var country=$("#category_id");
222     var regex = /^[a-zA-Z0-9\s.?!]{5,200}$/g
223
224
225     brojGresaka = 0;
226     if (!regex.test($("#text").val())) {
227         brojGresaka++;
228         $("#text").addClass("is-invalid");
229         $("#newpostheading").show();
230     }
231     else {
232         $("#text").removeClass("is-invalid");
233         $("#newpostheading").hide();
234     }
235     if($("#country").val()==0){
236         brojGresaka++;
237         $("#country").addClass("is-invalid");
238         $("#newpostcategory").show();
239     }
240 }
```

**JS post.js** X

assets > js > **JS** post.js > ready0 callback > on("submit") callback

```
238     $("#" + newpostcategory).show();
239 } else{
240     $(country).removeClass("is-invalid");
241     $("#" + newpostcategory).hide();
242 }
243
244 if (brojGresaka != 0) {
245     e.preventDefault();
246 }
247 })
248 $("#" + editcategoryerr).hide();
249 $("#" + editcategoryform).on("submit", function (e) {
250
251     var text = $("#" + categoryname);
252     var regex = /^[A-Z][a-z]{3,}$/g
253
254
255     brojGresaka = 0;
256     if (!regex.test($(text).val())) {
257         brojGresaka++;
258         $(text).addClass("is-invalid");
259         $("#" + editcategoryerr).show();
260     }
261     else {
262         $(text).removeClass("is-invalid");
263         $("#" + editcategoryerr).hide();
264     }
265     if (brojGresaka != 0) {
266         e.preventDefault();
267     }
268 })
269 $("#" + addcategoryerr).hide();
270 $("#" + addcatform).on("submit", function (e) {
271
272     var text = $("#" + catname);
273     var regex = /^[A-Z][a-z]{3,}$/g
274
275
276     brojGresaka = 0;
277     if (!regex.test($(text).val())) {
278         brojGresaka++;
279         $(text).addClass("is-invalid");
280         $("#" + addcategoryerr).show();
281     }
282     else {
283         $(text).removeClass("is-invalid");
284         $("#" + addcategoryerr).hide();
285     }
286     if (brojGresaka != 0) {
```

```
284     |         $( "#addcategoryerr" ).hide();
285     |     }
286     |     if (brojGresaka != 0) {
287     |         e.preventDefault();
288     |     }
289     | });
290
291
292
293 });
294
295
```