

Dokumentacija za sajt iz praktikuma PHP

Nemanja Jovičić 326/14

Sadržaj

1. Uvod

- 1.1 Programski jezici
- 1.2 Opis funkcionalnosti

2. Mapa sajta

- 2.1 Baza podataka
- 2.2 Sitemap
- 2.3 Struktura projekta
- 2.4 Slike stanica i funkcionalnosti
 - 2.4.1 Index
 - 2.4.2 News
 - 2.4.3 Single news
 - 2.4.4 Author
 - 2.4.5 Login
 - 2.4.6 Register
 - 2.4.7 Admin panel Index
 - 2.4.8 Admin panel Insert/Edit

3. Kodovi

- 3.1 Config
 - 3.1.1 Config.php
 - 3.1.2 Connection.php
- 3.2 index.php
- 3.3 head.php
- 3.4 navigation.php

- 3.5 footer.php
- 3.6 status_403.php
- 3.7 admin.js
- 3.8 adminChart.js
- 3.9 logAndReg.js
- 3.10 main.js

3.2 Pages

3.2

Uvod

1.1 Korisceni programski jezici

Korisceni su : HTML, CSS, JS, BOOTSTRAP, JQUERY

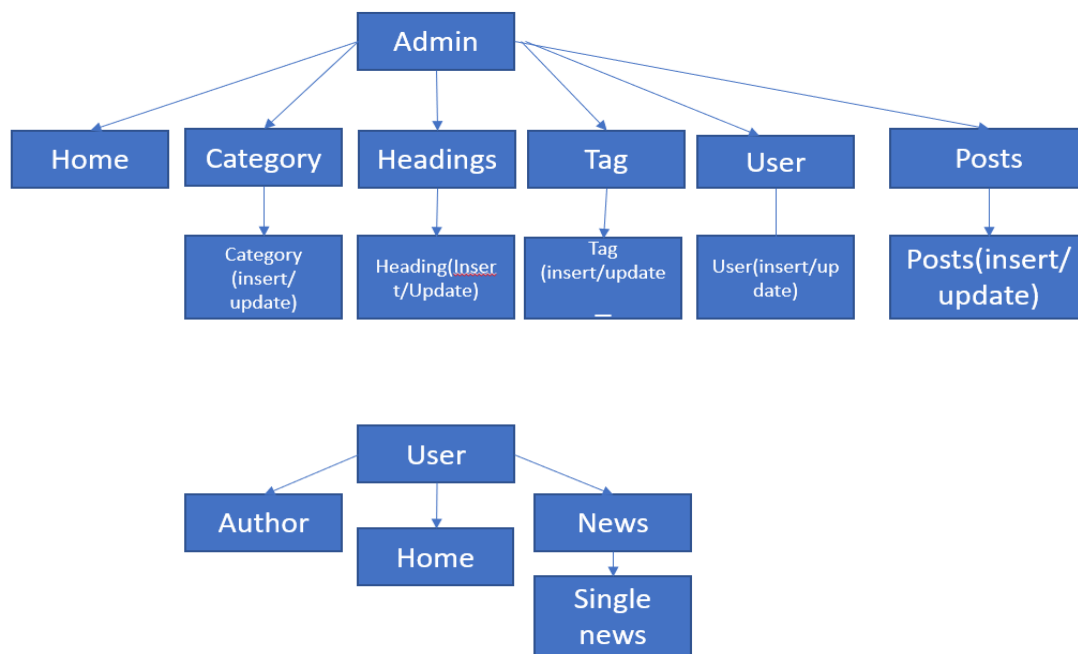
1.2 Opis funkcionalnosti

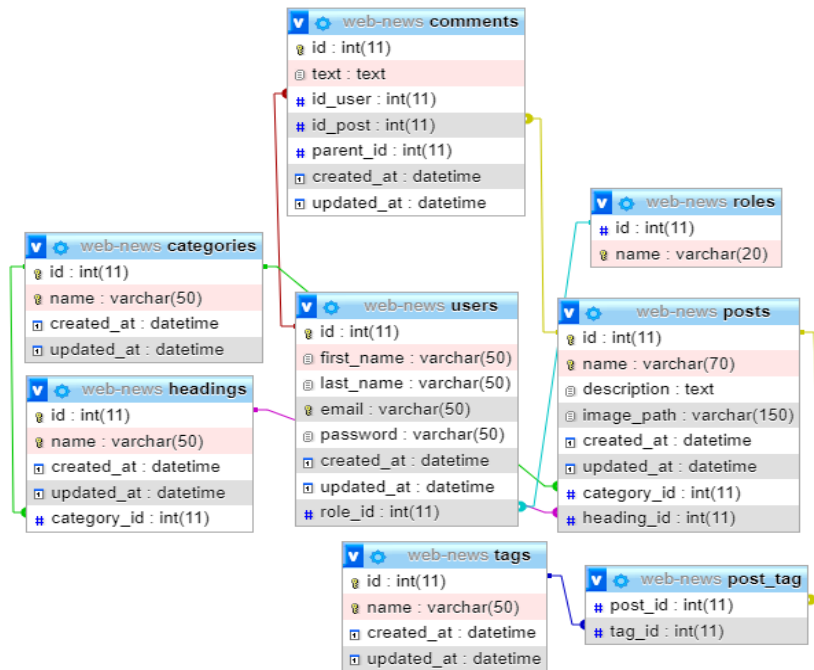
- Sadržaj koji se prikazuje se razlikuje od neulogovanog, ulogovanog i korisnika koji ima administratorsku ulogu
- Klijentska i serverska provera, prilikom unošenja elemenata na administratorskoj strani
- Filtriranje, sortiranje, pretraga kombinovani sa paginacijom na administratorskoj strani
- Registracija i registracija
- Pracenje statistike pristupa stranicama

- Sistem za komentarisanje
- Resize slike
- Svi ključni elementi su dinamički
- Korisnik može da unese komentar

2.1

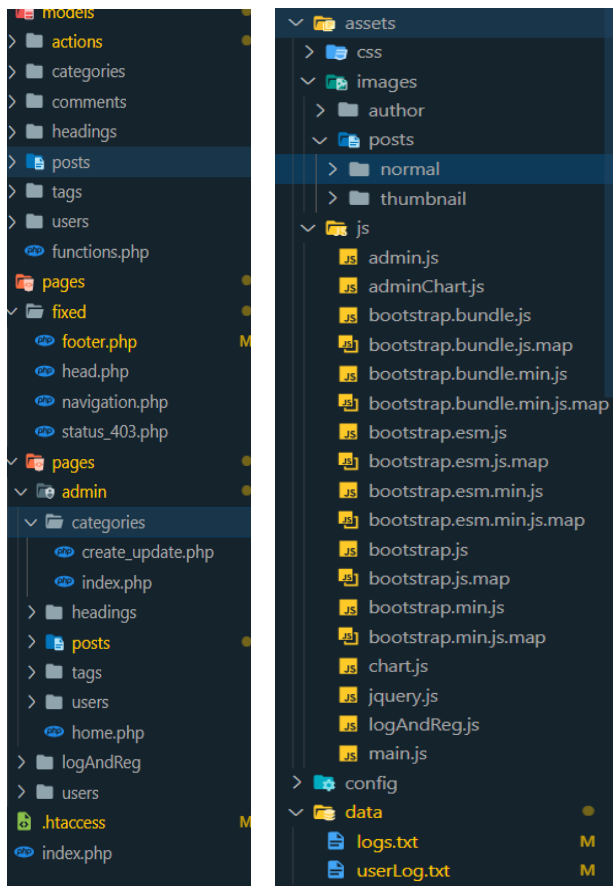
Mapa sajta





Baza podataka:

Struktura projekta:




INDEX.php

Prikaz index.php gde na osnovnoj stranici videmo sve


Odre]enu grupu postova

E-portal


Fudbal Kosarka Tenis Author Login Register




NASTAVLJAMO DA BROJIMO: NOVAK POČEO 336. NEDELJU NA PRVOM MESTU ATP LI




LEGENDARNA ŠTEFI GRAF NAVIJA ZA SRBINA: NADAM SE DA ĆE NOVAK USPETI, T




RONALDO ĆE POBESNETI ZBOG OVE IZJAVE: EVO ŠTA JE PEP GWARDIOLA REKAO O




MALI ĆE SE JEDNOG DANA PONOSITI OVOM FOTKOM: MESI DRŽAO U NARUČJU RAJK



RONALO PROŠAO LEKARSKE PREGLEDE: PORTUGALAC USKORO STIŽE NA OLD TRAFOR



DOŠAO IZ NBA I HTEO U AVION ZA KRAGUJEVAC! HRVATSKI KOŠARKAŠ ISPRIČAO




TEODOSIĆ BEZ TRAKE! SKARIOLO ZA KAPITENA VIRTUSA IZABRAO DRUGOG ASA

All right reserved


Page.php?categoryName=NazivKategorije

E-portal


Fudbal Kosarka Tenis Author Login Register




DOŠAO IZ NBA I HTEO U AVION ZA KRAGUJEVAC! Hrvatski košarkaš ispričao




TEODOSIĆ BEZ TRAKE! Skariolo za kapitena Virtusa izabrao drugog asa



TEODOSIĆ BEZ TRAKE! Skariolo za kapitena Virtusa izabrao drugog asa
12:44:51 30-08-2021



DOŠAO IZ NBA I HTEO U AVION ZA KRAGUJEVAC! Hrvatski košarkaš ispričao
12:22:32 30-08-2021



RONALO PROŠAO LEKARSKE PREGLEDE: Portugalac uskoro stiže na Old Trafor
12:18:42 30-08-2021

All right reserved


Prikaz stracine gde se filtriraju podaci na osnovu izabranie kategorije, dok sa strane vidimo odredjenu

Grupu elemenata koja predstavlja zajednicku celinu
Za sve stranice.

Autor.php

E-portal

FudbalKosarkaTenisAuthorLoginRegister



Nemanja Jovičić

What is Lorem Ipsum? Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Export to word

Login.php

E-portal

FudbalKosarkaTenisAuthorLoginRegister

Email:

Password:

Login

Register.php (stranica za registraciju korisnika)

E-portal

FudbalKosarkaTenisAuthorLoginRegister

First name:

Last name:

Email:

Password:

Register

Index Category.php

E-portal

CategoriesHeadingsTagsPostsUsersLogout

Add new category

Export to excel

#	Name	Created at	Updated at	Update	Delete
1	Fudbal	2021-08-25 15:57:14	2021-08-30 11:56:57	Update	Delete
2	Kosarka	2021-08-25 15:57:14	2021-08-30 11:03:11	Update	Delete
3	Tenis	2021-08-30 11:00:36	2021-08-30 11:03:22	Update	Delete

All right reserved

Pocetna stranica za kategorije koja sadrži prikaz osnovnih elemenata, Ima mogućnost eksportovanje naziva kategorija u excel file.

Category action-category.php

localhost/e-news-master/index.php?page=action-categories

E-portal

Categories Headings Tags Posts Users Logout

Name:

Save

All right reserved

Index Heading.php

localhost/e-news-master/index.php?page=headings

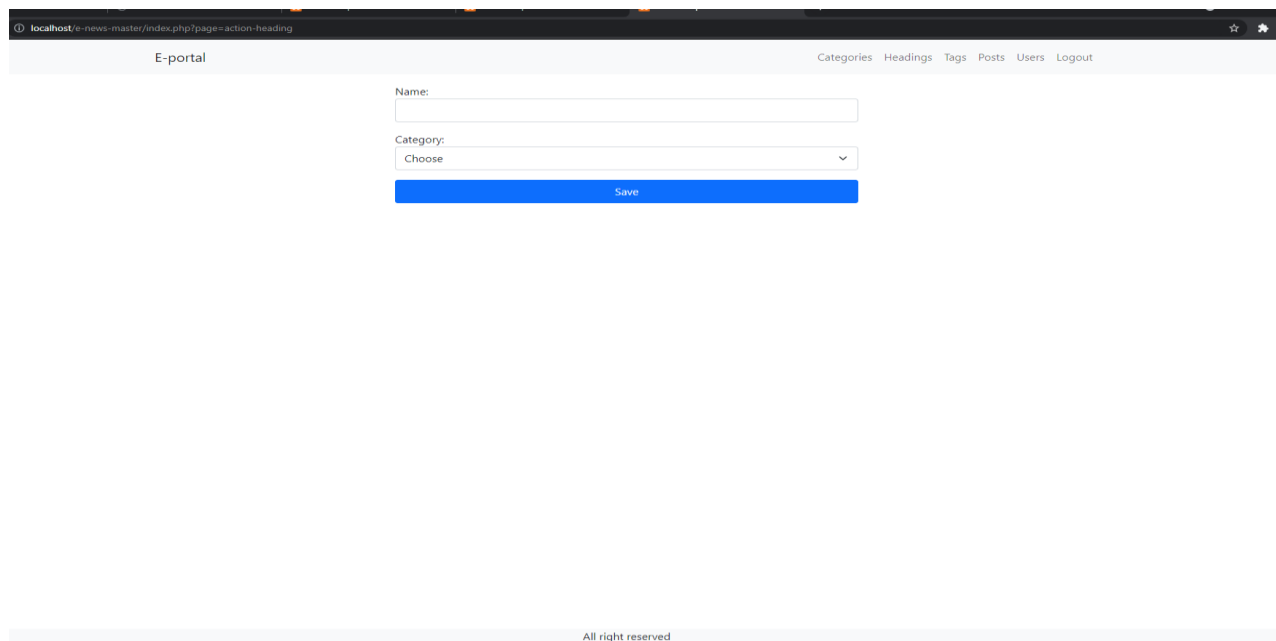
E-portal

Categories Headings Tags Posts Users Logout

Add new heading

#	Name	Category name	Created at	Updated at	Update	Delete
1	Premier Liga	Fudbal	2021-08-25 16:34:31	2021-08-30 11:05:12	Update	Delete
2	Primera	Fudbal	2021-08-25 18:06:17	2021-08-30 11:37:11	Update	Delete
3	Francuska Liga	Fudbal	2021-08-30 11:57:14	-	Update	Delete
4	Nba	Kosarka	2021-08-25 17:59:55	2021-08-30 11:36:56	Update	Delete
5	Liga Endessa	Kosarka	2021-08-30 12:21:03	-	Update	Delete
6	Euroliga	Kosarka	2021-08-30 12:43:13	-	Update	Delete
7	Wta	Tenis	2021-08-25 19:33:52	2021-08-30 11:37:19	Update	Delete
8	Atp	Tenis	2021-08-30 11:39:54	-	Update	Delete

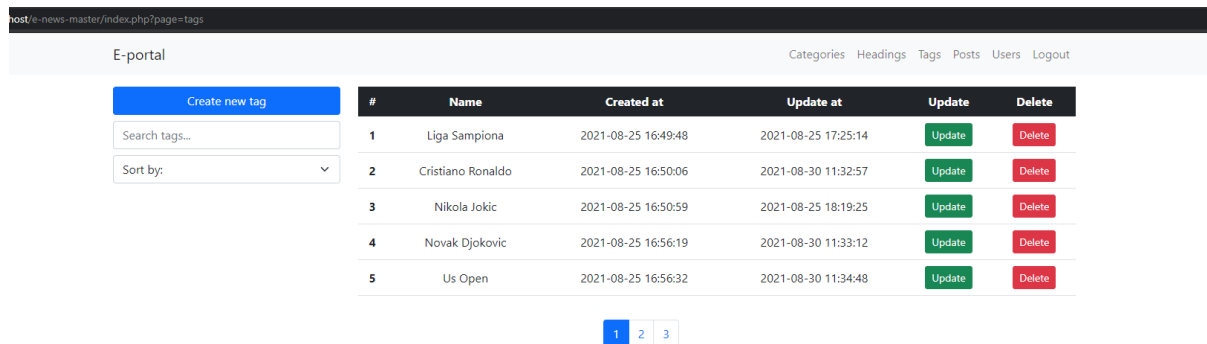
Prikaz rubrika gde kategorije koje su azurirane prikazan datum azuriranja



Heading action-heading.php

Je stranica na kojoj se vrši unos ili izmena odredjenih elemata.

Tag index.php



Je pocetna stranica sa tagovima, na kojoj se moze isvršiti filtriranje po Datumu I pretraga u kombinaciji sa paginacijom

Tag action-tag.php

e-news-master/index.php:page-action-tag

E-portal

CategoriesHeadingsTagsPostsUsersLogout

Name:

Save

All right reserved

Sluzi za izmenu tagova ili dodavanje novih.

Post index.php

E-portal

CategoriesHeadingsTagsPostsUsersLogout

Create new post

Search posts....

Categories:

☐ Fudbal

☐ Kosarka

☐ Tenis

Headings:

☐ Premier Liga

☐ Nba

☐ Primera

☐ Wta

☐ Atp

☐ Francuska Liga

☐ Liga Endessa

☐ Euroliga

Sort by date:

Choose:

#	Name	Category name	Heading name	Created at	Updated at	Update	Delete	Details
1	NASTAVLJAMO DA BROJIMO: Novak počeo 336. nedelju na prvom mestu ATP li	Tenis	Atp	18:33:42 25/08/2021	11:40:17 30/08/2021	<div>Update</div>	<div>Delete</div>	<div>Details</div>
2	LEGENDARNA ŠTEFI GRAF NAVIJA ZA SRBINA: Nadam se da će Novak uspeti, t	Tenis	Atp	19:30:58 25/08/2021	11:42:30 30/08/2021	<div>Update</div>	<div>Delete</div>	<div>Details</div>
3	RONALDO ĆE POBESNETI ZBOG OVE IZJAVE: Evo šta je Pep Gvardiola rekao o	Fudbal	Premier Liga	19:34:33 25/08/2021	11:55:54 30/08/2021	<div>Update</div>	<div>Delete</div>	<div>Details</div>
4	MALI ĆE SE JEDNOG DANA PONOSITI OVOM FOTKOM: Mesi držao u naručju Rajk	Fudbal	Francuska Liga	11:58:40 30/08/2021	/	<div>Update</div>	<div>Delete</div>	<div>Details</div>
5	RONALO PROŠAO LEKARSKE PREGLEDE: Portugalac uskoro stiže na Old Trafor	Fudbal	Premier Liga	12:18:42 30/08/2021	/	<div>Update</div>	<div>Delete</div>	<div>Details</div>

1

2

Na ovoj stranici možemo da filtriramo na osnovu kategorije, rubrike,

Po nazivu ili po datum kreiranja sa paginacijom

Post insert/update

E-portal

[Categories](#) [Headings](#) [Tags](#) [Posts](#) [Users](#) [Logout](#)

Name

Description:

Image:

Choose File

No file chosen

Category:

Choose

Heading:

Choose

Tags:


- ☐ Liga Sampiona
- ☐ Cristiano Ronaldo
- ☐ Nikola Jokic
- ☐ Novak Djokovic
- ☐ Us Open
- ☐ Munchester United
- ☐ Bayern Munica
- ☐ Lionel Messi
- ☐ Krunoslav Simon
- ☐ Milos Teodosic
- ☐ Carmelo Anthony

Submit

Detail-post.php prikaz jednog posta za administratora

Portali

Categories | News | Tags | RSS



TENIS 18:33:42 25-08-2021

NASTAVLJAMO DA BROJIMO: Novak počeo 336. nedelju na prvom mestu ATP li

Srpski teniser Novak Đoković započeo je danas rekordnu 336. nedelju na prvom mestu ATP liste, na kojoj nije bilo promena među najboljim teniserima sveta. Đoković je prvi na listi sa 11.113 bodova, uoči poslednjeg Grend slem turnira u sezoni, Otvorenog prvenstva SAD, koje počinje danas u Njujorku. Među prvih 20 tenisera nije bilo promena na svetskoj rang-listi, ruski teniser Danil Medvedev zauzima drugo mesto sa 9.980 bodova, a treće Grk Stefanos Cicipas sa 8.350. Nemački teniser Aleksandar Zverev je četvrti, a peto mesto drži Španac Rafael Nadal. Od srpskih tenisera Filip Krajinović zadržao je 38. mesto sa 1.589 bodova, a Dušan Lajović 40. mesto sa 1.565. Laslo Đere pokvario je plasman za jedno mesto i sada je 51. igrač sveta sa 1.307 bodova, a za jedno mesto pao je i Miomir Kecmanović i sada je 63. sa 1.050 bodova. Nikola Milojević je zadržao 160. mesto sa 567 bodova, a Danilo Petrović 177. poziciju sa 415. Viktor Troicki popravio je poziciju za dva mesta i sada je 218. igrač sveta sa 315 bodova. Kurir sport/Beta

Tags:

Novak Djokovic

Us Open

Vanja Jovicic

13:43:14 30-08-2021

Djokovic je najbolji

Vanja Jovicic

13:43:15 30-08-2021

Djokovic je najbolji

Vanja Jovicic

13:43:49 30-08-2021

Djokovic je najbolji

Vanja Jovicic

13:44:19 30-08-2021

Danas ce pobediti

Vanja Jovicic

13:44:42 30-08-2021

Nemanja

Vanja Jovicic

13:45:05 30-08-2021

asdas

Vanja Jovicic

13:45:44 30-08-2021

dasdass

Vanja Jovicic

14:40:43 30-08-2021

Danas je lep dan

Vanja Jovicic

14:41:05 30-08-2021

Na ovoj stranici mozemo da video jedan post, kada je korisnik ulogovan
Kao admin, sa komentarima za taj post.

Index.php (User)

E-portal

CategoriesHeadingsTagsPostsUsersLogout

Create a new user

Search users.....

Choose

#	Full name	Email	Created at	Updated at	Update	Delete
1	Paja Patak	pajapatak11@gmail.com	2021-08-25 16:05:08	-	<div>Update</div>	<div>Delete</div>
2	Mladen Bogdanovic	mladen99@gmail.com	2021-08-25 20:03:44	-	<div>Update</div>	<div>Delete</div>
3	Miki Patak	mikipatak13@gmail.com	2021-08-30 10:35:35	-	<div>Update</div>	<div>Delete</div>
4	Vanja Jovicic	vanjajovicic95@yahoo.com	2021-08-30 12:58:05	-	<div>Update</div>	<div>Delete</div>

User-action.php (na ovoj stranici mozemo da vršimo unos ili izmenu korisnika)

E-portal

CategoriesHeadingsTagsPostsUsersLogout

First name:

Last name:

Email:

Password:

Role:

Admin

User

Submit

3.Kodovi

Config

config.php

```
<?php
define("BASE_URL", $_SERVER['DOCUMENT_ROOT'] . "/");
define("ENV_FILE", BASE_URL . '/config/.env');
define("LOG_FILE", BASE_URL . '/data/logs.txt');

define("HOST", env('HOST'));
define('NAME', env('NAME'));
define('USERNAME', env('USERNAME'));
define('PASSWORD', env('PASSWORD'));

function env($name)
{
    $data = file(ENV_FILE);
    $value = "";
    foreach ($data as $row) {
        $configuration = explode("=", $row);
        if ($configuration[0] == $name) {
            $value = trim($configuration[1]);
        }
    }
    return $value;
}
```

Connection.php

```
<?php
require_once 'config.php';

pageAccess();
try {
```



```

    $conn = new PDO('mysql:host=' . HOST . ';dbname=' . NAME, USERNAME, PASSWORD)
;
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
} catch (PDOException $th) {
    echo json_encode($th->getMessage());
}

function pageAccess()
{
    $file = fopen(LOG_FILE, "a");
    if ($file) {
        $date = date('d-m-Y H:i:s');

        fwrite(
            $file,
            "{$_SERVER['PHP_SELF']}?{$_SERVER['QUERY_STRING']}\t{$date}\t{$_SERVER['REMOTE_ADDR']}\t\n"
        );
        fclose($file);
    }
}

```

Index.php

```

<?php
session_start();
require_once 'config/connection.php';
include 'models/functions.php';
include 'pages/fixed/head.php';
include 'pages/fixed/navigation.php';
$page = '';
if (isset($_GET['page'])) {
    $page = $_GET['page'];
    switch ($page) {
        case 'login':
            include 'pages/pages/logAndReg/login.php';
            break;
        case 'register':

```

```
        include 'pages/pages/logAndReg/register.php';
        break;
    case 'admin_home':
        include 'pages/pages/admin/home.php';
        break;
    case 'news':
        include 'pages/pages/users/posts.php';
        break;
    case 'single-post':
        include 'pages/pages/users/singlePost.php';
        break;
    case 'author':
        include 'pages/pages/users/author.php';
        break;
    case 'home':
        include 'pages/pages/users/home.php';
        break;

    case 'categories':
        include 'pages/pages/admin/categories/index.php';
        break;
    case 'action-categories':
        include 'pages/pages/admin/categories/create_update.php';
        break;
    case 'headings':
        include 'pages/pages/admin/headings/index.php';
        break;
    case 'action-heading':
        include 'pages/pages/admin/headings/action_heading.php';
        break;
    case 'tags':
        include 'pages/pages/admin/tags/index.php';
        break;
    case 'action-tag':
        include 'pages/pages/admin/tags/action_tag.php';
        break;
    case 'posts':
        include 'pages/pages/admin/posts/index.php';
        break;
    case 'action-post':
        include 'pages/pages/admin/posts/action_post.php';
        break;
    case 'post_details':
        include 'pages/pages/admin/posts/post_details.php';
        break;
```

```

        case 'users':
            include 'pages/pages/admin/users/index.php';
            break;
        case 'action-user':
            include 'pages/pages/admin/users/action-user.php';
            break;
        default:
            include 'pages/pages/users/home.php';
    }
}
include 'pages/fixed/footer.php';

```

head.php

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Electronic portal</title>
    <link rel="stylesheet" href="assets/css/bootstrap.min.css">
    <script src="assets/js/bootstrap.bundle.js"></script>
    <script src="assets/js/jquery.js"></script>
</head>

<body class="d-flex flex-column min-vh-100">

```

Navigation.php

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container">
        <?php
            if (isset($_SESSION['users']) && $_SESSION['users']-
>roleName == "Admin") :
                ?>
                    <a class="navbar-brand" href="index.php?page=admin_home">E-portal</a>

```

```

        <?php else : ?>
            <a class="navbar-brand" href="index.php?page=home">E-portal</a>
        <?php endif; ?>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
                <?php
                    if (isset($_SESSION['users']) && $_SESSION['users']-
>roleName == "Admin") :
                        ?>
                            <li class="nav-
item"><a href="index.php?page=categories" class="nav-link">Categories</a></li>
                            <li class="nav-
item"><a href="index.php?page=headings" class="nav-link">Headings</a></li>
                            <li class="nav-
item"><a href="index.php?page=tags" class="nav-link">Tags</a></li>
                            <li class="nav-
item"><a href="index.php?page=posts" class="nav-link">Posts</a></li>
                            <li class="nav-
item"><a href="index.php?page=users" class="nav-link">Users</a></li>
                        <?php

                            elseif ((isset($_SESSION['users']) && $_SESSION['users']-
>roleName == "User") || (!isset($_SESSION['users']))) :
                                $categories = getAll('categories');
                                foreach ($categories as $category) : ?>
                                    <li class="nav-
item"><a href="index.php?page=news&categoryName=<?= $category-
>name ?>" class="nav-link"><?= $category->name ?></a></li>
                                <?php
                                    endforeach; ?>
                                    <li class="nav-
item"><a href="index.php?page=author" class="nav-link">Author</a></li>
                                <?php endif; ?>
                            </ul>
                            <ul class="navbar-nav mb-2 mb-lg-0">
                                <?php
                                    if (!isset($_SESSION['users'])) :
                                        ?>

```

```

        <li class="nav-
item"><a href="index.php?page=login" class="nav-link">Login</a></li>
        <li class="nav-
item"><a href="index.php?page=register" class="nav-link">Register</a></li>
        <?php else : ?>
        <li class="nav-
item"><a href="models/actions/logout.php" class="nav-link">Logout</a></li>
        <?php endif; ?>
    </ul>
</div>
</div>
</nav>

```

Footer.php

```

<footer class="mt-auto bg-light">
    <div class="container">
        <div class="row">
            <div class="col-10 mx-auto">
                <?php
                    if (isset($_SESSION['users']) && $_SESSION['users']-
>roleName == "User") :
                    ?>
                    <div class="row my-2">
                        <div class="col-4">
                            <h2>E-news</h2>
                        </div>
                        <div class="col-6 d-flex">
                            <a href="data/logs.txt" class="nav-
link">Dokumentacija</a>
                            <a href="#" class="nav-link">Sitemap</a>
                        </div>
                    </div>
                <?php endif; ?>
            </div>
        </div>
        <div class="row pb-2">
            <p class="text-center">All right reserved</p>
        </div>
    </div>
</footer>

```

```

<script src="assets/js/logAndReg.js"></script>

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

<script src="assets/js/admin.js"></script>
</body>

</html>

```

Status_403.php

```

<section>
  <div class="container">
    <div class="row my-5">
      <div class="col-6 mx-auto my-5">
        <h1 class="text-danger fs-1 text-center">Access Forbiden</h1>
        <p class="text-muted fts-italic fw-bolder text-center">You try to access unauthorized area!</p>
      </div>
    </div>
  </div>
</section>

```

Admin.js

```

$(document).ready(function () {
  // categories
  $(document).on('click', '#btnCategory', function (e) {
    e.preventDefault();
    categoryFromRequestAndValidation()
  })
  $(document).on('click', '.delete-category', function (e) {
    e.preventDefault()
    let id = $(this).data('id')
    $.ajax({
      method: 'post',
      url: 'models/categories/delete.php',

```

```

        data: { id: id },
        dataType: "json",
        success: function (data, statusTxt, xhr) {

            if (xhr.status == 204) {

                getAllCategories()
            }
        },
        error: function (jqXHR, statusTxt, xhr) {
            printResponseMessages(jqXHR.status, jqXHR.responseJSON, '#categoryResponseMessages', 'warning,danger')
        }
    })
})

// headings
$(document).on('click', '.delete-heading', function (e) {
    e.preventDefault();
    let id = $(this).data('id')
    $.ajax({
        method: 'post',
        url: 'models/headings/delete.php',
        data: { id: id },
        dataType: 'json',
        success: function (data, statusTxt, xhr) {
            if (xhr.status == 204) {
                getAllHeadings()
            }
        },
        error: function (jqXHR, statusTxt, xhr) {
            printResponseMessages(jqXHR.code, jqXHR.responseJSON, '#showResponseErrorMessage', 'warning,danger')
        }
    })
})

$(document).on('click', '#btnHeading', function (e) {
    e.preventDefault()
    headingFromRequestAndValidation()
})

// tags

$(document).on('click', '.delete-tag', function (e) {

```

```

        e.preventDefault()
        let id = $(this).data('id')
        $.ajax({
            method: 'post',
            url: 'models/tags/delete.php',
            data: { id: id },
            dataType: 'json',
            success: function (data, statusTxt, xhr) {
                if (xhr.status == 204) {
                    getAllTags()
                }
            },
            errors: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseText, '#', 'warning', warning')
            }
        })
    })
    $(document).on('click', '#btnTag', function (e) {
        e.preventDefault()
        tagsFormValidationAndRequest()
    })
    $(document).on('keyup', '#searchTags', function (e) {
        e.preventDefault()
        filterTags()
    })
    $(document).on('change', '#sortByDateTag', function (e) {
        e.preventDefault()
        filterTags()
    })
    $(document).on("click", '.tag-pagination', function () {
        let limit = $(this).data('limit')
        filterTags(limit)
    })

    // posts
    $(document).on('click', '.delete-post', function (e) {
        e.preventDefault()
        let id = $(this).data('id')
        $.ajax({
            method: 'post',
            url: 'models/posts/delete.php',
            data: { id: id },
            dataType: 'json',

```



```

        success: function (data, statusTxt, xhr) {
            if (xhr.status == 204) {
                getAllPosts()
            }
        }, error: function (jqXHR, statusTxt, xhr) {
            printResponseMessages(jqXHR.status, jqXHR.responseText, "#postShowErrorMessages", 'warning, danger')
        }
    })
})
$(document).on('click', '#submitPost', function (e) {
    e.preventDefault()
    postFormVaildationAndRequest()
})
$(document).on('keyup', '#searchPost', function () {
    // e.preventDefault()
    filterPosts()
})
$(document).on('change', '.categories', function (e) {
    e.preventDefault()
    setCategories()
})
$(document).on('change', '.headings', function (e) {
    e.preventDefault()
    setHeadings()
})
$(document).on('change', '#postCategory', function (e) {
    e.preventDefault()
    let id = $(this).val()
    $.ajax({
        method: "GET",
        url: "models/posts/getHeadingsByCategory.php",
        data: { id: id },
        dataType: 'json',
        success: function (data, statusTxt, xhr) {
            printPostHeadings(data)
        },
        error: function (jqXHR, statusTxt, xhr) {
            console.log(xhr)
        }
    })
})
$(document).on('change', '#sortByDatePost', function (e) {
    e.preventDefault()
    filterPosts()
})

```

```

    })
    $(document).on('click', '.post-pagination', function () {

        let limit = $(this).data('limit')
        filterPosts(limit)
    })

    // users
    $(document).on('click', '.delete-user', function (e) {
        e.preventDefault()
        let id = $(this).data('id')
        $.ajax({
            method: 'post',
            url: 'models/users/delete.php',
            data: { id: id },
            dataType: 'json',
            success: function (data, statusTxt, xhr) {
                if (xhr.status == 204) {
                    getAllUsers()
                }
            },
            error: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseJSON, '#showUserErrorMessages', 'warning,danger')
            }
        })
    })

    $(document).on('click', '#btnSubmitUser', function (e) {
        e.preventDefault()
        userFormValidationAndRequest()
    })

    $(document).on('keyup', '#searchUser', function (e) {
        e.preventDefault()
        filterUser()
    })

    $(document).on('change', '#sortByDateUsers', function (e) {
        e.preventDefault()
        filterUser()
    })

    $(document).on("click", '.user-pagination', function () {
        let limit = $(this).data('limit')
        filterUser(limit)
    })

    const filterUser = (limit) => {

```

```

        filterAndSortUsers(limit)
    }
    const filterAndSortUsers = (limit) => {
        let text = document.querySelector('#searchUser').value
        let date = document.querySelector('#sortByDateUsers').value
        console.log(date)
        $.ajax({

            method: 'get',
            url: 'models/users/filter.php',
            data: {
                text: text,
                date: date,
                limit: limit
            }, dataType: 'json',
            success: function (data) {
                console.log(data.users)
                printAllUsers(data.users, limit)
                printPagination(data.pages, '#userPagination', limit, 'user-
pagination')
            },
            error: function (jqXHR, statusTxt, xhr) { }

        })
    }

    // category functions
    const getAllCategories = () => {
        $.ajax({
            method: 'get',
            url: 'models/categories/getAll.php',
            dataType: 'json',
            success: function (data) {

                printAllCategories(data)
            },
            error: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseText, '#categoryResponseMessages', 'warning, danger')
            }
        })
    }

    const categoryFromRequestAndValidation = () => {
        const id = document.querySelector('#categoryId').value

```

```

const name = document.querySelector("#categoryName").value
if (id == "") {
    if (validationCategory().length == 0) {
        $.ajax({
            method: 'post',
            url: 'models/categories/insert.php',
            data: { name: name },
            dataType: 'json',
            success: function (data, statusTxt, xhr) {
                if (xhr.status == 201) {
                    window.location.href = 'index.php?page=categories'
                }
            },
            error: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseText, '
#showDbResponseErrorMessages', 'warning,danger')
            }
        })
    }
} else {
    if (validationCategory().length == 0) {
        $.ajax({
            method: 'post',
            url: 'models/categories/update.php',
            data: { name: name, id: id },
            dataType: 'json',
            success: function (data, statusTxt, xhr) {
                if (xhr.status == 204) {
                    window.location.href = "index.php?page=categories"
                }
            },
            error: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseText, '
#showDbResponseErrorMessages', 'warning,danger')
            }
        })
    }
}
}

const validationCategory = () => {
    const name = document.querySelector('#categoryName').value
    const reName = /^[A-Z][a-z]{3,15}$/
    const errors = []
    if (!reName.test(name)) {
        errors.push(name)
    }
}

```

```

        createValidationErrorMessage('#categoryNameErrorMessage', 'text-
danger', `Name of category isn't ok`)
      } else {
        removeValidationErrorMessage('#categoryNameErrorMessage', 'text-
danger')
      }

      return errors
    }

const printAllCategories = (categories) => {
  let ispis = ' ', rb = 1
  if (categories.length > 0) {
    categories.forEach(category => {
      ispis += printCategory(category, rb)
      rb++
    })
  } else {
    ispis += NoContent("No category at this time", '6')
  }
  $('#categories').html(ispis)
}

const printCategory = (category, rb) => {
  return `
    <tr>
      <th scope='row'>${rb++}</th>
      <td>${category.name}</td>
      <td>${category.created_at}</td>
      <td>${category.updated_at ? category.updated_at : '-'}</td>
      <td><a href="index.php?page=action-
categories&id=${category.id}" class="btn btn-sm btn-success">Update</a></td>
      <td><button type="button" class="btn btn-sm btn-danger delete-
category" data-id="${category.id}">Delete</button></td>
    </tr>
  `
}

// headings

const getAllHeadings = () => {
  $.ajax({
    method: 'get',
    url: 'models/headings/getAll.php',
    dataType: 'json',
  })
}

```

```

        success: function (data) {
            printAllHeadings(data)
        },
        error: function () { }
    })
}

const printAllHeadings = (headings) => {
    let ispis = '', rb = 1
    if (headings.length > 0) {
        headings.forEach(heading => {
            ispis += printHeading(heading, rb)
            rb++
        })
    } else {
        ispis += NoContent('No headings at the moment', '7')
    }
    $('#headings').html(ispis)
}

const printHeading = (heading, rb) => {
    return `
        <tr>
            <th scope='row'>${rb}</th>
            <td>${heading.name}</td>
            <td>${heading.categoryName}s</td>
            <td>${heading.created_at}</td>
            <td>${heading.updated_at ? heading.updated_at : '-'}</td>
            <td><a href="index.php?page=action-heading" class="btn btn-sm btn-success">Update</a></td>
            <td><button type="button" class="btn btn-sm btn-danger" data-id="${heading.id}">Delete</button></td>
        </tr>
    `
}

const headingFromRequestAndValidation = () => {
    const id = document.querySelector('#headingId').value
    const name = document.querySelector('#headingName').value
    const categoryId = document.querySelector('#headingCategory').value
    if (id == "") {
        if (validationHeading().length == 0) {
            $.ajax({
                method: 'post',
                url: 'models/headings/insert.php',
                data: { name: name, categoryId: categoryId },
                dataType: 'json',
                success: function (data, statusTxt, xhr) {

```

```

        if (xhr.status == 201) {
            window.location.href = 'index.php?page=headings'
        }
    },
    error: function (jqXHR, statusTxt, xhr) {
        printResponseMessages(jqXHR.status, jqXHR.responseText, '
#crudResponseErrorsMessages', 'warning,danger')
    }
    })
}
} else {
    if (validationHeading().length == 0) {
        $.ajax({
            method: 'post',
            url: 'models/headings/update.php',
            data: { name: name, categoryId: categoryId, id: id },
            dataType: 'json',
            success: function (data, statusTxt, xhr) {
                if (xhr.status == 204) {
                    window.location.href = 'index.php?page=headings'
                }
            },
            error: function (jqXHR, statusTxt, xhr) {
                printResponseMessages(jqXHR.status, jqXHR.responseText, '
#crudResponseErrorsMessages', 'warning,danger')
            }
        })
    }
}
}

const validationHeading = () => {
    const name = document.querySelector('#headingName').value
    const categoryId = document.querySelector('#headingCategory').value
    const reName = /^[A-Z][a-z]{2,15}(\s)?([A-Z][a-z]{2,15})*/$/
    const errors = []
    if (!reName.test(name)) {
        errors.push(name)
        createValidationErrorMessage('#headingNameErrorMessage', 'text-
danger', "Heading name isn't ok!")
    } else {
        removeValidationErrorMessage('#headingNameErrorMessage', 'text-
danger')
    }
    if (categoryId == 0) {
        errors.push(categoryId)
    }
}

```

```

        createValidationErrorMessage('#headingCategoryErrorMessage', 'text-
danger', "You must choose category");
    } else {
        removeValidationErrorMessage('#headingCategoryErrorMessage', 'text-
danger')
    }

    return errors
}

// tags
const tagsFormValidationAndRequest = () => {
    const name = document.querySelector("#tagName").value
    const id = document.querySelector("#tagId").value
    console.log(id)
    if (id == " ") {

        if (validationTags().length == 0) {
            $.ajax({
                method: 'post',
                url: 'models/tags/insert.php',
                data: { name: name },
                success: function (data, statusTxt, xhr) {

                    if (xhr.status == 201) {
                        window.location.href = "index.php?page=tags"
                    }
                },
                error: function (jqXHR, statusTxt, xhr) {
                    printResponseMessages(jqXHR.status, jqXHR.responseJSON, '
#crudShowErrorMessages', 'warning,danger')
                }
            })
        }
    } else {

        if (validationTags().length == 0) {
            $.ajax({
                method: 'post',
                url: 'models/tags/update.php',
                data: { name: name, id: id },
                success: function (data, statusTxt, xhr) {
                    if (xhr.status == 204) {
                        window.location.href = 'index.php?page=tags'
                    }
                }
            })
        }
    }
}

```



```

    },
    error: function (jqXHR, statusTxt, xhr) {
        printResponseMessages(jqXHR.status, jqXHR.responseJSON, '
#crudShowErrorMessage', 'warning,danger')
    }
})
}
}
}

const validationTags = () => {
    const name = document.querySelector('#tagName').value
    const reNameTag = /^[A-Z][a-z]{1,15}(\s[A-Z][a-z]{3,15})?$/
    const errors = []
    if (!reNameTag.test(name)) {
        errors.push(name)
        createValidationErrorMessage('#tagNameErrorMessage', 'text-
danger', "Tag name isn't ok")
    } else {
        removeValidationErrorMessage('#tagNameErrorMessage', 'text-danger')
    }
    return errors
}

const getAllTags = () => {
    $.ajax({
        method: 'get',
        url: "models/tags/getAll.php",
        dataType: 'json',
        success: function (data) { printAllTags(data) },
        error: function (jqXHR, statusTxt, xhr) {
            printResponseMessages(jqXHR.status, jqXHR.responseJSON, '#', 'dan
ger,warning')
        }
    })
}

const printAllTags = (tags, limit) => {

    let ispis = ''
    let rB = ''
    if (!limit) {
        rB += 1
    } else {
        rB += limit * 5 + 1
    }
    if (tags.length > 0) {

```

```

    tags.forEach(tag => {
      ispis += printTag(tag, rB, limit)
      rB++
    })
  } else {
    ispis += NoContent("At the moment we don't have tags", '6')
  }
  $('#tags').html(ispis)
}
const printTag = (tag, rB) => {

  return `
    <tr>
      <th scope='row'>${rB}</th>
      <td>${tag.name}</td>
      <td>${tag.created_at}</td>
      <td>${tag.updated_at ? tag.updated_at : '-'}</td>
      <td><a href="index.php?page=action-
tag&id=${tag.id}" class="btn btn-sm btn-success">Update</a></td>
      <td><button type="button" class="btn btn-sm btn-danger delete-
tag" data-id="${tag.id}">Delete</button></td>
    </tr>
  `

}
const filterTags = (limit) => {
  filterAndSortByTags(limit)
}
const filterAndSortByTags = (limit) => {
  let text = document.querySelector("#searchTags").value
  let sortByDate = document.querySelector("#sortByDateTag").value
  $.ajax({
    method: 'get',
    url: 'models/tags/filter.php',
    data: {
      text: text,
      sortByDate: sortByDate,
      limit: limit
    },
    dataType: 'json',
    success: function (data) {

      printAllTags(data.tags, limit)
      data.pages > 1 ? printPagination(data.pages, '#tagPagination', li
mit, 'tag-pagination') : document.querySelector("#tagPagination").innerHTML = ""

```

```

    },
    error: function (jqXHR, statusTxt, xhr) { }
  })
}

//users
const getAllUsers = () => {
  $.ajax({
    method: 'get',
    url: 'models/users/getAll.php',
    dataType: 'json',
    success: function (data) {
      printAllUsers(data)
    }
  })
}

const printAllUsers = (users) => {
  console.log(users)
  ispis = '', rb = 1
  if (users.length > 0) {
    users.forEach(user => {
      ispis += printUser(user, rb)
      rb++
    })
  } else {
    ispis += NoContent('At this moment we dont have any user', '8')
  }
  $('#users').html(ispis)
}

const printUser = (user, rb) => {
  return `
    <tr>
      <th>${rb}</th>
      <td>${user.first_name + ' ' + user.last_name}</td>
      <td>${user.email}</td>
      <td>${user.created_at}</td>
      <td>${user.updated_at ? user.updated_at : '-'}</td>
      <td><a href="index.php?page=action-
user&id=${user.id}" class="btn btn-success btn-sm">Update</a></td>
      <td><button type="button" class="btn btn-danger delete-user btn-
sm" data-id="${user.id}">Delete</button></td>
    </tr>
  `
}

```

```

const userFormValidationAndRequest = () => {
  const id = document.querySelector('#userId').value
  const first_name = document.querySelector('#userFirstName').value
  const last_name = document.querySelector('#userLastName').value
  const email = document.querySelector('#userEmail').value

  const password = id == "" ? document.querySelector("#userPassword").value
: ""

  const role = $('input[name="userRole"]:checked').val()

  if (id == "") {
    if (validationUserForm().length == 0) {
      $.ajax({
        method: 'post',
        url: 'models/users/insert.php',
        data: {
          first_name: first_name,
          last_name: last_name,
          email: email,
          password: password,
          role: role
        },
        success: function (data, statusTxt, xhr) {
          if (xhr.status == 201) {
            window.location.href = "index.php?page=users"
          }
        }, error: function (jqXHR, statusTxt, xhr) {
          printResponseMessages(jqXHR.status, jqXHR.responseJSON, '
#crudUserErrorMessage', 'danger,warning')
        }
      })
    }
  } else {
    if (validationUserForm().length == 0) {
      $.ajax({
        method: 'post',
        url: 'models/users/update.php',
        data: {
          id: id,
          first_name: first_name,
          last_name: last_name,
          email: email,
          role: role
        },
        success: function (data, statusTxt, xhr) {

```

```

        if (xhr.status == 204) {
            window.location.href = "index.php?page=users"
        }
    }, error: function (jqXHR, statusTxt, xhr) {
        printResponseMessages(jqXHR.status, jqXHR.responseJSON, '#crudUserErrorMessage', 'danger,warning')
    }
    })
}
}
}

const validationUserForm = () => {
    const id = document.querySelector('#userId').value
    const first_name = document.querySelector('#userFirstName').value
    const last_name = document.querySelector('#userLastName').value
    const email = document.querySelector('#userEmail').value

    const password = id == "" ? document.querySelector("#userPassword").value
: ""

    const role = $('input[name="userRole"]:checked').val()

    const reFirstLastName = /^[A-Z][a-z]{3,15}$/
    const reEmail = /^[\w-\.]@([\w-]+\.)+[\w-]{2,4}$/
    const rePassword = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/

    const errors = []

    if (!reFirstLastName.test(first_name)) {
        errors.push(first_name)
        createValidationErrorMessage('#userFirstNameErrorMessage', 'text-
danger', "First name of user isn't ok")
    } else {
        removeValidationErrorMessage('#userFirstNameErrorMessage', 'text-
danger')
    }
    if (!reFirstLastName.test(last_name)) {
        errors.push(last_name)
        createValidationErrorMessage('#userLastNameErrorMessage', 'text-
danger', "Last name of user isn't ok")
    } else {
        removeValidationErrorMessage("#userLastNameErrorMessage", 'text-
danger')
    }
    if (!reEmail.test(email)) {

```

```

        errors.push(email)
        createValidationErrorMessage('#userEmailErrorMessage', 'text-
danger', "Your email isn't ok")
    } else {
        removeValidationErrorMessage('#userEmailErrorMessage', 'text-
danger')
    }
    if (id == "") {
        if (!rePassword.test(password)) {
            errors.push(password)
            createValidationErrorMessage("#userPasswordErrorMessage", 'text-
danger', "Your password isn't ok")
        } else {
            removeValidationErrorMessage('#userPasswordErrorMessage', 'text-
danger')
        }
    }
    // if (!reUsername.test(username)) {
    //     errors.push(username)
    //     createValidationErrorMessage('#userUsernameErrorMessage', 'text-
danger', "Your username isn't ok")
    // } else {
    //     removeValidationErrorMessage('#userUsernameErrorMessage', 'text-
danger')
    // }
    if (!role) {
        errors.push(role)
        createValidationErrorMessage('#userRoleErrorMessage', 'text-
danger', 'You must choose role')
    } else {
        removeValidationErrorMessage('#userRoleErrorMessage', 'text-danger')
    }
    return errors
}
// posts

const getAllPosts = () => {
    $.ajax({
        method: "get",
        url: 'models/posts/getAll.php',
        dataType: 'json',
        success: function (data) {
            printAllPost(data)
        },
        error: function (jqXHR, statusTxt, xhr) { }
    })
}

```

```

    })
  }
  const printAllPost = (posts, limit) => {
    console.log(posts)
    let ispis = '', rb = 1
    if (posts.length > 0) {
      posts.forEach(post => {
        ispis += printPost(post, rb)
        rb++
      })
    } else {
      ispis += NoContent('At this time we dont have any posts', '8')
    }
    $('#posts').html(ispis)
  }
  const printPost = (post, rb) => {

    return `
      <tr>
        <th scope="row">${rb}</th>
        <td>${post.name}</td>
        <td>${post.categoryName}</td>
        <td>${post.headingName}</td>
        <td>${post.created_at}</td>
        <td>${post.updated_at ? post.updated_at : '-'}</td>
        <td><a href="index.php?page=action_post&id=${post.id}" class="btn
btn-sm btn-success">Update</a></td>
        <td><button type="button" class="btn btn-sm btn-danger" data-
id="${post.id}">Delete</button></td>
        <td><a href="index.php?page=post_details&id=${post.id}" class="bt
n btn-info btn-sm">Details</a></td>
        <td></td>
      </tr>
    `
  }

  const postFormVaildationAndRequest = () => {
    const id = document.querySelector("#postId").value
    const name = document.querySelector('#postName').value
    const postDesc = document.querySelector('#postDescription').value
    const postImage = document.getElementById('postImage').files
    //const postImageSrc = document.querySelector('#postImageSrc').src
    const postCategory = document.querySelector('#postCategory').value
    const postHeading = document.querySelector('#postHeading').value
    const tags = $('input[name="postTags"]:checked')
    const selectedTags = []
  }

```

```

for (let tag of tags) {
    selectedTags.push(tag.value)
}

let formData = new FormData()
formData.append('name', name)
formData.append('postDesc', postDesc)
formData.append('category', postCategory)
formData.append('heading', postHeading)
formData.append('tags', selectedTags)
if (id == "") {
    formData.append('image', $('#postImage')[0].files[0])
    if (postValidation().length == 0) {
        $.ajax({
            method: 'post',
            url: "models/posts/insert.php",
            data: formData,
            contentType: false,
            processData: false,
            success: function (data, statusTxt, xhr) {

                if (xhr.status == 201) {
                    window.location.href = "index.php?page=posts"
                }
            },
            error: function (jqXHR, statusTxt, xhr) {

                printResponseMessages(jqXHR.status, jqXHR.responseJSON, '
#crudPostErrorMessages', 'warning,danger')
            }
        })
    }
} else {
    if (postImage.length != 0) {
        formData.append('image', $('#postImage')[0].files[0])
    }
    formData.append('id', id)
    if (postValidation().length == 0) {
        $.ajax({
            method: 'post',
            url: 'models/posts/update.php',
            data: formData,
            processData: false,
            contentType: false,

```



```

        success: function (data, statusTxt, xhr) {
            if (xhr.status == 204) {
                window.location.href = "index.php?page=posts"
            }
        }, error: function (jqXHR, statusTxt, xhr) {
            printResponseMessages(jqXHR.status, jqXHR.responseJSON, '
#crudPostErrorMessages', 'warning,danger')
        }
    })
}
}
}

const postValidation = () => {
    const id = document.querySelector("#postId").value
    const name = document.querySelector('#postName').value
    const postDesc = document.querySelector('#postDescription').value
    const postImage = document.getElementById('postImage').files
    const postCategory = document.querySelector('#postCategory').value
    const postHeading = document.querySelector('#postHeading').value
    const tags = $('input[name="postTags"]:checked')

    const selectedTags = []
    for (let tag of tags) {
        selectedTags.push(tag.value)
    }

    const errors = []
    if (name == '') {
        errors.push(name)
        createValidationErrorMessage('#postNameErrorMessage', 'text-
danger', "Post name isn't ok!")
    } else {
        removeValidationErrorMessage("#postNameErrorMessage", 'text-
danger')
    }
    if (postDesc == '') {
        errors.push(postDesc)
        createValidationErrorMessage('#postDescErrorMessage', 'text-
danger', "Description can't be empty")
    }
    else {
        removeValidationErrorMessage('#postNameErrorMessage', 'text-
danger')
    }
}

```

```

        if (id == "") {
            if (postImage.length == 0) {
                errors.push(postImage)
                createValidationErrorMessage('#postImageErrorMessage', 'text-
danger', "You must choose image")
            } else {
                removeValidationErrorMessage('#postImageErrorMessage', 'text-
danger')
            }
        }
        if (postCategory == 0) {
            errors.push(postCategory)
            createValidationErrorMessage('#postCategoryErrorMessage', 'text-
danger', "You must choose category for the post!")
        } else {
            removeValidationErrorMessage('#postCategoryErrorMessage', 'text-
danger')
        }
        if (postHeading == 0) {
            errors.push(postHeading)
            createValidationErrorMessage('#postHeadingErrorMessage', 'text-
danger', 'You must choose heading for the post')
        } else {
            removeValidationErrorMessage('#postHeadingErrorMessage', 'text-
danger')
        }
        if (selectedTags.length == 0) {
            errors.push(selectedTags)
            createValidationErrorMessage("#postTagsErrorMessage", 'text-
danger', "You must choose at least one tag")
        } else {
            removeValidationErrorMessage('#postTagsErrorMessage', 'text-danger')
        }
        return errors
    }
    const setCategories = () => {
        let selectedCategories = []
        let categories = $('input[name="product_categories"]:checked')
        for (var category of categories) {
            selectedCategories.push(category.value)
        }
        filterPosts()
    }
    const setHeadings = () => {
        let selectedHeadings = []

```

```

        let headings = $('input[name="product_heading"]:checked')
        for (let heading of headings) {
            selectedHeadings.push(heading.value)
        }
        filterPosts()
    }
    const filterPosts = (limit) => {
        filterAndSortPosts(limit)
    }
    const filterAndSortPosts = (limit) => {
        let search = document.querySelector('#searchPost').value

        let selectedCategories = []
        let categories = $('input[name="product_categories"]:checked')
        for (var category of categories) {
            selectedCategories.push(category.value)
        }

        let selectedHeadings = []
        let headings = $('input[name="product_heading"]:checked')
        for (let heading of headings) {
            selectedHeadings.push(heading.value)
        }

        let date = document.querySelector('#sortByDatePost').value
        $.ajax({
            method: 'get',
            url: 'models/posts/filter.php',
            data: {
                text: search,
                categories: selectedCategories,
                headings: selectedHeadings,
                limit: limit,
                date: date,
            },
            dataType: 'json',
            success: function (data) {
                console.log(data)
                printAllPost(data.posts, limit)
                printPagination(data.pages, '#postPagination', limit, 'post-
pagination')
            },
        })
    }
}

```

```

const printPostHeadings = (headings) => {
  console.log(headings)
  let ispis = ''
  ispis += "<option value='0'>Choose</option>"
  if (headings.length > 1) {
    headings.forEach(heading => {
      ispis += `<option value="${heading.id}">${heading.name}</option>`
    })
  }
  $('#postHeading').html(ispis)
}

// utilites

const createValidationErrorMessage = (element, cls, text) => {
  const el = document.querySelector(element)
  el.classList.add(cls)
  el.textContent = text
}

const removeValidationErrorMessage = (element, cls, text) => {
  const el = document.querySelector(element)
  el.textContent = ''
  el.classList.remove(cls)
}

const printResponseMessages = (statusCode, message, whereToPlace, colors) =>
{
  const color = colors.split(',')
  const colorYellow = color[0]
  const colorDanger = color[1]
  switch (statusCode) {
    case 404:
      printMessage(message, whereToPlace, colorDanger)
      break;
    case 409:
      printMessage(message, whereToPlace, colorYellow)
      break;
    case 422:
      printMessage(message, whereToPlace, colorDanger)
      break;
    case 500:
      printMessage(message, whereToPlace, colorDanger)
      break;
  }
}

```

```

    }
  }
  const printMessage = (message, whereToPlace, color) => {

    let ispis = ''
    ispis += `
      <div class="alert alert-${color} alert-
dismissible fade show" role="alert">
        ${message}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
      </div>
    `

    $(whereToPlace).html(ispis)
  }

  const NoContent = (message, number, cls) => {
    return `
      <tr>
        <th scope="row" colspan="${number}">${message}</th>
      </tr>
    `
  }

  const printPagination = (numOfPages, whereToPlace, limit, cls) => {
    let ispis = ''
    let lim = 1
    if (limit) {
      lim += limit
    }
    for (let i = 0; i < numOfPages; i++) {
      ispis += ` <li class="page-
item ${lim == (i + 1) ? 'active' : ''}"><a class="page-
link ${cls}" href="#" data-limit="${i}">${i + 1}</a></li>`
    }
    $(whereToPlace).html(ispis)
  }
})

```

adminChart.js

```
$(document).ready(function () {

  function pieChartPages() {
    $.ajax({
      url: "models/actions/getStats.php",
      method: "GET",
      dataType: "JSON",
      success: function (data) {
        console.log(data)
        // console.log(data)
        newPie('viewsTotal', 'Total', data.overallViews)
        newPie('views24H', '24h', data.todayViews)

        $('#registeredUsers').html(data.registeredUsers.users)
        $('#todayLogins').html(data.todayLogin)
      }
    })
  }

  const newPie = (wherePie, whereData, data) => {
    var labels = Object.keys(data);
    var values = Object.values(data);
    var sum = values.reduce(function (a, b) {
      return a + b;
    }, 0);
    var text = whereData == "24h" ? 'in last 24h' : ''
    let html = `

# Total ${text}<span class="fs-4 bold">: ${sum}</span></h1>` $('#${wherePie}`).html(html) $('#${whereData}`).html(viewPieTBody(data, sum)) var ch = $('#chart' + whereData) var chart = new Chart(ch, { type: 'pie', data: { labels: labels, datasets: [{ data: values, backgroundColor: [ 'rgba(87, 196, 229)', 'rgba(221, 64, 58)', 'rgba(255, 208, 70)' ] }] } }) }


```

```

        'rgba(8, 164, 189)'
    ]
    }]
  }
})

}

const viewPieTBody = (data, sum) => {
  var html = "";
  for (const key in data) {
    html += `<tr><td class="text-start align-middle text-
uppercase">${key}</td>`
    if (sum !== 0)
      html += `<td class="text-end fs-4">${data[key]}
(${(data[key] * 100 / sum).toFixed(2)}%)</td>`
    else
      html += `<td class="text-end">0 (0%)</td>`
    html += `</tr>`
  }
  return html;
}

pieChartPages()
});

```

logAndReg.js

```

$(document).ready(function () {
  $(document).on('click', '#btnRegister', function (e) {
    e.preventDefault()
    userFormRegistrationAndValidation()
  })
  const userFormRegistrationAndValidation = () => {
    const firstName = document.querySelector('#userFirstName').value
    const lastName = document.querySelector('#userLastName').value
    const email = document.querySelector("#userEmail").value
    const password = document.querySelector('#userPassword').value

    if (validationForm().length == 0) {

```

```

$.ajax({
  method: 'post',
  url: 'models/actions/register.php',
  data: {
    firstName: firstName,
    lastName: lastName,
    email: email,
    password: password
  }, dataType: 'json',
  success: function (data, statusTxt, xhr) {
    if (xhr.status == 201) {
      window.location.href = 'index.php?page=login'
    }
  }, error: function (jqXHR, statusTxt, xhr) {
    printResponseMessages(jqXHR.status, jqXHR.responseJSON, '#registerResponseErrorMessage', 'warning,danger')
  }
})
}

}

const validationForm = () => {
  const firstName = document.querySelector('#userFirstName').value
  const lastName = document.querySelector('#userLastName').value
  const email = document.querySelector("#userEmail").value
  const password = document.querySelector('#userPassword').value

  const reFirstLastName = /^[A-Z][a-z]{3,15}$/
  const reEmail = /^[\w-\.\.]+\@([\w-]+\.\.)+[\w-]{2,4}$/
  const rePassword = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/

  const errors = []
  if (!reFirstLastName.test(firstName)) {
    errors.push(firstName)
    createValidationErrorMessage('#userFirstNameErrorMessage', 'text-danger', "First name of user isn't ok")
  } else {
    removeValidationErrorMessage('#userFirstNameErrorMessage', 'text-danger')
  }
  if (!reFirstLastName.test(lastName)) {
    errors.push(lastName)
    createValidationErrorMessage('#userLastNameErrorMessage', 'text-danger', "Last name of user isn't ok")
  }
}

```



```

    } else {
      removeValidationErrorMessage("#userLastNameErrorMessage", 'text-
danger')
    }
    if (!reEmail.test(email)) {
      errors.push(errors)
      createValidationErrorMessage('#userEmailErrorMessage', 'text-
danger', "Email of user isn't ok")
    } else {
      removeValidationErrorMessage('#userEmailErrorMessage', 'text-
danger')
    }
    if (!rePassword.test(password)) {
      errors.push(password)
      createValidationErrorMessage('#userPasswordErrorMessage', 'text-
danger', "Password of user isn't ok")
    } else {
      removeValidationErrorMessage('#userPasswordErrorMessage', 'text-
danger')
    }
    return errors
  }
  const createValidationErrorMessage = (element, cls, text) => {
    const el = document.querySelector(element)
    el.classList.add(cls)
    el.textContent = text
  }
  const removeValidationErrorMessage = (element, cls, text) => {
    const el = document.querySelector(element)
    el.textContent = ''
    el.classList.remove(cls)
  }

  const printResponseMessages = (statusCode, message, whereToPlace, colors) =>
{
  console.log(statusCode)
  const color = colors.split(',')
  const colorYellow = color[0]
  const colorDanger = color[1]
  switch (statusCode) {
    case 404:
      printMessage(message, whereToPlace, colorDanger)
      break;
    case 409:
      printMessage(message, whereToPlace, colorYellow)

```

```

        break;
    case 422:
        printMessage(message, whereToPlace, colorDanger)
        break;
    case 500:
        printMessage(message, whereToPlace, colorDanger)
        break;
    }
}
const printMessage = (message, whereToPlace, color) => {

    let ispis = ''
    ispis += `
        <div class="alert alert-${color} alert-
dismissible fade show" role="alert">
            ${message}
            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
        </div>
    `

    $(whereToPlace).html(ispis)
}

const NoContent = (message, number) => {
    return `
        <tr>
            <th scope="row" colspan="${number}">${message}</th>
        </tr>
    `
}

})

```

Main.js

```

$(document).ready(function () {
    $(document).on('click', '#btnSubmit', function (e) {
        e.preventDefault();
        let text = document.querySelector('.textarea-commnet').value
        let post_id = $(this).data('post')
        $.ajax({
            method: "POST",
            url: "models/comments/insert.php",

```

```

        data: { text: text, post_id: post_id },
        dataType: 'json',
        success: function () {
            getComments(post_id)
            document.querySelector('.textarea-commnet').value = " "
        },
        error: function (jqXHR, statusTxt, xhr) { }
    })
})

const getComments = (post_id) => {
    $.ajax({
        method: "get",
        url: "models/comments/getComments.php",
        dataType: 'json',
        data: { id: post_id },
        success: function (data) {
            printComments(data)
        }
    })
}

const printComments = (comments) => {
    let ispis = ''
    if (comments.length == 1) {
        ispis += printComment(comments)
    } else {
        comments.forEach(comment => {
            console.log(comment)
            ispis += printComment(comment)
        })
    }
    $('#comments').html(ispis)
}

const printComment = (comment) => {
    console.log(comment)
    return `
    <div class="card mb-1">
        <div class="card-header">
            <em class="float-start fts-italic text-
uppercase">${comment.firstName + ' ' + comment.lastName}</em>
            <em class="float-end fts-italic text-uppercase text-muted fw-
bold">${comment.created_at}</em>

```

```

        </div>
        <div class="card-body">
            <p class="fts-italic">${comment.text}</p>
        </div>
    </div>
    `
}
});

```

Pages/admin/categories/index.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
$categories = getAll("categories");
?>
<section>
    <div class="container">
        <div class="row mt-3">
            <div class="col-lg-3">
                <div class="d-grid"><a href="index.php?page=action-
categories" class="btn btn-primary">Add new category</a></div>
            </div>
            <div class="col-lg-3 mt-3 mt-lg-0">
                <div class="d-grid">
                    <a href="models/actions/exportData.php?action=excel" class="b
tn btn-danger">Export to excel</a>
                </div>
            </div>
        </div>
        <div class="row my-3">
            <div class="col">
                <div id="categoryResponseMessages"></div>
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead class="table-dark">
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Name</th>
                                <th scope="col">Created at</th>

```

```

        <th scope="col">Updated at</th>
        <th scope="col">Update</th>
        <th scope="col">Delete</th>
    </tr>
</thead>
<tbody id="categories">
    <?php
    if (count($categories) == 0) :
    ?>
        <th scope="col" colspan="6">No one category at th
is time</th>

    <?php else : ?>
        <?php
        $rb = 1;

        foreach ($categories as $category) :
        ?>
            <tr>
                <th scope="row"><?= $rb++ ?></th>
                <td><?= $category->name ?></td>
                <td><?= $category->created_at ?></td>
                <td><?= $category-
>updated_at ? $category->updated_at : '-' ?></td>
                <td><a href="index.php?page=action-
categories&id=<?= $category->id ?>" class="btn btn-sm btn-
success">Update</a></td>
                <td><button type="button" class="btn btn-
sm btn-danger delete-category" data-id="<?= $category-
>id ?>">Delete</button></td>
            </tr>
            <?php endforeach; ?>
        <?php endif; ?>

    </tbody>
</table>
</div>
</div>
</div>
</div>
</section>

```

Pages/admin/categories/create_update.php

```
<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-6 mx-auto">
                <div id="showDbResponseErrorMessage"></div>
                <?php
                if (isset($_GET['id'])) {
                    $category = getDataWithFetch('categories', 'id', $_GET['id'])
;
                }
                ?>
                <form action="#" method="#">
                    <input type="hidden" name="categoryId" id="categoryId" value=
"= isset($category) ? $category-&gt;id : '' ?&gt;"&gt;
                    &lt;div class="mb-3"&gt;
                        &lt;label for="categoryName"&gt;Name:&lt;/label&gt;
                        &lt;input type="text" name="categoryName" id="categoryName"
class="form-control" value="<?= isset($category) ? $category-&gt;name : '' ?&gt;"&gt;
                        &lt;em id="categoryNameErrorMessage"&gt;&lt;/em&gt;
                    &lt;/div&gt;
                    &lt;div class="d-grid"&gt;
                        &lt;button class="btn btn-
primary" type="button" id="btnCategory"&gt;<?= isset($_GET['id']) ? "Update" : "Save
" ?&gt;&lt;/button&gt;
                    &lt;/div&gt;
                &lt;/form&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/section&gt;</pre
```

Pages/admin/headings/index.php

```
<?php
```

```

$headings = getHeadingsWithCategories();
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
    <div class="container">
        <div class="row mt-3">
            <div class="col-lg-3">
                <div class="d-grid"><a href="index.php?page=action-
heading" class="btn btn-primary">Add new heading</a></div>
            </div>
        </div>
        <div class="row my-3">
            <div class="col">
                <div id="showResponseErrorMessages"></div>
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead class="table-dark">
                            <tr>
                                <th>#</th>
                                <th>Name</th>
                                <th>Category name</th>
                                <th>Created at</th>
                                <th>Updated at</th>
                                <th>Update</th>
                                <th>Delete</th>
                            </tr>
                        </thead>
                        <tbody id='headings'>
                            <?php
                                $rb = 1;
                                $headings = $conn-
>query("SELECT h.*, c.name as categoryName FROM headings h JOIN categories c ON h
.category_id = c.id")->fetchAll();
                                if (count($headings) > 0) :
                                    foreach ($headings as $heading) :
                                        ?>
                                            <tr>
                                                <th><?= $rb++ ?></th>
                                                <td><?= $heading->name ?></td>
                                                <td><?= $heading->categoryName ?></td>
                                                <td><?= $heading->created_at ?></td>

```

```

<td><?= $heading->updated_at ? $heading-
>updated_at : '-' ?></td>
<td><a href="index.php?page=action-
heading&id=<?= $heading->id ?>" class=" btn btn-sm btn-success">Update</a></td>
<td><button type="button" class="btn btn-
danger btn-sm delete-heading" data-id="<?= $heading->id ?>">Delete</button></td>
</tr>
<?php endforeach;
else :
?>
<tr>
<th scope="row" colspan="7">No headings at th
is moment</th>
</tr>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</section>

```

Pages/admin/headings/action_heading.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
<div class="container">
<div class="row my-3">
<div class="col-6 mx-auto">
<em id="crudResponseErrorsMessages"></em>
<?php
if (isset($_GET['id'])) {
    $heading = getDataWithFetch('headings', 'id', $_GET['id']);
}
?>
<form action="#" method="#">

```



```

        <input type="hidden" name="headingId" id="headingId" value="<
?<= isset($heading) ? $heading->id : '' ?>">
        <div class="mb-3">
            <label for="headingName">Name:</label>
            <input type="text" name="headingName" id="headingName" cl
ass="form-control" value="<?<= isset($heading) ? $heading->name : '' ?>">
            <em id="headingNameErrorMessage"></em>
        </div>
        <div class="mb-3">
            <label for="headingCategory">Category:</label>
            <select name="headingCategory" id="headingCategory" class
="form-select">
                <option value="0">Choose</option>
                <?php
                $categories = getAll('categories');
                foreach ($categories as $category) :
                ?>
                    <option value="<?<= $category->
id ?>" <?php if (isset($category) && isset($heading) && $category->
id == $heading->
category_id) : ?> selected <?php else : ?> <?php endif; ?>><?<= $category->
name ?></option>
                    <?php endforeach ?>
                </select>
                <em id="headingCategoryErrorMessage"></em>
            </div>
            <div class="d-grid">
                <button class="btn btn-
primary" type="button" id="btnHeading"><?<= isset($_GET['id']) ? "Update" : "Save"
?></button>
            </div>
        </form>
    </div>
</div>
</div>
</section>

```

Pages/admin/posts/index.php

```

<?php

if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {

```

```

        header("Location:index.php?page=status_403");
    } else if (!isset($_SESSION['users'])) {
        header("Location:index.php?page=status_403");
    }
    $posts = postPagination();
    ?>
<section>
    <div class="container">

        <div class="row my-3">

            <div class="col-lg-3">
                <div class="d-grid">
                    <a href="index.php?page=action-post" class="btn btn-
primary">Create new post</a>
                </div>
                <!-- -->
                <div class="my-2">
                    <input type="text" name="searchPost" id="searchPost" class="f
orm-control" placeholder="Search posts....">
                </div>
                <div class="mb-3">
                    <h3 class="fs-5">Categories:</h3>
                    <div class="mb-1">
                        <?php
                            $categories = getAll('categories');
                            foreach ($categories as $category) :
                                ?>
                                    <div class="form-check">
                                        <input class="form-check-
input categories" type="checkbox" value="<?= $category->
id ?>" id="gerne_<?= $category->id ?>" name="product_categories">
                                        <label class="form-check-
label" for="<?= $category->id ?>">
                                            <?= $category->name ?>
                                        </label>
                                    </div><?php endforeach; ?>
                                </div>
                            </div>
                        <div class="mb-3">
                            <h3 class="fs-5">Headings:</h3>
                            <?php
                                $headings = getAll('headings');
                                foreach ($headings as $heading) :
                                    ?>

```



```

        ?>
        <tr>
            <th><?= $rb++ ?></th>
            <td><?= $post->name ?></td>
            <td><?= $post->categoryName ?></td>
            <td><?= $post->headingName ?></td>
            <td><?= date("H:i:s d/m/Y", strtotime($post->created_at)) ?></td>
            <td><?= $post->updated_at ? date("H:i:s d/m/Y", strtotime($post->updated_at)) : '/' ?></td>
            <td><a href="index.php?page=action-post&id=<?= $post->id ?>" class="btn btn-sm btn-success">Update</a></td>
            <td><button type="button" class="btn btn-sm btn-danger delete-post" data-id="<?= $post->id ?>">Delete</button></td>
            <td><a href="index.php?page=post_details&id=<?= $post->id ?>" class="btn btn-sm btn-info">Details</a></td>
        </tr>
    <?php endforeach;
    else : ?>
        <tr>
            <th scope="row" colspan="8">No post at the moment</th>
        </tr>
    <?php endif; ?>
</tbody>
</table>
</div>
</div>

</div>
<?php if (count($posts) >= 5) : ?>
    <div class="row">
        <div class="col-4 mx-auto">
            <nav aria-label="...">
                <ul class="pagination" id="postPagination">
                    <?php
                        $postPages = numberOfPostPages();
                        for ($i = 0; $i < $postPages; $i++) :
                            if ($i == 0) :
                                ?>

```

```

                                <li class="page-
item active"><a class="page-link post-pagination" href="#" data-
limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                                <?php
                                endif;
                                if ($i != 0) :
                                ?>
                                <li class="page-item"><a class="page-
link post-pagination" href="#" data-limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                                <?php
                                endif;
                                endfor;
                                ?>

                                </ul>
                                </nav>
                                </div>
                                </div>
                                <?php endif; ?>
                                </div>
                                </div>
                                </div>
                                </section>

```

Pages/admin/posts/action_post.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
$categories = getAll('categories');
$tags = getAll('tags');
?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-6 mx-auto">
                <div id="crudPostErrorMessages"></div>
                <?php
                if (isset($_GET['id'])) {
                    $post = getDataWithFetch('posts', 'id', $_GET['id']);

```

```

        $tag_arr = [];
        $post_tag = getPostTag($post->id);
        foreach ($post_tag as $pt) {
            array_push($tag_arr, $pt->id);
        }
    }
    ?>
    <form action="#" method="#">

        <input type="hidden" name="postId" id="postId" value="<?= isset($post) ? $post->id : '' ?>">

        <div class="mb-3">
            <label for="postName" class="mb-1">Name</label>
            <input type="text" name="postName" id="postName" class="form-control" value="<?= isset($post) ? $post->name : '' ?>">
            <em id="postNameErrorMessage"></em>
        </div>
        <div class="mb-3">
            <label for="postDescription" class="mb-1">Description:</label>
            <textarea name="postDescription" id="postDescription" cols="30" rows="10" class="form-control"><?= isset($post) ? $post->description : '' ?></textarea>
            <em id="postDescErrorMessage"></em>
        </div>
        <div class="mb-3">
            <label for="postImage" class="mb-1">Image:</label>
            <input type="file" name="postImage" id="postImage" class="form-control">
            <em id="postImageErrorMessage"></em>
        </div>
        <div class="mb-3">
            <label for="postCategory" class="mb-1">Category:</label>
            <select name="postCategory" id="postCategory" class="form-control">
                <option value="0">Choose</option>
                <?php
                foreach ($categories as $category) :
                    ?>
                        <option value="<?= $category->id ?>" <?php if (isset($categories) && isset($post) && $category->id == $post->category_id) : ?> selected <?php else : ?> <?php endif; ?>><?= $category->name ?></option>
                <?php

```

```

        endforeach;
        ?>
    </select>
    <em id="postCategoryErrorMessage"></em>
</div>
<div class="mb-3">
    <label for="postHeading" class="mb-1">Heading:</label>
    <select name="postHeading" id="postHeading" class="form-
select">

        <option value="0">Choose</option>
        <?php
            $headings = getAll('headings');
            foreach ($headings as $heading) :
                ?>
                    <option value="<?= $heading-
>id ?>" <?php if (isset($headings) && isset($post) && $heading->id == $post-
>heading_id) : ?> selected <?php else : ?> <?php endif; ?>><?= $heading-
>name ?></option>

                <?php endforeach; ?>
            </select>
            <em id="postHeadingErrorMessage"></em>
        </div>
        <div class="mb-3">
            <label for="postTags" class="mb-3">Tags:</label>
            <?php
                foreach ($tags as $tag) :
                    ?>
                        <div class="col-6">
                            <div class="form-check">
                                <input class="form-check-
input" type="checkbox" value="<?= $tag->id ?>" id="tag_<?= $tag-
>id ?>" name="postTags" <?php if (isset($tags) && isset($tag_arr) && in_array($ta
g->id, $tag_arr)) : ?> checked <?php endif ?>>
                                    <label class="form-check-
label" for="tag_<?= $tag->id ?>">

                                        <?= $tag->name ?>
                                    </label>
                                </div>
                            </div>
                        </div>
                    <?php
                        endforeach;
                    ?>
                    <em id="postTagsErrorMessage"></em>
                </div>
            <?php

```

```

        if (isset($_GET['id'])) :
        ?>
            
            <?php endif ?>
            <div class="d-grid"><button class="btn btn-
primary" type="button" id="submitPost">Submit</button></div>
        </form>
    </div>
</div>
</div>
</section>

```

Pages/admin/posts/post_details.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
    <div class="container">
        <?php
        if (isset($_GET['id'])) {
            $id = $_GET['id'];
            $post = $conn-
>query("SELECT p.*, c.name as categoryName FROM posts p JOIN categories c ON p.ca
tategory_id = c.id WHERE p.id = $id")->fetch();
            $post_tag = getPostTag($post->id);
            // var_dump($post_tag);
            // $postTag_arr = [];
        } ?>

        <div class="row my-3">

            <div class="col-lg-10 mx-auto">
                <div class="row">
                    <div class="col-lg-6">
                        

```



```

        <small class="text-danger text-uppercase"><?= $post-
>categoryName ?></small>
        <small><?= date("H:i:s d-m-Y", strtotime($post-
>created_at)) ?> </small>
        <h1 class="fs-2"><?= $post->name ?></h1>
        <p class="fs-5"><?= $post->description ?></p>
        <h2 class="fs-5">Tags:</h2>
        <div class="row mb-3">
            <?php
                foreach ($post_tag as $tag) :
                    ?>
                        <div class="col-4">
                            <div class="d-
grid"><button type="button" class="btn btn-dark btn-sm py-3"><?= $tag-
>name ?></button></div>
                                </div>
                            <?php endforeach; ?>
                        </div>
                    </div>
                <div class="col-lg-6">
                    <?php
                        $getComments = getComments($post->id);
                        if (count($getComments) > 0) :
                            foreach ($getComments as $comment) :
                                ?>
                                    <div class="card mt-1">
                                        <div class="card-body">
                                            <h5 class="card-title"><?= $comment-
>firstName . ' ' . $comment->lastName ?></h5>
                                            <p class="fs-6 text-
muted"><?= date("H:i:s d-m-Y", strtotime($comment->created_at)) ?></p>
                                            <p class="card-text"><?= $comment-
>text ?></p>
                                        </div>
                                    </div>
                                </div>
                            <?php endforeach;
                        else :
                            ?>
                                <p class="text-center fw-bolder fs-
3">We dont have any comment at this time!</p>
                            <?php endif; ?>
                        </div>
                    </div>
                </div>
            </div>

```

```

        </div>
    </div>
</section>

<!-- Modal -->
<div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-
keyboard="false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="staticBackdropLabel">Modal title</h5>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                ...
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary">Understood</button>
            </div>
        </div>
    </div>
</div>

```

Pages/admin/tags/index.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-lg-3">
                <div class="d-grid">

```

```

        <a href="index.php?page=action-tag" class="btn btn-
primary">Create new tag</a>
    </div>
    <div class="mt-2">
        <input type="text" name="searchTags" id="searchTags" class="f
orm-control" placeholder="Search tags...">
    </div>
    <div class="my-2">
        <select name="sortByDateTags" id="sortByDateTag" class="form-
select">
            <option value="0">Sort by:</option>
            <option value="1">Opadajucem</option>
            <option value="2">Rastucem</option>
        </select>
    </div>
</div>
<div class="col-lg-9">
    <div class="row">
        <div class="table-responsive-sm table-responsive-md">
            <table class="table text-center align-middle">
                <thead class="table-dark">
                    <tr>
                        <th>#</th>
                        <th>Name</th>
                        <th>Created at</th>
                        <th>Update at</th>
                        <th>Update</th>
                        <th>Delete</th>
                    </tr>
                </thead>
                <tbody id="tags">
                    <?php
                    $rb = 1;
                    $tags = tagPagination();
                    if (count($tags) > 0) :
                        foreach ($tags as $tag) :
                            ?>
                                <tr>
                                    <th><?= $rb++ ?></th>
                                    <td><?= $tag->name ?></td>
                                    <td><?= $tag->created_at ?></td>
                                    <td><?= $tag->updated_at ? $tag-
>updated_at : '-' ?></td>
                                    <td><a href="index.php?page=action-
tag&id=<?= $tag->id ?>" class="btn btn-success btn-sm">Update</a></td>

```

```

                <td><button type="button" class="btn
btn-danger btn-sm delete-tag" data-id="<?= $tag->id ?>">Delete</button></td>
            </tr>
        <?php endforeach;
        else : ?>
        <?php endif; ?>
    </tbody>
</table>
</div>
</div>
<?php
if (count($tags) >= 5) :
?>
    <div class="row mt-3">
        <div class="col-4 mx-auto">
            <nav aria-label="...">
                <ul class="pagination" id="tagPagination">
                    <?php
                    $tagPages = getNumOfPagesTags();
                    for ($i = 0; $i < $tagPages; $i++) :
                        if ($i == 0) :
                            <li class="page-
item active"><a class="page-link tag-pagination" href="#" data-
Limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                            <?php
                            endif;
                            if ($i != 0) :
                                <li class="page-item"><a class="page-
link tag-pagination" href="#" data-Limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                                <?php
                                endif;
                            endif;
                        </ul>
                    </nav>
                </div>
            </div>
        <?php endif; ?>
    </div>
</div>
</div>
</section>

```

Pages/admin/tags/action-tag.php

```
<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
?>
<section>
    <div class="container">
        <div class="row my-3">
            <?php
            if (isset($_GET['id'])) {
                $tag = getDataWithFetch('tags', 'id', $_GET['id']);
            }
            ?>
            <div class="col-6 mx-auto">
                <div id="crudShowErrorMessage"></div>
                <form method="POST">
                    <input type="hidden" name="tagId" id="tagId" value="<?= isset
($_GET['id']) ? $tag->id : '' ?> ">
                    <div class="mb-3">
                        <label for="tagName">Name:</label>
                        <input type="text" name="tagName" id="tagName" class="for
m-control" value="<?= isset($tag) ? $tag->name : '' ?>">
                        <em id="tagNameErrorMessage"></em>
                    </div>
                    <div class="d-grid">
                        <button class="btn btn-
primary" id="btnTag" type='button'><?= isset($_GET['id']) ? "Update" : "Save" ?><
/button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</?>
```

Pages/admin/users/index.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
}
if (isset($_GET['id'])) {
    $user = getDataWithFetch('users', 'id', $_GET['id']);
}
?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-6 <?php if (!isset($_GET['id'])) : ?> mx-
auto <?php endif; ?>">

                <em id="crudUserErrorMessage"></em>
                <form action="#" method="#">
                    <input type="hidden" name="userId" id="userId" value="<?= iss
et($user) ? $user->id : '' ?>">
                    <div class="mb-3">
                        <label for="userFirstName">First name:</label>
                        <input type="text" name="userFirstName" id="userFirstName
" class="form-control" value="<?= isset($user) ? $user->first_name : '' ?>">
                        <em id="userFirstNameErrorMessage"></em>
                    </div>
                    <div class="mb-3">
                        <label for="userLastName">Last name:</label>
                        <input type="text" name="userLastName" id="userLastName"
class="form-control" value="<?= isset($user) ? $user->last_name : '' ?>">
                        <em id="userLastNameErrorMessage"></em>
                    </div>
                    <div class="mb-3">
                        <label for="userEmail">Email:</label>
                        <input type="email" name="userEmail" id="userEmail" class
="form-control" value="<?= isset($user) ? $user->email : '' ?>">
                        <em id="userEmailErrorMessage"></em>
                    </div>

                    <?php
                    if (!isset($_GET['id'])) :
                        ?>
                        <div class="mb-3">
                            <label for="userPassword">Password:</label>

```

```

        <input type="password" name="userPassword" id="userPa
ssword" class="form-control">
        <em id="userPasswordErrorMessage"></em>
    </div>
    <?php endif ?>
    <div class="mb-3">
        <label for="userRole">Role:</label>
        <?php
        $roles = getAll('roles');
        foreach ($roles as $role) :
        ?>
            <div class="form-check">
                <input class="form-check-
input" type="radio" name="userRole" id="role_<?= $role->id ?>" value="<?= $role-
>id ?>" <?php if (isset($roles) && isset($user) && ($role->id == $user-
>role_id)) : ?> checked <?php else : ?> <?php endif; ?>>
                <label class="form-check-
label" for="role_<?= $role->id ?>">
                    <?= $role->name ?>
                </label>
            </div>
        <?php endforeach; ?>
        <em id="userRoleErrorMessage"></em>
    </div>
    <div class="d-grid"><button class="btn btn-
primary" type="button" id="btnSubmitUser">Submit</button></div>
</form>
</div>
<?php
if (isset($_GET['id'])) :
?>
    <div class="col-6">
        <form action="#" method="#">
            <input type="hidden" name="userResetId" id="userResetId"
value=<?= $user->id ?>>
            <div class="mb-3">
                <label for="userFirstNameLast">Change user for:</labe
l>
                <input type="text" name="userFirstLastName" id="yserF
irstLastName" class="form-control" value="<?= $user->first_name . ' ' . $user-
>last_name ?>" disabled>
            </div>
            <div class="mb-3">
                <label for="userPassword">Password:</label>

```

```

                <input type="password" name="userChangePassword" id="
userChangePassword" class="form-control">
            </div>
            <div class="d-grid"><button class="btn btn-
primary" id="btnChangePassword">Change password</button></div>
        </form>
    </div>
    <?php endif; ?>
</div>
</div>
</section>

```

Pages/admin/users/action-user.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {
    header("Location:index.php?page=status_403");
} ?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-lg-3">
                <div class="my-2">
                    <div class="d-grid">
                        <a href="index.php?page=action-user" class="btn btn-
primary">Create a new user</a>
                    </div>
                </div>
                <div class="my-2">
                    <input type="text" name="searchUser" id="searchUser" class="f
orm-control" placeholder="Search users....">
                </div>
                <div class="my-2">
                    <select name="sortByDateUsers" id="sortByDateUsers" class="fo
rm-select">
                        <option value="0">Choose</option>
                        <option value="1">Opadajucem</option>
                        <option value="2">Rastucem</option>
                    </select>
                </div>
            </div>
        </div>
    </div>

```



```

<div class="col-lg-9">
    <div class="row">
        <div class="table-responsive-sm table-responsive-md">
            <table class="table text-center align-middle">
                <thead class="table-dark">
                    <tr>
                        <th>#</th>
                        <th>Full name</th>
                        <th>Email</th>
                        <th>Created at</th>
                        <th>Updated at</th>
                        <th>Update</th>
                        <th>Delete</th>
                    </tr>
                </thead>
                <tbody id="users">
                    <?php
                    $rb = 1;
                    $users = getAll('users');
                    if (count($users) > 0) :
                        foreach ($users as $user) :
                            ?>
                                <tr>
                                    <th scope="row"><?= $rb++ ?></th>
                                    <td><?= $user->first_name . ' ' . $user->last_name ?></td>
                                    <td><?= $user->email ?></td>
                                    <td><?= $user->created_at ?></td>
                                    <td><?= $user->updated_at ? $user->updated_at : '-' ?></td>
                                    <td><a href="index.php?page=action-user&id=<?= $user->id ?>" class="btn btn-sm btn-success">Update</a></td>
                                    <td><button type="button" class="btn btn-sm btn-danger delete-user" data-id="<?= $user->id ?>">Delete</button></td>
                                </tr>
                            <?php endforeach;
                        else : ?>
                            <tr>
                                <th scope="row" colspan="7">We dont have any user!</th>
                            </tr>
                        <?php endif; ?>
                    </tbody>
                </table>
            </div>
        </div>
    </div>

```

```

        </div>
        <?php
        if (count($users) >= 5) :
        ?>
            <div class="row">
                <div class="col-4 mx-auto">
                    <nav aria-label="...">
                        <ul class="pagination" id="userPagination">
                            <?php
                            $userPages = getPagesOfUsers();
                            for ($i = 0; $i < $userPages; $i++) :
                                if ($i == 0) :
                                    <li class="page-
item active"><a class="page-link post-pagination" href="#" data-
Limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                                    <?php
                                    endif;
                                    if ($i != 0) :
                                        <li class="page-
link post-pagination" href="#" data-Limit="<?= $i ?>"><?= ($i + 1) ?></a></li>
                                        <?php
                                        endif;
                                    endfor;
                                ?>
                            </ul>
                        </nav>
                    </div>
                </div>
            <?php endif; ?>
        </div>
    </div>
</section>

```

Pages/admin/home.php

```

<?php
if (isset($_SESSION['users']) && $_SESSION['users']->roleName != "Admin") {
    header("Location:index.php?page=status_403");
} else if (!isset($_SESSION['users'])) {

```

```

    header("Location:index.php?page=status_403");
}

?>
<section>
    <div class="container">
        <div class="row my-3">
            <div class="col-xs-6 col-sm-6 col-md-6 col-lg-3 mb-2">
                <div class="card">
                    <div class="card-body">
                        <h1 class="fs-5" id="viewsTotal"></h1>
                        <canvas id="chartTotal"></canvas>
                    </div>
                    <div class="card-footer">
                        <table class="table">
                            <tbody id="Total"></tbody>
                        </table>
                    </div>
                </div>
            </div>

            <div class="col-6 col-lg-3">
                <div class="card">
                    <div class="card-body">
                        <h1 class="fs-5" id="views24H"></h1>
                        <canvas id="chart24h"></canvas>
                    </div>
                    <div class="card-footer">
                        <table class="table">
                            <tbody id="24h"></tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-lg-3 mt-sm-2 mt-md-0 mt-lg-0">
            <div class="card">
                <div class="card-header">
                    <h1 class="fts-bold fs-3">
                        Registred users:
                    </h1>
                </div>
                <div class="card-body">
                    <h2 class="fts-bold fs-1 text-center" id="registeredUsers"></h2>

```

```

        </div>
    </div>

    </div>
    <div class="col-lg-3 mt-sm-2 mt-md-0 mt-lg-0">
        <div class="card">
            <div class="card-header">
                <h1 class="fts-bold fs-3">
                    Today logins:
                </h1>
            </div>
            <div class="card-body">
                <h2 class="fts-bold fs-1 text-center" id="todayLogins"></h2>
            </div>
        </div>
    </div>
</div>
</section>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.5.0/chart.min.js">
</script>

<?php if (isset($_SESSION['users']) && $_SESSION['users']-
>roleName = "Admin") : ?>
    <script src="assets/js/adminChart.js"></script>
<?php endif; ?>

```

Models/function.php

```

<?php
function getHeadingsWithCategories()
{
    global $conn;
    $query = $conn-
>query("SELECT h.*, c.name as categoryName FROM headings h JOIN categories c ON h
.category_id = h.id")->fetchAll();
    return $query;
}
function getComments($post_id)
{
    global $conn;

```

```

    $query = $conn-
>query("SELECT c.*, p.id as postId, u.first_name as firstName, u.last_name as las
tName FROM comments c JOIN users u ON c.id_user=u.id LEFT JOIN posts p ON c.id_po
st=p.id WHERE p.id='$post_id' AND parent_id = 0")->fetchAll();
    return $query;
}
function getAll($table)
{
    global $conn;
    $query = $conn->query("SELECT * FROM $table")->fetchAll();
    return $query;
}

function getOneWithoutFetch($table, $column, $value, $column2 = null, $value2 = n
ull)
{
    global $conn;
    if ($column2 != null) {
        $query = $conn-
>prepare("SELECT * FROM $table WHERE $column = ? AND $column2 = ?");
        $query->execute([$value, $value2]);
    } else {
        $query = $conn->prepare("SELECT * FROM $table WHERE $column = ?");
        $query->execute([$value]);
    }
    return $query;
}

function getDataWithFetch($table, $column, $value, $column2 = null, $value2 = nul
l)
{
    global $conn;
    if ($column2 != null) {
        $query = $conn-
>prepare("SELECT * FROM $table WHERE $column = ? AND $column2 = ?");
        $query->execute([$value, $value2]);
    } else {
        $query = $conn->prepare("SELECT * FROM $table WHERE $column = ?");
        $query->execute([$value]);
    }
    if ($query->rowCount() > 1) {
        $rez = $query->fetchAll();
    } else {
        $rez = $query->fetch();
    }
}

```

```

        return $rez;
    }

function deleteData($table, $column, $id)
{
    global $conn;
    $query = $conn->prepare("DELETE FROM $table WHERE $column=?");
    $query->execute([$id]);
    return $query;
}

function resizeImage($file, $normal_path, $small_path)
{
    $image = $file;
    $new_image_name = time() . '_' . $image['name'];
    $tmp_name = $image['tmp_name'];
    $new_normal_path = $normal_path . $new_image_name;
    $new_small_path = $small_path . $new_image_name;
    move_uploaded_file($tmp_name, $new_normal_path);
    $getImageDimensions = getimagesize($new_normal_path);
    //var_dump($getImageDimensions);
    $width = $getImageDimensions[0];
    $height = $getImageDimensions[1];

    $newWidth = 100;
    $newHeight = $height / ($width / $newWidth);
    $imageExtension = pathinfo($new_normal_path, PATHINFO_EXTENSION);
    if ($imageExtension == 'png') {
        $resource = imagecreatefrompng($new_normal_path);
        $canvars = imagecreatetruecolor($newWidth, $newHeight);
        imagecopyresampled($canvars, $resource, 0, 0, 0, 0, $newWidth, $newHeight,
, $width, $height);
        imagepng($canvars, $new_small_path);
    } else {
        $resource = imagecreatefromjpeg($new_normal_path);
        $canvars = imagecreatetruecolor($newWidth, $newHeight);
        imagecopyresampled($canvars, $resource, 0, 0, 0, 0, $newWidth, $newHeight,
, $width, $height);
        imagejpeg($canvars, $new_small_path);
    }

    return $new_image_name;
}

function getPostTag($post_id)
{

```

```

    global $conn;
    $query = "SELECT * FROM tags WHERE id IN ";
    $query .= "(SELECT tag_id FROM post_tag WHERE post_id =
        ?)";
    $result = $conn->prepare($query);
    $result->execute([$post_id]);
    $data = $result->fetchAll();
    return $data;
}

function getColumnIngridients()
{
    global $conn;
    $query = "SELECT COLUMN_NAME
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE
        TABLE_SCHEMA = Database()
        AND TABLE_NAME = 'categories'"; // "SELECT COUNT(COLUMN_NAME) FROM INFORMATION
    _SCHEMA.COLUMNS WHERE
    //TABLE_NAME = $table ";
    $result = $conn->query($query);
    return $result->fetchAll();
}

define("OFFSET", 5);
function postPagination($limit = 0)
{
    global $conn;
    $query = $conn-
>prepare("SELECT p.*, c.name as categoryName, h.name as headingName FROM posts p
JOIN categories c ON p.category_id = c.id JOIN headings h ON p.heading_id = h.id
LIMIT :limit, :offset");
    $limit = ((int)$limit) * OFFSET;
    $query->bindParam(":limit", $limit, PDO::PARAM_INT);
    $offset = OFFSET;
    $query->bindParam(":offset", $offset, PDO::PARAM_INT);
    $query->execute();

    $rez = $query->fetchAll();
    return $rez;
}

function getNumOfPosts()
{
    global $conn;

```

```

    $rez = $conn->query("SELECT COUNT(*) AS numOfPosts FROM posts")->fetch();
    return $rez;
}

function numberOfPostPages()
{
    $postsNum = getNumOfPosts();
    $pagesCount = ceil($postsNum->numOfPosts / OFFSET);
    return $pagesCount;
}

function tagPagination($limit = 0)
{
    global $conn;
    $query = $conn->prepare("SELECT * FROM tags LIMIT :limit, :offset");
    $limit = ((int)$limit) * OFFSET;
    $offset = OFFSET;
    $query->bindParam(":limit", $limit, PDO::PARAM_INT);
    $query->bindParam(":offset", $offset, PDO::PARAM_INT);
    $query->execute();
    $rez = $query->fetchAll();
    return $rez;
}

function getNumOfTag()
{
    global $conn;
    $rez = $conn->query("SELECT COUNT(*) AS numOfTags FROM tags")->fetch();
    return $rez;
}

function getNumOfPagesTags()
{
    $tagNum = getNumOfTag();
    $pagesCount = ceil($tagNum->numOfTags / OFFSET);
    return $pagesCount;
}

function userPagination($limit = 0)
{
    global $conn;
    $query = $conn->prepare("SELECT * FROM users LIMIT :limit, :offset");
    $limit = ((int)$limit) * OFFSET;
    $offset = OFFSET;

```



```

$query->bindParam(":limit", $limit, PDO::PARAM_INT);
$query->bindParam(":offset", $offset, PDO::PARAM_INT);
$query->execute();
$rez = $query->fetchAll();
return $rez;
}

function getNumOfUsers()
{
    global $conn;
    $rez = $conn->query("SELECT COUNT(*) AS numOfUsers FROM users")->fetch();
    return $rez;
}

function getPagesOfUsers()
{
    $tagUser = getNumOfUsers();
    $pagesCount = ceil($tagUser->numOfUsers / OFFSET);
    return $pagesCount;
}

function inTheLast24H($date)
{
    $todayMax = time() + 86400;
    $todayMin = time() - 86400;
    $check = strtotime($date);
    if ($todayMin <= $check && $check <= $todayMax)
        return true;
    else
        return false;
}

function loggedInToday()
{
    $br = 0;
    $fileU = fopen("../data/userLog.txt", "r");
    $rowsU = file("../data/userLog.txt");
    fclose($fileU);
    foreach ($rowsU as $row) {
        $data = explode("\t", $row);
        $action = $data[1];
        $date = date("d m y", strtotime($data[2]));
        $today = date("d m y");
    }
}

```

```

        //echo json_encode($date);
        // echo json_encode($today);
        if ($action == "Login" && $today == $date)
            $br++;
    }
    // echo json_encode($br);
    return $br;
}

function registeredusers()
{
    global $conn;
    $query = "SELECT COUNT(*) as users FROM users";
    $result = $conn->query($query);
    $result = $result->fetch(PDO::FETCH_ASSOC);
    return $result;
}

function insertUserActivity($email, $action)
{
    $file = fopen("../data/userLog.txt", "a");
    $date = date('d-m-Y H:i:s');
    fwrite($file, "{$email}\t{$action}\t{$date}\n");
    fclose($file);
}

function insertNewCategory($table, $name)
{
    global $conn;
    $queryInsert = $conn->prepare("INSERT INTO $table (name) VALUES(?)");
    $queryInsert->execute([$name]);
    return $queryInsert;
}

function updateCategory($table, $name, $updated_at, $id)
{
    global $conn;
    $queryUpdate = $conn->
    >prepare("UPDATE $table SET name=?, updated_at=? WHERE id=?");
    $queryUpdate->execute([$name, $updated_at, $id]);
    return $queryUpdate;
}

function insertNewHeading($table, $name, $category_id)
{
    global $conn;

```

```

    $queryInsert = $conn-
>prepare("INSERT INTO $table (name, category_id) VALUES(?,?)");
    $queryInsert->execute([$name, $category_id]);
    return $queryInsert;
}

function updateHeading($table, $name, $category_id, $date, $id)
{
    global $conn;
    $queryUpdate = $conn-
>prepare("UPDATE $table SET name=?, updated_at=?, category_id=? WHERE id=?");
    $queryUpdate->execute([$name, $date, $category_id, $id]);
    return $queryUpdate;
}

function insertNewTag($table, $name)
{
    global $conn;
    $queryInsert = $conn->prepare("INSERT INTO $table (name) VALUES(?)");
    $queryInsert->execute([$name]);
    return $queryInsert;
}

function updateTag($table, $name, $date, $id)
{
    global $conn;
    $queryUpdate = $conn-
>prepare("UPDATE $table SET name=?, updated_at=? WHERE id=?");
    $queryUpdate->execute([$name, $date, $id]);
    return $queryUpdate;
}

function insertNewUser($table, $first_name, $last_name, $email, $password, $role_id)
{
    global $conn;

    $queryInsert = $conn-
>prepare("INSERT INTO $table (first_name, last_name, email, password, role_id) VA
LUES(?,?,?,?,?)");
    $queryInsert-
>execute([$first_name, $last_name, $email, md5($password), $role_id]);
    return $queryInsert;
}

```

```

function updateUser($table, $first_name, $last_name, $email, $role_id, $date, $id)
{
    global $conn;
    $queryUpdate = $conn-
>prepare("UPDATE $table SET first_name=?, last_name=?, email=?, updated_at=?,role
_id=? WHERE id=?");
    $queryUpdate-
>execute([$first_name, $last_name, $email, $date, $role_id, $id]);
    return $queryUpdate;
}

function createNewPost($table1, $table2, $postName, $postDescription, $postImage,
$category_id, $heading_ig, $tags_arr)
{
    global $conn;

    $queryInsert = $conn-
>prepare("INSERT INTO $table1 (name, description, image_path, category_id, headin
g_id) VALUES(?,?,?,?,?)");
    $queryInsert-
>execute([$postName, $postDescription, $postImage, $category_id, $heading_ig]);
    $id = $conn->lastInsertId();
    if (count($tags_arr) > 0) {
        $queryParams = [];
        $values = [];
        foreach ($tags_arr as $tag) {
            $queryParams[] = "(?,?)";
            $values[] = (int)$id;
            $values[] = (int)$tag;
        }

        $post_tag_insert = $conn-
>prepare("INSERT INTO $table2 VALUES" . implode(", ", $queryParams));
        $post_tag_insert->execute($values);
    }
}

function updatePosts($table1, $table2, $postName, $postDesc, $image = "", $catego
ry_id, $heading_ig, $tags_arr)
{
    global $conn;
    if ($image == "") {

```

```

        $queryUpdate = $conn-
>prepare("UPDATE $table1 SET name=?, description=?, image_path=?, updated_at=?, c
ategory_id=?, heading_id=?");
        $queryUpdate->execute();
    }
}

function insertNewComment($table, $id_post, $id_user, $text, $parent_comment, $da
te)
{
    global $conn;
    //echo json_encode($text);
    $query = $conn-
>prepare("INSERT INTO $table (id_post, id_user, text, parent_id,created_at) VALUE
S(?,?,?,?,?)");
    $query->execute([$id_post, $id_user, $text, $parent_comment, $date]);
    return $query;
}

function readComments($table1, $table2, $post_id, $comment_id)
{
    global $conn;
    $query = $conn-
>query("SELECT c.*, u.first_name as firstName, u.last_name as postName FROM '$tab
le1' c JOIN ` $table2` u ON c.id_user = u.id WHERE parent_comment = '$comment_id'
AND id_post='$post_id' ")->fetchAll();
    return $query;
}

function checkIsReacted($table, $post_id, $comment_id, $user_id)
{
    // var_dump($post_id, $user_id, $comment_id);
    global $conn;
    $query = $conn-
>query("SELECT * FROM $table WHERE user_id = $user_id AND post_id = $post_id AND
comment_id = $comment_id");
    return $query;
}

function insertReaction($table, $action, $post_id, $comment_id, $user_id, $vote)
{
    global $conn;

```

```

    $query = '';
    if ($action == "like") {
        $query = $conn-
>prepare("INSERT INTO $table (comment_id, post_id, user_id, likes) VALUES(?,?,?,?)");
        $query->execute([$comment_id, $post_id, $user_id, $vote]);
    } else {
        $query = $conn-
>prepare("INSERT INTO $table (comment_id, post_id, user_id, disslikes) VALUES(?,?,?,?)");
        $query->execute([$comment_id, $post_id, $user_id, $vote]);
    }
    return $query;
}

function deleteVote($table, $action, $post_id, $comment_id, $user_id)
{
    global $conn;
    $query = '';
    if ($action == 'like') {
        $query = $conn-
>prepare("DELETE FROM $table WHERE post_id=? AND user_id=? AND comment_id=?");
        $query->execute([$post_id, $user_id, $comment_id]);
    } else {
        $query = $conn-
>prepare("DELETE FROM $table WHERE post_id=? AND user_id=? AND comment_id=?");
        $query->execute([$post_id, $user_id, $comment_id]);
    }
}

function voteChange($table, $from, $to, $like, $disslike, $comment_id, $post_id, $user_id)
{
    global $conn;
    // $queryUpdate = $conn-
>prepare("UPDATE reactions SET likes=?, disslikes=? WHERE comment_id=? AND user_id=?");
    // $queryUpdate->execute([REACT_ON, REACT_OFF, $comment_id, $id]);
    if ($from == 'like' && $to == "disslike") {
        // echo json_encode("od Like to disslake");
        $queryUpdate = $conn-
>prepare("UPDATE $table SET likes=?, disslikes=? WHERE post_id=? AND user_id=? AND comment_id=? ");
        $queryUpdate-
>execute([$like, $disslike, $post_id, $user_id, $comment_id]);

```

```

    } else {
        $queryUpdate = $conn-
>prepare("UPDATE $table SET likes=?, disslikes=? WHERE post_id=? AND user_id=? A
ND comment_id=? ");
        $queryUpdate-
>execute([$like, $disslike, $post_id, $user_id, $comment_id]);
    }

    return $queryUpdate;
}

function getReactions($post_id, $user_id)
{
    global $conn;
    $query = $conn-
>query("SELECT * FROM reactions WHERE post_id= $post_id AND user_id = $user_id")
->fetchAll();
    return $query;
}

function postFilter($text, $categories, $headings, $pagination, $date)
{
    $query = '';

    $queryA = "SELECT COUNT(p.id) as NumOfPost FROM posts p JOIN categories c ON
p.category_id = c.id JOIN headings h ON p.heading_id = h.id ";
    $baseQuery = "SELECT p.*, h.name as headingName, c.name as categoryName FROM
posts p JOIN categories c ON p.category_id = c.id JOIN headings h ON p.heading_id
= h.id ";
    if ($text) {
        $query .= " WHERE";
        $query .= " p.name LIKE '%$text%'";
    }
    if ($categories) {
        if ($text == "") {
            $query .= " WHERE p.category_id IN (" . implode(',', $categories) .
")";
        }
        $query .= " AND p.category_id IN (" . implode(',', $categories) . ")";
    }
    if ($headings) {
        if ($text && $categories) {
            $query .= " AND p.heading_id IN (" . implode(',', $headings) . ")";
        } else if ($text) {

```

```

        $query .= " AND p.heading_id IN (" . implode(',', $headings) . ")";
    } else if ($categories) {
        $query .= " AND p.heading_id IN (" . implode(',', $headings) . ")";
    } else {
        $query .= " WHERE p.heading_id IN (" . implode(',', $headings) . ")"
;
    }
}

$queryA = $queryA . $query;

if ($date != 0) {
    if ($date == 1) {
        $query .= " ORDER BY created_at DESC ";
    } else {
        $query .= " ORDER BY created_at ASC";
    }
}

if ($pagination && $pagination > 0) {
    $limit = ((int)$_GET['limit']) * OFFSET;
    $offset = OFFSET;
    $query .= " LIMIT $limit, $offset";
} else if ($pagination == 0 || !$pagination) {
    $limit = 0;
    $offset = OFFSET;
    $query .= " LIMIT $limit, $offset";
}
$baseQuery = $baseQuery . $query;
// echo json_encode($baseQuery);
return [
    $baseQuery, $queryA
];
}

function userWithoutRole()
{
    global $conn;
    $getAll = $conn->
    >query("SELECT u.*, r.name as roleName FROM users u JOIN roles r ON u.role_id = r
.id WHERE r.name = 'Admin'");
    return $getAll;
}

```



```

function getPosts()
{
    global $conn;
    $rez = $conn->
>query("SELECT p.*, c.name as categoryName, h.name as headingName FROM posts p JO
IN categories c ON p.category_id = c.id JOIN headings h ON p.heading_id = h.id ")
->fetchAll();
    return $rez;
}

```

Models/users/delete.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    try {
        require_once '../config/connection.php';
        require_once '../functions.php';
        $delete = deleteData('users', 'id', $_POST['id']);
        http_response_code(204);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    http_response_code(404);
}

```

Models/users/filter.php

```

<?php
header("Content-type:application/json");
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    $text = isset($_GET['text']) ? $_GET['text'] : '';
    $date = isset($_GET['date']) ? $_GET['date'] : '';
    $pagination = isset($_GET['limit']) ? $_GET['limit'] : '';
    require_once '../config/connection.php';

```

```
} else {  
    http_response_code(404);  
}
```

Models/users/getAll.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER['REQUEST_METHOD'] == "GET") {  
    require_once '../../config/connection.php';  
    $getAll = userWithoutRole();  
    echo json_encode($getAll);  
} else {  
    http_response_code(404);  
}
```

Models/users/insert.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER['REQUEST_METHOD'] == "POST") {  
    $first_name = $_POST['first_name'];  
    $last_name = $_POST['last_name'];  
  
    $email = $_POST['email'];  
    $password = $_POST['password'];  
    $role = $_POST['role'];  
  
    $reFirstLastName = '/^[A-Z][a-z]{3,15}$/';  
    $rePassword = '/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/';  
  
    $errors = [];  
    if (!preg_match($reFirstLastName, $first_name)) {  
        array_push($errors, $first_name);  
    }  
    if (!preg_match($reFirstLastName, $last_name)) {  
        array_push($errors, "Last name of user isn't ok");  
    }  
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
```

```

        array_push($errors, "Email isn't ok");
    }

    if (!preg_match($rePassword, $password)) {
        array_push($errors, "Your password isn't ok");
    }
    if ($role == 0) {
        array_push($errors, "Must choose one role");
    }
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(422);
        }
    } else {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $checkEmail = getOneWithoutFetch('users', 'email', $email);
        if ($checkEmail->rowCount() > 0) {
            echo json_encode("This email is already taken!");
            http_response_code(409);
        } else {
            try {
                insertNewUser('users', $first_name, $last_name, $email, $password
, $role);

                echo json_encode("User is successfull created!");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
} else {
    http_response_code(404);
}
}

```

Models/users/update.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {

```

```

$first_name = $_POST['first_name'];
$last_name = $_POST['last_name'];

$email = $_POST['email'];
$password = $_POST['password'];
$role = $_POST['role'];

$reFirstLastName = '/^[A-Z][a-z]{3,15}$/';
$rePassword = '/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/';

$errors = [];
if (!preg_match($reFirstLastName, $first_name)) {
    array_push($errors, $first_name);
}
if (!preg_match($reFirstLastName, $last_name)) {
    array_push($errors, "Last name of user isn't ok");
}
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    array_push($errors, "Email isn't ok");
}

if (!preg_match($rePassword, $password)) {
    array_push($errors, "Your password isn't ok");
}
if ($role == 0) {
    array_push($errors, "Must choose one role");
}
if (count($errors) > 0) {
    foreach ($errors as $error) {
        echo json_encode("Error: " . $error);
        http_response_code(422);
    }
} else {
    require_once '../config/connection.php';
    require_once '../functions.php';
    $checkEmail = getOneWithoutFetch('users', 'email', $email);
    if ($checkEmail->rowCount() > 0) {
        echo json_encode("This email is already taken!");
        http_response_code(409);
    } else {
        try {
            insertNewUser('users', $first_name, $last_name, $email, $password
, $role);

            echo json_encode("User is successfull created!");
            http_response_code(201);

```

```

        } catch (PDOException $th) {
            echo json_encode($th->getMessage());
            http_response_code(500);
        }
    }
}
} else {
    http_response_code(404);
}

```

Models/tags/delete.php

```

<?php
header("Content-type:application/json");
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    try {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $delete = deleteData('tags', 'id', $_POST['id']);
        http_response_code(204);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    http_response_code(404);
}

```

Models/tags/filter.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "GET") {
    $text = trim(isset($_GET['text'])) ? $_GET['text'] : '';
    $date = isset($_GET['sortByDate']) ? $_GET['sortByDate'] : '';
    $pagination = isset($_GET['limit']) ? $_GET['limit'] : '';

    require_once '../../config/connection.php';

```

```

$query = '';
define("OFFSET", 5);
$queryA = "SELECT COUNT(*) AS numOfTags FROM tags";
$baseQuery = "SELECT * FROM tags";
if ($text) {
    $query .= " WHERE name LIKE '%$text%'";
}
if ($date != 0) {
    if ($date == 1) {
        $query .= " ORDER BY created_at DESC ";
    } else {
        $query .= " ORDER BY created_at ASC";
    }
}
$queryA = $queryA . $query;

if ($pagination && $pagination > 0) {
    $limit = ((int)$_GET['limit']) * OFFSET;
    $offset = OFFSET;
    $query .= " LIMIT $limit, $offset";
} else if ($pagination == 0 || !$pagination) {
    $limit = 0;
    $offset = OFFSET;
    $query .= " LIMIT $limit, $offset";
}
$baseQuery = $baseQuery . $query;

//echo json_encode($queryA);
$rez = $conn->query($baseQuery)->fetchAll();
$res = $conn->query($queryA)->fetch(PDO::FETCH_ASSOC);
//echo json_encode($res);
// echo json_encode($rezPages['numOfTags']);
// echo json_encode(gettype($rezPages['numOfTags']));
$numOfPages = ceil($res['numOfTags'] / OFFSET);
//echo json_encode($numOfPages);

// $numOfPages = ceil((int)$rezPages['numOfTags'] / OFFSET);
// // echo json_encode($rezPages);
echo json_encode([
    'tags' => $rez,
    'pages' => $numOfPages
]);

```

```

    //echo json_encode($rez);
} else {
    http_response_code(404);
}

```

Models/tags/getAll.php

```

<?php
header("Content-type:application/json");
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    require_once '../config/connection.php';
    require_once '../functions.php';
    $getAll = getAll('tags');
    echo json_encode($getAll);
} else {
    http_response_code(500);
}

```

Models/tags/insert.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $reName = '/^[A-Z][a-z]{1,15}(\s[A-Z][a-z]{3,15})?$/';
    $errors = [];
    if (!preg_match($reName, $_POST['name'])) {
        array_push($errors, "Name of the tag isn't ok!");
    }
    if (count($errors) == 0) {
        require_once '../config/connection.php';
        require_once '../functions.php';

        $checkName = getOneWithoutFetch('tags', 'name', $_POST['name']);
        if ($checkName->rowCount() > 0) {
            echo json_encode("Name of tag is already taken!");
            http_response_code(409);
        } else {
            try {
                insertNewTag('tags', $_POST['name']);
            } catch (Exception $e) {
                http_response_code(500);
            }
        }
    }
}

```

```

        echo json_encode("Tag is successfull created!");
        http_response_code(201);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
} else {
    foreach ($errors as $error) {
        echo json_encode("Error: " . $error);
        http_response_code(500);
    }
}
} else {
    http_response_code(404);
}
}

```

Models/tags/update.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $reName = '/^[A-Z][a-z]{1,15}(\s[A-Z][a-z]{3,15})?$/';
    $errors = [];
    if (!preg_match($reName, $_POST['name'])) {
        array_push($errors, "Name of the tag isn't ok!");
    }
    if (count($errors) == 0) {
        require_once '../config/connection.php';
        require_once '../functions.php';
        $date = date("Y-m-d H:i:s");
        // echo json_encode($date);
        $checkName = getOneWithoutFetch('tags', 'name', $_POST['name']);
        if ($checkName->rowCount() > 0) {

            echo json_encode($date);
            $getTag = getDataWithFetch('tags', 'name', $_POST['name']);
            if ($getTag->id == $_POST['id'] && $getTag->name == $_POST['tag']) {
                try {
                    updateTag('tags', $_POST['name'], $date, $_POST['id']);
                    http_response_code(204);
                } catch (PDOException $th) {

```



```

        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    echo json_encode("Tag name is already taken! Please choose another one!");
    http_response_code(409);
}
} else {
    try {
        updateTag('tags', $_POST['name'], $date, $_POST['id']);
        http_response_code(204);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
} else {
    foreach ($errors as $error) {
        echo json_encode("Error: " . $error);
        http_response_code(500);
    }
}
} else {
    http_response_code(404);
}
}

```

Models/posts/delete.php

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    require_once '../config/connection.php';
    require_once '../functions.php';
    try {
        deleteData('posts', 'id', $id);
    } catch (PDOException $th) {
        echo json_encode("This data can't delete");
    }
} else {

```

```
    http_response_code(404);  
}
```

Models/posts/filter.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER['REQUEST_METHOD'] == "GET") {  
    $text = trim(isset($_GET['text'])) ? $_GET['text'] : '';  
    $categories = isset($_GET['categories']) ? $_GET['categories'] : '';  
    $headings = isset($_GET['headings']) ? $_GET['headings'] : '';  
    $date = isset($_GET['sortByDate']) ? $_GET['sortByDate'] : '';  
    $pagination = isset($_GET['limit']) ? $_GET['limit'] : '';  
  
    require_once '../config/connection.php';  
    require_once '../functions.php';  
  
    $f = postFilter($text, $categories, $headings, $pagination, $date);  
    $rez = $conn->query($f[0])->fetchAll();  
  
    $res = $conn->query($f[1])->fetch(PDO::FETCH_ASSOC);  
    $numOfPages = ceil($res['NumOfPost'] / 5);  
    // ech ojson  
    echo json_encode([  
        'posts' => $rez,  
        'pages' => $numOfPages  
    ]);  
} else {  
    http_response_code(404);  
}
```

Models/posts/getAll.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER["REQUEST_METHOD"] == "GET") {  
    require_once '../config/connection.php';  
    $getAllPosts = $getAllPosts();  
}
```

```
    echo json_encode($getAllPosts);  
} else {  
    http_response_code(404);  
}
```

Models/posts/getHeadingByCategory.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER['REQUEST_METHOD'] == "GET") {  
    require_once '../config/connection.php';  
    require_once '../functions.php';  
  
    $getHeadings = getDataWithFetch('headings', 'category_id', $_GET['id']);  
    echo json_encode($getHeadings);  
} else {  
    http_response_code(404);  
}
```

Models/posts/insert.php

```
<?php  
header("Content-type:application/json");  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $name = $_POST['name'];  
    $desc = $_POST['postDesc'];  
    $image = $_FILES['image'];  
    $category = $_POST['category'];  
    $heading = $_POST['heading'];  
    $tags = $_POST['tags'];  
    $tags_arr = explode(",", $tags);  
  
    $errors = [];  
    if ($image['size'] > 3000000) {  
        array_push($errors, "Image size must be less than 3mb");  
    }  
    if ($category == 0) {  
        array_push($errors, "You must choose category for the post!");  
    }  
}
```

```

    }
    if ($heading == 0) {
        array_push($errors, "You must choose heading for the post");
    }
    if (count($tags_arr) == 0) {
        array_push($errors, "You must choose at least one tag!");
    }
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(422);
        }
    } else {
        require_once '../config/connection.php';
        require_once '../functions.php';
        $checkName = getOneWithoutFetch('posts', 'name', $_POST['name']);
        if ($checkName->rowCount() > 0) {
            echo json_encode('Name of the post is already taken');
            http_response_code(409);
        } else {
            try {
                $image = resizeImage($_FILES['image'], "../assets/images/posts/normal/", "../assets/images/posts/thumbnail/");
                $conn->beginTransaction();
                createNewPost('posts', 'post_tag', $name, $desc, $image, $category, $heading, $tags_arr);
                $conn->commit();
                http_response_code(201);
                echo json_encode("Post is successfully created");
            } catch (PDOException $th) {
                $conn->rollback();
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
}
}

```

Models/posts/update.php

```

<?php
header("Content-type:application/json");

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $desc = $_POST['postDesc'];

    $category = $_POST['category'];
    $heading = $_POST['heading'];
    $tags = $_POST['tags'];
    $tags_arr = explode(",", $tags);

    $errors = [];
    if (isset($_FILES['image'])) {
        $image = $_FILES['image'];
        if ($image['size'] > 3000000) {
            array_push($errors, "Image size must be less than 3mb");
        }
    }

    if ($category == 0) {
        array_push($errors, "You must choose category for the post!");
    }
    if ($heading == 0) {
        array_push($errors, "You must choose heading for the post");
    }
    if (count($tags_arr) == 0) {
        array_push($errors, "You must choose at least one tag!");
    }
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(422);
        }
    } else {
        $date = date("Y-m-d H:i:s");
        require_once '../config/connection.php';
        require_once '../functions.php';
        $checkName = getOneWithoutFetch('posts', 'name', $_POST['name']);
        if ($checkName->rowCount() > 0) {
            $post = getDataWithFetch('posts', 'name', $name);
            if ($post->id == $id && $post->name == $name) {
                if (isset($_FILES['image'])) {
                    try {
                        $image = resizeImage($_FILES['image'], '../assets/images/posts/thumbnail/');
                    } catch (Exception $e) {
                        array_push($errors, $e->getMessage());
                    }
                }
            }
        }
    }
}

```

```

        $conn->beginTransaction();
        $updateQuery = $conn-
>prepare("UPDATE posts SET name=?, description=?,image_path=?, updated_at=?, cate
gory_id=?, heading_id=? WHERE id=?");
        $updateQuery-
>execute([$name, $desc, $image, $date, $category, $heading, $id]);
        // delete from post tag
        $deletePostTag = deleteData('post_tag', 'post_id', $id);
        if (count($tags_arr) > 0) {
            $queryParams = [];
            $values = [];
            foreach ($tags_arr as $tag) {
                $queryParams[] = "(?,?)";
                $values[] = (int)$id;
                $values[] = (int)$tag;
            }
            $update_post_tag = $conn-
>prepare("INSERT INTO post_tag (post_id, tag_id) VALUES" . implode(',', $queryPar
ams));
            $update_post_tag->execute($values);
        }
        $conn->commit();
        http_response_code(204);
    } catch (PDOException $th) {
        $conn->rollBack();
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    try {
        $conn->beginTransaction();
        $updateQuery = $conn-
>prepare("UPDATE posts SET name=?, description=?, updated_at=?, category_id=?, h
eading_id=? WHERE id=?");
        $updateQuery-
>execute([$name, $desc, $date, $category, $heading, $id]);
        // delete from post tag
        $deletePostTag = deleteData('post_tag', 'post_id', $id);
        if (count($tags_arr) > 0) {
            $queryParams = [];
            $values = [];
            foreach ($tags_arr as $tag) {
                $queryParams[] = "(?,?)";
                $values[] = (int)$id;
                $values[] = (int)$tag;
            }
        }
    }
}

```

```

        }
        $update_post_tag = $conn-
>prepare("INSERT INTO post_tag (post_id, tag_id) VALUES" . implode(',', $queryPar
ams));

        $update_post_tag->execute($values);
    }
    $conn->commit();
    http_response_code(204);
} catch (PDOException $th) {
    $conn->rollBack();
    echo json_encode($th->getMessage());
    http_response_code(500);
}
}
} else if ($post->id != $id && $post->name == $name) {
    echo json_encode("This name is already taken!");
    http_response_code(409);
}
} else {
    if (isset($_FILES['image'])) {
        $image = resizeImage($_FILES['image'], '../../assets/images/posts
/normal/', '../../assets/images/posts/thumbnail/');
        try {
            $conn->beginTransaction();
            $updateQuery = $conn-
>prepare("UPDATE posts SET name=?, description=?,image_path=?, updated_at=?, cate
gory_id=?, heading_id=? WHERE id=?");
            $updateQuery-
>execute([$name, $desc, $image, $date, $category, $heading, $id]);
            // delete from post tag
            $deletePostTag = deleteData('post_tag', 'post_id', $id);
            if (count($tags_arr) > 0) {
                $queryParams = [];
                $values = [];
                foreach ($tags_arr as $tag) {
                    $queryParams[] = "(?,?)";
                    $values[] = (int)$id;
                    $values[] = (int)$tag;
                }
                $update_post_tag = $conn-
>prepare("INSERT INTO post_tag (post_id, tag_id) VALUES" . implode(',', $queryPar
ams));

                $update_post_tag->execute($values);
            }
            $conn->commit();

```

```

        http_response_code(204);
    } catch (PDOException $th) {
        $conn->rollBack();
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    try {
        $conn->beginTransaction();
        $updateQuery = $conn-
>prepare("UPDATE posts SET name=?, description=?, updated_at=?, category_id=?, heading_id=? WHERE id=?");
        $updateQuery-
>execute([$name, $desc, $date, $category, $heading, $id]);
        // delete from post tag
        $deletePostTag = deleteData('post_tag', 'post_id', $id);
        if (count($tags_arr) > 0) {
            $queryParams = [];
            $values = [];
            foreach ($tags_arr as $tag) {
                $queryParams[] = "(?,?)";
                $values[] = (int)$id;
                $values[] = (int)$tag;
            }
            $update_post_tag = $conn-
>prepare("INSERT INTO post_tag (post_id, tag_id) VALUES" . implode(',', $queryParams));
            $update_post_tag->execute($values);
        }
        $conn->commit();
        http_response_code(204);
    } catch (PDOException $th) {
        $conn->rollBack();
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
}
}
}
}

```

Models/headings/delete.php


```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    try {
        require_once '../config/connection.php';
        require_once '../functions.php';
        $delete = deleteData('headings', 'id', $_POST['id']);
        http_response_code(204);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    http_response_code(404);
}

```

Models/headings/getAll.php

```

<?php
header("Content-type:application/json");
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    require_once '../config/connection.php';
    require_once '../functions.php';
    $headings = getHeadingsWithCategories();
    echo json_encode($headings);
} else {
    http_response_code(404);
}

```

Models/headings/insert.php

```

<?php
header("content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $reName = "/^[A-Z][a-z]{2,15}(\s)?([A-Z][a-z]{2,15})*$/";
    $errors = [];
    if (!preg_match($reName, $_POST['name'])) {
        array_push($errors, "Name of the headings isn't ok!");
    }
}

```

```

if ($_POST['categoryId'] == 0) {
    array_push($errors, "You must choose heading");
}
if (count($errors) > 0) {
    foreach ($errors as $error) {
        echo json_encode("Error: " . $error);
        http_response_code(500);
    }
} else {
    require_once '../config/connection.php';
    require_once '../functions.php';
    $checkName = getOneWithoutFetch('headings', 'name', $_POST['name']);
    if ($checkName->rowCount() > 0) {
        echo json_encode("That name is already taken!");
        http_response_code(409);
    } else {
        try {
            insertNewHeading('headings', $_POST['name'], $_POST['categoryId']
);
            echo json_encode("Heading is successfull created");
            http_response_code(201);
        } catch (PDOException $th) {
            echo json_encode($th->getMessage());
            http_response_code(500);
        }
    }
}
} else {
    http_response_code(404);
}
}

```

Models/headings/update.php

```

<?php
header("content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $reName = "/^[A-Z][a-z]{2,15}(\s)?([A-Z][a-z]{2,15})*$/";
    $errors = [];
    if (!preg_match($reName, $_POST['name'])) {
        array_push($errors, "Name of the headings isn't ok!");
    }
    if ($_POST['categoryId'] == 0) {

```

```

        array_push($errors, "You must choose heading");
    }
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(500);
        }
    } else {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $date = date("Y-m-d H:i:s");
        echo json_encode($date);
        $check = getOneWithoutFetch('headings', 'name', $_POST['name'], $_POST['id']);
        if ($check->rowCount() > 0) {
            echo json_encode("Koristimp postojeci naziv");
        } else {
            try {
                updateHeading('headings', $_POST['name'], $_POST['categoryId'], $date, $_POST['id']);
                http_response_code(204);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
} else {
    http_response_code(404);
}

```

Models/comments/getComments.php

```

<?php
session_start();
header("Content-type:application/json");
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    require_once '../../config/connection.php';
    require_once '../functions.php';
    $post_id = $_GET['id'];

```

```

    $comments = getComments($post_id);
    echo json_encode($comments);
} else {
    http_response_code(404);
}

```

Models/comments/insert.php

```

<?php
session_start();
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $comment_id = isset($_POST['comment_id']) ? $_POST['comment_id'] : 0;
    $post_id = $_POST['post_id'];
    $text = $_POST['text'];

    $user_id = isset($_SESSION['users']) ? $_SESSION['users']->id : '';
    $date = date("Y-m-d H:i:s");

    try {
        require_once '../config/connection.php';
        require_once '../functions.php';
        insertNewComment('comments', $post_id, $user_id, $text, $comment_id, $date);
        echo json_encode("Comment is successfully added");
        http_response_code(201);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else {
    http_response_code(404);
}

```

Models/comments/vote.php

```

<?php
session_start();
header("Content-type:application/json");

```

```

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    require_once '../../config/connection.php';
    require_once '../functions.php';
    $action = $_POST['action'];
    $post_id = $_POST['post'];
    // echo json_encode($post_id);
    $comment_id = $_POST['comment'];

    $id_user = isset($_SESSION['users']) ? $_SESSION['users']->id : '';
    // echo json_encode($post_id);
    // echo json_encode($comment_id);
    // echo json_encode($action);

    $q = checkIsReacted('reactions', $post_id, $comment_id, $id_user);
    //var_dump($q);
    define("REACT_ON", 1);
    define("REACT_OFF", 0);
    if ($q->rowCount() == 0) {

        if ($action == "like") {
            try {
                insertReaction('reactions', 'like', $post_id, $comment_id, $id_us
er, REACT_ON);
                echo json_encode("Commet is added");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        } else {
            try {
                insertReaction('reactions', 'dislike', $post_id, $comment_id, $i
d_user, REACT_ON);
                echo json_encode("Dislike is added");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    } else {
        $res = $q->fetch();
        // echo json_encode($res);
        if ($action == "like") {
            // echo json_encode('like');

```

```

        if ($res->likes > 0) {
            try {
                deleteVote('reactions', 'like', $post_id, $comment_id, $id_us
er);

                http_response_code(204);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        } else if ($res->likes == 0 && $res->disslikes > 0) {
            try {
                voteChange('reactions', 'dislike', 'like', REACT_ON, REACT_O
FF, $comment_id, $post_id, $id_user);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    } else {
        if ($res->disslikes > 0) {
            try {
                deleteVote('reactions', 'dislike', $post_id, $comment_id, $i
d_user);

                http_response_code(204);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
            }
        } else if ($rez->disslikes == 0 && $res->likes > 0) {
            try {
                voteChange('reactions', 'like', 'dislike', REACT_OFF, REACT_
ON, $comment_id, $post_id, $id_user);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
}
}
} else {
    http_response_code(404);
}

<?php
session_start();
header("Content-type:application/json");

```

```

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    require_once '../../config/connection.php';
    require_once '../functions.php';
    $action = $_POST['action'];
    $post_id = $_POST['post'];
    // echo json_encode($post_id);
    $comment_id = $_POST['comment'];

    $id_user = isset($_SESSION['users']) ? $_SESSION['users']->id : '';
    // echo json_encode($post_id);
    // echo json_encode($comment_id);
    // echo json_encode($action);

    $q = checkIsReacted('reactions', $post_id, $comment_id, $id_user);
    //var_dump($q);
    define("REACT_ON", 1);
    define("REACT_OFF", 0);
    if ($q->rowCount() == 0) {

        if ($action == "like") {
            try {
                insertReaction('reactions', 'like', $post_id, $comment_id, $id_us
er, REACT_ON);
                echo json_encode("Commet is added");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        } else {
            try {
                insertReaction('reactions', 'dislike', $post_id, $comment_id, $i
d_user, REACT_ON);
                echo json_encode("Dislike is added");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    } else {
        $res = $q->fetch();
        // echo json_encode($res);
        if ($action == "like") {
            // echo json_encode('like');

```

```

        if ($res->likes > 0) {
            try {
                deleteVote('reactions', 'like', $post_id, $comment_id, $id_us
er);

                http_response_code(204);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        } else if ($res->likes == 0 && $res->disslikes > 0) {
            try {
                voteChange('reactions', 'dislike', 'like', REACT_ON, REACT_O
FF, $comment_id, $post_id, $id_user);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    } else {
        if ($res->disslikes > 0) {
            try {
                deleteVote('reactions', 'dislike', $post_id, $comment_id, $i
d_user);

                http_response_code(204);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
            }
        } else if ($res->disslikes == 0 && $res->likes > 0) {
            try {
                voteChange('reactions', 'like', 'dislike', REACT_OFF, REACT_
ON, $comment_id, $post_id, $id_user);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
} else {
    http_response_code(404);
}

```


Models/categories/delete.php

```
<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    try {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $delete = deleteData('categories', 'id', $_POST['id']);
        http_response_code(204);
    } catch (PDOException $th) {
        echo json_encode("You cant delete this data, because is related with other data!");
        http_response_code(500);
    }
} else {
    http_response_code(404);
}
```

Models/categories/getAll.php

```
<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == "GET") {
    require_once '../../config/connection.php';
    require_once '../functions.php';
    $categories = getAll('categories');
    echo json_encode($categories);
} else {
    http_response_code(404);
}
```

Models/categories/insert.php

```
<?php
header("Content-type:application/json");
```

```

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $rename = '/^[A-Z][a-z]{3,15}$/';
    $name = $_POST['name'];
    $errors = [];
    if (!preg_match($rename, $name)) {
        array_push($errors, "Category name isn't ok!");
    }
    if (count($errors) == 0) {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $nameCheck = getOneWithoutFetch('categories', 'name', $name);
        if ($nameCheck->rowCount() == 0) {
            try {
                insertNewCategory('categories', $name);
                echo json_encode("Category is successfull created");
                http_response_code(201);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        } else {
            http_response_code(409);
            echo json_encode('That name is already taken!');
        }
    } else {
        foreach ($errors as $error) {
            echo json_encode('Error: ' . $error);
            http_response_code(422);
        }
    }
} else {
    http_response_code(404);
}

```

Models/categories/update.php

```

<?php
header("Content-type:application/json");
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $rename = '/^[A-Z][a-z]{3,15}$/';
    $errors = [];
    if (!preg_match($rename, $_POST['name'])) {

```

```

        array_push($errors, "Category isn't good");
    }
    if (count($errors) == 0) {
        require_once '../config/connection.php';
        require_once '../functions.php';
        $date = date("Y-m-d H:i:s");
        $checkName = getOneWithoutFetch('categories', 'name', $_POST['name']);
        if ($checkName->rowCount() > 0) {
            $category = getDataWithFetch('categories', 'name', $_POST['name']);
            if ($category->name == $_POST['name'] && $category->id == $_POST['id']) {
                try {
                    updateCategory('categories', $_POST['name'], $date, $_POST['id']);
                } catch (PDOException $th) {
                    echo json_encode($th->getMessage());
                    http_response_code(500);
                }
            } else {
                echo json_encode("This data is already in use! Choose another name.");
                http_response_code(409);
            }
        } else {
            try {
                updateCategory('categories', $_POST['name'], $date, $_POST['id']);
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    } else {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(422);
        }
    }
} else {
    http_response_code(404);
}

```

Models/actions/exportData.php

```
<?php
require "../config/connection.php";
require "../functions.php";
if ($_GET['action'] == 'excel') {

    //var_dump($field);
    $field = getColumnIngridients();
    $data = getAll('categories');
    $timestamp = time();
    $filename = 'ingredientsNJ326' . $timestamp . '.xls';
    header("Content-Type: application/vnd.ms-excel");
    header("Content-Disposition: attachment; filename=\"$filename\"");
    foreach ($field as $d)
        echo $d->COLUMN_NAME . "\t";
    echo "\n";
    foreach ($data as $d)
        echo $d->id . "\t" . $d->name . "\t" . $d->created_at . "\t" . $d->updated_at . "\t\n";

    exit();

    exit();;
} elseif ($_GET['action'] == 'word') {

    $timestamp = time();
    $filename = 'documentNJ326' . $timestamp . '.doc';
    header("Content-type: application/vnd.ms-word");
    header("Content-Disposition: attachment; filename=\"$filename\"");
    echo "Nemanja Jovicic 326/14 \n";
    echo "I'm a twenty-something web developer, a level thirty-
six Pokémon trainer
and a somewhat proud mother of two dumb dogs and an annoying cat.\n
    I enjoy making websites and I'm currently studying at the ICT College of
Vocational Studies in Belgrade. I like working as a team member as well as
independently; a team leader and a team player. I am curious and constantly learn
ing.";

    exit();
```

```
} else {  
    echo "NO";  
}
```

Models/action/getStats.php

```
<?php  
include "../../config/connection.php";  
include "../functions.php";  
$file = fopen("../../data/logs.txt", "r");  
$rows = file("../../data/logs.txt");  
fclose($file);  
$stats = [];  
$total = 0;  
$total24 = 0;  
$mainPages = ["home" => 0, "news" => 0, "author" => 0];  
$mainPages24 = ["home" => 0, "news" => 0, "author" => 0];  
$visitedPages = [];  
foreach ($rows as $row) {  
  
    $data = explode("\t", $row);  
    $url = $data[0];  
    $url = explode("?", $url);  
    $mainUrl = $url[0];  
    $visited = explode("/", $mainUrl);  
    $index = $visited[count($visited) - 1];  
    if ($index != "index.php") {  
        continue;  
    }  
    if (count($url) == 1) {  
        $mainPages["home"]++;  
        $total++;  
    } else {  
        $page = $url[1];  
        $page = explode("&", $page)[0];  
        $page = explode("=", $page);  
        if ($page[0] == "page" && count($page) > 1) {  
            if (array_key_exists($page[1], $mainPages)) {  
                $mainPages[$page[1]]++;  
                $total++;  
                if (inTheLast24H($data[1]))  
                    $mainPages24[$page[1]]++;  
            }  
        }  
    }  
}
```

```

    }
}
}
}
$logins = loggedInToday();
$registeredUsers = registeredusers();
$stats = ["overallViews" => $mainPages, "todayViews" => $mainPages24, "todayLogin"
" =>
$logins, "registeredUsers" => $registeredUsers];
echo json_encode($stats);

```

Models/action/getVotes.php

```

<?php
session_start();
if ($_SERVER['REQUEST_METHOD'] == "GET") {
    $comment_id = $_GET['comment'];
    require_once '../../config/connection.php';
    $id = isset($_SESSION['users']) ? $_SESSION['users']->id : '';
    $userLikes = $conn-
>query("SELECT user_id FROM reactions WHERE comment_id = $comment_id AND user_id
= $id AND likes != 0")->fetch();
    $userDislike = $conn-
>query("SELECT user_id FROM reactions WHERE comment_id = $comment_id AND user_id
= $id AND disslikes > 0")->fetch();
    $countLikes = $conn-
>query("SELECT COUNT(likes) as likes FROM reactions WHERE comment_id = $comment_id
AND likes > 0 LIMIT 1")->fetch();
    $countDislike = $conn-
>query("SELECT COUNT(disslikes) as disslake FROM reactions WHERE comment_id = $co
mment_id AND disslikes > 0 LIMIT 1")->fetch();
    echo json_encode([
        'user_id' => $id,
        'userLikes' => $userLikes,
        'userDisslikes' => $userDislike,
        'countLikes' => $countLikes,
        'countDisslike' => $countDislike
    ]);
} else {
    http_response_code(404);
}

```

Models/action/login.php

```
<?php
session_start();
if (isset($_POST['btnSubmitLogin'])) {
    $email = $_POST['loginEmail'];
    $password = $_POST['loginPassword'];

    $errors = [];
    $rePassword = '/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/';
    if (!preg_match($rePassword, $password)) {
        array_push($errors, "Password isn't ok");
    }
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        array_push($errors, "Email of user isn't ok!");
    }
    if (count($errors) == 0) {
        try {
            require_once '../config/connection.php';
            require_once '../functions.php';
            $queryCheck = $conn-
>prepare("SELECT u.*, r.id as roleId, r.name as roleName FROM users u JOIN roles
r ON u.role_id = r.id WHERE email=? AND password=?");
            $queryCheck->execute([$email, md5($password)]);
            if ($queryCheck->rowCount() > 0) {
                $user = $queryCheck->fetch();

                $_SESSION['users'] = $user;
                insertUserActivity($email, "Login");
                if ($user->roleName == "Admin") {
                    header("Location: ../../index.php?page=admin_home");
                } else if ($user->roleName == "User") {
                    header("Location: ../../index.php?page=home");
                }
            } else {
                $_SESSION['errors'] = "Your credentials is not ok!";
                header("Location: ../../index.php?page=login");
            }
        } catch (PDOException $th) {
            echo $th->getMessage();
        }
    } else {
```

```

        $_SESSION['errors'] = "Your credentials isn't ok";
        header("Location:../../index.php?page=login");
    }
}

```

Models/action/logout.php

```

<?php
session_start();

if (isset($_SESSION['users'])) {
    require_once '../functions.php';
    $email = $_SESSION['users']->email;
    insertUserActivity($email, "Logout");
    unset($_SESSION['users']);
    header("Location:../../index.php?page=login");
}

```

Models/action/register.php

```

<?php
header("Content-type:applicaton/json");
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $first_name = $_POST['firstName'];
    $last_name = $_POST['lastName'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    $errors = [];
    $reFirstLastName = '/^[A-Z][a-z]{3,15}$/';
    $rePassword = '/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[a-zA-Z]).{8,}$/';

    if (!preg_match($reFirstLastName, $first_name)) {
        array_push($errors, "First name of user isn't ok");
    }
    if (!preg_match($reFirstLastName, $last_name)) {
        array_push($errors, "Last name of user isn't ok");
    }
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {

```



```

        array_push($errors, "Email of user isn't ok!");
    }
    if (!preg_match($rePassword, $password)) {
        array_push($errors, "Password of user isn't ok");
    }
    if (count($errors) > 0) {
        foreach ($errors as $error) {
            echo json_encode("Error: " . $error);
            http_response_code(422);
        }
    } else {
        require_once '../../config/connection.php';
        require_once '../functions.php';
        $checkEmail = getOneWithoutFetch('users', 'email', $email);
        if ($checkEmail->rowCount() > 0) {
            http_response_code(409);
            echo json_encode("That email is already taken");
        } else {
            try {
                define("USER_ROLE", 2);
                insertNewUser('users', $first_name, $last_name, $email, $password
, USER_ROLE);
                echo json_encode("User is successfull created");
                http_response_code(201);
                insertUserActivity($email, "Register");
            } catch (PDOException $th) {
                echo json_encode($th->getMessage());
                http_response_code(500);
            }
        }
    }
} else {
    http_response_code(404);
}

```