

# **VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE**

## **PRAKTIKUM PHP**



## **DOKUMENTACIJA**

**[techtalksict.epizy.com](http://techtalksict.epizy.com)**

**Aleksandar Stanković 33/17**

## Contents

UVOD.....	5
KORIŠĆENE TEHNOLOGIJE.....	5
FUNKCIONALNOSTI .....	5
TEMPLATE .....	6
HOME .....	6
AUTHOR .....	7
ACCOUNT SETTINGS.....	8
ADMIN.....	9
TOPIC.....	12
SHOW THREAD .....	12
NEW POST .....	13
NEW THREAD .....	13
SHOW USER.....	14
BAZA PODATAKA.....	15
STRUKTURA.....	16
KOD .....	17
INDEX.PHP .....	17
ROBOTS.TXT .....	19
.htaccess.....	19
TOP.PHP .....	19
BOTTOM.PHP .....	22
ADMIN_PANELS_NAV.PHP.....	24
DASHBOARD.PHP .....	25
FORUM.....	26
REGLOGFORMS.PHP .....	29
THREADS.PHP .....	31
WELCOMEMSG.PHP .....	33
404.PHP .....	34
ACCESS_FORBIDDEN.PHP .....	34
ACCOUNT_SETTINGS.PHP .....	34
ADMIN.PHP .....	35

AUTHOR.PHP .....	46
HOME.PHP .....	47
NEW_POST.PHP .....	47
NEW_THREAD.PHP .....	47
SHOW_THREAD.PHP .....	48
SHOW_USER.PHP .....	48
TOPIC.PHP .....	50
ADD_POST.PHP .....	53
ADD_SECTION.PHP.....	54
ADD_THREAD.PHP .....	55
ADD_TOPIC.PHP .....	57
BAN_USER.PHP .....	58
DELETE_ROW.PHP.....	60
EXCEL_EXPORT2.PHP .....	62
EXPORT_AUTHOR.PHP.....	64
FIND_SECTION.PHP .....	65
FIND_THREAD.PHP .....	66
FIND_TOPIC.PHP .....	68
FIND_USER.PHP .....	69
GET_OVERVIEW_DATA.PHP.....	70
GET_SECTIONS_FOR_SELECT.PHP .....	73
LOAD_THREAD.PHP .....	74
LOGIN.PHP .....	76
LOGOUT.PHP.....	78
REGISTER.PHP .....	79
UPDATE_ROLE.PHP .....	81
UDPATE_USER.PHP .....	82
USERNAME_AND_EMAIL_CHECK.PHP.....	84
.htaccess (ostali, ispod root-a) .....	85
SITEMAP.XML.....	85
FUNCTIONS.PHP.....	88
DBCONN.PHP .....	98

CONFIG.PHP ..... 99

.ENV..... 100

MAIN.CSS ..... 100

MAIN.JS ..... 143

## UVOD

### KORIŠĆENE TEHNOLOGIJE

- Alati
  - Atom
  - XAMPP
- Jezici
  - HTML5
  - CSS3
  - Javascript (ES6)
  - PHP
  - MySQL
- Prenos i čuvanje podataka
  - MySQL
  - AJAX
- Javascript biblioteke
  - jQuery 3.3.1

### FUNKCIONALNOSTI












Sajt je klasičan forum, sa osnovnim mogućnostima. Korisnici započinju teme, odgovaraju na tuđe, itd.

Korisnici imaju mogućnost da promene svoju profilnu sliku i da dodaju svoju kratku biografiju.

Administratori imaju mogućnost brisanja korisnika, banovanja korisnika, dodavanja novih sekcija i tema, brisanja svog sadržaja, uvid u statistiku, itd.


## TEMPLATE

### HOME

 <a href="#">Home</a> <a href="#">Author</a> <a href="#">Log in</a>		
General		
	Threads / Posts	Last post
 <b>Forum rules</b> Forum rules and permissions	Threads: 1 Posts: 1	Diskusija ... by <b>Catman95</b> 2019-06-14
 <b>About forum</b> Brief history and basic info about forum	Threads: 1 Posts: 1	Da li kona... by <b>Catman95</b> 2019-06-14
Computer hardware		
	Threads / Posts	Last post
 <b>Processors</b> Discussing computer processors	Threads: 0 Posts: 0	No posts here
 <b>Graphic cards</b> All about graphic cards	Threads: 0 Posts: 0	No posts here
 <b>Motherboards</b> Here we talk about motherboards	Threads: 0 Posts: 0	No posts here
 <b>Storage</b> Data storing devices	Threads: 0 Posts: 0	No posts here
 <b>Power supplies</b> Which to buy, how to fix, and so on...	Threads: 0 Posts: 0	No posts here
 <b>Optical drives</b> Discussing optical drives	Threads: 0 Posts: 0	No posts here
Software		
	Threads / Posts	Last post
 <b>Productivity</b> Programs which help you with work	Threads: 0 Posts: 0	No posts here
 <b>Games</b> All about games and gaming	Threads: 0 Posts: 0	No posts here


Prikaz sekcija foruma, naziv i opis, koliko svaka ima thread-ova i postova, poslednja poruka u sekciji, i ko i kada je napisao. Header sadrži logo i navigaciju. Pri manjim dimenzijama ekrana, pojavljuje se burger meni. Klikom na log in, otvara se login forma. Ako je lozinka pogrešna, korisniku se šalje mail (Ne funkcioniše na ovom hostingu)

## AUTHOR

**tech**  **talks**


Home Author Log in

### Aleksandar Stankovic







The author was born and raised in Mladenovac, Belgrade district, Central Serbia. He curses the day he decided to make this forum because it took hell of a lot of work. Yet, he knew that making a forum would be a great test of his skills. He studies at ICT college in Belgrade, and he is a freelance ESL teacher.

**E-mail:** [aleksandar.stankovic95@protonmail.com](mailto:aleksandar.stankovic95@protonmail.com)  
**Location:** Serbia  
**DOB:** 1995-11-25

[Export to word](#) 

Credits  
Documentation  
Sitemap

Aleksandar Stanković 33/17  
[aleksandar.stankovic.33.17@ict.edu.rs](mailto:aleksandar.stankovic.33.17@ict.edu.rs)  
Some street XY  
Mladenovac  
060/7606-916

© Copyright: Aleksandar Stanković 2019. Zero rights reserved. This is just a decoraton.

Osnovni podaci o autoru, i link za eksport u word. (Ne funkcioniše na ovom hostingu). Na ovoj slici se može videti i footer sajta, u kome se nalazi par linkova, i kontakt podaci.

## ACCOUNT SETTINGS


Logged in as test10

Account Admin

tech talks

[Home](#) [Author](#) [Log out](#)

### test10's account



Change avatar  No file chosen

**ATTENTION:** Make sure the image is at least 250px wide and high for decent quality

#### Biography

Short biography. (Optional)

Podešavanja naloga. Korisnik može da promeni profilnu sliku i doda kratku biografiju. Iznad headera se mogu videti linkovi ka admin stranici (ukoliko je korisnik admin) i ka stranici za podešavanja naloga. Slika se prvo smanjuje na odgovarajuću veličinu, a onda se višak kroji.




## ADMIN


Logged in as test10


Account Admin


tech talks


Home Author Log out

 Panels


 Overview

 Users


 Content

 Overview


Users

 7


Posts

 2


Threads

 2


Visits

 76

Currently online

 1

Banned

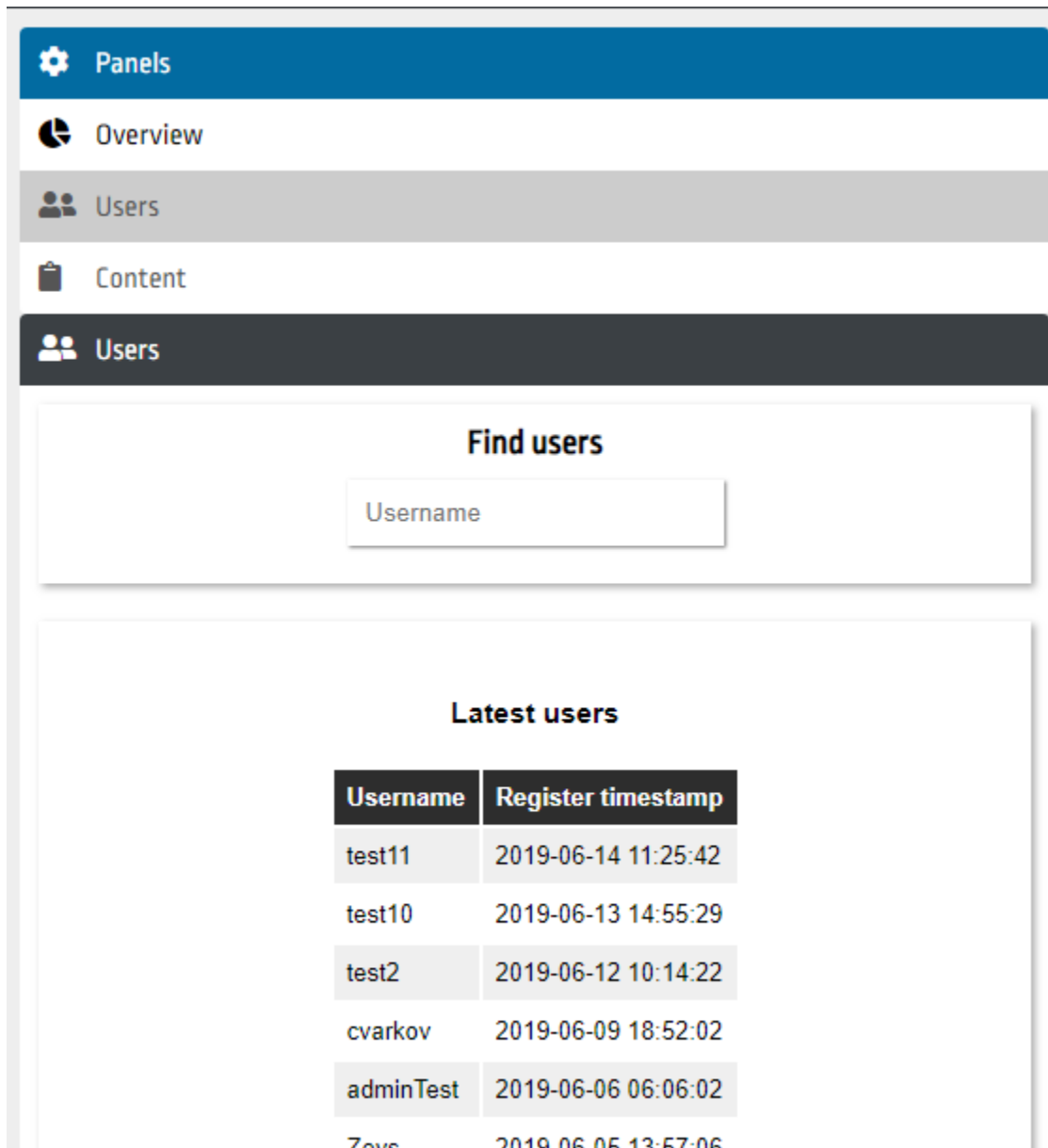
 2

Page visits

Page	Last 24h	Total	Percentage
Home	38	38	50.00%
Admin	25	26	34.21%
Account settings	1	1	1.32%
New post	1	1	1.32%
New thread	0	0	0.00%

9

Ovde imamo 3 panela. Prvi je uvid u statistiku sajta i statistiku pristupa stranicama.



**Find users**

Username

**Latest users**

Username	Register timestamp
test11	2019-06-14 11:25:42
test10	2019-06-13 14:55:29
test2	2019-06-12 10:14:22
cvarkov	2019-06-09 18:52:02
adminTest	2019-06-06 06:06:02
7ave	2019-06-05 13:57:06

Drugi panel je za pronalaženje korisnika. Možemo ih brisati, banovati, i menjati im privilegije. Prikaz najnovijih korisnika i export u excel (Koji takođe ne radi na ovom hostinguu)

Treći panel je rukovanje sadržajem foruma.

**Add section**

**Add topic**

**Find sections**

**Find topics**

**Find threads**

## TOPIC

Logged in as test10

Account Admin

tech talks


Home Author Log out

About forum

Posts

Last post

New thread

 Proba 2  
By Catman95

Posts: 1

by Catman95  
2019-06-14 -  
12:37

Credits  
Documentation

Aleksandar Stanković 33/17  
aleksandar.stankovic.33.17@ict.edu.rs  
Some street XY

f in

Ova stranica se otvara klikom na neku od tema na početnoj strani. Prikazuje thread-ove koje su korisnici započeli, koliko postova imaju, i podatke o poslednjem postu. Iznad je link za dodavanje novog thread-a.


## SHOW THREAD

tech talks

Home Author Log out

Proba 2

2019-06-14 12:37



Catman95  
Offline  
Registered: 2019-06-05  
Posts: 2

Da li konacno radi?

Page: 1

1


Answer

Credits  
Documentation

Postovi u thread-u i podaci o korisniku. Dodavanje odgovora i AJAX paginacija. 10 postova po strani.

## NEW POST

Logged in as test10 Account Admin

tech  talks Home Author Log out

You are answering as: test10


Your answer goes here

Submit

Credits  
Documentation  
Feedback

Odgovor na thread. Polje za unos teksta, i dugme za slanje.

## NEW THREAD

tech  talks

Start a new thread

Title


This will be the first post of the thread

Submit

Dodavanje novog thread-a. Naslov i tekst. Taster za slanje.

## SHOW USER

**Catman95**  
Offline



Proba

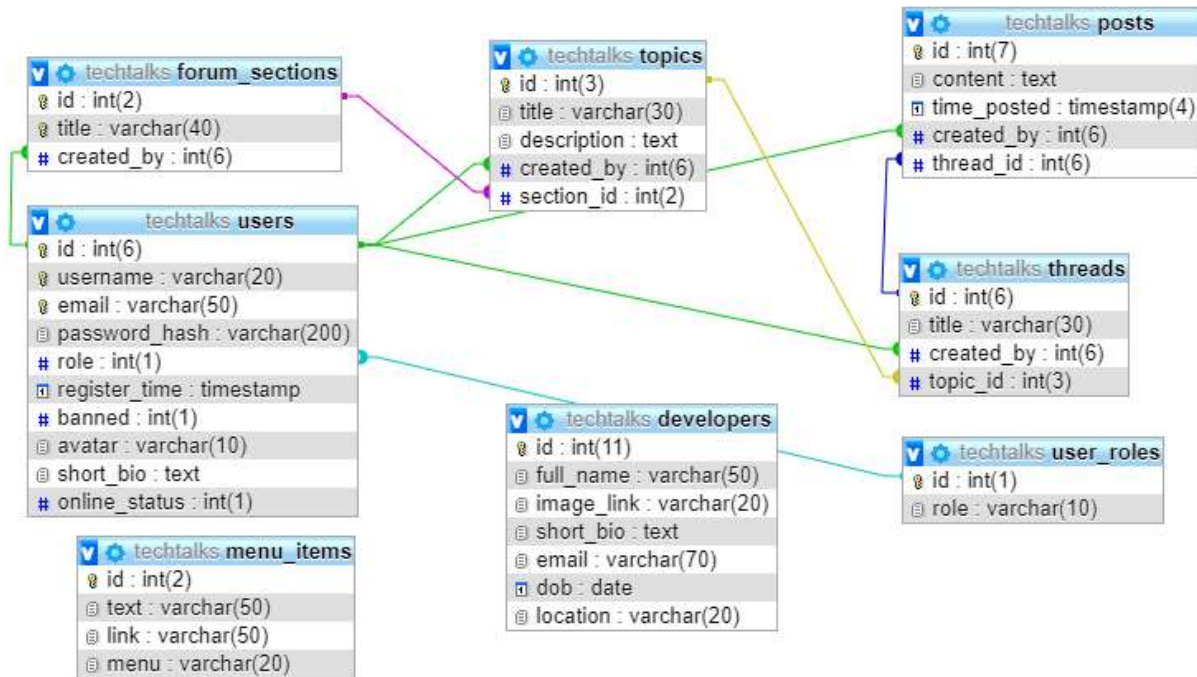
**E-mail:** aleksandar.stankovic95@protonmail.com  
**Member since:** 2019-06-05 10:58:41  
**Role:** Admin

**Latest posts**

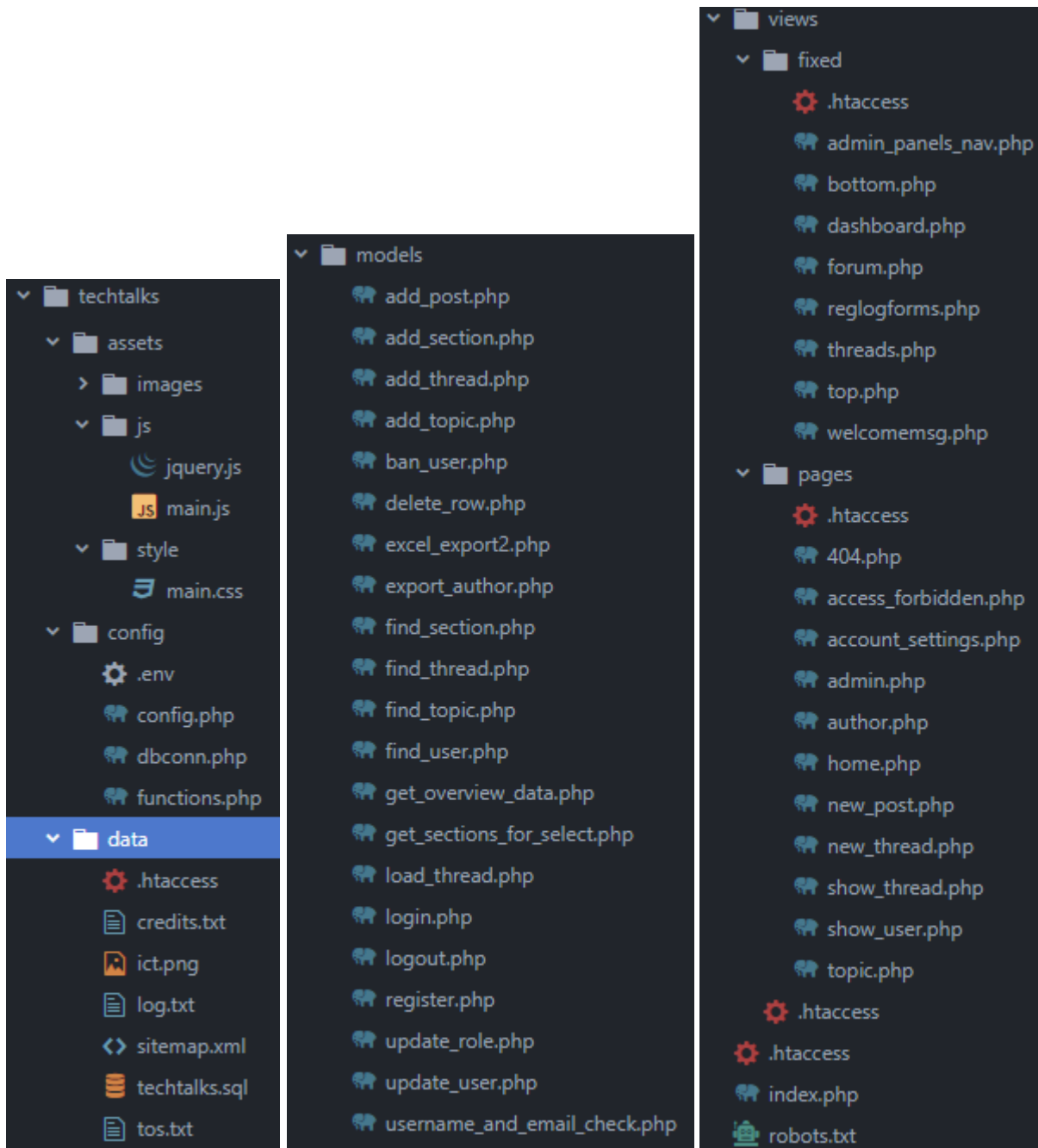
Text	Date
Da li konacno radi?	2019-06-14

Prikaz korisnika. Osnovni podaci i najnoviji postovi.

## BAZA PODATAKA



## STRUKTURA





## KOD

### INDEX.PHP

<?php

```
require 'config/config.php';
```

```
include 'views/fixed/top.php';
```

```
if(isset($_GET['page'])){  
    page_access_log();  
  
    $logged_in = (isset($_SESSION['username'])? true : false);  
  
    $admin = ($logged_in && $_SESSION['role'] == 1? true: false);  
  
    switch($_GET['page']){  
  
        case 'home':  
  
            include "views/pages/home.php";  
  
            break;  
  
        case 'admin':  
  
            redirect('home');  
  
            adminOnly();  
  
            include "views/pages/admin.php";  
  
            break;  
  
        case 'show_user':  
  
            include "views/pages/show_user.php";  
  
            break;  
  
        case 'author':  
  
            include "views/pages/author.php";
```

```
        break;

    case 'topic':

        include "views/pages/topic.php";

        break;

    case 'new_thread':

        forceIndex();

        include "views/pages/new_thread.php";

        break;

    case 'show_thread':

        include "views/pages/show_thread.php";

        break;

    case 'new_post':

        forceIndex();

        include "views/pages/new_post.php";

        break;

    case 'account_settings':

        forceIndex();

        include "views/pages/account_settings.php";

        break;

    case 'not_found':

        include "views/pages/404.php";

        break;

    case 'access_forbidden':

        include "views/pages/access_forbidden.php";

        break;
```

```

    }
}
else {
    header("Location: index.php?page=home");
}

```

```
include 'views/fixed/bottom.php';
```

```
?>
```

## ROBOTS.TXT

```
User-agent: *
```

```
Disallow:
```

## .htaccess

```
ErrorDocument 404 http://techtalksict.epizy.com/index.php?page=not_found
```

```
ErrorDocument 403 http://techtalksict.epizy.com/index.php?page=access\_forbidden
```

## TOP.PHP

```
<!DOCTYPE html>
```

```
<html lang="en" dir="ltr">
```

```
    <head>
```

```
        <meta charset="utf-8">
```

```
        <meta name="author" content="Aleksandar Stanković">
```

```
        <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
```

```
        <title><?php echo 'Tech talks - ' . ucfirst($_GET['page']); ?></title>
```

```
        <link rel="stylesheet" href="assets/style/main.css">
```

```
        <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+Pj2ysiDJxpsK1OYPAYjqT085Qq/1cq5FLXAZQ7Ay"
crossorigin="anonymous">
```

```
        <link href="https://fonts.googleapis.com/css?family=Ropa+Sans&display=swap" rel="stylesheet">
```

```

<link rel="shortcut icon" type="image/png" href="assets/images/ico2.png"/>

</head>

<body>

<div id="wrapper">

    <?php

        if(isset($_SESSION['username'])){

            include "views/fixed/dashboard.php";

        }

    ?>

    <header id="header">

        <div id="logoHolder">

            <a href="index.php?page=home"></a>

        </div>

        <div id="burger">

            <i class="fas fa-bars"></i>

        </div>

        <nav id="nav">

            <ul>

                <?php

                    $nav_links = retrieveData("SELECT * FROM menu_items WHERE menu = 'nav'");

                    foreach($nav_links as $row){

                        if($row->link == 'models/logout.php'){

                            if(!isset($_SESSION['username'])){

                                echo "<li><a href='#\" class='\"loginOpener\">Log in</a></li>";

```

```

        }else {

            echo "<li><a href=\"\$row->link\">\$row->text</a></li>";

        }

    }else {

        echo "<li><a href=\"\$row->link\">\$row->text</a></li>";

    }

}

?>

</ul>

</nav>

</header>

<div id="drawer">

    <nav id="drawerNav">

        <ul>

            <?php

                $drawerLinks = retrieveData("SELECT * FROM menu_items WHERE menu = 'nav'");

                foreach($drawerLinks as $row){

                    if($row->link == 'models/logout.php'){

                        if(!isset($_SESSION['username'])){

                            echo "<li><a href=\"#\" class=\"loginOpener\">Log in</a></li>";

                        }else {

                            echo "<li><a href=\"\$row->link\">\$row->text</a></li>";

                        }

                    }

                }

            }

        }

    }

}

```

```

        }else {
            echo "<li><a href=\"\$row->link\">\$row->text</a></li>";
        }
    }

    ?>

</ul>

</nav>

</div>

<?php include 'views/fixed/reglogforms.php'; ?>

<main id="main">

```

## BOTTOM.PHP

```

</main>

<footer id="footer">

    <div id="footerTop">

        <div id="footerPages">

            <ul>

                <?php

                    $footer_links = retrieveData("SELECT * FROM menu_items WHERE menu = 'footer'");

                    foreach($footer_links as $row){

                        echo "<li><a href=\"\$row->link\" target=\"_blank\">\$row->text</a></li>";

                    }

                ?>

            </ul>

```

```

</div>

<div id="footerContact">

    <ul>

        <li>Aleksandar Stanković 33/17</li>

        <li>aleksandar.stankovic.33.17@ict.edu.rs</li>

        <li>Some street XY</li>

        <li>Mladenovac</li>

        <li>060/7606-916</li>

    </ul>

</div>

<div id="footerSocial">

    <?php

        $social_links = retrieveData("SELECT * FROM menu_items WHERE menu = 'social'");

        foreach($social_links as $row){

            echo "<a href=\"\$row->link\" target=\"\_blank\">\$row->text</a>";

        }

    ?>

</div>

</div>

<div id="footerBottom">

    <p>&copy; Copyright: Aleksandar Stanković 2019. Zero rights reserved. This is just a decoraton.</p>

</div>

</footer>

```

```
</div>
```

```
<script src="assets/js/jquery.js"></script>
```

```
<script src="assets/js/main.js"></script>
```

```
</body>
```

```
</html>
```

```
<?php $conn = null ?>
```

## ADMIN\_PANELS\_NAV.PHP

```
<div id="adminPanelsNav">
```

```
    <div class="adminPanelsHead">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-cog"></i></div>Panels
```

```
    </div>
```

```
    <div class="adminPanelsLink" data-id="0" style="background-color:#ccc">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-chart-pie"></i></div>Overview
```

```
    </div>
```

```
    <div class="adminPanelsLink" data-id="1">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-user-friends"></i></div>Users
```

```
    </div>
```

```
    <div class="adminPanelsLink" data-id="2">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-clipboard"></i></div>Content
```

```
    </div>
```

```
    <!--COMING SOON
```

```
    <div class="adminPanelsLink" data-id="3">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-file-alt"></i></div>Logs
```

```
    </div>
```

```
    <div class="adminPanelsLink" data-id="4">
```

```
        <div class="adminPanelsLinkIcon"><i class="fas fa-bug"></i></div> Bug reports
```



```

</div>

<div class="adminPanelsLink" data-id="5">

    <div class="adminPanelsLinkIcon"><i class="fas fa-drum"></i></div> Announcements

</div>

<div class="adminPanelsLink" data-id="6">

    <div class="adminPanelsLinkIcon"><i class="fas fa-comments"></i></div> Admin chatroom

</div>-->

</div>

```

## DASHBOARD.PHP

```

<div id="dashboard">

    <div id="userData">

        <p>Logged in as <?= $_SESSION['username']; ?></p>

    </div>

    <div id="dashboardLinks">

        <?php

            $dashboard_links = retrieveData("SELECT * FROM menu_items WHERE menu = 'dashboard'");

            foreach($dashboard_links as $row){

                if($row->link == 'index.php?page=admin'){

                    if($_SESSION['role'] == 1){

                        echo "<a href=\"\$row->link\">\$row->text</a>";

                    }

                }else {

                    echo "<a href=\"\$row->link\">\$row->text</a>";

                }

            }

        }

    </div>

</div>

```

```

    }

    ?>

</div>

</div>

```

## FORUM

```
<?php
```

```

$get = $conn->query("SELECT * FROM forum_sections");
foreach($get as $section){
    echo '
        <div class="forumSection">
            <div class="sectionHead">
                <div class="sectionTitle">
                    <p>' . $section->title . '</p>
                </div>
                <div class="postsLabel obsolete">
                    <p>Threads / Posts</p>
                </div>
                <div class="lastPostLabel obsolete">
                    <p>Last post</p>
                </div>
            </div>
        ';

    $getTopics = $conn->query("SELECT * FROM topics WHERE section_id = $section->id");
    if($getTopics->rowCount() > 0){

```

```

foreach($getTopics as $topic){

    $countThreads = $conn->query("SELECT COUNT(*) as count FROM threads WHERE topic_id =
    $topic->id");

    $no_of_threads = $countThreads->fetch()->count;

    $countPosts = $conn->query("SELECT COUNT(*) as count FROM posts p INNER JOIN threads th
    ON p.thread_id = th.id WHERE th.topic_id = $topic->id");

    $no_of_posts = $countPosts->fetch()->count;

    $lastPostData = $conn->query("SELECT u.username, u.id, p.time_posted,
    SUBSTRING(p.content, 1, 10) as content FROM posts p INNER JOIN users u ON p.created_by = u.id INNER
    JOIN threads th ON p.thread_id = th.id INNER JOIN topics tp ON tp.id = th.topic_id WHERE tp.id = $topic-
    >id ORDER BY p.time_posted DESC LIMIT 1");

    $lastPost = $lastPostData->fetch();

    $lastPostLabel = '';

    if($lastPostData->rowCount() == 1){

        $lastPostLabel = '<p>' . $lastPost->content . '...</p>'

        <p>by <a href="index.php?page=show_user&id=' . $lastPost->id . "'>' . $lastPost->username .
        '</a></p>'

        <p>' . substr($lastPost->time_posted, 0, 11) . '</p>';

    }else {

        $lastPostLabel = "<p>No posts here</p>";

    }

    echo '

    <div class="topic">

        <div class="sectionTitle">

            <div class="topicIcon">

            </div>

            <div class="topicTitleAndDescription">

```

```

        <div class="topicTitle">

            <a href="index.php?page=topic&id=' . $topic->id . '"><p>' . $topic->title . '</p></a>

        </div>

        <div class="topicDescription">

            <p>' . $topic->description . '</p>

        </div>

    </div>

</div>

<div class="postsLabel obsolete">

    <p>Threads: ' . $no_of_threads . '</p>

    <p>Posts: ' . $no_of_posts . '</p>

</div>

<div class="lastPostLabel obsolete">' . $lastPostLabel . '</div>

</div>

';

}

}else {

    echo "<p style='padding: 10px'>Section empty ¶</p>";

}

echo "</div>";

}

?>

```

## REGLOGFORMS.PHP

```
<div id="loginForm">

    <form>

        <i class="fas fa-window-close loginCloser"></i>

        <p class="formTitle">MEMBER LOGIN</p>

        <div class="inputHolder">

            <input type="email" placeholder="E-mail*" id="loginEmail" autocomplete="on"
spellcheck="false"><i class="fas fa-at"></i>

        </div>

        <div class="inputHolder">

            <input type="password" placeholder="Password*" id="loginPassword"><i class="fas fa-key"></i>

        </div>

        <button id="loginBtn">Log in</button><br>

        <div class="formLinkHolder">

            <a href="#">Forgot password?</a>

            <a href="#" class="registerOpener">Register</a>

        </div>

    </form>

    <div id="loginImage">

        <p>If you don't have an account and you want to join the discussion, you'll have to register
first.</p>

    </div>

</div>

<div id="registerForm">

    <form>

        <i class="fas fa-window-close registerCloser"></i>

        <p class="formTitle">NEW ACCOUNT</p>
```

```

<div class="inputHolder">

    <input type="text" id="registerUsername" placeholder="Username*" spellcheck="false"
autocomplete="off"><i class="fas fa-user"></i>

</div>

<div class="inputHolder">

    <input type="email" placeholder="E-mail*" id="registerEmail" autocomplete="on"
spellcheck="false"><i class="fas fa-at"></i>

</div>

<div class="inputHolder">

    <input type="password" placeholder="Password*" id="registerPassword"><i class="fas fa-
key"></i>

</div>

<div class="tosHolder">

    <label for="#tosCheck">I accept the <a href="data/tos.txt">terms of service</a></label>

    <input type="checkbox" id="tosCheck">

</div>

<button id="registerBtn">Register</button><br>

<div class="formLinkHolder">

    <a href="#" class="loginOpener">Already have an account?</a>

</div>

</form>

<div id="registerImage">

    <p>When you register, you'll become a member of the forum, and you'll be able to start new
threads, and post in others.</p>

</div>

</div>

```

## THREADS.PHP

```
<?php

$topic_id = $_GET['id'];

$topicInfo = $conn->prepare("SELECT * FROM topics WHERE id = ?");
$topicInfo->bindValue(1, $topic_id);
$topicInfo->execute();
$Row = $topicInfo->fetch();
$title = $Row->title;

echo "

<div class=\"sectionHead\">

    <div class=\"sectionTitle\">

        <p>\" . $title . "\"</p>

    </div>

    <div class=\"postsLabel\">

        <p>Posts</p>

    </div>

    <div class=\"lastPostLabel\">

        <p>Last post</p>

    </div>

</div>";

if(isset($_SESSION['user_id'])){

    echo "<div><button id=\"newThreadBtn\" class=\"addBtn btn btn-dark\" data-id='$topic_id'>New
thread <i class=\"fas fa-clipboard\"></i></button></div>";
```

```
}
```

```
$getThreads = $conn->query("SELECT th.*, u.username FROM threads th INNER JOIN users u ON  
th.created_by = u.id WHERE topic_id = $topic_id");
```

```
if($getThreads->rowCount() > 0){
```

```
    foreach($getThreads as $thread){
```

```
        $countPosts = $conn->query("SELECT COUNT(*) as count FROM posts WHERE thread_id = $thread-  
>id");
```

```
        $no_of_posts = $countPosts->fetch()->count;
```

```
        $checkLastPost = $conn->query("SELECT p.*, u.username, u.id as user_id FROM posts p INNER JOIN  
users u ON p.created_by = u.id WHERE thread_id = $thread->id ORDER BY time_posted DESC LIMIT 1");
```

```
        $lastPost = $checkLastPost->fetch();
```

```
        $last_post_by = $lastPost->username;
```

```
        $last_time_posted = $lastPost->time_posted;
```

```
        echo "
```

```
<div class=\"topic\">
```

```
    <div class=\"sectionTitle\">
```

```
        <div class=\"topicIcon\">
```

```
            <img src=\"assets/images/document.png\" alt=\"Topic icon\">
```

```
        </div>
```

```
    <div class=\"topicTitleAndDescription\">
```

```
        <div class=\"topicTitle\">
```

```
            <a href=\"index.php?page=show_thread&id=\" . $thread->id . "\"><p>\" . $thread->title .  
\"</p></a>
```

```
        </div>
```

```
    <div class=\"topicDescription\">
```



```

        <p>By <a href='index.php?page=show_user&id=" . $thread->created_by . "'>" . $thread-
>username . "</a></p>

    </div>

</div>

</div>

<div class=\"postsLabel\">

    <p>Posts: " . $no_of_posts . "</p>

</div>

<div class=\"lastPostLabel\">

    <p>by <a href=\"index.php?page=show_user&id=" . $lastPost->user_id . "\">" . $last_post_by .
"</a></p>

    <p>" . substr($last_time_posted, 0, 11) . " - " . substr($last_time_posted, 11, 5) . "</p>

</div>

</div>

";

}

}else {

    echo "<p style='padding: 30px'>No threads here</p>";

}

?>

```

## WELCOMEMSG.PHP

```

<div id="welcomeMessage">

    <div id="welcomeMessageTop">

        <p>Welcome to ITalks!</p>

    </div>

    <div id="welcomeMessageBottom">

```

<p>This is a simple, and easy to use forum (Related to computers) which was born as a result of a college assignment. It was inspired by <a href="https://www.sk.rs/forum/">this amazing forum</a>. If you don't have an account, you can register <a href="#" class="registerOpener">here</a>.</p>

</div>

</div>

## 404.PHP

<p class="errorCode">404</p>

<p class="errorMessageGeneric">Not found :</p>

<p class="errorMessageCustom">The page you are looking for doesn't exist</p>

## ACCESS\_FORBIDDEN.PHP

<p class="errorCode">403</p>

<p class="errorMessageGeneric">Access forbidden!</p>

<p class="errorMessageCustom">You're not allowed to access this area</p>

## ACCOUNT\_SETTINGS.PHP

<?php

    \$User = getUserData(\$\_SESSION['user\_id']);

?>

<div id="accountSettingsHolder">

    <h1><?= \$User->username ?>'s account</h1>

    <form action="models/update\_user.php" method="post" enctype="multipart/form-data">

        <div id="avatarSettings">

            <div class="avatarHolder">

            </div>

        <label for="avatarInput">Change avatar</label>

        <input type="file" id="avatarInput" name="avatar\_input">

</div>

<p><span class="validationError">ATTENTION:</span> Make sure the image is at least 250px wide and high<br>for decent quality</p>

<h3>Biography</h3>

<textarea rows="14" cols="80" placeholder="Short biography. (Optional)" spellcheck="false" name="shortBioInput"><?= \$User->short\_bio ?></textarea>

<input type="submit" value="Update account" name="updateSubmit">

</form>

</div>

## ADMIN.PHP

<div id="adminHolder">

<?php include "views/fixed/admin\_panels\_nav.php"; ?>

<div class="adminPanel" id="overviewPanel">

<div class="adminPanelHead">

<div class="adminPanelsLinkIcon"><i class="fas fa-chart-pie"></i></div>Overview

</div>

<div class="adminPanelContent">

<div id="overviewData">

<div class="overviewDataItem">

<div class="overviewDataItemHead">

<p>Users</p>

</div>

<div class="overviewDataItemBottom">

<div class="overviewDataItemBottomLeft">

<i class="fas fa-users"></i>

</div>

```

        <div class="overviewDataItemBottomRight">
            <p class="totalUsers"></p>
        </div>
    </div>
</div>
<div class="overviewDataItem">
    <div class="overviewDataItemHead">
        <p>Posts</p>
    </div>
    <div class="overviewDataItemBottom">
        <div class="overviewDataItemBottomLeft">
            <i class="fas fa-pencil-alt"></i>
        </div>
        <div class="overviewDataItemBottomRight">
            <p class="totalPosts"></p>
        </div>
    </div>
</div>
<div class="overviewDataItem">
    <div class="overviewDataItemHead">
        <p>Threads</p>
    </div>
    <div class="overviewDataItemBottom">
        <div class="overviewDataItemBottomLeft">
            <i class="fas fa-clipboard"></i>

```

```

</div>

<div class="overviewDataItemBottomRight">

    <p class="totalThreads"></p>

</div>

</div>

</div>

<div class="overviewDataItem">

    <div class="overviewDataItemHead">

        <p>Visits</p>

    </div>

    <div class="overviewDataItemBottom">

        <div class="overviewDataItemBottomLeft">

            <i class="fas fa-chart-bar"></i>

        </div>

        <div class="overviewDataItemBottomRight">

            <p class="totalVisits">

                <?php

                    $totalVisits = 0;

                    foreach(readLogFile() as $log){

                        if(substr($log, 0, 11) == 'Page access'){

                            $totalVisits++;

                        }

                    }

                }

            </p>

        </div>

    </div>


```

```

        echo $totalVisits;

        ?>
    </p>
</div>
</div>
</div>
<div class="overviewDataItem">
    <div class="overviewDataItemHead">
        <p>Currently online</p>
    </div>
    <div class="overviewDataItemBottom">
        <div class="overviewDataItemBottomLeft">
            <i class="fas fa-plug"></i>
        </div>
        <div class="overviewDataItemBottomRight">
            <p class="currentlyOnline"></p>
        </div>
    </div>
</div>
<div class="overviewDataItem">
    <div class="overviewDataItemHead">
        <p>Banned</p>
    </div>

```

```

<div class="overviewDataItemBottom">

    <div class="overviewDataItemBottomLeft">

        <i class="fas fa-user-slash"></i>

    </div>

    <div class="overviewDataItemBottomRight">

        <p class="bannedUsers"></p>

    </div>

</div>

</div>

</div>

<div id="overviewBottom">

    <div id="pageAccess">

        <h3>Page visits</h3>

        <?php

```

//Prvi broj je broj poseta u poslednja 24h, drugi je broj poseta uopšte, a treći je udeo u procentima

```

$pages = ['home' => [0, 0, 0, 'Home'], 'admin' => [0, 0, 0, 'Admin'], 'account_settings' =>
[0, 0, 0, 'Account settings'], 'new_post' => [0, 0, 0, 'New post'], 'new_thread' => [0, 0, 0, 'New thread'],

```

```

'show_user' => [0, 0, 0, 'Show user'], 'author' => [0, 0, 0, 'Author'], 'show_thread' => [0,
0, 0, 'Show thread'], 'topic' => [0, 0, 0, 'Topic'], 'not_found' => [0, 0, 0, 'Not found'], 'access_forbidden' =>
[0, 0, 0, 'Access forbidden']];

```

```

$totalVisits = 0;

```

```

foreach(readLogFile() as $log){

    if(substr($log, 0, 11) == 'Page access'){

```

```

        $timestamp = substr($log, strpos($log, 'Timestamp: ') + 11, 20);
        $time = strtotime($timestamp);
        $first_bracket = strpos($log, '[');
        $first_comma = strpos($log, ',');
        $length = $first_comma - $first_bracket;
        $page = substr($log, $first_bracket + 1, $length - 1);
        mark_visit($page, $time);
    }
}

show_page_visits($pages);

?>
</div>
</div>
<div id="logFileDiv">
    <h3>Logs</h3>
    <p id="log">
        <?php

        foreach(readLogFile() as $log){

            echo $log . "<br>";

        }

    </p>
?>

```



```

        </p>
    </div>

</div>
</div>
<div class="adminPanel" id="usersPanel">
    <div class="adminPanelHead">
        <div class="adminPanelsLinkIcon"><i class="fas fa-user-friends"></i></div>Users
    </div>
    <div id="givePrivDiv">
        <h3>Choose role</h3>
        <select>
            <?php
                try {
                    $roles = retrieveData("SELECT * FROM user_roles");
                    foreach($roles as $row){
                        echo "<option value=$row->id>" . ucfirst($row->role) . "</option>";
                    }
                } catch (PDOException $e) {
                    log_error($e);
                    echo $e->getMessage();
                }
            ?>
        </select>

        <p>Users with this role will be able to ban, delete, and change roles of other users (except
admins), to add sections, topics, and delete everything.</p>

```

```

<button class="btn addBtn btn-dark submitGivePriv">Submit</button>

<button class="btn delBtn btn-dark cancelGivePriv">Cancel</button>

</div>

<div class="adminPanelContent">

    <div class="adminPanelContentHolder">

        <h3>Find users</h3>

        <input type="text" id="findUser" class="adminSearchBar" placeholder="Username">

        <table class="customTable" id="usersTable">

            </table>

    </div>

    <div class="adminPanelContentHolder">

        <div class="customTable" id="latestUsersTable">

            <h3>Latest users</h3>

            <table>

                <thead>

                    <tr>

                        <th class="obsolete">ID</th>

                        <th>Username</th>

                        <th class="obsolete">E-mail</th>

                        <th>Register timestamp</th>

                    </tr>

                </thead>

                <tbody></tbody>

            </table>

            <a href="models/excel_export2.php" class="exportLink excel">Export to excel <i class="fas fa-file-excel"></i></a>

```

```

        </div>

    </div>

</div>

</div>

<div class="adminPanel" id="contentPanel">

    <div class="adminPanelHead">

        <div class="adminPanelsLinkIcon"><i class="fas fa-clipboard"></i></div>Content

    </div>

    <div class="adminPanelContent">

        <div class="adminPanelContentHolder">

            <form class="addContent" id="addSectionForm">

                <h3>Add section</h3>

                <input type="text" placeholder="Section title" id="addSectionTitle">

                <button id="addSectionBtn" class="addBtn btn btn-dark">Add</button>

            </form>

        </div>

        <div class="adminPanelContentHolder">

            <form class="addContent" id="addTopicForm">

                <h3>Add topic</h3>

                <input type="text" placeholder="Topic title" id="addTopicTitle">

                <input type="text" id="addTopicDescription" placeholder="Short description">

                <select id="selectSection">

                </select>

                <button id="addTopicBtn" class="addBtn btn btn-dark">Add</button>

            </form>

```

```

</div>

<div class="adminPanelContentHolder">

    <h3>Find sections</h3>

    <input type="text" id="findSection" class="adminSearchBar" placeholder="Section title">

    <table class="customTable" id="sectionTable">

    </table>

</div>

<div class="adminPanelContentHolder">

    <h3>Find topics</h3>

    <input type="text" id="findTopic" class="adminSearchBar" placeholder="Topic title">

    <table class="customTable" id="topicTable">

    </table>

</div>

<div class="adminPanelContentHolder">

    <h3>Find threads</h3>

    <input type="text" id="findThreads" class="adminSearchBar" placeholder="Thread title">

    <table class="customTable" id="threadTable">

    </table>

</div>

</div>

<!--COMING SOON

<div class="adminPanel" id="logsPanel">

    <div class="adminPanelHead">

        <div class="adminPanelsLinkIcon"><i class="fas fa-file-alt"></i></div>Logs

```

```

</div>

<div class="adminPanelContent">

</div>

</div>

<div class="adminPanel" id="bugPanel">

  <div class="adminPanelHead">

    <div class="adminPanelsLinkIcon"><i class="fas fa-bug"></i></div>Bug reports

  </div>

  <div class="adminPanelContent">

</div>

</div>

<div class="adminPanel" id="announcementPanel">

  <div class="adminPanelHead">

    <div class="adminPanelsLinkIcon"><i class="fas fa-drum"></i></div>Announcements

  </div>

  <div class="adminPanelContent">

</div>

</div>

<div class="adminPanel" id="chatroomPanel">

  <div class="adminPanelHead">

    <div class="adminPanelsLinkIcon"><i class="fas fa-comments"></i></div>Admin chatroom

  </div>

```

```

        <div class="adminPanelContent">

        </div>

    </div>

    <div class="adminPanel" id="databasePanel">

        <div class="adminPanelHead">

            <div class="adminPanelsLinkIcon"><i class="fas fa-database"></i></div>Browse database

        </div>

        <div class="adminPanelContent">

        </div>

    </div>-->

</div>

```

## AUTHOR.PHP

```

<?php

try {

    $author = getRow("SELECT * FROM developers WHERE id = 1");

} catch (PDOException $e){

    echo $e->getMessage();

    log_error($e);

}

?>

<div id="showUserHolder">

    <h1><?= $author['full_name'] ?></h1>

    <div id="avatarAndBio">

```

```

        <p><?= $author['short_bio'] ?></p>

    </div>

    <div id="showUserData">

        <p><b>E-mail: </b><?= $author['email'] ?></p>

        <p><b>Location: </b><?= $author['location'] ?></p>

        <p><b>DOB: </b><?= $author['dob'] ?></p>

    </div>

    <a href="models/export_author.php" class="exportLink word">Export to word <i class="fas fa-file-word"></i></a>

</div>

```

## HOME.PHP

```
<?php include 'views/fixed/forum.php'; ?>
```

## NEW\_POST.PHP

```

<div id="postAnswer">

    <p>You are answering as: <b><?= $_SESSION['username'] ?></b></p>

    <textarea name="name" rows="8" cols="80" placeholder="Your answer goes here"
id="postAnswerText"></textarea>

    <button id="postAnswerBtn" class="btn addBtn btn-dark" data-id=<?= $_GET['thread_id']
?>>Submit</button>

</div>

```

## NEW\_THREAD.PHP

```

<div id="newThreadDiv">

    <h2>Start a new thread</h2>

    <form>

        <input type="text" placeholder="Title" id="newThreadTitle">

```

```
<textarea name="name" rows="8" cols="80" placeholder="This will be the first post of the thread" id="newThreadPost"></textarea>
```

```
<button id="submitThreadBtn">Submit</button>
```

```
</form>
```

```
</div>
```

## SHOW\_THREAD.PHP

```
<?php
```

```
$thread_id = $_GET['id'];
```

```
$getPost = $conn->query("SELECT * FROM posts WHERE thread_id = $thread_id");
```

```
$getThreadTitle = $conn->query("SELECT title FROM threads WHERE id = $thread_id");
```

```
$threadTitle = $getThreadTitle->fetch()->title;
```

```
echo "<div class='threadHead'>
```

```
<p>" . $threadTitle . "</p>
```

```
</div>";
```

```
echo "<div id='postsHolder'></div>";
```

```
echo "<div id='pageNumber'><p></p></div>";
```

```
echo "<div id='paginationLinks'></div>";
```

```
if(isset($_SESSION['user_id']) && $_SESSION['banned'] == 0){
```

```
    echo "<button data-id=$thread_id class='btn btn-dark addBtn' id='newPostBtn'>Answer <i class='fas fa-clipboard'></i></button>";
```

```
}else {
```

```
    echo "<p class='banMsg'>You can't post because you're not logged in or you're banned</p>";
```

```
}
```

```
?>
```

## SHOW\_USER.PHP

```
<?php
```



```

$user_id = $_GET['id'];

$user = getUserData($user_id);

?>

```

```

<div id="showUserHolder">

    <h1><?= $User->username ?></h1>

    <p><?= ($User->online_status == 1? '<i class="fas fa-circle"></i> Online' : '<i class="far fa-circle"></i> Offline') ?></p>

    <div id="avatarAndBio">

        <p><?= $User->short_bio ?></p>

    </div>

    <div id="showUserData">

        <p><b>E-mail: </b><?= $User->email ?></p>

        <p><b>Member since: </b><?= $User->register_time ?></p>

        <p><b>Role: </b><?= getUserRole($User->role) ?></p>

    </div>

    <div id="latestUserPosts">

        <h3>Latest posts</h3>

        <?php

            $usersPosts = getUsersPosts($User->id, 3);

            if(sizeof($usersPosts) > 0){

                echo "<table
class='customTable'><thead><tr><th>Text</th><th>Date</th></tr></thead><tbody>";

                foreach($usersPosts as $row){

```

```

        $date = substr($row->time_posted, 0, 11);

        echo "<tr>

            <td>$row->content</td>

            <td>$date</td>

        </tr>";

    }

    echo "</tbody></table>";

} else {

    echo "<p>This user has never posted</p>";

}

?>

</div>

</div>

```

## TOPIC.PHP

```

<?php

$topic_id = $_GET['id'];

$topicInfo = $conn->prepare("SELECT * FROM topics WHERE id = ?");

$topicInfo->bindValue(1, $topic_id);

$topicInfo->execute();

$Row = $topicInfo->fetch();

$title = $Row->title;

```

```

echo "

<div class=\"sectionHead\">

    <div class=\"sectionTitle\">

        <p>\" . $title . \"</p>

    </div>

    <div class=\"postsLabel obsolete\">

        <p>Posts</p>

    </div>

    <div class=\"lastPostLabel obsolete\">

        <p>Last post</p>

    </div>

</div>";

if(isset($_SESSION['user_id'])){

    echo "<div><button id=\"newThreadBtn\" class=\"addBtn btn btn-dark\" data-id='$topic_id'>New
thread <i class=\"fas fa-clipboard\"></i></button></div>";

}

$getThreads = $conn->query("SELECT th.*, u.username FROM threads th INNER JOIN users u ON
th.created_by = u.id WHERE topic_id = $topic_id");

if($getThreads->rowCount() > 0){

    foreach($getThreads as $thread){

        $countPosts = $conn->query("SELECT COUNT(*) as count FROM posts WHERE thread_id = $thread-
>id");

        $no_of_posts = $countPosts->fetch()->count;

```

```
$checkLastPost = $conn->query("SELECT p.*, u.username, u.id as user_id FROM posts p INNER JOIN  
users u ON p.created_by = u.id WHERE thread_id = $thread->id ORDER BY time_posted DESC LIMIT 1");
```

```
$lastPost = $checkLastPost->fetch();
```

```
$last_post_by = $lastPost->username;
```

```
$last_time_posted = $lastPost->time_posted;
```

```
echo "
```

```
<div class=\"topic\">
```

```
    <div class=\"sectionTitle\">
```

```
        <div class=\"topicIcon\">
```

```
            <img src=\"assets/images/document.png\" alt=\"Topic icon\">
```

```
        </div>
```

```
    <div class=\"topicTitleAndDescription\">
```

```
        <div class=\"topicTitle\">
```

```
            <a href='index.php?page=show_thread&id=' . $thread->id . "&part=1'><p>" . $thread->title . "</p></a>
```

```
        </div>
```

```
    <div class=\"topicDescription\">
```

```
        <p>By <a href='index.php?page=show_user&id=' . $thread->created_by . "'>" . $thread->username . "</a></p>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
<div class=\"postsLabel obsolete\">
```

```
    <p>Posts: " . $no_of_posts . "</p>
```

```
</div>
```

```
<div class=\"lastPostLabel obsolete\">
```

```

        <p>by <a href=\"index.php?page=show_user&id=\" . $lastPost->user_id . "\">\" . $last_post_by .
    "</a></p>

```

```

        <p>" . substr($last_time_posted, 0, 11) . " - " . substr($last_time_posted, 11, 5) . "</p>

```

```

    </div>

```

```

</div>

```

```

    ";

```

```

}

```

```

}else {

```

```

    echo "<p style='padding: 30px'>No threads here</p>";

```

```

}

```

```

?>

```

## ADD\_POST.PHP

```

<?php

```

```

header("Content-Type: application/json");

```

```

require "../config/config.php";

```

```

if($_SERVER['REQUEST_METHOD'] == "POST" && $_SESSION['banned'] == 0){

```

```

    $user_id = $_SESSION['user_id'];

```

```

    $thread_id = $_POST['thread_id'];

```

```

    $text = $_POST['text'];

```

```

    $response = [];

```

```

    try {

```

```

        $add_thread = $conn->prepare("INSERT INTO posts(content, created_by, thread_id)
VALUES(?,?,?)");

```

```

        $add_thread->bindValue(1, $text);

```

```

        $add_thread->bindValue(2, $user_id);

        $add_thread->bindValue(3, $thread_id);

        $add_thread->execute();

        $response['result'] = 'Success';

        $status = 201;
    }

    catch(PDOException $e){

        $response['result'] = $e->getMessage();

        log_error($e);

        $status = 500;
    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

} else {

    header("Location: ../index.php");

}

?>

```

## ADD\_SECTION.PHP

```
<?php
```

```

header("Content-Type: application/json");

require "../config/config.php";

if($_SERVER['REQUEST_METHOD'] == "POST"){

    $title = $_POST['title'];

```

```

$response = [];

try {
    $add_section = $conn->prepare("INSERT INTO forum_sections(title, created_by) VALUES(?,?)");
    $add_section->bindValue(1, $title);
    $add_section->bindValue(2, $_SESSION['user_id']);
    $add_section->execute();
    $response['result'] = 'Success';
    $status = 201;
}

catch(PDOException $e){
    $status = 500;
    if($e->getCode() == '23000'){
        $response['result'] = 'Error: Section with such a name already exists';
    }
    log_error($e);
}

http_response_code($status);
echo json_encode($response, JSON_PRETTY_PRINT);
}else {
    header("Location: ../index.php");
}

?>

```

## ADD\_THREAD.PHP

```
<?php
```

```
header("Content-Type: application/json");
```

```

require "../config/config.php";

if($_SERVER['REQUEST_METHOD'] == "POST"){

    $created_by = $_SESSION['user_id'];

    $title = $_POST['title'];

    $text = $_POST['text'];

    $topic_id = $_POST['topic_id'];

    $response = [];

    try {

        $conn->beginTransaction();

        $add_thread = $conn->prepare("INSERT INTO threads(title, created_by, topic_id)
VALUES(?,?,?)");

        $add_thread->bindValue(1, $title);

        $add_thread->bindValue(2, $created_by);

        $add_thread->bindValue(3, $topic_id);

        $add_thread->execute();

        $thread_id = $conn->lastInsertId();

        $add_post = $conn->prepare("INSERT INTO posts(content, created_by, thread_id)
VALUES(?,?,?)");

        $add_post->bindValue(1, $text);

        $add_post->bindValue(2, $created_by);

        $add_post->bindValue(3, $thread_id);

        $add_post->execute();

        $conn->commit();

        $response['redirect'] = "index.php?page=show_thread&id=$thread_id&part=1";
    }
}

```



```

        $status = 201;
    }

    catch(PDOException $e){

        $status = 500;

        //$conn->rollBack();

        $response['result'] = $e->getMessage();

        log_error($e);

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

}

else {

    header("Location: ../index.php");

}

?>

```

## ADD\_TOPIC.PHP

```
<?php
```

```

header("Content-Type: application/json");

require "../config/config.php";

if($_SERVER['REQUEST_METHOD'] == "POST"){

    $title = $_POST['title'];

    $section_id = $_POST['section_id'];

    $description = $_POST['description'];

    $response = [];

```

```

try {
    $add_topic = $conn->prepare("INSERT INTO topics(title, created_by, section_id, description)
VALUES(?,?,?,?)");

    $add_topic->bindValue(1, $title);

    $add_topic->bindValue(2, $_SESSION['user_id']);

    $add_topic->bindValue(3, $section_id);

    $add_topic->bindValue(4, $description);

    $add_topic->execute();

    $response['result'] = 'added';

    $status = 201;
}

catch(PDOException $e){
    $status = 500;

    log_error($e);

    $response['result'] = 'Error: ' . $e->getMessage();
}

http_response_code($status);

echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}

?>

```

**BAN\_USER.PHP**

```

<?php

```

```

require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $id = $_POST['id'];

    $response = [];

    $check = getRow("SELECT * FROM users WHERE id = $id AND banned = 1");

    $is_banned = ($check == false? false : true);

    if(!$is_banned){

        try {

            $ban = $conn->prepare("UPDATE users SET banned = 1 WHERE id = ?");

            $ban->bindValue(1, $id);

            $ban->execute();

            $response['result'] = 'banned';

            $status = 201;

        } catch (PDOException $e) {

            $status = 500;

            $response['error'] = "Error: " . $e->getMessage();

            log_error($e);

        }

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

```

```
}
```

```
?>
```

## DELETE\_ROW.PHP

```
<?php
```

```
require "../config/config.php";
```

```
header("Content-Type: application/json");
```

```
if($_SERVER['REQUEST_METHOD'] == 'POST'){
```

```
    $response = [];
```

```
    $row_id = $_POST['row_id'];
```

```
    $table = $_POST['table'];
```

```
    switch ($table) {
```

```
        case 'users':
```

```
            $query = "DELETE FROM users WHERE id = ?";
```

```
            break;
```

```
        case 'forum_sections':
```

```
            $query = "DELETE FROM forum_sections WHERE id = ?";
```

```
            break;
```

```
        case 'topics':
```

```
            $query = "DELETE FROM topics WHERE id = ?";
```

```
            break;
```

```

        case 'threads':

            $query = "DELETE FROM threads WHERE id = ?";

            break;

        case 'posts':

            $query = "DELETE FROM posts WHERE id = ?";

            break;

    }

    try {

        $delete = $conn->prepare($query);

        $delete->bindValue(1, $row_id);

        $delete->execute();

        $response['result'] = 'Success';

        $status = 201;

    }catch(PDOException $e){

        $status = 500;

        log_error($e);

        $response['result'] = "Error: " . $e->getMessage();

    }

    http_response_code($status);

    echo json_encode($response);

}

else {

    header("Location: ../index.php");

}

```

?>

## EXCEL\_EXPORT2.PHP

```
<?php
```

```
require_once "../config/connection.php";
```

```
$latestUsers = retrieveData("SELECT * FROM users ORDER BY register_time");
```

```
$excel = new COM("Excel.Application");
```

```
$excel->Visible = 1;
```

```
$excel->DisplayAlerts = 1;
```

```
$workbook = $excel->Workbooks->Open("http://stankovic95.epizy.com/data/latest_user.xlsx") or  
die('Did not open filename');
```

```
$sheet = $workbook->Worksheets('Sheet1');
```

```
$sheet->activate;
```

```
$br = 1;
```

```
foreach($latestUsers as $user){
```

```
    $cell = $sheet->Range("A{$br}");
```

```
    $cell->activate;
```

```
    $cell->value = $user['id'];
```

```
    $cell = $sheet->Range("B{$br}");
```

```
    $cell->activate;
```

```
    $cell->value = $user['username'];
```

```
    $cell = $sheet->Range("C{$br}");
```

```

$cell->activate;

$cell->value = $user['email'];

$cell = $sheet->Range("D{$br}");

$cell->activate;

$cell->value = $user['register_time'];


$br++;
}

$cell = $sheet->Range("E{$br}");

$cell->activate;

$cell->value = count($latestUsers);


$workbook->_SaveAs("http://stankovic95.epizy.com/data/latest_user.xlsx", -4143);

$workbook->Save();


$workbook->Saved=true;

$workbook->Close;


$excel->Workbooks->Close();

$excel->Quit();


unset($sheet);

unset($workbook);

unset($excel);

```

?>

## EXPORT\_AUTHOR.PHP

<?php

```
require "../config/config.php";
```

```
try {
```

```
    $author = getRow("SELECT * FROM developers WHERE id = 1");
```

```
    //var_dump($author);
```

```
    $wordApp = new COM("Word.Application") or die("Unable to instantiate Word");
```

```
    $wordApp->Visible = true;
```

```
    $wordApp->Documents->Add();
```

```
    $wordApp->Selection->TypeText($author['full_name'] . "\n" . $author['short_bio'] . "\n" . "E-mail: " . $author['email'] . "\n" . "Location: " . $author['location'] . "\n" . "DOB: " . $author['dob']);
```

```
    $wordApp->Documents[1]->SaveAs("about_author.doc");
```

```
    header("../index.php?page=author");
```

```
}catch (PDOException $e){
```

```
    echo $e->getMessage();
```

```
    log_error($e);
```

```
}
```



?>

## FIND\_SECTION.PHP

<?php

```
require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $title = str_replace("_", "\_", $_POST['title']);

    $response = [];

    if($title != ""){

        try {

            $find_section = retrieveData("SELECT s.*, u.username FROM forum_sections s INNER JOIN
users u ON s.created_by = u.id WHERE title LIKE '$title%'");

            if(count($find_section) > 0){

                foreach($find_section as $row){

                    array_push($response, $row);

                }

            }else {

                $response['result'] = 'empty';

            }

            $status = 201;

        } catch (PDOException $e) {

            $status = 500;

            $response['result'] = "Error: " . $e->getMessage();

            log_error($e);

        }

    }

}
```

```

    }

    }else {

        $status = 201;

        $response['result'] = 'empty';

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

}

header("Location: ../index.php");

}

?>

```

## FIND\_THREAD.PHP

```

<?php

require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $title = str_replace("_", "\_", $_POST['title']);

    $response = [];

    if($title != ""){

        try {

            $find_thread = retrieveData("SELECT th.id, th.title, tp.title as topic, u.username FROM threads
th INNER JOIN topics tp ON th.topic_id = tp.id INNER JOIN users u ON th.created_by = u.id WHERE
th.title LIKE '$title%'");

```

```

        if(count($find_thread) > 0){

            foreach($find_thread as $row){

                array_push($response, $row);

            }

        }else {

            $response['result'] = 'empty';

        }

        $status = 201;

    } catch (PDOException $e) {

        log_error($e);

        $response['result'] = "Error: " . $e->getMessage();

        $status = 500;

    }

}

}else {

    $response['result'] = 'empty';

    $status = 201;

}

http_response_code($status);

echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}

```

## FIND\_TOPIC.PHP

<?php

```
require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $title = str_replace("_", "\_", $_POST['title']);

    $response = [];

    if($title != ""){

        try {

            $find_topic = retrieveData("SELECT t.id, t.title, t.created_by, s.title as section, u.username
FROM topics t INNER JOIN forum_sections s ON t.section_id = s.id INNER JOIN users u ON t.created_by =
u.id WHERE t.title LIKE '$title%'");

            if(count($find_topic) > 0){

                foreach($find_topic as $row){

                    array_push($response, $row);

                }

                $status = 201;

            }else {

                $status = 201;

                $response['result'] = 'empty';

            }

        } catch (PDOException $e) {

            $response['result'] = "Error: " . $e->getMessage();

            log_error($e);

            $status = 500;

        }

    }

}
```

```

    }

    }else {

        $status = 201;

        $response['result'] = 'empty';

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

}

header("Location: ../index.php");

}

?>

```

## FIND\_USER.PHP

```

<?php

require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    try {

        $username = str_replace("_", "\_", $_POST['username']);

        $result = $conn->query("SELECT u.*, r.role as userRole

                                FROM users u INNER JOIN user_roles r

                                ON u.role = r.id

                                WHERE u.username LIKE '$username%'");

        $response = [];
    }
}

```

```

foreach($result as $row){

    $posts = getRow("SELECT COUNT(id) as count FROM posts WHERE created_by = $row->id");

    $no_of_posts = $posts['count'];

    $row->posts = $no_of_posts;

    array_push($response, $row);

}

if($username == "" || $result->rowCount() == 0){

    $response['empty'] = true;

}

$status = 201;

} catch (PDOException $e) {

    $response['error'] = $e->getMessage();

    log_error($e);

    $status = 500;

}

$conn = NULL;

http_response_code($status);

echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}

?>

```

## GET\_OVERVIEW\_DATA.PHP

```
<?php
```

```

require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    try {

        //First part of the query

        $count_all = "SELECT COUNT(*) AS count FROM";

        //Queries the database to get total numbers of some attributes

        $conn->beginTransaction();

        $ttl_usr = getRow($count_all . " users");

        $ttl_ban = getRow($count_all . " users WHERE banned = 1");

        $ttl_pst = getRow($count_all . " posts");

        $ttl_thr = getRow($count_all . " threads");

        $ttl_onl = getRow($count_all . " users WHERE online_status = 1");

        //Getting the data about the latest users

        $latest_users = retrieveData("SELECT id, username, email, register_time FROM users ORDER BY
register_time DESC LIMIT 10");

        $conn->commit();

        //Data for admin panel overview. Total numbers of certain stuff...

        $overview_data = ['totalUsers' => $ttl_usr['count'],

```

```

        'totalPosts' => $ttl_pst['count'],
        'totalThreads' => $ttl_thr['count'],
        'currentlyOnline' => $ttl_onl['count'],
        'totalBanned' => $ttl_ban['count']);

//Preparing the data to be sent as an ajax response

$response = ['latestUsers' => $latest_users, 'otherData' => $overview_data];

$status = 201;

} catch (PDOException $e) {

    $response['error'] = "Database error: " . $e->getMessage();

    log_error($e);

    $status = 500;

}

//Closes the connection

$conn = NULL;

//Echoing the data

http_response_code($status);

echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}

```



?>

## GET\_SECTIONS\_FOR\_SELECT.PHP

<?php

```
require "../config/config.php";

header("Content-Type: application/json");

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    $response = [];

    try {

        $get_sections = retrieveData("SELECT * FROM forum_sections");

        foreach($get_sections as $section){

            array_push($response, $section);

        }

        $status = 201;

    } catch (PDOException $e) {

        $response['result'] = $e->getMessage();

        log_error($e);

        $status = 500;

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}
```

?>

## LOAD\_THREAD.PHP

<?php

```
header("Content-Type: application/json");

require "../config/config.php";

if($_SERVER['REQUEST_METHOD']){

    $thread_id = $_POST['thread_id'];

    $page_number = $_POST['page_number'];

    $response = ['data' => [], 'pagination_links' => [], 'current_page' => $page_number];

    try {

        //Pagination

        $posts_per_page = 10;

        $post_count = getRow("SELECT COUNT(*) as count FROM posts WHERE thread_id =
$thread_id")['count'];

        $needed_pages = ceil($post_count / $posts_per_page);

        $range_start = ($page_number - 1) * $posts_per_page;

        for($i = 1; $i <= $needed_pages; $i++){

            array_push($response['pagination_links'], "<button class='paginationButton btn' data-
part=$i>$i</button>");

        }

        //Getting data

        $posts = retrieveData("SELECT * FROM posts WHERE thread_id = $thread_id LIMIT $range_start,
$posts_per_page");

        $get_thread = getRow("SELECT * FROM threads WHERE id = $thread_id");
```

```

$thread_title = $get_thread['title'];

foreach($posts as $row){

    $user = getRow("SELECT * FROM users WHERE id = $row->created_by");

    $no_of_posts = getRow("SELECT COUNT(*) as count FROM posts WHERE created_by = $row->created_by")['count'];

    $post_data = [];

    $post_data['username'] = $user['username'];

    $post_data['online_status'] = $user['online_status'];

    $post_data['register_time'] = $user['register_time'];

    $post_data['no_of_posts'] = $no_of_posts;

    $post_data['avatar'] = $user['avatar'];

    $post_data['post_time'] = $row->time_posted;

    $post_data['content'] = $row->content;

    $post_data['post_id'] = $row->id;

    $post_data['user_id'] = $user['id'];

    $post_data['is_admin_or_mod'] = (isset($_SESSION['role']) && ($_SESSION['role'] == 1 || $_SESSION['role'] == 2)? true : false);

    $post_data['is_last_post'] = (getRow("SELECT COUNT(*) as count FROM posts WHERE thread_id = $thread_id")['count'] == 1? true : false);

    array_push($response['data'], $post_data);

}

$status = 201;

} catch (PDOException $e) {

    $response['error'] = "Error: " . $e->getMessage();

    log_error($e);

    $status = 500;

```

```

    }

    http_response_code($status);

    echo json_encode($response, JSON_PRETTY_PRINT);

} else {

    header("Location: ../index.php");

}

?>

```

## LOGIN.PHP

```

<?php

header("Content-Type: application/json");

require "../config/config.php";

if($_SERVER['REQUEST_METHOD'] == "POST"){

    $password = $_POST['password'];

    $email = $_POST['email'];

    $response = [];

    try {

        $findUser = $conn->prepare("SELECT * FROM users WHERE email = ?");

        $findUser->bindValue(1, $email);

        $findUser->execute();
    }
}

```

```

if($findUser->rowCount() > 0){

    $Row = $findUser->fetch(PDO::FETCH_ASSOC);

    $user = $Row['username'];

    $hash = $Row['password_hash'];

    $user_id = $Row['id'];

    $role = $Row['role'];

    $banned = $Row['banned'];

    if(password_verify($password, $hash)){

        $_SESSION['role'] = $role;

        $_SESSION['user_id'] = $user_id;

        $_SESSION['username'] = $user;

        $_SESSION['email'] = $email;

        $_SESSION['banned'] = $banned;

        $onlineStatus = $conn->prepare("UPDATE users SET online_status = 1 WHERE username =
?");

        $onlineStatus->bindValue(1, $_SESSION['username']);

        $onlineStatus->execute();

        $conn = NULL;

        $response['redirect'] = "index.php?page=home";

    }else {

        $response['error'] = 'Wrong password';

        mail($email, "Tech Talks Log in", "There was an attempt of logging into your tech talks
account with an incorrect password.");

    }

}
}

```

```

        $response['error'] = 'No such account';
    }

    $status = 201;
} catch (PDOException $e) {

    $status = 500;

    $response['error'] = $e->getMessage();

    log_error($e);
}

http_response_code($status);
echo json_encode($response);
}else {

    header("Location: ../index.php");
}

?>

```

## LOGOUT.PHP

```
<?php
```

```

require "../config/config.php";

try {

    $onlineStatus = $conn->prepare("UPDATE users SET online_status = 0 WHERE username = ?");

    $onlineStatus->bindValue(1, $_SESSION['username']);

    $onlineStatus->execute();
} catch (PDOException $e) {

    log_error($e);
}

```

```
die("Error: " . $e->getMessage());  
}
```

```
session_start();  
  
session_destroy();  
  
header("Location: ../index.php");
```

?>

## REGISTER.PHP

<?php

```
header("Content-Type: application/json");  
  
require "../config/config.php";
```

```
if($_SERVER['REQUEST_METHOD'] == "POST"){
```

```
    $username = $_POST['username'];
```

```
    $password = $_POST['password'];
```

```
    $email = $_POST['email'];
```

```
    $usernameRegex = "/^[a-zA-Z0-9_]{3,20}$/";
```

```
    $emailRegex = "/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,6})+$/";
```

```
    $usernameOk = (preg_match($usernameRegex, $username)? true : false);
```

```
    $emailOk = (preg_match($emailRegex, $email)? true : false);
```

```
    $passwordOk = (strlen($password) > 8? true: false);
```

```

$response = [];

if($usernameOk && $emailOk && $passwordOk){

    try {

        $register = $conn->prepare("INSERT INTO users (username, email, password_hash) VALUES (?, ?, ?)");

        $register->bindValue(1, $username);

        $register->bindValue(2, $email);

        $register->bindValue(3, password_hash($password, PASSWORD_DEFAULT));

        $register->execute();

        $response['result'] = "success";

        $status = 501;

    } catch (PDOException $e) {

        $status = 500;

        $response['error'] = $e->getMessage();

        log_error($e);

    }

}

}

else {

    $response['validationError'] = true;

}

echo json_encode($response, JSON_PRETTY_PRINT);

}

else {

    header("Location: ../index.php");

}

```



?>

## UPDATE\_ROLE.PHP

<?php

```
include "../config/config.php";
```

```
if($_SERVER['REQUEST_METHOD'] == 'POST'){
```

```
    $user_id = $_POST['user_id'];
```

```
    $role_id = $_POST['role_id'];
```

```
    $response = [];
```

```
    try {
```

```
        $update = $conn->prepare("UPDATE users SET role = ? WHERE id = ?");
```

```
        $update->bindValue(1, $role_id);
```

```
        $update->bindValue(2, $user_id);
```

```
        $update->execute();
```

```
        $status = 201;
```

```
        $response['result'] = 'Success';
```

```
    } catch (PDOException $e) {
```

```
        $status = 500;
```

```
        $response['result'] = "Error: " . $e->getMessage();
```

```
        log_error($e);
```

```
    }
```

```
    http_response_code($status);
```

```
    echo json_encode($response, JSON_PRETTY_PRINT);
```

```
}else {  
  
    header("Location: ../index.php");  
  
}
```

?>

## UPDATE\_USER.PHP

```
<?php  
  
include "../config/config.php";  
  
if($_SERVER['REQUEST_METHOD'] == 'POST'){  
  
    $user_id = $_SESSION['user_id'];  
  
  
    //Gets data about the image  
  
    $file = $_FILES['avatar_input'];  
  
    $file_name = $file['name'];  
  
    $tmp_name = $file['tmp_name'];  
  
    $file_size = $file['size'];  
  
    $file_type = $file['type'];  
  
    $file_error = $file['error'];  
  
    //If the user has chosen a file  
  
    if(!empty($file_name)){  
  
        $file_name_explored = explode(".", $file_name);  
  
        $file_extension = end($file_name_explored);  
  
        $upload_dir = '../assets/images/avatars/';  
  
        $new_file_name = $user_id . "." . $file_extension;  
  
        $file_path = $upload_dir . $new_file_name;
```

```
$crop_dimensions = 250;
```

```
try {  
    prepare_image($tmp_name, $crop_dimensions, $file_path, $file_type);  
} catch (Exception $e) {  
    log_error($e);  
    die ("Error: " . $e->getMessage());  
}
```

```
try {  
    $update_avatar = $conn->prepare("UPDATE users SET avatar = ? WHERE id = ?");  
    $update_avatar->bindValue(1, $new_file_name);  
    $update_avatar->bindValue(2, $user_id);  
    $update_avatar->execute();  
} catch(PDOException $e) {  
    log_error($e);  
    die ("Error: " . $e->getMessage());  
}  
}
```

```
try {  
    $update_bio = $conn->prepare("UPDATE users SET short_bio = ? WHERE id = ?");  
    $update_bio->bindValue(1, $_POST['shortBioInput']);  
    $update_bio->bindValue(2, $user_id);  
    $update_bio->execute();  
}
```

```

    } catch (PDOException $e) {

        log_error($e);

        die ("Error: " . $e->getMessage());
    }

```

```

header("Location: ../index.php?page=account_settings");

```

```

}else {

    header("Location: ../index.php");

}

```

```

?>

```

## USERNAME\_AND\_EMAIL\_CHECK.PHP

```

<?php

```

```

header("Content-Type: application/json");

```

```

require "../config/config.php";

```

```

if($_SERVER['REQUEST_METHOD'] == "POST"){

```

```

    $whatToCheck = $_POST['whatToCheck'];

```

```

    $value = $_POST['value'];

```

```

    $query = ($whatToCheck == 'username'? "SELECT * FROM users WHERE username = ?" : "SELECT *
FROM users WHERE email = ?");

```

```

    $response = [];

```

```

try {

    $check = $conn->prepare($query);

    $check->bindValue(1, $value);

    $check->execute();

    $response['exists'] = ($check->rowCount() > 0? true: false);

    $status = 201;

} catch(PDOException $e){

    $response['error'] = "Error: " . $e->getMessage();

    $status = 500;

    log_error($e);

}

http_response_code($status);

echo json_encode($response, JSON_PRETTY_PRINT);

}else {

    header("Location: ../index.php");

}

?>

```

## **.htaccess (ostali, ispod root-a)**

Deny from all

## **SITEMAP.XML**

```

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

    <url>

        <loc>http://techtalksict.epizy.com/index.php?page=home</loc>

        <lastmod>2019-06-15</lastmod>

        <changefreq>monthly</changefreq>

        <priority>1</priority>

```

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=admin</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=show\_user</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=author</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=topic</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=show\_thread</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=new\_thread</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=new\_post</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

<loc>http://techtalksict.epizy.com/index.php?page=account\_settings</loc>

<lastmod>2019-06-15</lastmod>

<changefreq>monthly</changefreq>

<priority>0.5</priority>

</url>

<url>

```

        <loc>http://techtalksict.epizy.com/index.php?page=not_found</loc>

        <lastmod>2019-06-15</lastmod>

        <changefreq>monthly</changefreq>

        <priority>0.3</priority>
    </url>

    <url>

        <loc>http://techtalksict.epizy.com/index.php?page=access_forbidden</loc>

        <lastmod>2019-06-15</lastmod>

        <changefreq>monthly</changefreq>

        <priority>0.3</priority>
    </url>
</urlset>

```

## FUNCTIONS.PHP

```

<?php

function redirect($page){
    header("Location: index.php?page=$page");
}

function forceIndex(){
    if(!isset($_SESSION['username'])){
        redirect('home');
    }
}

function protectPageFromGetRequest(){

```



```

if(!isset($_GET['page']) && $_SERVER['PHP_SELF'] != '/techtalks/index.php'){
    header("../index.php");
}
}

```

```

function adminOnly(){
    if($_SESSION['role'] != 1){
        redirect('home');
    }
}

```

```

function retrieveData($query){
    global $conn;

    $getData = $conn->query($query);

    return $getData->fetchAll();
}

```

```

function getUserData($id){
    global $conn;

    $getData = $conn->query("SELECT * FROM users WHERE id = $id");

    return $getData->fetch();
}

```

```

function getRow($query){
    global $conn;

```

```

$getRow = $conn->query($query);

if($getRow->rowCount() == 0){

    return false;

}else {

    return $getRow->fetch(PDO::FETCH_ASSOC);

}

}

```

```

function getUserRole($role_id){

    global $conn;

    $getRole = $conn->query("SELECT role FROM user_roles WHERE id = $role_id");

    return ucfirst($getRole->fetch()->role);

}

```

```

function getUsersPosts($user_id, $no_of_posts){

    global $conn;

    $posts = retrieveData("SELECT * FROM posts WHERE created_by = $user_id ORDER BY time_posted
DESC LIMIT $no_of_posts");

    return $posts;

}

```

```

function page_access_log(){

    @ $log_file = fopen("data/log.txt", "ab");

    if(!$log_file){

        echo "Something went wrong";

    }else {

```

```

    @ $user = (null != $_SESSION['username'])?$_SESSION['username']:'guest');

    $date = date("d-M-Y");

    $time = date("H:i:s");

    $page_code = $_GET['page'];

    if(!isset($_GET['page'])){

        $page_code = 'index';

    }

    $ip_address = $_SERVER['REMOTE_ADDR'];

    fwrite($log_file, "Page access: [$page_code, User: $user, IP: $ip_address, Timestamp: $date
$time] [endline]\r\n");

    fclose($log_file);

}

}

```

```

function log_error($e){

    @ $log_file = fopen("../data/log.txt", "a");

    if(!$log_file){

        echo "Something went wrong";

    }else {

        @ $user = (null != $_SESSION['username'])?$_SESSION['username']:'guest');

        $date = date("d-M-Y");

        $time = date("H:i:s");

        $error_code = $e->getCode();

        $error_message = $e->getMessage();

        $ip_address = $_SERVER['REMOTE_ADDR'];

        fwrite($log_file, "Error: [Code = $error_code; Message = $error_message] [endline]\r\n");
    }
}

```

```

        fclose($log_file);
    }
}

function prepare_image($tmp_name, $crop_dimensions, $file_path, $file_type) {

    list($src_width, $src_height) = getimagesize($tmp_name);

    $image_orientation = ($src_width == $src_height? 'square' : ($src_width > $src_height? 'landscape'
: 'portrait'));

    if($image_orientation == 'portrait'){

        //Nova dužina

        $new_width = $crop_dimensions;

        //Proverava za koliko procenata se slika mora suziti da bi bila jednaka novoj dužini

        $resize_percentage = round(100 - $new_width / $src_width * 100);

        //Za toliko procenata ćemo smanjiti i visinu

        $new_height = round($src_height - $src_height * $resize_percentage / 100);

        switch($file_type){

            case "image/jpeg":

                $img_handler = imagecreatefromjpeg($tmp_name);

                break;

            case "image/png":

                $img_handler = imagecreatefrompng($tmp_name);

                break;

        }
    }
}

```

```

$empty_to_resize = imagecreatetruecolor($new_width, $new_height);

imagecopyresized($empty_to_resize, $img_handler, 0, 0, 0, 0, $new_width, $new_height,
$src_width, $src_height);

$resized = $empty_to_resize;

$empty_to_crop = imagecreatetruecolor($crop_dimensions, $crop_dimensions);

imagecopyresampled($empty_to_crop, $resized, 0, 0, 0, ($new_height - $crop_dimensions) / 2,
$crop_dimensions, $crop_dimensions, $crop_dimensions, $crop_dimensions);

$cropped_image = $empty_to_crop;

$compression = 100;

switch($file_type){

    case "image/jpeg":

        imagejpeg($cropped_image, $file_path, $compression);

        break;

    case "image/png":

        imagepng($cropped_image, $file_path);

        break;

}

}else if($image_orientation == 'landscape'){

    //Nova dužina

    $new_height = $crop_dimensions;

    //Proverava za koliko procenata se slika mora suziti da bi bila jednaka novoj dužini

    $resize_percentage = round(100 - $new_height / $src_height * 100);

    //Za toliko procenata ćemo smanjiti i visinu

    $new_width = round($src_width - $src_width * $resize_percentage / 100);

```

```

switch($file_type){

    case "image/jpeg":

        $img_handler = imagecreatefromjpeg($tmp_name);

        break;

    case "image/png":

        $img_handler = imagecreatefrompng($tmp_name);

        break;

}

$empty_to_resize = imagecreatetruecolor($new_width, $new_height);

imagecopyresized($empty_to_resize, $img_handler, 0, 0, 0, 0, $new_width, $new_height,
$src_width, $src_height);

$resized = $empty_to_resize;

$empty_to_crop = imagecreatetruecolor($crop_dimensions, $crop_dimensions);

imagecopyresampled($empty_to_crop, $resized, 0, 0, ($new_width - $crop_dimensions) / 2, 0,
$crop_dimensions, $crop_dimensions, $crop_dimensions, $crop_dimensions);

$cropped_image = $empty_to_crop;

$compression = 100;

switch($file_type){

    case "image/jpeg":

        imagejpeg($cropped_image, $file_path, $compression);

        break;

    case "image/png":

```

```

        imagepng($cropped_image, $file_path);

        break;
    }
}
else {
    switch($file_type){
        case "image/jpeg":
            $img_handler = imagecreatefromjpeg($tmp_name);

            break;

        case "image/png":
            $img_handler = imagecreatefrompng($tmp_name);

            break;
    }

    $empty_to_resize = imagecreatetruecolor($crop_dimensions, $crop_dimensions);

    imagecopyresized($empty_to_resize, $img_handler, 0, 0, 0, 0, $crop_dimensions,
    $crop_dimensions, $src_width, $src_height);

    $resized = $empty_to_resize;

    $compression = 100;

    switch($file_type){
        case "image/jpeg":
            imagejpeg($resized, $file_path, $compression);

            break;

        case "image/png":
            imagepng($resized, $file_path);

            break;
    }
}

```

```
}  
}
```

```
function readLogFile(){  
    $log_file = fopen("data/log.txt", 'r');  
    $content = "";  
    if($log_file){  
  
        while(!feof($log_file)){  
            $buffer = fread($log_file, 4096);  
            $content = $content . $buffer;  
        }  
  
        fclose($log_file);  
  
        $logs = explode("[endline]\r\n", $content);  
        return $logs;  
    }  
}
```

```
function mark_visit($page, $time){  
    global $totalVisits, $pages;  
    $yesterday = time() - 84600;  
    if($time > $yesterday){  
        $pages[$page][0]++;  
    }  
}
```



```

    }

    $pages[$page][1]++;

    $totalVisits++;
}

```

```

function show_page_visits($pages){

    global $totalVisits;

    echo "<table class=\"customTable\">

        <thead>

            <tr>

                <th>Page</th>

                <th>Last 24h</th>

                <th>Total</th>

                <th>Percentage</th>

            </tr>

        </thead>

        <tbody>";

    foreach($pages as $page){

        $page[2] = $page[1] / $totalVisits * 100;

        echo "<tr>

            <td>" . $page[3] . "</td>

            <td>" . $page[0] . "</td>

            <td>" . $page[1] . "</td>

            <td>" . number_format($page[2], 2, '.', '') . "%</td>

```

```

        </tr>";
    }

    echo "</tbody></table>";
}

?>

```

## DBCONN.PHP

```
<?php
```

```

/*$host = 'sql308.epizy.com';

$dbname = "epiz_24006859_techtalks";

$user = "epiz_24006859";

$pass = "lggRNN5Ry2AID8y";*/

// Ostala podesavanja
define("ENV_FAJL", "http://" . $_SERVER['HTTP_HOST'] . "/config/.env");
define("LOG_FAJL", "http://" . $_SERVER['HTTP_HOST'] . "/data/log.txt");

// Podesavanja za bazu
define("SERVER", env("SERVER"));
define("DATABASE", env("DBNAME"));
define("USERNAME", env("USERNAME"));
define("PASSWORD", env("PASSWORD"));

function env($name){

    $open = fopen(ENV_FAJL, "r");

    $data = file(ENV_FAJL);

    $value = "";

```

```

        foreach($data as $row){

            $divided = explode("=", $row);

            if($divided[0] == $name){

                return trim($divided[1]);

            }

        }

    }

    try {

        $conn = new PDO("mysql:host=".SERVER.";dbname=".DATABASE, USERNAME, PASSWORD);

        $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    }catch(PDOException $e){

        log_error($e);

        echo $e->getMessage();

        die("<h2 style='text-align:center'>Couldn't connect to the database. Rest assured we're working on
it.</h2><img style='display:block;margin:auto' src='assets/images/error.jpg'>");

    }

?>

```

## CONFIG.PHP

```
<?php
```

```

/*****

require "dbconn.php";

require "functions.php";

if(!isset($_SESSION)){

```

```

//session_set_cookie_params(1800,"/");

/*$newVisit = $conn->prepare("INSERT INTO visits (ip) VALUES (?)");

$newVisit->bindValue(1, $_SERVER['REMOTE_ADDR']);

$newVisit->execute();*/

session_start();

$status;

}

```

?>

## **.ENV**

DBNAME=epiz\_24049202\_techtalks

SERVER=sql208.epizy.com

USERNAME=epiz\_24049202

PASSWORD=5edkqS8fbVFIP4

## **MAIN.CSS**

```

* {

    padding: 0;

    margin: 0;

    box-sizing: border-box;

}

```

```

body, html {

    /*height: 100%;*/

    width: 100%;

    height: 100%;

}

```

```
.errorCode {  
    font-size: 300px;  
    text-align: center;  
}
```

```
.errorMessageCustom {  
    font-size: 20px;  
    text-align: center;  
    padding: 40px;  
}
```

```
.errorMessageGeneric {  
    padding: 40px;  
    font-size: 50px;  
    text-align: center;  
}
```

```
.postData {  
    display: flex;  
    flex-direction: row;  
    justify-content: space-between;  
    align-items: center;  
}
```

```
.exportLink {  
    margin: auto;  
    padding: 10px;  
}
```

```
.excel {  
    color: #207244;  
}
```

```
.word {  
    color: #2A5599;  
}
```

```
.postData p {  
    padding-left: 5px;  
}
```

```
#pageNumber p {  
    text-align: center;  
    padding: 5px;  
}
```

```
#paginationLinks {  
    display: flex;
```

```
flex-direction: row;

justify-content: center;

}
```

```
#logFileDiv {

display: flex;

flex-direction: column;

align-items: center;

padding: 40px;

}
```

```
#givePrivDiv {

position: absolute;

padding: 10px;

background-color: #ccc;

display: none;

flex-direction: column;

align-items: center;

width: 300px;

left: 50%;

transform: translateX(-50%);

}
```

```
#givePrivDiv select {

padding: 5px;
```

```
    margin: 10px;
}
```

```
#givePrivDiv p {
    text-align: justify;
    word-wrap: break-word;
}
```

```
#logFileDiv h3 {
    padding: 10px;
}
```

```
#log {
    height: 400px;
    width: 600px;
    overflow-y: scroll;
    background-color: #000;
    color: #fff;
    padding: 5px 10px;
}
```

```
.paginationButton {
    border: 1px solid #000;
}
```



```
.banMsg {  
    text-align: center;  
    padding: 5px;  
}
```

```
.btn {  
    padding: 5px;  
    margin: 5px;  
    border-radius: 2px;  
    border: none;  
    cursor: pointer;  
}
```

```
#pageAccess {  
    padding-top: 30px;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}
```

```
.delBtn {  
    background-color: #CA2323;  
}
```

```
.addBtn {
```

```
background-color: #2997F9;  
}
```

```
#newPostBtn {  
background-color: #1189FE;  
position: relative;  
left: 50%;  
transform: translateX(-50%);  
}
```

```
.btn-dark {  
color: #fff;  
}
```

```
::-webkit-scrollbar {  
display: none;  
}
```

```
.fa-circle {  
color: #63A924;  
}
```

```
#avatarSettings {  
display: flex;  
flex-direction: row;
```

```
    align-items: center;
    padding: 30px;
}
```

```
.avatarHolder {
    width: 250px;
    height: 250px;
    border-radius: 100%;
    overflow: hidden;
    margin-right: 20px;
}
```

```
.avatarHolder img {
    width: 250px;
}
```

```
#latestUserPosts {
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 10px;
}
```

```
#latestUserPosts h3 {
    padding-bottom: 20px;
```

```
}
```

```
#showUserHolder {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    padding-top: 10px;  
}
```

```
#avatarAndBio {  
    display: flex;  
    flex-direction: row;  
    align-items: center;  
    padding: 20px;  
}
```

```
#showUserData {  
    padding: 20px;  
    font-size: 20px;  
}
```

```
#avatarAndBio p {  
    width: 400px;  
    text-align: justify;  
    color: #444;
```

```
font-size: 20px;  
padding: 20px;  
}
```

```
#avatarAndBio img {  
    width: 250px;  
    height: 250px;  
    border-radius: 100%;  
}
```

```
#accountSettingsHolder {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    padding: 10px;  
}
```

```
#accountSettingsHolder form {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}
```

```
#accountSettingsHolder input {  
    padding: 5px;
```

```
}
```

```
body {  
    background: url('../images/bcg.jpg');  
    background-position: center;  
    background-attachment: fixed;  
    background-size: cover;  
}
```

```
#wrapper {  
    visibility: visible;  
    width: 1200px;  
    margin: auto;  
}
```

```
#main {  
    background-color: #eee;  
    font-family: 'Ropa Sans', 'Arial';  
    height: auto;  
}
```

```
.total-users {  
    background-color: #27A9E3;  
}
```

```
.total-threads {  
    background-color: #28B779;  
}
```

```
.total-messages {  
    background-color: #E23636;  
}
```

```
.total-posts {  
    background-color: #FFB849;  
}
```

```
#header {  
    width: 100%;  
    background-color: #3B4044;  
    display: flex;  
    flex-direction: row;  
    justify-content: space-between;  
    align-items: center;  
    padding: 5px;  
}
```

```
#logoHolder {  
    display: inline;  
}
```

```
#logoHolder img {  
    width: 200px;  
    display: block;  
    margin: 0;  
}
```

```
#nav ul {  
    list-style-type: none;  
    display: flex;  
    flex-direction: row;  
}
```

```
#nav ul li a {  
    color: #73D2FF;  
    font-family: 'Ropa Sans', 'Arial';  
    text-decoration: none;  
    margin: 0 5px;  
    font-size: 20px;  
    transition: background-color 0.3s, color 0.3s;  
    border-radius: 2px;  
    padding: 5px;  
}
```

```
#nav ul li a:hover {
```



```
    color: #2d2d2d;

    background-color: #73D2FF;
}
```

```
#welcomeMessage, .forumSection {

    background-color: #eee;
}
```

```
#welcomeMessageTop, .sectionHead {

    color: #fff;

    background-color: #000000;

    font-family: 'Ropa Sans', 'Arial';

    padding: 5px;
}
```

```
.sectionHead, .sectionTitle {

    display: flex;

    flex-direction: row;
}
```

```
#welcomeMessageBottom {

    padding: 5px;

    font-family: 'Arial';
}
```

```
.topic {  
    padding: 5px;  
    height: 70px;  
    display: flex;  
    flex-direction: row;  
    font-family: 'Ropa Sans', 'Arial';  
    background: linear-gradient(180deg, #fff, #eee);  
    font-size: 18px;  
}
```

```
.topic:hover {  
    background: linear-gradient(180deg, #fff, #ccc);  
}
```

```
.topic a {  
    text-decoration: none;  
    font-weight: bold;  
    color: #000;  
}
```

```
.topic a:hover {  
    text-decoration: underline;  
}
```

```
.topicIcon {
```

```
display: flex;

flex-direction: row;

align-items: center;

}
```

```
.topicIcon img {

width: 40px;

height: 40px;

display: block;

}
```

```
.topicTitleAndDescription {

display: flex;

flex-direction: column;

padding-left: 5px;

justify-content: center;

}
```

```
.sectionTitle {

width: 70%;

}
```

```
.postsLabel {

width: 15%;

border-left: 1px solid #fff;
```

```
padding-left: 5px;

display: flex;

flex-direction: column;

justify-content: center;
}
```

```
.lastPostLabel {

padding-left: 5px;

border-left: 1px solid #fff;
}
```

```
.topicDescription p {

font-family: 'Ropa Sans', 'Arial';
}
```

```
.topicDescription a {

color: #129DF8;
}
```

```
.post {

display: flex;

flex-direction: column;

background-color: #fff;
}
```

```
.postTop {  
    background-color: #026BA1;  
    color: #fff;  
}
```

```
.postTop p {  
    padding: 2px;  
}
```

```
.postBottom {  
    display: flex;  
    flex-direction: row;  
    justify-content: flex-start;  
}
```

```
.postLeft img {  
    width: 60px;  
    height: 60px;  
    border-radius: 100%;  
}
```

```
.postLeft, .postRight {  
    background-color: #fff;  
    padding: 10px;  
}
```

```
.postRight {  
    width: 600px;  
}
```

```
.postLeft {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    text-align: center;  
    width: 200px;  
    background-color: #F5F5F5;  
}
```

```
.threadHead {  
    background-color: #000;  
    color: #fff;  
    padding: 5px;  
}
```

```
#loginForm, #registerForm {  
    position: fixed;  
    top: 10%;  
    left: 50%;  
    transform: translateX(-50%);
```

```
display: none;

flex-direction: row;

box-shadow: 0px 0px 20px #000;
}
```

```
#registerForm {

    visibility: visible;

}
```

```
#loginImage, #registerImage {

    background: url('../images/lock.jpg');

    background-position: center;

    background-size: cover;

    width: 300px;

    height: auto;

}
```

```
#loginImage p, #registerImage p {

    color: #fff;

    text-align: center;

    font-size: 20px;

    font-family: 'Ropa Sans', 'Arial';

    padding: 30px;

}
```

```
.formTitle {  
    font-size: 20px;  
    margin: 20px 0;  
}
```

```
.fa-window-close {  
    color: #B80000;  
    cursor: pointer;  
    float: right;  
    padding: 5px;  
    font-size: 20px;  
}
```

```
#loginForm form, #registerForm form {  
    font-family: 'Ropa Sans', 'Arial';  
    background-color: #F4F7F5;  
    padding: 10px;  
    border-radius: 2px;  
}
```

```
#loginForm form a, #registerForm form a {  
    color: #004151;  
    text-decoration: none;  
}
```



```
form a:hover {  
    text-decoration: underline;  
}
```

```
.formLinkHolder {  
    display: flex;  
    flex-direction: row;  
    justify-content: space-between;  
    margin-bottom: 20px;  
}
```

```
.inputHolder i {  
    color: #757575;  
    padding-left: 5px;  
}
```

```
.inputHolder:first-child {  
    margin-top: 0;  
}
```

```
.inputHolder {  
    border-bottom: 1px solid gray;  
    padding-bottom: 5px;  
    margin-top: 20px;  
}
```

```
.tosHolder {  
    margin-top: 25px;  
}
```

```
#loginForm input, #registerForm input {  
    outline: none;  
    border: none;  
    font-size: 15px;  
    background-color: transparent;  
    padding-top: 5px;  
    color: #757575;  
}
```

```
#loginBtn, #registerBtn {  
    padding: 10px;  
    outline: none;  
    border: none;  
    margin: 20px 0;  
    border-radius: 5px;  
    color: #fff;  
    background-color: #246977;  
    cursor: pointer;  
    transition: background-color 0.3s;  
}
```

```
#loginForm form, #registerForm form {  
    width: 240px;  
}
```

```
#loginBtn:hover, #registerBtn:hover {  
    background-color: #004151;  
}
```

```
.validationError {  
    color: #FB3B41;  
    padding-top: 2px;  
    word-wrap: break-word;  
    margin-bottom: 10px;  
}
```

```
#dashboard {  
    background-color: #3B4044;  
    display: flex;  
    color: #fff;  
    flex-direction: row;  
    font-family: 'Ropa Sans', 'Arial';  
    justify-content: space-between;  
    align-items: center;  
    border-bottom: 2px solid #777;
```

```
}
```

```
#dashboardLinks {  
    display: flex;  
    flex-direction: row;  
    align-items: center;  
}
```

```
#dashboard p {  
    padding: 4px;  
}
```

```
#dashboardLinks a {  
    text-decoration: none;  
    padding: 4px;  
    color: #fff;  
    transition: background-color 0.3s;  
}
```

```
#dashboardLinks a:hover {  
    background-color: #777777;  
}
```

```
#dashboardLinks img {  
    width: 20px;
```

```
    display: block;
}
```

```
/*ADMIN*/
```

```
#adminHolder {
    padding: 10px;
    display: flex;
    flex-direction: row;
}
```

```
.adminPanel {
    margin-left: 10px;
    display: none;
    flex-direction: column;
    width: 970px;
    border-radius: 4px;
    overflow: hidden;
}
```

```
.adminPanelHead {
    background-color: #3B4044;
    padding: 10px;
    color: #fff;
}
```

```
.adminPanelContent {  
    background-color: #fff;  
    padding: 10px;  
}
```

```
#adminPanelsNav {  
    display: inline-block;  
    width: 200px;  
    background-color: #fff;  
    border-radius: 4px;  
    overflow: hidden;  
}
```

```
.adminPanelsHead {  
    background-color: #026BA1;  
    color: #fff;  
    padding: 10px;  
    cursor: default;  
}
```

```
.adminPanelsLink {  
    padding: 10px;  
    color: #555;  
}
```

```
.adminPanelsLink:nth-child(1) {  
    color: red;  
}
```

```
.adminPanelsLinkIcon {  
    display: inline-block;  
    width: 30px;  
}
```

```
.adminPanelsLink:hover {  
    color: #000;  
    cursor: pointer;  
}
```

```
/*ADMIN OVERVIEW*/
```

```
#overviewData {  
    display: grid;  
    grid-template-columns: 200px 200px 200px;  
    grid-row-gap: 50px;  
    grid-column-gap: 50px;  
    justify-content: center;  
    padding-top: 30px;  
}
```

```
.overviewDataItem {  
    width: 200px;  
    color: #fff;  
    text-align: center;  
}
```

```
.overviewDataItem:first-child {  
    background-color: #2CA6FF;  
}
```

```
.overviewDataItem:nth-child(2) {  
    background-color: #EB3427;  
}
```

```
.overviewDataItem:nth-child(3) {  
    background-color: #FDC006;  
}
```

```
.overviewDataItem:nth-child(4) {  
    background-color: #BCD64E;  
}
```

```
.overviewDataItem:nth-child(5) {  
    background-color: #8320AD;  
}
```



```
}
```

```
.overviewDataItem:last-child {  
    background-color: #F76B05;  
}
```

```
.overviewDataItemHead {  
    background-color: rgba(0, 0, 0, 0.2);  
    padding: 5px;  
}
```

```
.overviewDataItemBottom {  
    display: flex;  
    flex-direction: row;  
    font-size: 40px;  
    padding: 20px;  
}
```

```
.overviewDataItemBottom div {  
    width: 100px;  
}
```

```
#overviewBottom {  
    display: flex;  
    flex-direction: row;
```

```
    justify-content: space-around;
    padding-top: 40px;
}
```

```
.customTable {
    font-family: 'Arial';
    font-size: 17px;
    margin-top: 10px;
}
```

```
.customTable h3 {
    text-align: center;
    padding: 20px;
}
```

```
.customTable td, .customTable th {
    padding: 7px;
}
```

```
.customTable thead {
    background-color: #2d2d2d;
    color: #fff;
}
```

```
.customTable tbody tr:nth-child(odd) {
```

```

    background-color: #efefef;
}

.adminSearchBar, .addContent input[type=text], #newThreadTitle {
    padding: 10px;
    border: none;
    box-shadow: 1px 1px 2px gray;
    font-family: 'Arial';
    margin-top: 10px;
}

#newThreadDiv {
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 10px;
}

#newThreadDiv form {
    display: flex;
    flex-direction: column;
    align-items: center;
}

textarea {

```

```
padding: 10px;

font-family: 'Arial';

margin: 10px 0;

border: none;

box-shadow: 2px 2px 2px gray;

resize: vertical;

}
```

```
#postAnswer {

    display: flex;

    flex-direction: column;

    align-items: center;

    padding: 10px;

}
```

```
#selectSection {

    padding: 8px;

}
```

```
.addContent {

    margin-bottom: 10px;

    display: flex;

    flex-direction: column;

    align-items: center;

}
```

```
.addContent h3 {  
    color: #000;  
    margin-bottom: 5px;  
}
```

```
.addContent input, .addContent select {  
    padding: 3px;  
    margin-bottom: 10px;  
}
```

```
.adminPanelContentHolder {  
    box-shadow: 2px 2px 4px #aaa;  
    padding: 10px;  
    margin-bottom: 20px;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}
```

```
#addSectionForm, #addTopicForm {  
    display: flex;  
    flex-direction: column;  
    align-items: baseline;  
}
```

```
.adminPanelContentHolder select, .adminPanelContentHolder input {  
    width: 200px;  
}
```

```
/*FOOTER*/
```

```
#footer {  
    font-family: 'Ropa Sans', 'Arial';  
    background-color: #1d1d1d;  
    width: 100%;  
    padding: 40px;  
    color: #fff;  
}
```

```
#footerTop {  
    display: flex;  
    flex-direction: row;  
    justify-content: space-around;  
    align-items: center;  
    width: 600px;  
    margin: auto;  
}
```

```
#footerPages ul, #footerContact ul {
```

```
list-style-type: none;  
}
```

```
#footerContact {  
    color: #ccc;  
}
```

```
#footerPages a {  
    color: #fff;  
    text-decoration: none;  
}
```

```
#footerPages a:hover {  
    text-decoration: underline;  
}
```

```
#footerSocial i {  
    font-size: 30px;  
    color: #aaa;  
    margin: 5px;  
    transition: color 0.3s;  
}
```

```
#footerSocial i:hover {  
    color: #fff;
```

```
}
```

```
#footerBottom p {  
    text-align: center;  
    padding-top: 20px;  
    color: #ccc;  
}
```

```
#drawer {  
    display: none;  
    background-color: #eee;  
    text-align: center;  
}
```

```
#drawer ul {  
    list-style-type: none;  
}
```

```
#drawer a {  
    text-decoration: none;  
    font-family: 'Ropa Sans';  
    color: #242424;  
    font-size: 30px;  
}
```



```
#drawer li {  
    padding: 8px;  
    background: linear-gradient(0deg, #ddd, #fff);  
}
```

```
#burger {  
    display: none;  
    cursor: pointer;  
}
```

```
#burger i {  
    font-size: 30px;  
}
```

```
@media only screen and (max-width: 1220px) {  
    #wrapper {  
        width: 100%;  
    }  
}
```

```
@media only screen and (max-width: 1000px) {  
    #adminPanelsNav {  
        width: 100%;  
    }  
}
```

```

/*#usersPanel, #contentPanel {
    display: flex;
}*/

#adminHolder {
    flex-direction: column;
}

.adminPanel {
    margin: 0;
    width: 100%;
}
}

@media only screen and (max-width: 910px) {
    .customTable {
        font-size: 13px;
    }
}

@media only screen and (max-width: 750px) {
    #overviewData {
        grid-row-gap: 5px;
        grid-column-gap: 5px;
    }
}

```

```
#footerTop {  
    display: flex;  
    flex-direction: column;  
    width: auto;  
    text-align: center;  
}
```

```
.obsolete {  
    display: none;  
}
```

```
#avatarAndBio {  
    flex-direction: column;  
    width: auto;  
}
```

```
#avatarSettings {  
    flex-direction: column;  
}
```

```
#accountSettingsHolder textarea {  
    resize: horizontal;  
    width: 500px;  
}
```

```
}
```

```
@media only screen and (max-width: 650px) {
```

```
  .overviewDataItem {
```

```
    width: 150px;
```

```
  }
```

```
#postAnswerText, #newThreadPost {
```

```
  resize: both;
```

```
  width: 100%;
```

```
}
```

```
#overviewData {
```

```
  grid-template-columns: 150px 150px 150px;
```

```
}
```

```
#logFileDiv {
```

```
  display: none;
```

```
}
```

```
}
```

```
@media only screen and (max-width: 570px) {
```

```
  #loginImage, #registerImage {
```

```
    display: none;
```

```
}
```

```
}
```

```
@media only screen and (max-width: 520px) {
```

```
  #accountSettingsHolder textarea {
```

```
    resize: horizontal;
```

```
    width: 350px;
```

```
  }
```

```
p {
```

```
  word-wrap: break-word;
```

```
}
```

```
}
```

```
@media only screen and (max-width: 490px) {
```

```
  .overviewDataItem {
```

```
    width: 150px;
```

```
}
```

```
#overviewData {
```

```
  grid-template-columns: 150px 150px;
```

```
}
```

```
.errorCode {
```

```
  font-size: 120px;
```

```
}
```

```
}
```

```
@media only screen and (max-width: 440px) {
```

```
  #nav {  
    display: none;  
  }
```

```
  #burger {  
    display: inline-block;  
    color: #fff;  
  }
```

```
  #showUserHolder h1 {  
    font-size: 18px;  
  }
```

```
  #avatarAndBio p, #showUserData p {  
    font-size: 15px;  
    width: auto;  
  }
```

```
  #avatarAndBio img {  
    width: 60px;  
    height: 60px;  
  }
```

```

td, th {

    padding: 0;

}

.adminPanelContent {

    padding: 0;

}

```

## MAIN.JS

```

$(document).ready(function(){

    let url = window.location.href;

    let pageCode = url.split("page=")[1].split("&")[0];

    //Login and register form show and hide

    $(".loginOpener").click(function(e){

        e.preventDefault();

        $("#loginForm").css("display", "flex");

        $("#registerForm").css("display", "none");

    });

    $(".loginCloser").click(function(e){

        e.preventDefault();

        $("#loginForm").css("display", "none");

    });

    $(".registerOpener").click(function(e){

```

```

    e.preventDefault();

    $("#registerForm").css("display", "flex");

    $("#loginForm").css("display", "none");

});

$(".registerCloser").click(function(e){

    e.preventDefault();

    $("#registerForm").css("display", "none");

});

//Register form validation

let emailRegex = /^[\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,6})+$/;

let usernameRegex = /^[a-zA-Z0-9_]{3,20}$/;

let emailOk = false;

let loginEmailOk = false;

let usernameOk = false;

let passwordOk = false;

let tosOk = false;


//Checks if the username and the email already exists

//It will take two arguments

//1. Whether the username or email are being checked

//2. The element to get data from

//Both arguments are strings

function usernameAndEmailCheck(whatToCheck, element){

    $.ajax({

```



```

url: "models/username_and_email_check.php",

method: "post",

dataType: "json",

data: {

    whatToCheck: whatToCheck,

    value: $(element).val()

},

success: function(response){

    if(response.exists){

        if(whatToCheck == 'email'){

            emailOk = false;

        }else {

            usernameOk = false;

        }

        if($(element).parent().next().hasClass("validationError")){

            $(element).parent().next().remove();

            $(element).parent().after("<p class='validationError'>Taken</p>");

        }else {

            $(element).parent().after("<p class='validationError'>Taken</p>");

        }

    }else {

        if(whatToCheck == 'email'){

            emailOk = true;

        }else {

            usernameOk = true;

        }

    }

}

```

```

    }

    if($(element).parent().next().hasClass("validationError")){

        $(element).parent().next().remove();

    }

}

},

error: function(jqXHR, status, message){

    console.log(jqXHR);

}

});

}

```

//Funkcija koja prikazuje grešku ispod prosleđenog elementa

```

function showValidationError(message, that, element = undefined){

    if(!$(that).parent().next().hasClass("validationError")){

        $(that).parent().after('<p class="validationError">${message}</p>');

    }

    switch(element){

        case 'username':

            usernameOk = false;

            break;

        case 'email':

            emailOk = false;

            break;

        case 'password':

```

```

        passwordOk = false;

        break;

    case 'loginEmail':

        loginEmailOk = false;

        break;

    }

}

```

//Fukcija koja sklanja greške ispod prosleđenog elementa

```

function removeValidationError(that){

    if($(that).parent().next().hasClass("validationError")){

        $(that).parent().next().remove();

    }

}

```

//Na blur proverava format korisničkog imena, i da li je već u upotrebi

```

$("#registerUsername").blur(function(){

    if($(this).val() != ""){

        if(!usernameRegex.test($(this).val())){

            showValidationError("Must be at least 3 characters long! Can include only letters, numbers and underscore", this, 'username');

        }else {

            usernameAndEmailCheck('username', '#registerUsername');

        }

    }else {

        usernameOk = false;

    }

}

```

```

        removeValidationError(this);
    }
});

//Na blur proverava format email-a, i da li je već u upotrebi
$("#registerEmail").blur(function(){
    if($(this).val() != ""){
        if(!emailRegex.test($(this).val())){
            showValidationError("Wrong e-mail format!", this, 'email');
        }else {
            usernameAndEmailCheck('email', '#registerEmail');
        }
    }else {
        emailOk = false;
        removeValidationError(this);
    }
});

```

```

//Na blur proverava dužinu lozinke i prikazuje potencijalnu grešku
$("#registerPassword").blur(function(){
    if($(this).val() != ""){
        if($(this).val().length < 8){
            showValidationError("Must contain at least 8 characters", this, 'password');
        }else {
            passwordOk = true;
        }
    }
});

```

```

        removeValidationError(this);
    }
}
else {
    passwordOk = false;
    removeValidationError(this);
}
});

```

//Na klik, smatra se da korisnik prihvata uslove, i greška se uklanja

```

$("#tosCheck").click(function(){
    if($(this).parent().next().hasClass("validationError")){
        $(this).parent().next().remove();
        tosOk = true;
    }
});

```

//Na klik se radi validacija, i podaci se šalju ka fajlu za registraciju

```

$("#registerBtn").click(function(e) {
    e.preventDefault();
    if(usernameOk && passwordOk && emailOk){
        if($("#tosCheck").prop("checked")){
            $.ajax({
                url: 'models/register.php',
                method: 'post',
                dataType: 'json',

```

```

data: {
    username: $("#registerUsername").val(),
    email: $("#registerEmail").val(),
    password: $("#registerPassword").val()
},
success: function(response){
    if(response.result == 'success'){
        $("#registerForm").css("display", "none");
        $("#registerForm form").trigger("reset");
        $("#loginForm").css("display", "flex");
        $("#loginForm #loginImage p").text("Registration successful. Now, you can log in with your credentials.");
    }else if (response.validationError) {
        alert("There was a validation error on the server side. Please try again.");
    }
},
error: function(jqXHR, message){
    console.log(jqXHR.responseText);
}
});
}else {
    $("#tosCheck").parent().after('<p class="validationError">You have to accept the terms of service</p>');
}
}else {
    if(!$(this).next().hasClass("validationError")){

```

```
$(this).after('<p class="validationError">Make sure no fields are empty and you agree with  
terms of service!</p>')
```

```
}
```

```
}
```

```
});
```

```
//Na klik, radi se validacija i podaci se šalju ka fajlu za logovanje
```

```
$("#loginBtn").click(function(e){
```

```
    e.preventDefault();
```

```
    if(emailRegex.test($("#loginEmail").val())){
```

```
        if($("#loginEmail").parent().next().hasClass("validationError")){
```

```
            $("#loginEmail").parent().next().remove();
```

```
        }
```

```
        $.ajax({
```

```
            url: 'models/login.php',
```

```
            method: 'post',
```

```
            dataType: 'json',
```

```
            data: {
```

```
                email: $("#loginEmail").val(),
```

```
                password: $("#loginPassword").val()
```

```
            },
```

```
            success: function(response){
```

```
                if(response.redirect !== undefined){
```

```
                    window.location = response.redirect;
```

```
                }
```

```
                if(response.error !== undefined){
```

```

        if(response.error == 'Wrong password'){
            if(!$("#loginPassword").parent().next().hasClass("validationError")){
                $("#loginPassword").parent().after('<p class='validationError'>${response.error}</p>');
            }
        }else {
            if(!$("#loginEmail").parent().next().hasClass("validationError")){
                $("#loginEmail").parent().after('<p class='validationError'>${response.error}</p>');
            }
        }

    },

    error: function(message){
        console.log(message);
    }

});

}else {

    if(!$("#loginEmail").parent().next().hasClass("validationError")){
        $("#loginEmail").parent().after('<p class='validationError'>Wrong email format</p>');
    }

}

});

```

//Kada se izađe iz email polja kod logovanja, radi se validacija i prikazuju greške

```

$("#loginEmail").blur(function(){

```



```

if(emailRegex.test($("#loginEmail").val()) || ($("#loginEmail").val() == ""){
    removeValidationError(this);
}else {
    showValidationError('Wrong e-mail format', this, 'loginEmail');
}
});

```

```

$("#loginPassword").blur(function(){
    if($("#loginPassword").val() == ""){
        removeValidationError($("#loginPassword"));
    }
});

```

//Popunjavanje overview panel odgovarajućim podacima

```

function fillOverviewPanel() {
    $.ajax({
        url: "models/get_overview_data.php",
        dataType: "json",
        method: "post",
        success: function(response){
            let ispis = "";
            if(response.error == undefined){
                for(x of response.latestUsers){
                    ispis += `<tr>
                        <td class="obsolete">${x.id}</td>

```

```

        <td>${x.username}</td>

        <td class="obsolete">${x.email}</td>

        <td>${x.register_time}</td>

    </tr>`;
}

$("#latestUsersTable table tbody").html(ispis);

$(".totalUsers").text(response.otherData.totalUsers);

$(".totalPosts").text(response.otherData.totalPosts);

$(".totalThreads").text(response.otherData.totalThreads);

$(".currentlyOnline").text(response.otherData.currentlyOnline);

$(".bannedUsers").text(response.otherData.totalBanned);

}else {

    console.log(response.error);

}

},

error: function(jqXHR){

    console.log(jqXHR);

}

});

}

```

```

function getSectionsForSelect(){

$.ajax({

    url: 'models/get_sections_for_select.php',

    method: 'post',

```

```

    dataType: 'json',
    success: function(response){
        let ispis = "";
        for(x of response){
            ispis += `
                <option value=${x.id}>${x.title}</option>
            `;
        }
        $("#selectSection").html(ispis);
    },
    error: function(jqXHR){
        console.log(jqXHR);
    }
});
}

```

```

if(pageCode != 'show_thread' && pageCode != 'new_thread' && pageCode != 'new_post'){
    localStorage.setItem("last_pagination_part", 'null');
}

```

```

function loadThread(threadId, paginationPart) {
    let pageNumber = paginationPart;
    if(localStorage.getItem("last_pagination_part") != 'null'){
        pageNumber = localStorage.getItem("last_pagination_part");
    }
    $.ajax({

```

```

url: 'models/load_thread.php',
method: 'post',
dataType: 'json',
data: {
    thread_id: threadId,
    page_number: pageNumber
},
success: function(response) {
    if(response.error == undefined) {
        let ispis = "";
        for(x of response.data){
            ispis += `
<div class="post">
    <div class="postTop">
        <div class="postData">
            <p>${x.post_time.substring(0, 16)}</p>
            ${x.is_admin_or_mod && !x.is_last_post? "<button class=\"delPostBtn btn delBtn btn-dark\" data-id=\" + x.post_id + \">Remove post</button>\" : ""}
        </div>
    </div>
    <div class="postBottom">
        <div class="postLeft">
            <div class="postCreatorAvatar"></div>
            <div class="postCreatorData">
                <p><a
href='index.php?page=show_user&id=${x.user_id}'>${x.username}</a></p>

```

```
        <p>${x.online_status == 1? '<i class="fas fa-circle"></i> Online' : '<i class="far fa-circle"></i> Offline'}</p>
```

```
        <p>Registered: ${ x.register_time.substr(0, 11)}</p>
```

```
        <p>Posts: ${x.no_of_posts}</p>
```

```
    </div>
```

```
</div>
```

```
<div class="postRight">
```

```
    <p>${x.content}</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
`;
```

```
}
```

```
$("#postsHolder").html(ispis);
```

```
$(".delPostBtn").click(function(e){
```

```
    deleteRow(e, 'posts');
```

```
    loadThread(threadId, pageNumber);
```

```
});
```

```
let paginacijasps = "";
```

```
for(x of response.pagination_links){
```

```
    paginacijasps += x;
```

```
}
```

```
$("#paginationLinks").html(paginacijasps);
```

```
$("#pageNumber p").text("Page: " + response.current_page);
```

```
$(".paginationButton").click(function(){
```

```

        let lastPaginationPart = localStorage.setItem("last_pagination_part", $(this).attr('data-
part'));

        loadThread(threadId, $(this).attr('data-part'));

    });

    }else {

        console.log(response.error);

    }

},

error: function(jqXHR){

    console.log(jqXHR);

}

});

}

```

//Poziva funkciju za popunjavanje overview panela ako je stranica 'admin', na svakih deset sekundi

```

if(pageCode == "admin"){

    fillOverviewPanel();

    setInterval(function(){

        fillOverviewPanel();

    }, 10000);

    getSectionsForSelect();

    setInterval(function(){

        getSectionsForSelect();

    }, 3000);

}

```

```

if(pageCode == "show_thread"){

    let threadId = Number(url.split("page=")[1].split("&")[1].split("=")[1]);

    //let paginationPart = Number(url.split("part=")[1]);

    loadThread(threadId, 1);

}


//Admin panels shifting


//Niz koji sadrži elemente panela

let adminPanels = [$("#overviewPanel"), $("#usersPanel"), $("#contentPanel"), $("#logsPanel"),
    $("#bugPanel"), $("#announcementPanel"), $("#chatroomPanel"), $("#databasePanel")];


//Čuva redni broj trenutno aktivnog panela

let activePanel = 0;


//Prikazuje aktivni panel na početku. To je overview panel.

loadActivePanel(activePanel);


//Klikom na neki od tastera, menja mu se boja, i otvara se odgovarajući panel

$(".adminPanelsLink").click(function(){

    activePanel = $(this).attr('data-id');

    loadActivePanel(activePanel);

    $(".adminPanelsLink").css('background-color', '#fff');

    $(this).css('background-color', '#ccc');

```

```
});
```

```
//Sakriva sve panele izuzev trenutno aktivnog
```

```
function loadActivePanel(panel) {
```

```
    //if(window.innerWidth > 1000){
```

```
        $(".adminPanel").css('display', 'none');
```

```
        adminPanels[panel].css('display', 'flex');
```

```
    //}
```

```
}
```

```
$("#burger").click(function(){
```

```
    if($("#drawer").css("display") == "block"){
```

```
        $("#drawer").slideUp();
```

```
    }else {
```

```
        $("#drawer").slideDown();
```

```
    }
```

```
});
```

```
//Vrednost koja se nalazila u polju za pretragu korisnika, kako bih osvežio tabelu sa istim vrednostima
```

```
//Pronalazi korisnike kojima se početni deo korisničkog imena poklapa sa unetom vrednošću
```

```
function findUser(username){
```

```
    //Čuva unetu vrednost
```

```
    findUserVal = username;
```



```

if(/^^[a-zA-Z0-9\_]*$/ .test($("#findUser").val())){

    removeValidationError($("#findUser"));

    $.ajax({

        url: 'models/find_user.php',

        method: 'post',

        dataType: 'json',

        data: {

            username: username

        },

        success: function(response){

            let ispis = `

```

```

`;

ispis += `

<tr>

    <td class="obsolete">${x.id}</td>

    <td><a href="index.php?page=show_user&id=${x.id}">${x.username}</a></td>

    <td class="obsolete">${x.email}</td>

    <td><b>${x.userRole}</b></td>

    <td class="obsolete">${x.online_status == 1?'online':'offline'}</td>

    <td class="obsolete">${x.posts}</td>

    ${x.role == 1?"":buttons}

</tr>

`;

}

ispis += '</tbody>';

}else {

    ispis = '<p style="color:#aaa;padding-top:5px">No results</p>';

}

$("#usersTable").html(ispis);

$(".deleteUserBtn").click(function(e){

    deleteRow(e, 'users', findUserVal);

});

$(".givePrivBtn").click(function(){

```

```

let user_id = $(this).attr('data-id');

$("#givePrivDiv").css("display", "flex");

$(".submitGivePriv").click(function(){

    updateRole($("#givePrivDiv select").val(), user_id);

});

});

```

```

$(".banUserBtn").click(function(){

$.ajax({

    url: 'models/ban_user.php',

    method: 'post',

    dataType: 'json',

    data: {

        id: $(this).attr('data-id')

    },

    success: function(response){

        if(response.result == 'banned'){

            findUser(findUserVal);

        }else {

            console.log(response);

        }

    },

    error: function(jqXHR){

        console.log(jqXHR);

    }

}

```

```

        });

    });

}

});

}else {

    showValidationError("Wrong format", $("#findUser"));

}

}

```

```

$("#givePrivDiv select").change(function(){

    let textValue = "";

    let value = ($("#givePrivDiv select").val());

    console.log(value);

    switch(value){

        case '1':

            textValue = "Users with this role will be able to ban, delete, and change roles of other users  
(except admins), to add sections, topics, and delete everything.";

            break;

        case '2':

            textValue = "Moderators can only delete posts for now.";

            break;

        case '3':

            textValue = "Basic users.";

            break;

    }

    $("#givePrivDiv p").text(textValue);

```

```
});
```

```
function updateRole(role, user){  
    $.ajax({  
        url: 'models/update_role.php',  
        method: 'post',  
        dataType: 'json',  
        data: {  
            role_id: role,  
            user_id: user  
        },  
        success: function(response){  
            //console.log(response);  
        },  
        error: function(jqXHR){  
            console.log(jqXHR);  
        }  
    });  
    $("#givePrivDiv").css('display', 'none');  
    findUser(findUserVal);  
}
```

```
$("#cancelGivePriv").click(function(){  
    $("#givePrivDiv").css("display", "none");  
});
```

```

function banUser(){
    $.ajax({
        url: 'models/ban_user.php',
        method: 'post',
        dataType: 'json',
        data: {
            id: $(this).attr('data-id')
        },
        success: function(response){
            if(response.result == 'banned'){
                findUser(findUserVal);
            }
        },
        error: function(jqXHR){
            console.log(jqXHR);
        }
    });
}

```

```

$("#findUser").keyup(function(){
    let findUserVal = $(this).val();
    findUser(findUserVal);
});

```

```
//Content management
```

```
let sectionTitleRegex = /^[A-Z]{1}[a-z]*(\s[a-z]*)*$/;
```

```
let topicTitleRegex = /^[A-Z]{1}[a-z]*(\s[a-z]*)*$/;
```

```
let topicDescriptionRegex = /^[A-Z]{1}[a-z]*(\s[a-z]*)*$/;
```

```
$("#addSectionBtn").click(function(e){  
    e.preventDefault();  
    if(sectionTitleRegex.test($("#addSectionTitle").val())){  
        $.ajax({  
            url: "models/add_section.php",  
            method: "post",  
            dataType: "json",  
            data: {  
                title: $("#addSectionTitle").val()  
            },  
            success: function(response){  
                alert(response.result);  
                $("#addSectionTitle").val("");  
            },  
            error: function (jqXHR){  
                console.log(jqXHR);  
            }  
        });  
    }else {
```

```

        showValidationError("Only words. First letter capitalized", $("#addSectionTitle"));
    }

});

```

```

function findSection(val){
    if(!/[A-Za-z0-9?!.\s]/.test(val) && val != ""){
        showValidationError("Wrong format", $("#findSection"));
    }else {
        removeValidationError($("#findSection"));
        $.ajax({
            url: 'models/find_section.php',
            method: 'post',
            dataType: 'json',
            data: {
                title: val
            },
            success: function(response){
                let ispis = `
                <thead>
                <tr>
                <th class="obsolete">ID</th>
                <th>Title</th>
                <th class="obsolete">Creator</th>
                </tr>

```



```

</thead>

<tbody>

`;

if(response.result != undefined && response.result == 'empty'){

    ispis = '<p style="color:#777">No results</p>';

}else {

    for(x of response){

        ispis += `

        <tr>

            <td class="obsolete">${x.id}</td>

            <td>${x.title}</td>

            <td class="obsolete">${x.username}</td>

            <td><button class="deleteSectionBtn delBtn btn btn-dark" data-id=${x.id}>Del <i
class="far fa-trash-alt"></i></button></td>

        </tr>

        `;

    }

    ispis += '</tbody>';

}

$("#sectionTable").html(ispis);

$(".deleteSectionBtn").click(function(e){

    deleteRow(e, 'forum_sections', val);

});

},

error: function(jqXHR){

    console.log(jqXHR);

```

```

    }
  });
}
}

```

```

function findTopic(val){
  if(!/[A-Za-z0-9?!.\s]/.test(val) && val != ""){
    showValidationError("Wrong format", $("#findTopic"));
  }else {
    removeValidationError($("#findTopic"));
    $.ajax({
      url: 'models/find_topic.php',
      method: 'post',
      dataType: 'json',
      data: {
        title: val
      },
      success: function(response){
        let ispis = `
        <thead>
        <tr>
        <th class="obsolete">ID</th>
        <th>Title</th>
        <th class="obsolete">Creator</th>
        <th class="obsolete">Section</th>

```

```

        </tr>

    </thead>

    <tbody>

        `;

        if(response.result != undefined && response.result == 'empty'){

            ispis = '<p style="color:#777">No results</p>';

        }else {

            for(x of response){

                ispis += `

                <tr>

                    <td class="obsolete">${x.id}</td>

                    <td>${x.title}</td>

                    <td class="obsolete">${x.username}</td>

                    <td class="obsolete">${x.section}</td>

                    <td><button class="deleteTopicBtn delBtn btn btn-dark" data-id=${x.id}>Del <i
class="far fa-trash-alt"></i></button></td>

                </tr>

                `;

            }

            ispis += '</tbody>';

        }

        $("#topicTable").html(ispis);

        $(".deleteTopicBtn").click(function(e){

            deleteRow(e, 'topics', val);

        });

    },

```

```

        error: function(jqXHR){
            console.log(jqXHR);
        }
    });
}
}

```

```

function findThread(val){
    if(!/[A-Za-z0-9?!.\s]/.test(val) && val != ""){
        showValidationError("Wrong format", $("#findThreads"));
    }else {
        removeValidationError($("#findThreads"));
        $.ajax({
            url: 'models/find_thread.php',
            method: 'post',
            dataType: 'json',
            data: {
                title: val
            },
            success: function(response){
                let ispis = `
                <thead>
                <tr>
                <th class="obsolete">ID</th>
                <th>Title</th>

```

```

        <th class="obsolete">Creator</th>

        <th class="obsolete">Topic</th>

    </tr>

</thead>

<tbody>

`;

if(response.result != undefined && response.result == 'empty'){

    ispis = '<p style="color:#777">No results</p>';

}else {

    for(x of response){

        ispis += `

        <tr>

            <td class="obsolete">${x.id}</td>

            <td>${x.title}</td>

            <td class="obsolete">${x.username}</td>

            <td class="obsolete">${x.topic}</td>

            <td><button class="deleteThreadBtn delBtn btn btn-dark" data-id=${x.id}>Del <i
class="far fa-trash-alt"></i></button></td>

        </tr>

        `;

    }

    ispis += '</tbody>';

}

$("#threadTable").html(ispis);

$(".deleteThreadBtn").click(function(e){

    deleteRow(e, 'threads', val);

```

```

        });

    },

    error: function(jqXHR){

        console.log(jqXHR);

    }

});

}

}

```

```

$("#findSection").keyup(function(){

    let findSectionVal = $(this).val();

    findSection(findSectionVal);

});

```

```

$("#findTopic").keyup(function(){

    let findTopicVal = $(this).val();

    findTopic(findTopicVal);

});

```

```

$("#findThreads").keyup(function(){

    let findThreadVal = $(this).val();

    findThread(findThreadVal);

});

```

```

$("#addTopicBtn").click(function(e){
    e.preventDefault();
    if(topicTitleRegex.test($("#addTopicTitle").val())){
        removeValidationError($("#addTopicTitle"));
        removeValidationError($("#addTopicDescription"));
        if(topicDescriptionRegex.test($("#addTopicDescription").val())){
            $.ajax({
                url: 'models/add_topic.php',
                method: 'post',
                dataType: 'json',
                data: {
                    title: $("#addTopicTitle").val(),
                    section_id: $("#selectSection").val(),
                    description: $("#addTopicDescription").val()
                },
                success: function(response){
                    if(response.result == 'added'){
                        alert("Success");
                        $("#addTopicTitle").val("");
                        $("#addTopicDescription").val("");
                    }else {
                        console.log(response.result);
                    }
                },
                error: function(jqXHR){

```

```

        console.log(jqXHR);

    }

    });

    }else {

        showValidationError("Description needed. Only words. First letter capitalized",
$("#addTopicDescription"));

    }

}

    }else {

        showValidationError("Only words. First letter capitalized", $("#addTopicTitle"));

    }

});

```

//Funkcija omogućava brisanje redova. Prihvata element handler, naziv tabele, i opcionalno, pretraživanu vrednost u slučaju da je obrisano

//red iz neke html tabele, kako bi tabela mogla da se osveži istim vrednostima

```

function deleteRow(e, table, searchVal = undefined){

    let id = parseInt(e.currentTarget.attributes['data-id'].value);

    $.ajax({

        url: 'models/delete_row.php',

        method: 'post',

        dataType: 'json',

        data: {

            table: table,

            row_id: id

```



```

},
success: function(response){
    if(table == 'users'){
        if(searchVal != undefined){
            findUser(searchVal);
        }
        $("#findUser").val("");
    }else if(table == 'forum_sections'){
        if(searchVal != undefined){
            findSection(searchVal);
        }
        $("#findSection").val("");
    }else if(table == 'topics'){
        if(searchVal != undefined){
            findTopic(searchVal);
        }
        $("#findTopic").val("");
    }else {
        if(searchVal != undefined){
            findThread(searchVal);
        }
        $("#findThread").val("");
    }
},
error: function(jqXHR){

```

```
        console.log(jqXHR);
    }
});
}
```

//Klikom na taster, otvara se stranica za dodavanje novog thread-a na forumu

```
$("#newThreadBtn").click(function(){
    lastPaginationPart = 'null';
    window.location = `index.php?page=new_thread&topic_id=${$(this).attr('data-id')}`;
});
```

//Klikom na dugme, dodaje se novi post u bazu

```
$("#postAnswerBtn").click(function(){
    let thread_id = $(this).attr("data-id");
    $.ajax({
        url: 'models/add_post.php',
        method: 'post',
        dataType: 'json',
        data: {
            thread_id: thread_id,
            text: $("#postAnswerText").val()
        },
        success: function(response){
            alert(response.result);
        }
    });
});
```

```

        window.location.href = `index.php?page=show_thread&id=${thread_id}`;
    },
    error: function(jqXHR){
        console.log(jqXHR);
    }
});
});

```

```

//Klikom na dugme, dodaje se novi thread
$("#submitThreadBtn").click(function(e){
    e.preventDefault();

    function getTopicId(){
        return Number(url.substring(url.indexOf('id') + 3));
    }

    let textRegex = /^\\w+\\/;

    let topic_id = getTopicId();

    let title = $("#newThreadTitle").val();

    let text = $("#newThreadPost").val();

    if(textRegex.test(text)){
        $.ajax({
            url: 'models/add_thread.php',
            method: 'post',
            dataType: 'json',
            data: {
                topic_id: topic_id,

```

```

        title: title,
        text: text
    },
    success: function(response){
        if(response.redirect != undefined){
            window.location.href = response.redirect;
        }
    },
    error: function(jqXHR){
        console.log(jqXHR);
    }
});
}
});

```

//Klikom na dugme, otvara se strana za dodavanje novog posta

```

$("#newPostBtn").click(function(){
    let thread_id = $(this).attr('data-id');
    window.location.href = `index.php?page=new_post&thread_id=${thread_id}`;
});

});

```