

Nemanja Jovicic 247/21

AVANTURISTA

Contents

Korisceni elementi za izgradnju sajta	5
Funkcionalnosti sajta.....	6
Sema baze podataka	8
Sema sajta	8
Sitemap	9
Struktura projekta	10
Stranice	12
Home stranica	12
.....	12
Stranica sa destinacijama	13
Stranica o pojedinačnoj destinaciji	14
Stranica za logovanje.....	14
.....	15
Stranica za registraciju	16
.....	16
Admin (Home) Page	17
.....	17
Stranica za kategorije	17
Stranica za ankete	18
Stranica za dodavanje novih pitanja	18
Stranica za dodavanje novih odgovora	19
.....	19
Stranica za dodavanje novih destinacija	19
.....	19
Stranica za dodavanje novih datuma polaska	20
Stranica za prikazivanje poruka koje dolaze iz kontakt forme.....	20
.....	20
Kodovi	21
index.php	21
admin.php	22
includes/partials/head.php.....	24
include/partials/navigation.php.....	24

include/partials/footer.php	25
includes/pages/errors.php	27
includes/pages/user/home.php	27
includes/pages/user/tours.php	28
includes/pages/user/contact.php	32
includes/pages/user/author.php	33
includes/pages/admin/home.php	33
includes/pages/admin/categories.php	34
includes/pages/admin/survey.php	36
includes/pages/admin/questions.php	38
includes/pages/admin/answers.php	40
includes/pages/admin/posts.php	41
includes/pages/admin/dates.php	44
includes/pages/admin/messages.php	46
includes/pages/auth/login.php	48
models/functions.php	50
models/getData.php	62
models/utilities.php	63
models/validations.php	65
models/answers/delete.php	67
models/answers/getOne.php	68
models/answers/store.php	68
models/answers/update.php	69
models/auth/register.php	69
models/auth/login.php	70
models/auth/logout.php	71
models/categories/getOne.php	71
models/categories/delete.php	72
models/dates/delete.php	74
models/dates/store.php	74
models/dates/edit.php	75
models/dates/update.php	75
models/messages/getOne.php	76

models/messages/pagination.php.....	77
models/messages/store.php.....	77
models/questions/delete.php	78
models/questions/getOne.php.....	78
models/question/store.php	79
models/questions/update.php	79
models/reservations/store.php	80
models/survey/delete.php.....	81
models/survey/store.php.....	82
models/survey/update.php	82
models/survey/getOne.php	83
models/survey/vote.php.....	83
models/tours/delete.php.....	84
models/tours/edit.php.....	85
models/tours/store.php.....	85
models/tours/update.php	86
models/tours/filter.php.....	87
assets/js/auth.js	88
assets/js/utilities.js.....	90
assets/js/validation.js.....	93
assets/js/admin.js	96
assets/js/user.js.....	119

Korisceni elementi za izgradnju sajta

Korisceni programski jezici

- PHP
- JS

Korisceni tip baze

- SQL

Koriscene biblioteke

- JQuery
- Bootstrap

Ostale elemtni koji su korisceni

- CkEditor
- Herolcons
- HTML

Funkcionalnosti sajta

Stranice i sadrzaj koji se prikazuje zavisi od : u slucaju da korisnik nije ulogovan, da jeste i da je uloga tipa admin ili da je obican korisnik

- Navigacioni meni se ispisuje iz baze

Korsnicki deo:

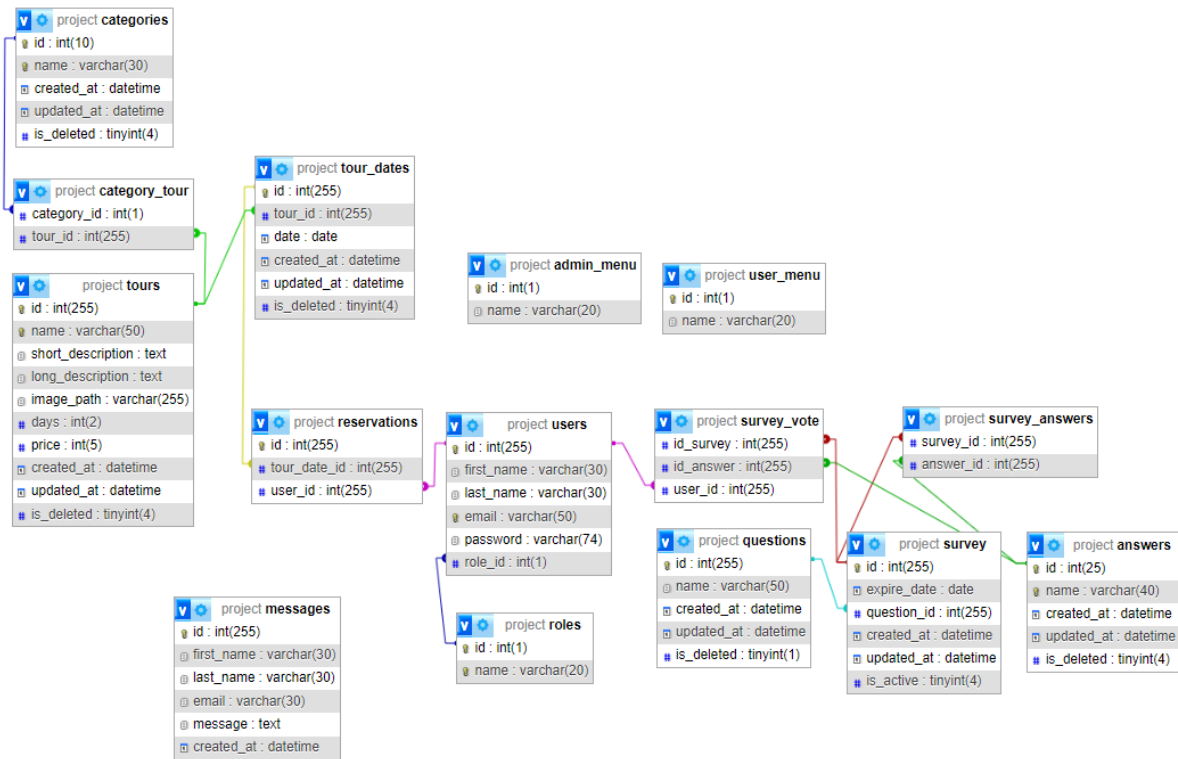
- Autentifikacija korisnika (logovanje i registracija)

- Mogucnost popunjavanja ankete, moguće je samo jednom po anketi
- Kreiranje rezervacija
- Filtiranje i pretraga kroz moguće destinacije
- Kontaktiranje administratora putem kontakt stranice

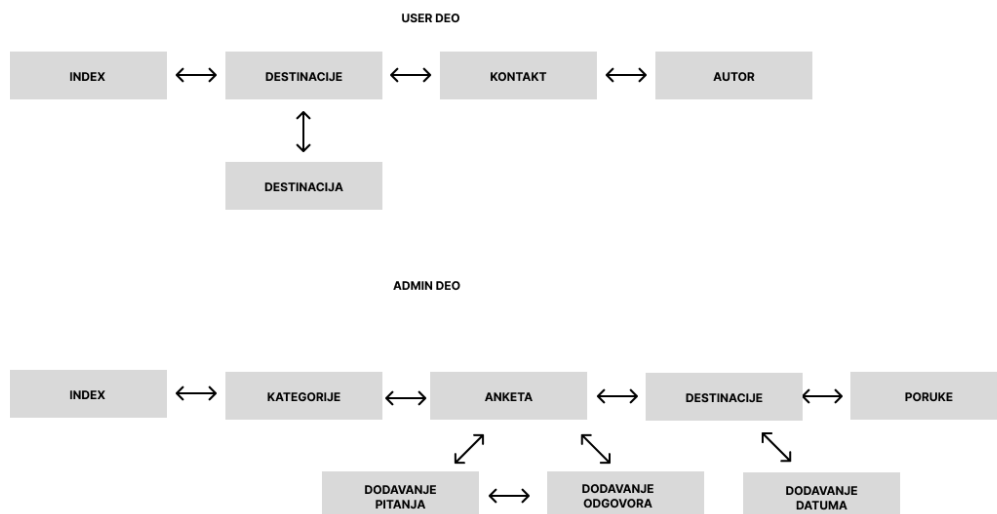
Admin deo:

- Manipulacijom kategorijama (celokupan crud)
- Manipulacijom anketama, gde je moguće dodati odgovore i pitanja (celokupan crud)
- Manipulacija turama (celokupan crud) i dodavanje novih termina za te ture (celokupan crud)

Sema baze podataka

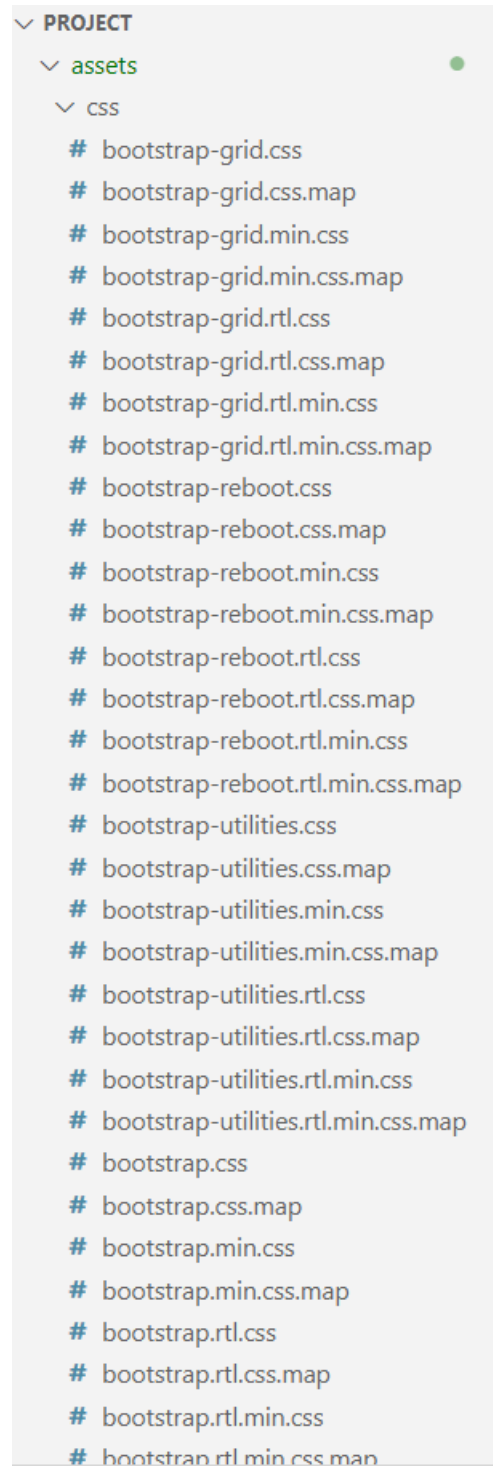


Sema sajta



Sitemap

Struktura projekta



img		
	127789955.jpeg	U
	1717148371_26_6511d842d0...	U
	1717148443_ovcarsko-kablar...	U
	1717148581_magija-pester...	U
	1717148658_43_61fc3f0988d...	U
	1717148746_196_6399c4cc7...	U
	1717148827_gornje-poduna...	U
	1717148930_karpati-muntele...	U
	1717149031_181653932_235...	U
	1717149136_11_610303f0efd...	U
	1717149215_golubac-voznja...	U
	1717149328_viber-image-20...	U
	1717149405_Document_Na...	U
	1717149494_188_638f95346...	U
	1717149565_img-e205a1e0d...	U
	photo-1458442310124-dde6...	U
js		
JS	admin.js	M
JS	auth.js	M
JS	bootstrap.bundle.js	
JS	bootstrap.bundle.js.map	
JS	bootstrap.bundle.min.js	
JS	bootstrap.bundle.min.js.map	
JS	bootstrap.esm.js	
JS	bootstrap.esm.js.map	
JS	bootstrap.esm.min.js	
JS	bootstrap.esm.min.js.map	
JS	bootstrap.js	
JS	bootstrap.js.map	

dates		
	store.php	M
	update.php	M
messages		
	getOne.php	
	pagination.php	
	store.php	M
questions		
	delete.php	
	getOne.php	
	store.php	M
	update.php	M
reservations		
	store.php	U
survey		
	delete.php	U
	getOne.php	
	store.php	M
	update.php	M
	vote.php	U
tours		
	delete.php	
	edit.php	
	filter.php	U
	store.php	M
	update.php	M
	functions.php	M
	getData.php	U
	utilities.php	M
	validations.php	M
	admin.php	
	index.php	M
OUTLINE		
TIMELINE		
aster* ↺ 0 ⚠ 0 🗨 0		

Stranice

Home stranica

Istrazite svet sa nama

Hodanje je najsavršeniji oblik kretanja
za osobu koja želi da otkrije pravi život.
– Thoreau. .



Izdvajamo iz ponude



NP SEMENIK - Jezero Valiug |
Rumunija



Ovčarsko-kablarska klisura:
KAJAK – BIKE – HIKE



MAGIJA PEŠTERA - Vrh Trojan
i Rezervat prirode Gut



SICEVAČKA KLISURA -
rimskim putem Via Militaris

Brzi linkovi

AVANTURISTA

„Najbolji način da doživite prirodu je peške. –
Muir

[Pocetna](#)

[Destinacije](#)

[Kontakt](#)



[Dokumnetacija](#)



avanturista@gmail.com

Stranica sa destinacijama

Stranica koja sadrži sve ture koje su u ponudi. Moguće je filtrirati po nazivu, po kategoriji ili u kombinaciji oba kriterijuma

AVANTURISTA

PocetnaDestinacijeKontaktAutor

Log inRegister

Categories


☐Kampovanje

☐Krstarenje


☐Pesacenje

☐Voznja bicikla


☐Voznja kajaka




NP SEMENIK -
Jezero Valiug |
Rumunija




Ovčarsko-kablarska
klisura: KAJAK –
BIKE – HIKE




MAGIJA PEŠTERA -
Vrh Trojan i Rezervat
prirode Gut




SIĆEVAČKA KLISURA
- rimskim putem Via
Militaris




Zagajička brda -
tvrđava Ram |
bajkoviti predeo Sr




GORNJE
PODUNAVLJE -
"Evropski Amazon" |
bajs / hi




KARPATI - Caraku i
Muntele Mik |
Rumunija




NP Šar-planina |
Piribreg (2522 mnv.)
i Jažinačko




PROKLETIJE | biser
prirode Balkana



GOLUBAC -
krstarenje, bajs i
pešačka tura



E-BIKE & HIKE
AVANTURA |
BREZOVICA - ŠAR
PLANINA



VIDOVDANSKI
NOĆNI USPON NA
RTANJ

12

AVANTURISTA


„Najbolji način da doživite prirodu je peške. – Muir


Brzi linkovi

Pocetna

Destinacije

Kontakt

Dokumentacija

avanturista@gmail.com


Stranica o pojedinačnoj destinaciji

Prikaz pojedinačne destinacije, sa mogućnošću rezervacije određenih tura i mogućnosti glasanja na anketi ukoliko korisnik nije glasao.

AVANTURISTA

PocetnaDestinacijeKontaktAutor

Logout



Cena: 3200

Broj dana: 1

Termini:

06/06, 14/06

Rezervacije

Odaberite datum polaska

Datum polaska

▼

Rezervisi

NP SEMENIK - Jezero Valiug | Rumunija

Izazovite avanturu i upoznajte nacionalni park Semenik. Nacionalni park Semenik nalazi se na jugozapadu Rumunije, u centru okruga Karaš-Severin, i prostire se na dve planinske celine: Semeničke planine i Anine planine. Nacionalni park Semenik pokriva površinu od 36.051 ha, a najveću površinu čine šume i to bukove šume (80%). Zapanjujuća je raznolikost impresivnih pećina, šuma starih više vekova, živopisnih klisura, nestvarnih livada koje stvaraju spektakularne pejzaže ovog brdovitog dela Rumunije. Park ima 10 rezervata prirode, 9 obeleženih turističkih staza u okviru zaštićenog područja i 2 biciklističke staze. Od 2017. godine, rezervat prirode "Izvoarele Nere" u Semeniku proglašen je UNESCO-om svetskom baštinom kao deo "Izuzetno značajnih šuma Karpat i drugih regiona Evrope". Ovo područje je Jedno od najznačajnijih područja netaknutih starih šuma u Evropi. Reprezentativni su devičanski ekosistemi čiste bukve, gde se može naći samo nekoliko primeraka drugih vrsta drveta. Ovde se mogu naći impresivna stara stabala, prečnika preko 1,5 m, visine preko 50 m, a starosti preko 400 godina. Staza koju smo odabrali spada u lakše i duž cele staze nema opasnih deonica!

TEHNIČKI OPIS STAZE

Dužina staze - 13,5 km

Najniža tačka - 597 m

Najviša tačka - 1470 m

Kondiciona težina - 4/10

Okvirna vremensko trajanje uspona 6 sati (zavisi od spremnosti grupe)

PLAN I PROGRAM

06.00

Polazak iz Beograda

Vožnja preko Pančeva, Kovina, Bele Crkve, Oravice, Anine, Rešice do Valiuga.

10.30

Dolazak do jezera Valiug (jezero Grozna) i početka žičare odakle započinje pešačka staza.

Stranica za logovanje

Sadrži klijentsku i serversku validaciju

Prijavi se

Email

Password

Log in

Nemate nalog? [Registrujte se](#)

AVANTURISTA

„Najbolji način da doživite prirodu je peške. –
Muir

Brzi linkovi

[Pocetna](#)

[Destinacije](#)

[Kontakt](#)

[Autor](#)



Dokumnetacija



avanturista@gmail.com

Stranica za registraciju

Sadrzi klijentsku I serversku validaciju

AVANTURISTA

[Pocetna](#) [Destinacije](#) [Kontakt](#) [Autor](#)

[Log in](#) [Register](#)

Registruj se

First name

Last name

Email

Password

Registruj se

Vec imate nalog? [Log in](#)

AVANTURISTA

„Najbolji način da doživite prirodu je peške. –
Muir

Brzi linkovi

[Pocetna](#)

[Destinacije](#)

[Kontakt](#)

[Autor](#)



Dokumnetacija



avanturista@gmail.com

Admin (Home) Page

AVANTURISTA

Pocetna Kategorije Ankete Destinacije Poruke

Logout

#	Name	Date	Number of reservations
1	Ovčarsko-kablariska klisura: KAJAK – BIKE – HIKE	31/05/2024	2
2	KARPATI - Carku i Muntele Mik Rumunija	08/06/2024	1

Broj registrovanih korisnika

1

Broj aktivnih destinacija

14

Stranica za kategorije

Stranica za dodavanje kategorija, sadrzi klijentsku I serversku validaciju

AVANTURISTA

Pocetna Kategorije Ankete Destinacije Poruke

Logout

#	Name	Created at	Updated at	Edit	Delete
1	Pesacenje	30/05/2024 04:10:21	30/05/2024 04:10:33	Edit	Delete
2	Voznja bicikla	30/05/2024 04:10:44	-	Edit	Delete
3	Kampovanje	30/05/2024 04:10:51	-	Edit	Delete
4	Voznja kajaka	30/05/2024 04:10:59	-	Edit	Delete
5	Krstarenje	30/05/2024 21:42:46	-	Edit	Delete

Name

Save

Reset

Stranica za ankete

Stranica za dodavanje novih anketa, gde se pitanja I odgovri dodaju dinamicki

AVANTURISTA

PocetnaKategorijeAnketeDestinacijePoruke

Logout

Add question

Add a answer

#	Question	Exire at	Created at	Updated at	Edit	Delete
1	Sta bi ste novo zeleli u ponudi?	07/06/2024	31/05/2024 12:11:05	-	Edit	Delete

Date

mm/dd/yyyy

Question

Izaberite

Answers

☐ Avanture koje traju vise dana

☐ Bolje ture

☐ Avanture na moru

☐ Nesto novo u ponudi

☐ Jedan

☐ Dva

☐ Tri

Save

Reset

Stranica za dodavanje novih pitanja

Dodavanje novih pitanja za ankete

AVANTURISTA

PocetnaKategorijeAnketeDestinacijePoruke

Logout

#	Name	Created at	Updated at	Edit	Delete
1	Sta bi ste novo zeleli u ponudi?	31/05/2024 12:08:17	-	Edit	Delete

Question

Save

Reset

Stranica za dodavanje novih odgovora

Stranica za dodavanje novih odgovora za ankete

AVANTURISTA

PocetnaKategorijeAnketeDestinacijePoruke

Logout

#	Name	Created at	Updated at	Edit	Delete
1	Avanture koje traju vise dana	31/05/2024 12:04:15	-	Edit	Delete
2	Bolje ture	31/05/2024 12:04:43	-	Edit	Delete
3	Avanture na moru	31/05/2024 12:04:51	-	Edit	Delete
4	Nesto novo u ponudi	31/05/2024 12:06:39	-	Edit	Delete
5	Jedan	31/05/2024 12:06:59	-	Edit	Delete
6	Dva	31/05/2024 12:07:26	-	Edit	Delete
7	Tri	31/05/2024 12:07:55	-	Edit	Delete

Answer

Save

Reset

Stranica za dodavanje novih destinacija

Na ovoj stranici moguće je dodati nove destinacije

AVANTURISTA

PocetnaKategorijeAnketeDestinacijePoruke

Logout

#	Naziv	Cena	Broj dana	Kategorije	Dodaj termine	Datum kreiranja	Datum izmene	Izmeni	Obrisi
1	NP SEMENIK - Jezero Valug Rumunija	3200	1	Pesacenje	Dodaj	31/05/2024 11:39:31	31/05/2024 11:41:29	Edit	Delete
2	Ovčarsko-kablariska klisura: KAJAK – BIKE – HIKE	6990	1	Pesacenje, Voznja bicikla, Voznja kajaka	Dodaj	31/05/2024 11:40:43	31/05/2024 11:41:35	Edit	Delete
3	MAGUA PEŠTERA - Vrh Trojan i Rezervat prirode Gut	7990	2	Pesacenje	Dodaj	31/05/2024 11:43:01	-	Edit	Delete
4	SIČEVAČKA KLISURA - rimskim putem Via Militaris	3400	1	Pesacenje	Dodaj	31/05/2024 11:44:18	-	Edit	Delete
5	Zagajička brda - tvrđava Ram bajkoviti predeo Sr	2600	1	Pesacenje	Dodaj	31/05/2024 11:45:46	-	Edit	Delete

123

Naziv

Cena

Slika

Choose FileNo file chosen

Trajanje (u danima)

Kategorija

☐ Kampovanje

☐ Krstarenje

☐ Pesacenje

☐ Voznja bicikla

☐ Voznja kajaka

Kratak opis

Opis putovanja

Paragraph

B

I

Save

Reset

Stranica za dodavanje novih datuma polaska

AVANTURISTA

Pocetna Kategorije Ankete Destinacije Poruke

Logout

NP SEMENIK - Jezero Valiug | Rumunija

#	Datum polaska	Datum kreiranja	Datum izmene	Izmeni	Izbrisi
1	06/06/2024	31/05/2024 11:59:42	-	Edit	Delete
2	14/06/2024	31/05/2024 11:59:54	-	Edit	Delete

Date

mm/dd/yyyy

Save

Reset

Stranica za prikazivanje poruka koje dolaze iz kontakt forme

AVANTURISTA

Pocetna Destinacije Kontakt Autor

Log in Register

Contact us

First name

Last name

Email

Message

[Contact us](#)

Kodovi

[index.php](#)

```
<?php
    session_start();

    require_once 'config/connection.php';
    include 'models/functions.php';

    include 'includes/partials/head.php';
    include 'includes/partials/navigation.php';
    $page = '';
    if(isset($_GET['page'])){
        $page = $_GET['page'];
        switch($page){
            case 'login':
                include 'includes/pages/auth/login.php';
                break;
            case 'register':
                include 'includes/pages/auth/register.php';
                break;
            case 'pocetna':
                include 'includes/pages/user/home.php';
                break;
            case 'destinacije':
                include 'includes/pages/user/tours.php';
                break;
            case 'kontakt':
                include 'includes/pages/user/contact.php';
                break;
            case 'tour':
                include 'includes/pages/user/tour.php';
                break;
            case 'autor':
                include 'includes/pages/user/author.php';
                break;
            default :
                include 'includes/pages/errors.php';
                break;
        }
    }else{
        include 'includes/pages/user/home.php';
    }
    include 'includes/partials/footer.php';
```

admin.php

```
<?php
```

```
session_start();

require_once 'config/connection.php';
include 'models/functions.php';

include 'includes/partials/head.php';
include 'includes/partials/navigation.php';
$page = '';
if(isset($_GET['page'])){
    $page = $_GET['page'];
    switch($page){
        case 'kategorije':
            include 'includes/pages/admin/categories.php';
            break;
        case 'destinacije':
            include 'includes/pages/admin/posts.php';
            break;
        case 'dates':
            include 'includes/pages/admin/dates.php';
            break;
        case 'ankete':
            include 'includes/pages/admin/survey.php';
            break;
        case 'poruke':
            include 'includes/pages/admin/messages.php';
            break;
        case 'home':
            include 'includes/pages/admin/home.php';
            break;
        case 'add-question':
            include 'includes/pages/admin/questions.php';
            break;
        case 'add-answer':
            include 'includes/pages/admin/answers.php';
            break;
        default:
            include 'includes/pages/errors.php';
    }
}else{
    include 'includes/pages/admin/home.php';
}
include 'includes/partials/footer.php';
```

includes/partials/head.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>
        <?= isset($_GET['page']) ? ucfirst($_GET['page']) : 'Pocetna'; ?>
    </title>
    <link rel="stylesheet" href="assets/css/bootstrap.min.css">
    <link rel="stylesheet" href="assets/css/style.css">

    <meta name="description" content="Find the best adventure">
    <meta name="keywords" content="Hiking, Cicling, Kayaking">

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

include/partials/navigation.php

```
<?php
$homeUrl = "";
$navigation = "";
$route = '';
    if(!isset($_SESSION['user']) || isset($_SESSION['user']) && $_SESSION['user']->role_id == 2){
        $homeUrl = "index.php";
        $navigation = getUserMenu();
    }else if(isset($_SESSION['user']) && $_SESSION['user']->role_id == 1){
        $homeUrl = "admin.php";
        $navigation = getAdminMenu();
    }
?>
<nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container">
        <a class="navbar-brand fw-bold text-uppercase"
href="<?=$homeUrl?>">Avanturista</a>
        <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 foreach($navigation as $home):?>
```



```

        <li class="nav-item"><a href="<?=$homeUrl?>?page=<?=$home->name?>"
class="nav-link <? = isset($_GET['page']) && $_GET['page'] == $home->name ?
'active border-bottom' : ''?>"><?=ucfirst($home->name)?></a></li>
    <?php endforeach;?>
</ul>
    <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <?php if(!isset($_SESSION['user'])):?>
            <li class="nav-item"><a href="index.php?page=login" class="nav-link <?=
isset($_GET['page']) && $_GET['page'] == 'login'? 'active border-
bottom': ''?>">Log in</a></li>
            <li class="nav-item"><a href="index.php?page=register" class="nav-
link <? = isset($_GET['page']) && $_GET['page'] == 'register'? 'active border-
bottom': ''?>">Register</a></li>
            <?php else:~>
                <li class="nav-item"><a href="models/auth/logout.php" class="nav-
link">Logout</a></li>
            <?php endif;?>
        </ul>
    </div>
</div>
</nav>

```

include/partials/footer.php

```

<?php
    $userNav = getUserMenu();
?>
<footer class="mt-auto bg-body-tertiary">
    <div class="container py-3 px-1">
        <?php if(!isset($_SESSION['user']) || isset($_SESSION['user']) &&
$_SESSION['user']->role_id == 2):?>
            <div class="row px-2 py-1 text-center align-items-center ">
                <div class="col-lg-4 text-center mb-2 mb-lg-0 align-items-
center">
                    <h2> <a href="index.php" class="nav-link text-uppercase fw-
bold">Avanturista</a></h2>
                    <p class="fs-5">
                        „Najbolji način da doživite prirodu je peške. – Muir
                    </p>
                </div>
                <div class="col-lg-4">
                    <p class="mb-2 fs-3 fw-bold">Brzi linkovi</p>
                    <div class="navbar-nav d-flex flex-column gap-1">
                        <?php foreach($userNav as $nav):?>

```

```

        <li class="nav-item"><a href="index.php?page=<?=$nav-
>name?>" class="nav-link"><?=ucfirst($nav->name)?></a></li>
        <?php endforeach;?>
    </div>
</div>
<div class="col-lg-4">
    <div class="d-flex flex-column gap-2">
        <div class="d-flex align-items-center gap-2 fw-bold ">
            <svg xmlns="http://www.w3.org/2000/svg"
fill="none" viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor"
style="width:36px; height:36px;" class="bg-dark rounded-circle px-2 py-2 text-
light">
                <path stroke-linecap="round" stroke-
linejoin="round" d="M19.5 14.25v-2.625a3.375 3.375 0 0 0-3.375-3.375h-1.5A1.125
1.125 0 0 1 13.5 7.125v-1.5a3.375 3.375 0 0 0-3.375-3.375H8.25m2.25 0H5.625c-.621
0-1.125.504-1.125 1.125v17.25c0 .621.504 1.125 1.125 1.125h12.75c.621 0 1.125-
.504 1.125-1.125V11.25a9 9 0 0 0-9-9Z" />
            </svg>
            <span><a href="models/docs.pdf" class="nav-
link">Dokumnetacija</a></span>
        </div>
        <div class="d-flex align-items-center gap-2 fw-bold ">
            <svg xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke-width="1.5" stroke="currentColor" style="width:36px;
height:36px;" class="bg-dark rounded-circle px-2 py-2 text-light">
                <path stroke-linecap="round" stroke-linejoin="round" d="M21.75 6.75v10.5a2.25
2.25 0 0 1-2.25 2.25h-15a2.25 2.25 0 0 1-2.25-2.25V6.75m19.5 0A2.25 2.25 0 0 0
19.5 4.5h-15a2.25 2.25 0 0 0-2.25 2.25m19.5 0v.243a2.25 2.25 0 0 1-1.07 1.916l-
7.5 4.615a2.25 2.25 0 0 1-2.36 0L3.32 8.91a2.25 2.25 0 0 1-1.07-1.916V6.75" />
            </svg> <span>avanturista@gmail.com</span>
        </div>
    </div>
</div>
</div>
</div>
<?php endif;?>
</div>
</footer>
<script src="assets/js/jquery.min.js"></script>
<?php if(!isset($_SESSION['user'])):?>
<script src="assets/js/auth.js"></script>
<?php endif;?>
<script src="assets/js/utilities.js"></script>
<script src="assets/js/validation.js"></script>
<?php if(isset($_SESSION['user']) && $_SESSION['user']->role_id == 1):?>
<script src="assets/js/admin.js"></script>

```

```

<?php endif;?>
<script src="assets/js/bootstrap.bundle.min.js"></script>
<script
src="https://cdn.ckeditor.com/ckeditor5/41.4.2/classic/ckeditor.js"></script>
<?php if(!isset($_SESSION['user']) ||isset($_SESSION['user']) &&
$_SESSION['user']->role_id == 2):?>
<script src="assets/js/user.js"></script>
<?php endif;?>
</body>
</html>

```

includes/pages/errors.php

```

<?php
    $codeError = isset($_GET['code']) ? $_GET['code'] : '';
    $code = '';
    $message = '';
    switch($codeError){
        case '403':
            $code = 403;
            $message='Neovlasceni pristup';
        default:
            $code = 404;
            $message = "Stranica nije pronadjena";
    }
?>
<main>
    <section class="container">
        <div class="row mt-5 text-center">
            <h1><?=$code?></h1>
            <p><?=$message?></p>
        </div>
    </section>
</main>

```

includes/pages/user/home.php

```

<?php
$getTours = getMainTours();?>
<main>
    <section class="container">
        <div class="row py-5 align-items-center">
            <div class="col-lg-4">
                <h1 class="fw-bold mb-5 ">Istrazite svet sa nama</h1>

```

```

        <p class="fs-4">Hodanje je najsavršeniji oblik kretanja za osobu koja
želi da otkrije pravi život. – Thoreau. . </p>
    </div>
    <div class="col-lg-7 ms-auto">
        
    </div>
</div>
</section>
<section class="container mb-5">
    <h2 class="mb-4">Izdvajamo iz ponude</h2>
    <div class="row">
        <?php foreach($getTours as $tour):?>
            <div class="col-lg-3">
                <div class="card h-100">
                    name?">
                    <div class="card-body text-center">
                        <h5 class="card-title">
                            <a href="index.php?page=tour&id=<?=$tour->id?"
class="nav-link"><?=$tour->name?></a>
                        </h5>
                    </div>
                </div>
            </div>
        <?php endforeach;?>
    </div>
</section>
</main>

```

includes/pages/user/tours.php

```

<?php
    $getAvailableCategories = getAvailableCategories();
    $getAvailableTours = getAvailableTours();
    $toursPagination = getAvaliableToursPagination();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div id="tour_response_message" class="mb-2"></div>
            <div class="col-lg-3">
                <input type="text" class="form-control mb-2" id="search_text">
                <div class="mb-2">
                    <label for="category" class="mb-1 fw-bold">Categories</label>

```

```

        <?php foreach($getAvailableCategories as $category):?>
            <div class="form-check">
                <input class="form-check-input" type="checkbox"
value="<?=$category->id?>" id="check_<?=$category->id?>" name="categories">
                <label class="form-check-label" for="check_<?=$category->id?>">
                    <?=$category->name?>
                </label>
            </div>
        <?php endforeach;?>
    </div>
</div>
<div class="col-lg-9">
    <div class="row " id="tours">
        <?php foreach($getAvailableTours as $tour):?>
            <div class="col-lg-3 mb-2">
                <div class="card h-100">
                    name?>">
                    <div class="card-body text-center">
                        <h2 class="fs-5"> <a
href="index.php?page=tour&id=<?=$tour->id?>" class="nav-link"><?=$tour->name?></a></h2>
                    </div>
                </div>
            </div>
        <?php endforeach;?>
    </div>
    <div class="d-flex justify-content-center">
        <nav aria-label="Page navigation example">
            <ul class="pagination" id="pages-tours">
                <?php for($i =0 ;$i<$toursPagination;$i++):?>
                    <li class="page-item"><a data-limit = '<?=$i?>'
class="page-link page-tours <?=$i == 0 ? 'active' : ''?>"
href="#"><?=$i+1?></a></li>
                <?php endfor;?>
            </ul>
        </nav>
    </div>
</div>
</div>
</section>
</main>

```

```

includes/pages/user/tour.php <?php

$id = $_GET['id'];
$tourDates = '';
$tourData = getTourData($id);

$tourDates = getAvailableTourDate($id);
$getQuestion = getAvailableQuestion();

$getAnswers = getSurveyOptions($getQuestion->id);
if(isset($_SESSION['user'])){
    $checkVote = checkIfUserVote($_SESSION['user']->id, $getQuestion->id);
}
?>
<main>
    <section>
        <div class="container">
            <div class="row mt-5">
                <div id="tour_reservation_response_message"></div>
                <div class="<?=isset($_SESSION['user']) ? 'col-lg-6 mx-auto mb-2
mb-lg-0 ' : 'col-lg-6 mx-auto'?>">
                    
                    <h2 class="mb-3"><?= $tourData->name?></h2>
                    <p class="lh-base fs-6"><?=$tourData->short_description?></p>
                    <?=$tourData->long_description?>
                </div>

                <div class="col-lg-4 ms-auto">

                    <p class="fs-3 fw-bold">Cena: <span class="ms-2 text-
danger"><?=$tourData->price?></span></p>
                    <p class="fs-3">Broj dana: <span class="ms-2"><?=$tourData-
>days?></span></p>
                    <p class="fs-3 fw-bold">Termini:</p>
                    <?php foreach($tourDates as $index => $tourDate):?>
                        <span class="fs-4"><?= array_key_last($tourDates) !==
$index ? date('d/m', strtotime($tourDate->date)).', ' : date('d/m',
strtotime($tourDate->date))?></span>
                    <?php endforeach;?>
                    <?php if(isset($_SESSION['user'])):?>
                    <p class="fs-3 mt-3 fw-bold">Rezervacije</p>

                    <form action="#">

```

```

        <input type="hidden" name="tour_id" id="tour_id"
value="<?=$id?>">
        <div class="mb-3">
            <label for="date" class="mb-1">Odaberite datum
polaska</label>
            <select name="date" id="date" class="form-select">
                <option value="0">Datum polaska</option>
                <?php foreach($tourDates as $tourDate):?>
                    <option value="<?=$tourDate-
>id?>"><?=date('d/M/Y',strtotime($tourDate->date))?></option>
                    <?php endforeach;?>
                </select>
                <div id="tour_date_error"></div>
            </div>
            <div class="d-grid gap-1">
                <button class="btn btn-sm btn-primary" type="button"
id="btnReserveTour">Rezervisi</button>
            </div>
        </form>
        <div id='survey'>
            <?php if(!$checkVote):?>
            <p class="fs-3 my-2">Anketa</p>
            <p class="fs-5"><?=$getQuestion->name?></p>
            <form action="#">
                <input type="hidden" name="survey_id" id="survey_id"
value="<?=$getQuestion->id?>">
                <div class="mb-2">
                    <?php foreach($getAnswers as $answer):?>
                    <div class="form-check">
                        <input class="form-check-input" type="radio"
name="answers" id="answer_<?=$answer->answer_id?>" value="<?=$answer-
>answer_id?>">
                        <label class="form-check-label"
for="answer_<?=$answer->answer_id?>">
                            <?=$answer->name?>
                        </label>
                    </div>
                    <?php endforeach;?>
                    <div id="vote_error"></div>
                </div>
                <div class="d-grid">
                    <button class="btn btn-sm btn-primary" type="button"
id="btnVote">Glasaj</button>
                </div>
            </form>
        </div>
    </div>

```

```

        </form>
    </div>
    <?php endif;
        endif;?>
</div>

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

```

includes/pages/user/contact.php

```

<main>
    <section class="container">
        <div class="row mt-5">
            <div class="col-lg-6 mx-auto">
                <h2 class="mb-3">Contact us</h2>
                <div id="contact_response_message" class="mb-2"></div>
                <form action="#">
                    <div class="mb-3">
                        <label for="first_name" class="mb-1">First name</label>
                        <input type="text" name="first_name" id="first_name"
class="form-control mb-1">
                        <div id="first_name_error"></div>
                    </div>
                    <div class="mb-3">
                        <label for="last_name" class="mb-1">Last name</label>
                        <input type="text" name="last_name" id="last_name"
class="form-control">
                        <div id="last_name_error"></div>
                    </div>
                    <div class="mb-3">
                        <label for="email" class="mb-1">Email</label>
                        <input type="email" name="email" id="email" class="form-
control">
                        <div id="email_error"></div>
                    </div>
                    <div class="mb-3">
                        <label for="message" class="mb-1">Message</label>
                        <textarea name="message" id="message" class="form-
control"></textarea>
                        <div id="message_error"></div>
                    </div>
                </form>
            </div>
        </div>
    </section>
</main>

```



```

        <button class="btn btn-sm btn-primary" id="btnContactUs"
type="button" name="btnContactUs">Contact us</button>
    </form>
</div>
</div>
</section>
</main>

```

includes/pages/user/author.php

```

<main>
    <section class="container">
        <div class="row mt-5">
            <div class="col-lg-4 mx-auto">
                
                <p class="text-center my-3">
                    Moje ime je <span class="fw-bold">Nemanja Jovici</span>
dolazim
                    iz Uba. Moji hobiji su planinarenje i igranje video igara.
                </p>
            </div>
        </div>
    </section>
</main>

```

includes/pages/admin/home.php

```

<main>
    <section class="container">
        <div class="row mt-5">
            <div class="col-lg-8 mb-2 mb-lg-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Name</th>
                                <th scope="col">Date</th>
                                <th scope="col">Number of reservations</th>
                            </tr>
                        </thead>
                        <tbody id="reservations">

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

```

```

        <div class="col-lg-4">
            <div class="d-grid gap-1">
                <div class="col mb-2">
                    <div class="card text-center py-3">
                        <h5 class="mb-2">Broj registrovanih korisnika</h5>
                        <h6 id="numberOfUsers" class="display-4 fw-
bold"></h6>
                    </div>
                </div>
            </div>
            <div class="col">
                <div class="card text-center py-3">
                    <h5 class="mb-2">Broj aktivnih destinacija</h5>
                    <h6 id="numberOfTours" class="display-4 fw-
bold"></h6>
                </div>
            </div>
        </div>
    </div>
</section>
</main>

```

includes/pages/admin/categories.php

```

<?php
    if(!isset($_SESSION['user']) || isset($_SESSION['user']) && $_SESSION['user']-
>role_id != 1)
        header("Location: index.php?page=errors&code=403");

    $categories = getAllCategories();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div id="category_response_message" class="mb-2"></div>
            <div class="col-lg-8 mb-2 mb-lg-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <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">Edit</th>

```

```

        <th scope="col">Delete</th>
    </tr>
</thead>
<tbody id="categories">
    <?php foreach($categories as $index => $category):?>
        <tr id="category_<?=$index?>">
            <th scope="row"><?=$index + 1?></th>
            <td><?=$category->name?></td>
            <td><?= date('d/m/Y H:i:s',
strtotime($category->created_at));?></td>
            <td><?= $category->updated_at != null ?
date("d/m/Y H:i:s",strtotime($category->updated_at)) : "-"?></td>
            <td><button class="btn btn-sm btn-success
btn-edit-category" type="button" data-id="<?=$category->id?>" data-
index="<?=$category->id?>">Edit</button></td>
            <td><button class="btn btn-sm btn-danger btn-
delete-category" type="button" data-id="<?=$category->id?>" data-
index="<?=$index?>" data-status="<?=$category->is_deleted?>"><?=$category-
>is_deleted == 0 ? 'Delete' : 'Activate'?></button></td>
        </tr>
    <?php endforeach;?>
</tbody>
</table>
</div>
</div>
<div class="col-lg-4">
    <form action="#">
        <input type="hidden" name="category_id" id="category_id">
        <input type="hidden" name="category_index"
id="category_index">
        <div class="mb-3">
            <label for="name" class="mb-1">Name</label>
            <input type="text" name="name" id="name" class="form-
control mb-1">
            <div id="name_error"></div>
        </div>
        <div class="d-grid gap-1">
            <button class="btn btn-sm btn-primary"
id="btnSaveCategory" name="btnSaveCategory" type="button">Save</button>
            <button class="btn btn-sm btn-danger"
id="btnResetCategory" name="btnResetCategory"
type="btnResetCategory">Reset</button>
        </div>
    </form>
</div>

```

```

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

```

includes/pages/admin/survey.php

```

<?php
    $questions = getAllQuestions();
    $answers = getAllAnswers();
    $surveys = getAllSurveys();
?>
<main>
    <section class="container">

        <div class="row mt-5">
            <div id="survey_message" class="mb-2"></div>
            <div class="col-lg-3 mb-2 mb-lg-0">
                <div class="d-grid"><a href="admin.php?page=add-question"
class="btn btn-sm btn-primary">Add question</a></div>
            </div>
            <div class="col-lg-3">
                <div class="d-grid"><a href="admin.php?page=add-answer"
class="btn btn-sm btn-primary">Add a answer</a></div>
            </div>
        </div>
        <div class="row mt-2">
            <div class="col-lg-8 mb-2 mb-lg-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Question</th>
                                <th scope="col">Exire at</th>
                                <th scope="col">Created at</th>
                                <th scope="col">Updated at</th>
                                <th scope="col">Edit</th>
                                <th scope="col">Delete</th>
                            </tr>
                        </thead>
                        <tbody id="surveys">
                            <?php foreach($surveys as $index => $survey):?>
                                <tr id="survey_<?=$index?>">
                                    <th scope="row"><?=$index + 1?></th>
                                    <td><?=$survey->questionName?></td>

```

```
 <?= date("d/m/Y", strtotime($survey- >expire_date))?></td>  <?= date("d/m/Y H:i:s", strtotime($survey->created_at))?></td>  <?= $survey->updated_at != null ? date("d/m/Y H:i:s", strtotime($survey->updated_at)) : "-"?></td>  <button class="btn btn-sm btn-success btn-edit-survey" type="button" data-id="<?=$survey->id?>" data-index="<?=$index?>">Edit</button></td>  <button class="btn btn-sm btn-danger btn-delete-survey" type="button" data-id="<?=$survey- >id?>" data-index="<?=$index?>" data-active="<?=$survey- >is_active?>"><?=$survey->is_active == 1 ? "Delete" : "Activate"?></button> </td> </tr> <?php endforeach;?> </tbody> </table> </div> </div> <div class="col-lg-4"> <form action="#"> <input type="hidden" name="survey_id" id="survey_id"> <input type="hidden" name="survey_index" id="survey_index"> <div class="mb-3"> <label for="date" class="mb-1">Date</label> <input type="date" name="date" id="date" class="form- control mb-1"> <div id="date_error"></div> </div> <div class="mb-3"> <label for="question" class="mb-1">Question</label> <select name="question" id="question" class="form- select"> <option value="0">Izaberite</option> <?php foreach($questions as $question):?> <option value="<?=$question- >id?>"><?=$question->name?></option> <?php endforeach;?> </select> | | | | |
```

```

        <div id="question_error"></div>
    </div>
    <div class="mb-3">
        <label for="answers" class="mb-1">Answers</label>
        <?php foreach($answers as $answer):?>
            <div class="form-check mb-1">
                <input class="form-check-input" type="checkbox"
value="<?=$answer->id?>" id="answer_<?=$answer->id?>" name="answers">
                <label class="form-check-label"
for="answer_<?=$answer->id?>">
                    <?=$answer->name?>
                </label>
            </div>
            <?php endforeach;?>
            <div id="answer_error"></div>
        </div>
        <div class="d-grid gap-1">
            <button class="btn btn-sm btn-primary"
id="btnSaveSurvey" name="btnSaveSurvey" type="button">Save</button>
            <button class="btn btn-sm btn-danger"
id="btnResetSurvey" name="btnResetSurvey" type="button">Reset</button>
        </div>
    </form>
</div>
</div>
</section>
</main>

```

includes/pages/admin/questions.php

```

<?php
    if(!isset($_SESSION['user']) || isset($_SESSION['user']) &&
$_SESSION['user']->role_id != 1)
        header("Location: index.php?page=errors&code=403");
    $questions = getAllQuestions();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div id="question_response_message" class="mb-2"></div>
            <div class="col-lg-8 mb-2 mb-lg-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <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">Edit</th>
        <th scope="col">Delete</th>
    </tr>
</thead>
<tbody id="questions">
    <?php foreach($questions as $index => $question) :?>
        <tr id="question_<?=$index?>">
            <th scope="row"><?=$index + 1?></th>
            <td><?=$question->name?></td>
            <td><?=date('d/m/Y H:i:s', strtotime($question-
>created_at));?></td>
            <td><?=$question->updated_at != null ?
date('d/m/Y H:i:s', strtotime($question->updated_at)) : '-'?></td>
            <td><button class="btn btn-sm btn-success btn-
edit-question" type="button" data-id="<?=$question->id?>" data-
index="<?=$index?>">Edit</button></td>
            <td><button class="btn btn-sm btn-danger btn-
delete-question"
                                type="button" data-index="<?=$index?>" data-
status = "<?=$question->is_deleted?>" data-id="<?=$question->id?>"
                                <?=$question->is_deleted == 0 ? "Delete" :
"Activate"?></button></td>
        </tr>
    <?php endforeach;?>
</tbody>
</table>
</div>
</div>
<div class="col-lg-4">
    <form action="#">
        <input type="hidden" name="question_index"
id="question_index">
        <input type="hidden" name="question_id" id="question_id">
        <div class="mb-3">
            <label for="question" class="mb-1">Question</label>
            <input type="text" name="question" id="question"
class="form-control">
            <div id="question_error"></div>
        </div>
        <div class="d-grid gap-1">

```

```

        <button class="btn btn-sm btn-primary"
id="btnSaveQuestion" type="button" name="btnSaveQuestion">Save</button>
        <button class="btn btn-sm btn-danger"
id="btnResetQuestion" type="reset" name="btnResetQuestion">Reset</button>
    </div>
</form>
</div>
</div>
</section>
</main>

```

includes/pages/admin/answers.php

```

<?php
    if(!isset($_SESSION['user']) || isset($_SESSION['user']) && $_SESSION['user']->role_id != 1)
    header("Location: index.php?page=errors&code=403");
    $answers = getAllAnswers();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div id="answer_response_message" class="mb-2"></div>
            <div class="col-lg-8 mb-2 mb-lg-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Name</th>
                                <th scope="col">Created at</th>
                                <th scope="col">Updated at</th>
                                <th scope="col">Edit</th>
                                <th scope="col">Delete</th>
                            </tr>
                        </thead>
                        <tbody id="answers">
                            <?php foreach($answers as $index=>$answer):?>
                                <tr id="answer_<?=$index?>">
                                    <th scope="row"><?=$index + 1?></th>
                                    <td><?=$answer->name?></td>
                                    <td><?= date('d/m/Y H:i:s',
strtotime($answer->created_at))></td>
                                    <td><?= $answer->updated_at != null ?
date("d/m/Y H:i:s", strtotime($answer->updated_at)):"-"?></td>

```



```

                <td><button class="btn btn-sm btn-success
btn-edit-answer" type="button" data-id="<?=$answer->id?>" data-
index="<?=$index?>">Edit</button></td>
                <td><button class="btn btn-sm btn-danger btn-
delete-answer" type="button" data-id ="<?=$answer->id?>" data-index="<?=$index?>"
data-status="<?=$answer->is_deleted?>"><?=$answer->is_deleted == 0 ? "Delete" :
"Activate"?></button></td>
            </tr>
        <?php endforeach;?>
    </tbody>
</table>
</div>
</div>
<div class="col-lg-4">
    <form action="#">
        <input type="hidden" name="answer_id" id="answer_id">
        <input type="hidden" name="answer_index" id="answer_index">
        <div class="mb-3">
            <label for="answer" class="mb-1">Answer</label>
            <input type="text" name="answer" id="answer" class="form-
control mb-1">
            <div id="answer_error"></div>
        </div>
        <div class="d-grid gap-1">
            <button class="btn btn-sm btn-primary" id="btnSaveAnswer"
type="button" name="btnSaveAnswer">Save</button>
            <button class="btn btn-sm btn-danger" id="btnResetAnswer"
type="button" name="btnResetAnswer">Reset</button>
        </div>
    </form>
</div>
</div>
</section>
</main>

```

includes/pages/admin/posts.php

```

<?php

if(!isset($_SESSION['user']) || isset($_SESSION['user']) && $_SESSION['user']-
>role_id != 1)
header("Location: index.php?page=errors&code=403");

$categories = getAvailableCategories();
$tours = getAllTours();
$pagination = getAllToursPagination();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div class="mb-2" id="tour_response_message"></div>
            <div class="col-lg-8 mb-2 mb-g-0">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Naziv</th>
                                <th scope="col">Cena</th>
                                <th scope="col">Broj dana</th>
                                <th scope="col">Kategorije</th>
                                <th scope="col">Dodaj termine</th>
                                <th scope="col">Datum kreiranja</th>
                                <th scope="col">Datum izmene</th>
                                <th scope="col">Izmeni</th>
                                <th scope="col">Obrisi</th>
                            </tr>
                        </thead>
                        <tbody id="tours">

                        </tbody>
                    </table>
                </div>
                <div class="d-flex justify-content-center mt-2">
                    <nav aria-label="Page navigation example">
                        <ul class="pagination" id="tour-pages">

                        </ul>
                    </nav>
                </div>
            </div>
            <div class="col-lg-4">

```

```

        <form action="#">
            <input type="hidden" name="tour_index" id="tour_index">
            <input type="hidden" name="tour_id" id="tour_id">
            <div class="mb-3">
                <label for="name" class="mb-1">Naziv</label>
                <input type="text" name="name" id="name" class="form-
control">

                <div id="name_error"></div>
            </div>
            <div class="mb-3">
                <label for="price" class="mb-1">Cena</label>
                <input type="number" name="price" id="price" class="form-
control mb-1">

                <div id="price_error"></div>
            </div>
            <div class="mb-3">
                <label for="image" class="mb-1">Slika</label>
                <input type="file" name="image" id="image" class="form-
control">

                <div id="image_error"></div>
            </div>
            <div class="mb-3">
                <label for="duration" class="mb-1">Trajanje (u
daniminutima)</label>
                <input type="number" name="duration" id="duration"
class="form-control">

                <div id="duration_error"></div>
            </div>
            <div class="mb-3">
                <label for="category" class="mb-1">Kategorija</label>
                <div class="row">
                    <?php foreach($categories as $category):?>
                        <div class="col-4">
                            <div class="form-check">
                                <input class="form-check-input"
type="checkbox" value="<?=$category->id?" id="category_<?=$category->id?"
name="categories">

                                <label class="form-check-label"
for="category_<?=$category->id?">

                                    <?=$category->name?>
                                </label>
                            </div>
                        </div>
                    <?php endforeach;?>
                </div>
            </div>
        </form>

```

```

        <div id="category_error"></div>
    </div>
    <div class="mb-3">
        <label for="short_description" class="mb-1">Kratak
opis</label>
        <textarea name="short_descption" id="short_description"
class="form-control mb-1"></textarea>
        <div id="short_description_error"></div>
    </div>
    <div class="mb-3" >
        <label for="tour_description" class="mb-1">Opis
putovanja</label>
        <textarea name="tour_description" id="tour_description"
class="form-control"></textarea>
        <div id="tour_description_error"></div>
    </div>
    <div class="mb-3">
        <img src="" alt="" id="tour_img" class="d-none">
    </div>
    <div class="d-grid gap-1">
        <button class="btn btn-sm btn-primary"
id="btnSaveTour" type="button">Save</button>
        <button class="btn btn-sm btn-danger"
id="btnResetTour" type="button">Reset</button>
    </div>
</form>
</div>
</div>
</section>
</main>

```

includes/pages/admin/dates.php

```

<?php

    if(!isset($_SESSION['user']) || isset($_SESSION['user']) &&
$_SESSION['user']->role_id != 1)
        header("Location: index.php?page=errors&code=403");

    $id =$_GET['id'];
    $tourName = getTourName($id);
    $dates= getTourDates($id);

?>
<main>

```

```

<section class="container">
  <div class="row mt-5">
    <div class="col mb-3" id="date_response_message"></div>
  </div>
  <div class="row mt-2">
    <h2><?=$tourName->name?></h2>
    <hr>
    <div class="col-lg-8 mb-2 mb-lg-0">
      <div class="table-responsive-sm table-responsive-md">
        <table class="table text-center align-middle">
          <thead>
            <tr>
              <th scope="col">#</th>
              <th scope="col">Datum polaska</th>
              <th scope="col">Datum kreiranja</th>
              <th scope="col">Datum izmene</th>
              <th scope="col">Izmeni</th>
              <th scope="col">Izbrisi</th>
            </tr>
          </thead>
          <tbody id="dates">
            <?php foreach($dates as $index => $date):?>
              <tr id="date_<?=$index?>">
                <th scope="row"><?=(int)$index +1?></th>
                <td><?=date("d/m/Y", strtotime($date-
>date))?></td>
                <td><?= date("d/m/Y H:i:s", strtotime($date-
>created_at))?></td>
                <td><?= $date->updated_at != null ?
date("d/m/Y H:i:s", strtotime($date->updated_at)):"-"?></td>
                <td><button class="btn btn-sm btn-success
btn-edit-date" type="button" data-id="<?=$date->id?>" data-
index="<?=$index?>">Edit</button></td>
                <td><button class="btn btn-sm btn-danger btn-
delete-date" type='button' data-id="<?=$date->id?>" data-index="<?=$index?>"
data-status="<?=$date->is_deleted?>"><?=$date->is_deleted == 0 ? "Delete"
:"Activate"?></button></td>
              </tr>
            <?php endforeach;?>
          </tbody>
        </table>
      </div>
    </div>
    <div class="col-lg-4">

```

```

        <input type="hidden" name="tour_id" id="tour_id"
value="<?=$id?>">
        <input type="hidden" name="date_index" id="date_index">
        <input type="hidden" name="date_id" id="date_id">
        <div class="mb-3">
            <label for="date" class="mb-1">Date</label>
            <input type="date" name="date" id="date" class="form-control"
class="mb-1">
            <div id="date_error"></div>
        </div>
        <div class="d-grid gap-1">
            <button class="btn btn-sm btn-primary" id="btnSaveDate"
type="button">Save</button>
            <button class="btn btn-sm btn-danger" id="btnResetDate"
type="button">Reset</button>
        </div>
    </div>
</div>
</section>
</main>

```

includes/pages/admin/messages.php

```

<?php
    if(!isset($_SESSION['user']) || isset($_SESSION['user']) &&
$_SESSION['user']->role_id != 1)
        header("Location: index.php?page=errors&code=403");
    $messages = getAllMessages();
    $messagePage = messagePagination();
?>
<main>
    <section class="container">
        <div class="row mt-5">
            <div class="mt-5" id="message_response_message"></div>
            <div class="col">
                <div class="table-responsive-sm table-responsive-md">
                    <table class="table text-center align-middle">
                        <thead>
                            <tr>
                                <th scope="col">#</th>
                                <th scope="col">Full name</th>
                                <th scope="col">Email</th>
                                <th scope="col">Arrived at</th>
                                <th scope="col">Read</th>
                            </tr>

```

```

        </thead>
        <tbody id="messages">
            <?php foreach($messages as $index=>$message):?>
                <tr id="message_<?=$index?>">
                    <th scope="row"><?=$index + 1?></th>
                    <td><?=$message->first_name.' '.<?=$message->last_name?></td>
                    <td><?=$message->email?></td>
                    <td><?=date('d/m/Y H:i:s', strtotime($message->created_at))?></td>
                    <td>
                        <button type="button" class="btn btn-primary btn-sm btn-read-message" data-bs-toggle="modal" data-bs-target="#exampleModal" data-id="<?=$message->id?>">Read</button>
                    </td>
                </tr>
            <?php endforeach;?>
        </tbody>
    </table>
</div>
<div class="d-flex justify-content-center mt-2">
    <nav aria-label="Page navigation example">
        <ul class="pagination" id="message-pages">
            <?php for($i = 0; $i < $messagePage; $i++):?>
                <li class="page-item"><a class="message-page page-link" <?=$i==0 ? 'active' : ''?>" href="#" data-limit = "<?=$i?>"><?=$i+1?></a></li>
            <?php endfor?>
        </ul>
    </nav>
</div>
</div>
</div>
</section>
</main>
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="exampleModalLabel">Message</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <p class="mb-2">

```



```

        <button class="btn btn-sm btn-primary" id="btnLogin"
type="button" name="btnLogin">Log in</button>
        <div class="d-flex justify-content-center gap-2">
            <span>Nemate nalog?</span>
            <a href="index.php?page=register" class="nav-
item">Registrujte se</a>
        </div>
    </div>
</form>
</div>
</div>
</section>
</main>

```

```

includes/pages/auth/register.php <?php
    if(isset($_SESSION['user'])){
        header("Location:index.php?page=errors&code=403");
    }
?>
<main>
    <section class="mt-5">
        <div class="container">
            <div class="row">
                <div class="col-lg-4 mx-auto">
                    <div id="register_response_message"></div>
                    <h2 class="text-center mb-2">Registruj se</h2>
                    <hr>
                    <form action="#">
                        <div class="mb-3">
                            <label for="first_name" class="mb-1">First
name</label>
                            <input type="text" name="first_name" id="first_name"
class="form-control mb-1">
                            <div id="first_name_error"></div>
                        </div>
                        <div class="mb-3">
                            <label for="last_name" class="mb-1">Last name</label>
                            <input type="text" name="last_name" id="last_name"
class="form-control">
                            <div id="last_name_error"></div>
                        </div>
                        <div class="mb-3">
                            <label for="email" class="mb-1">Email</label>

```

```

        <input type="email" name="email" id="email"
class="form-control">
        <div id="email_error"></div>
    </div>
    <div class="mb-3">
        <label for="password" class="mb-1">Password</label>
        <input type="password" name="password" id="password"
class="form-control">
        <div id="password_error"></div>
    </div>
    <div class="d-grid gap-2">
        <button class="btn btn-sm btn-primary"
id="btnRegister" name="btnRegister">Registruj se</button>
        <div class="d-flex justify-content-center gap-2">
            <span>Vec imate nalog?</span>
            <a href="index.php?page=login" class="nav-
item">Log in</a>
        </div>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
</section>
</main>

```

models/functions.php

```

<?php
// global

define("ADMIN_OFFSET",5);
define("USER_OFFSET",12);

function getOne($query){
    global $connection;
    return $connection->query($query)->fetch();
}

function getAll($query){
    global $connection;
    return $connection->query($query)->fetchAll();
}

function changeStatus($table, $status, $id){

```

```

    global $connection;
    $status = $status == 0 ? 1 : 0;
    $query = "update $table set is_deleted = ? where id =?";
    $update = $connection->prepare($query);
    $update->execute([$status, $id]);
}

function checkEmail($email){
    global $connection;
    return $connection->query("select id from users where email = '$email'")->fetch();
}

function checkAccount($email, $password){
    global $connection;
    $query = "select id, role_id from users where email = ? and password = ?";
    // select id, role_id from users where email = ? and password = ?
    $result = $connection->prepare($query);
    $result->execute([$email, md5($password)]);
    return $result->fetch();
}

function createNewAccount($first_name, $last_name, $email, $password){
    global $connection;
    define("USER_ROLE_ID",2);
    $query = "insert into users (first_name, last_name, email, password, role_id)
values(?,?,?,?,?)";
    $insert = $connection->prepare($query);
    $insert->execute([$first_name, $last_name, $email, md5($password),
(int)USER_ROLE_ID]);
}

// categories
function getAllCategories(){
return getAll('select * from categories');}

function getCategory($id){
    return getOne("select id, name from categories where id = '$id'");
}

function getCategoryFullRow($id){
    return getOne("select * from categories where id = '$id'");
}

function checkCategoryName($name){

```

```

        return getOne("select id, name from categories where name = '$name'");
    }
function storeNewCategory($name){
    global $connection;
    $query = "insert into categories (name) values(?)";
    $insert = $connection->prepare($query);
    $insert->execute([$name]);
}

function updateCategory($id, $name){
    global $connection;
    $date = date("Y-m-d H:i:s");
    $query = "update categories set name =? , updated_at =? where id = ?";
    $update = $connection->prepare($query);
    $update->execute([$name, $date, $id]);
}

// tours
function getAvailableCategories(){
    return getAll("select id, name from categories");
}

function getAllTours($limit = 0){
    $queryTours = "select * from tours";

    $offset = (int) ADMIN_OFFSET;
    $limit = (int)$limit * $offset;
    $queryLimit = " limit $limit, $offset";
    $queryTours.= $queryLimit;
    $tours = getAll($queryTours);
    foreach($tours as $tour){
        $queryCategory = "select name from categories
        c join category_tour ct on c.id = ct.category_id
        where ct.tour_id = $tour->id";
        $tour->categories = getAll($queryCategory);
    }
    return $tours;
}

function getAllToursPagination(){
    $query = "select count(*) as numberOfElements from tours";
    $data = getOne($query);
    return ceil($data->numberOfElements / ADMIN_OFFSET);
}

```

```

function getOneTour($id){
    $query = "select id, name, short_description, long_description, image_path,
days, price from tours where id = '$id'";
    $tour = getOne($query);
    $categoryQuery = "select category_id from category_tour where tour_id = '$id'";
    $tour->categories = getAll($categoryQuery);
    return $tour;
}

function getOneTourFullRow($id){
    $query = "select id, name, days, price, created_at, updated_at, is_deleted
from tours where id = '$id'";
    $tour = getOne($query);
    $categoryQuery = "select name from category_tour ct join categories c on
ct.category_id = c.id where ct.tour_id = '$id'";
    $tour->categories = getAll($categoryQuery);
    return $tour;
}

function checkTourName($name){
    $query = "select id, name from tours where name = '$name'";
    return getOne($query);
}

function insertTour($name, $short_description, $long_description, $image, $days,
$price, $categories){
    global $connection;
    $queryInsert = "insert into tours (name, short_description, long_description,
image_path, days, price ) values(?,?,?,?,?,?)";
    $insert = $connection->prepare($queryInsert);
    $insert->execute([$name, $short_description, $long_description, $image,
$days, $price]);
    $id = $connection->lastInsertId();
    insertIntoPivotTable('category_tour', 'category_id', 'tour_id', $categories,
$id);
}

function updateTour($name, $shortDescription, $longDescription, $days, $price,
$categories, $id, $image = ""){
    global $connection;
    $date = date("Y-m-d H:i:s");

```

```

        $query = "update tours set name = ?, short_description = ?,
long_description = ?, ";
        $dataArray = [];
        array_push($dataArray, $name);
        array_push($dataArray, $shortDescription);
        array_push($dataArray, $longDescription);

        if($image != ""){
            $query.= "image_path = ?,";
            array_push($dataArray, $image);
        }
        $query.= "days =? , price =?, updated_at =? where id =?";

        array_push($dataArray, $days);
        array_push($dataArray, $price);
        array_push($dataArray, $date);
        array_push($dataArray, $id);
        $data = $connection->prepare($query);
        $data->execute($dataArray);
        deleteData('category_tour','tour_id', $id);
        insertIntoPivotTable('category_tour', 'category_id','tour_id', $categories,
$id);
    }

function insertIntoPivotTable($table, $columnOne, $columnTwo, $valueArray, $id){
    global $connection;
    $params = [];
    $values = [];
    foreach($valueArray as $value){
        $params[] = "(?,?)";
        $values[] = $value;
        $values[] = $id;
    }
    $queryInsert = "insert into $table ($columnOne, $columnTwo)
values".implode(',', $params);
    $insert = $connection->prepare($queryInsert);
    $insert->execute($values);
}

function deleteData($table, $columnWhere, $columnData){
    global $connection;
    $query = "delete from $table where $columnWhere = ?";
    $delete = $connection->prepare($query);

```

```

    $delete->execute([$columnData]);
}

// dates
function getTourName($tour_id){
    return getOne("select name from tours where id = '$tour_id'");
}
function getTourDates($tour_id){
    return getAll("select * from tour_dates where tour_id = '$tour_id'");
}
function getOneDate($id){
    return getOne("select id, date from tour_dates where id = '$id'");
}
function getOneDateFullRow($id){
    return getOne("select * from tour_dates where id = '$id'");
}
function checkTourDate($date, $tour_id){
    $query = "select id, date, tour_id from tour_dates where tour_id= '$tour_id'
and date= '$date' " ;
    return getOne($query);
}

function insertTourDate($date, $tour_id){
    global $connection;
    $query = "insert into tour_dates (tour_id, date) values(?,?)";
    $insert = $connection->prepare($query);
    $insert->execute([$tour_id, $date]);
}

function updateTourDate($date, $id){
    global $connection;
    $date_update= date("Y-m-d H:i:s");
    $queryUpdate = "update tour_dates set  date = ?, updated_at  = ? where id
=?";

    $update = $connection->prepare($queryUpdate);
    $update->execute([$date, $date_update, $id]);
}

// questions
function getAllQuestions(){
    return getAll('select * from questions');
}

```

```

function getOneQuestion($id){
    return getOne("select id, name from questions where id = '$id'");
}

function getQuestionFullRow($id){
    return getOne("select * from questions where id = '$id' ");
}

function checkQuestion($name){
    return getOne("select id, name from questions where name = '$name'");
}

function insertQuestion($name){
    global $connection;
    $query = "insert into questions (name) values(?)";
    $insert = $connection->prepare($query);
    $insert->execute([$name]);
}

function updateQuestion($name, $id){
    global $connection;
    $date = date("Y-m-d H:i:s");
    $query = "update questions set name =?, updated_at =? where id = ?";
    $update = $connection->prepare($query);
    $update->execute([$name, $date, $id]);
}

// messages
function storeNewMessage($first_name, $last_name, $email, $message){
    global $connection;
    $query = "insert into messages (first_name, last_name, email, message)
        values (?, ?, ?, ?)";
    $insert = $connection->prepare($query);
    $insert->execute([$first_name, $last_name, $email, $message]);
}

function getAllMessages($limit = 0){
    $baseQuery = "select * from messages";
    $orderBy = " ORDER BY created_at desc";
    $offset = (int)ADMIN_OFFSET;
    $limit = (int)$limit * $offset;
    $queryLimit = " limit $limit, $offset";

    $query = $baseQuery.$orderBy.$queryLimit;
    return getAll($query);
}

```



```

}

function messagePagination(){
    $query = "select count(*) as numberOfElements from messages";
    $elements = getOne($query);
    return ceil($elements->numberOfElements/ADMIN_OFFSET);
}

function getOneMessage($id){
    return getOne("select first_name, last_name, email, message, created_at from
messages where id = '$id'");
}

// answers

function getAllAnswers(){
    return getAll('select * from answers');
}

function getOneAnswer($id){
    return getOne("select id, name from answers where id = '$id'");
}

function getAnswerFullRow($id){
    return getOne("select * from answers where id = '$id'");
}

function checkAnswers($name){
    return getOne("select id, name from answers where name = '$name'");
}

function insertNewOption($name){
    global $connection;
    $query="insert into answers (name) values(?)";
    $insert = $connection->prepare($query);
    $insert->execute([$name]);
}

function updateOption($name, $id){
    global $connection;
    $date = date("Y-m-d H:i:s");
    $query = "update answers set name =?, updated_at =? where id =?";

```

```

$update= $connection->prepare($query);
$update->execute([$name, $date,$id]);
}

// survy
function getAvailableQuestions(){
    $query = "select id, name from questions where is_deleted = 0";
    return getAll($query);
}

function getAvailableAnswers(){
    $query = "select id, name from answers where is_deleted =0";
    return getAll($query);
}

function checkActiveSurvey(){
    $query = "select * from survey where is_active = 1";
    return getAll($query);
}

function updateStatusSurvey(){
    global $connection;
    $is_active = 0;
    $query = "update survey set is_active = ?";
    $updateQuery = $connection->prepare($query);
    $updateQuery->execute([$is_active]);
}

function insertSurvey($date, $question, $answers){
    global $connection;
    if(checkActiveSurvey()){
        updateStatusSurvey();
    }else{
        $query = "insert into survey (expire_date, question_id) values(?,?)";
        $insert = $connection->prepare($query);
        $insert->execute([$date, $question]);
        $last_id = $connection->lastInsertId();
        insertIntoSurveyAnswer($last_id, $answers);
    }
}

function insertIntoSurveyAnswer($survey_id, $answers){
    global $connection;
    $query = "insert into survey_answers (survey_id, answer_id)";
    $surveyParam = [];

```

```

    $surveyValues = [];
    foreach($answers as $answer){
        $surveyParam[] = "(?,?)";
        $surveyValues[] = (int)$survey_id;
        $surveyValues[] = (int)$answer;
    }
    $query.= " values".implode(',',$surveyParam);
    $insert = $connection->prepare($query);
    $insert->execute($surveyValues);
}

function getAllSurveys(){
    return getAll('select s.*, q.name as questionName
    from survey s join questions q on s.question_id = q.id');
}

function getOneSurvey($survey_id){
    $query = "select id, question_id, expire_date from survey where id=
'$survey_id'";
    $survey = getOne($query);
    $surveyOptionsQuery = "select answer_id from survey_answers where survey_id =
'$survey->id'";
    $survey->answers = getAll($surveyOptionsQuery);
    return $survey;
}

function updateSurvey($survey_id, $date, $question, $answers){
    global $connection;
    $update_date = date("Y-m-d H:i:s");
    $queryUpdate = "update survey set expire_date = ? , question_id = ?,
updated_at =?
    where id = ?";
    $update = $connection->prepare($queryUpdate);
    $update->execute([$date, $question, $update_date, $survey_id]);
    deleteData('survey_answers', 'survey_id', $survey_id);
    insertIntoSurveyAnswer($survey_id, $answers);
}

function getSurveyOneFullRow($id){
    return getOne("select s.*, q.name as questionName
    from survey s join questions q on s.question_id = q.id where s.id = '$id'");
}

function checkIfIsAnyActive(){

```

```

    $query = "select count(*) as numberOfSurveys from survey where is_active =
1";
    return getOne($query);
}

function changeStatusSurvey($id, $status){
    global $connection;
    $status = $status == 1 ? 0 : 1;
    $query = "update survey set is_active =? where id =?";
    $update = $connection->prepare($query);
    $update->execute([$status, $id ]);
}

// others
function numberOfRegisteredUsers(){
    $query = "select count(*) as numberOfUsers from users where role_id = 2";
    return getOne($query);
}

function numberOfTours(){
    $query = "select count(*) as numberOfTours from tours where is_deleted = 0";
    return getOne($query);
}

//
function getAvailableTours($limit = 0, $search = '', $categories= ''){
    // SELECT * from tours t join category_tour ct on t.id = ct.tour_id where
ct.category_id in (1);
    $queryWhere = " where is_deleted = 0";
    $baseQuery = "select t.id, t.name, t.image_path from tours t ";
    if($search != ""){
        $compareString = trim("%$search%");
        $queryWhere.=" and name like '$compareString'";
    }
    if($categories != ""){
        $baseQuery.=" join category_tour ct on t.id = ct.tour_id";
        $queryWhere .= " and ct.category_id in (". $categories .")";
    }
    $offset = (int)USER_OFFSET;
    $limit = (int)$limit * $offset;
    $queryLimit = " limit $limit, $offset";
    $query = $baseQuery.$queryWhere.$queryLimit;
    // return $query;
    return getAll($query);
}

```

```

function getAvaliableToursPagination($search="", $categories=""){
    $baseQuery = "select count(*) as numberOfElements from tours t";
    $queryWhere = " where is_deleted = 0";
    if($search != ""){
        $compareString = trim("%$search%");
        $queryWhere .= " and t.name like '$compareString'";
    }
    if($categories != ""){
        $baseQuery.= " join category_tour ct on t.id = ct.tour_id";
        $queryWhere .= " and ct.category_id in (". $categories .")";
    }
    $query = $baseQuery.$queryWhere;
    $data = getOne($query);
    return ceil($data->numberOfElements / USER_OFFSET);
}

function getTheMostVisitedTours(){
    $query = "select COUNT(*) as mostVisitedTours, t.name, td.date from
reservations utd join tour_dates td on utd.tour_date_id = td.id
join tours t on td.tour_id = t.id GROUP BY tour_date_id ORDER BY
mostVisitedTours DESC";
    return getAll($query);
}

// user
function getAvailableTourDate($tour_id){
    $query = "select id, date from tour_dates where is_deleted = 0 and tour_id =
'$tour_id' ORDER BY date ASC";
    return getAll($query);
}

function getTourData($tour_id){
    $query = "select * from tours where id= '$tour_id'";
    return getOne($query);
}

// nav fucntions
function getUserMenu(){
    return getAll('select * from user_menu');
}

function getAdminMenu(){
    return getAll("select * from admin_menu");
}

```

```

function storeNewReservation( $tour_date, $user_id){
    global $connection;
    $query = "insert into reservations (tour_date_id, user_id) values(?,?)";
    $insert = $connection->prepare($query);
    $insert->execute([$tour_date, $user_id]);
}

function getAvailableQuestion(){
    $query = "select s.id, q.name from survey s join questions q on s.question_id
= q.id where q.is_deleted = 0";
    return getOne($query);
}

function getSurveyOptions($survey_id){
    $query = "select sa.answer_id, a.name from survey_answers sa join answers a on
sa.answer_id = a.id where sa.survey_id = '$survey_id'";
    return getAll($query);
}

function checkIfUserVote($user_id, $survey_id){
    $query = "select * from survey_vote where user_id = '$user_id' and
id_survey='$survey_id'";
    return getOne($query);
}

function insertVote($survey_id, $option_id, $user_id){
    global $connection;
    $queryInsert = $connection->prepare("insert into survey_vote (id_survey,
id_answer, user_id) values(?,?,?) ");
    $queryInsert->execute([$survey_id,$option_id, $user_id]);
}

function getMainTours(){
    $baseQuery = "select t.id, t.name, t.image_path from tours t limit 4 ";
    return getAll($baseQuery);
}

```

[models/getData.php](#)

```

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

```

```

        echo json_encode([
            'numberOfUsers' => numberOfRegisteredUsers(),
            'numberOfTours' => numberOfTours(),
            'mostVisitedTours' => getTheMostVisitedTours()
        ]);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
} else http_response_code(404);

```

models/utilities.php

```

<?php
function inputValidation(&$errorArray, $errorMessage, $input, $regInput = ''){
    if(str_contains($input, "@")){
        if(!filter_var($input, FILTER_VALIDATE_EMAIL))
            array_push($errorArray , $errorMessage);
    }else{
        if(!preg_match($regInput, $input))
            array_push($errorArray , $errorMessage);
    }
}

function printErrorsMessage($errorArray){
    foreach($errorArray as $error)
    {
        echo json_encode($error);
        http_response_code(422);
    }
}

function checkBoxValidation($checkBoxArray, &$errorsArray, $errorMessage){
    if(count($checkBoxArray) == 0){
        array_push($errorsArray, $errorMessage);
    }
}

function imageFileValidation($image, &$errorsArray, $errorMessage){
    list($imageEmptyError, $imageTypeError, $imageSizeError) = $errorMessage;
    $image_size = $image['size'];
    $image_type = $image['type'];

    $image_type_array = ["image/png","image/jpeg","image/jpg"];

```

```

        if($image == "")
            array_push($errorsArray, $imageEmptyError);
        else if($image_size > 3 * 1024 * 1024)
            array_push($errorsArray, $imageSizeError);
        else if(!in_array($image_type, $image_type_array))
            array_push($errorsArray, $imageTypeError);
    }

    function uploadImage($image){
        $image_name = $image["name"];
        $image_tmp = $image["tmp_name"];

        $new_image_name = time()." ".$image_name;
        $image_path = "../assets/img/$new_image_name";
        move_uploaded_file($image_tmp, $image_path);
        return $new_image_name;
    }

    function validateInputDate($date, &$errorArray, $errorMessages){
        $currentDate = date("Y-m-d");
        $currentDateTimeStamp = strtotime($currentDate);
        $myDateTimeStamp = strtotime($date);

        list ($emptyDate, $invalidDate) = $errorMessages;
        if($date==""){
            array_push($errorArray, $emptyDate);
        }else if($currentDateTimeStamp > $myDateTimeStamp){
            array_push($errorArray, $invalidDate);
        }
    }

    function validateInputSelect($select, &$errors, $errorMessage){
        if($select == 0 || $select=="choose")
            array_push($errors, $errorMessage);
    }

    function radioINputValidation($radio, &$errors, $errorMessage){
        if($radio == null)
            array_push($errors, $errorMessage);
    }

```


models/validations.php

```
<?php

include 'utilities.php';

function registerFormValidation($first_name, $last_name, $email, $password){
    $errors = [];

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

    inputValidation($errors, "Ime nije u redu", $first_name, $reFirstLastName);
    inputValidation($errors, "Prezime nije u redu", $last_name, $reFirstLastName);
    inputValidation($errors, "Email nije u redu", $email);
    inputValidation($errors, "Lozinka nije u redu", $password, $rePassword);

    return $errors;
}

function loginFormValidation($email, $password){
    $errors = [];

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

    inputValidation($errors, "Email nije validan", $email);
    inputValidation($errors, "Lozinka nije validna", $password, $rePassword);

    return $errors;
}

function categoryFormValidation($name){
    $errors = [];
    $reName = "/[A-Z][\w\s\-,.!?' ]+/" ;
    inputValidation($errors, "Naziv nije u redu", $name, $reName);
    return $errors;
}

function tourFormValidation($name, $price, $duration, $categories,
$shortDescription, $longDescription, $image= ""){
    $errors = [];

    $reName = "/[A-Z][\w\s\-,.!?' ]+/" ;
    $rePrice = "/^[1-9][\d]{2,}$/";
    $reDuration = "/^[1-9](\[\d]{1,})*$/";
```

```

        inputValidation($errors, "Naziv nije u redu", $name, $reName);
        inputValidation($errors, "Cena nije u redu", $price, $rePrice);
        checkBoxValidation($categories, $errors, "Morate odabrati barem jednu
kategoriju");
        $image != "" ? imageFileValidation($image, $errors, ["Morate odabrati
sliku", "Tip slike nije u redu", "Velicina slikemora biti manja od 3mb"]) : "";
        inputValidation($errors, "Period tranajanja nije u redu", $duration,
$reDuration);
        inputValidation($errors, "Opis nije u redu", $shortDescription, $reName);
        if(strlen($longDescription) == 0){
            array_push($errors, "Opis nije u redu");
        }

        return $errors;
    }

function dateFormValidation($date){
    $errors = [];

    validateInputDate($date, $errors, ["Morate odabrati datum", "Datum nije
validan"]);

    return $errors;
}

function contactFormValidation($first_name, $last_name, $email, $message){
    $errors = [];

    $reFirstLastName = "/^([A-ZÀ-ÿ][-a-z. ']+[ ]*)+$/";
    $reMessage = "/[A-Z][\w\s\-,.!?' ]+/";

    inputValidation($errors, "Ime nije u redu", $first_name, $reFirstLastName);
    inputValidation($errors, "Prezime nije u redu", $last_name,
$reFirstLastName);
    inputValidation($errors, "Email nije validan", $email);
    inputValidation($errors, "Poruka nije u redu", $message, $reMessage);

    return $errors;
}

function questionFormValidation($name){
    $errors = [];
    $reName = "/[A-Z][\w\s\-,.!?' ]+/";
    inputValidation($errors, "Pitanje nije validno", $name, $reName);

```

```

        return $errors;
    }

function optionFormValidation($name){
    $errors = [];
    $reName = "/[A-Z][\w\s\-,.!?' ]+/" ;
    inputValidation($errors, "Odgovor nije validan", $name, $reName);
    return $errors;
}

function surveyFormValidation($date, $question, $answers){
    $errors = [];
    validateInputDate($date, $errors, ["Morate odabrati datum", "Datum nije validan"]);
    validateInputSelect($question, $errors, "Odaberite pitanje");
    checkBoxValidation($answers, $errors, "Morate odabrati barem jedan odgovor");
    return $errors;
}

function tourFormResValidation($date){
    $errors = [];
    validateInputSelect($date, $errors, "Morate odabrati datum polaska");
    return $errors;
}

function surveyVoteValidation($vote){
    $errors= [];
    radioInputValidation($vote, $errors, "Odaberite jednu od opcija");
    return $errors;
}

```

[models/answers/delete.php](#)

```

<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == "POST"){
    $id = $_POST['id'];
    $status = $_POST['status'];
    try {
        require_once '../config/connection.php';
        include '../functions.php';
        changeStatus('answers', $status, $id);
        echo json_encode(getAnswerFullRow($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
    }
}

```

```
        http_response_code(500);
    }
}
else http_response_code(404);
```

[models/answers/getOne.php](#)

```
<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'GET'){
    $id = $_GET['id'];
    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        echo json_encode(getOneAnswer($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
else http_response_code(404);
```

[models/answers/store.php](#)

```
<?php
header("Content-type:application/json");
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $option = $_POST['option'];
    include '../validations.php';
    $optionValidation = optionFormValidation($option);
    if(count($optionValidation) > 0)
        printErrorsMessage($optionValidation);
    else{
        try {
            require_once '../../config/connection.php';
            include '../functions.php';
            $checkAnswers = checkAnswers($option);
            if($checkAnswers){
                echo json_encode("Takav odgovor vec postoji");
                http_response_code(409);
            }else{
                insertNewOption($option);
                echo json_encode([
                    'message' => "Novi odgovor je dodat",
                    'options' => getAllAnswers()
                ]);
            }
        }
    }
}
```

```

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

```

models/answers/update.php

```

<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == "POST"){
    $name= $_POST['option'];
    $id = $_POST['id'];
    include '../validations.php';
    $answerValidation = optionFormValidation($name);
    if(count($answerValidation) > 0)
        printErrorsMessage($answerValidation);
    else{
        try {
            require_once '../config/connection.php';
            include '../functions.php';
            $checkAnswer = checkAnswers($name);
            if($checkAnswer && $checkAnswer->id != $id && $checkAnswer->name ==
$name){
                echo json_encode("Same answer allready exists");
                http_response_code(409);
            }else{
                updateOption($name, $id);
                echo json_encode(getAnswerFullRow($id));
            }
        } catch (PDOException $th) {
            echo json_encode($th->getMessage());
            http_response_code(500);
        }
    }
}
else http_response_code(404);

```

models/auth/register.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'];

    include '../validations.php';
    $registerFormValidation = registerFormValidation($first_name, $last_name,
$email, $password);
    if(count($registerFormValidation) > 0)
        printErrorMessage($registerFormValidation);
    else{
        try {
            require_once '../../config/connection.php';
            include '../functions.php';

            if(checkEmail($email)){
                echo json_encode("Takav email je vec u upotrebi");
                http_response_code(409);
            }else{
                createNewAccount($first_name, $last_name, $email, $password);
                echo json_encode("Novi nalog je kreiran");
                http_response_code(201);
            }
        } catch (PDOException $th) {
            echo json_encode($th->getMessage());
            http_response_code(500);
        }
    }
}
else http_response_code(404);

```

[models/auth/login.php](#)

```

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

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

    include '../validations.php';
    $loginFormValidation = loginFormValidation($email,$password);
    if(count($loginFormValidation) > 0)

```

```

        printErrorMessage($loginFormValidation);
    else{
        try {
            require_once '../../config/connection.php';
            include '../functions.php';

            if(!checkEmail($email)){
                echo json_encode("Nalog sa ovim email-om nepostoji");
                http_response_code(401);
            }else{

                $checkAccount = checkAccount($email, $password);
                if($checkAccount){
                    $_SESSION['user'] = $checkAccount;
                    echo json_encode($checkAccount->role_id);
                }else{
                    echo json_encode("Vasi kredencijali nisu u redu! Pokusajte ponovo");
                    http_response_code(401);
                }
            }

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

```

models/auth/logout.php

```

<?php
session_start();
if($_SESSION['user']){
    unset($_SESSION['user']);
    session_destroy();
    header("Location:../../index.php");
}

```

models/categories/getOne.php

```

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

```

```

        try {
            require_once '../config/connection.php';
            include '../functions.php';
            echo json_encode(getOneCategory($id));
        } 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"){
    $id = $_POST['id'];
    $status = $_POST['status'];

    try {
        require_once '../config/connection.php';
        include '../functions.php';

        changeStatus('categories', $status, $id);
        echo json_encode(getOneCategoryFullRow($id));

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

}
else
    http_response_code(404);

```

models/categories/store.php <?php

```

header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $name = $_POST['name'];
    include '../validations.php';
    $categoryFormValidation = categoryFormValidation($name);
    if(count($categoryFormValidation) > 0)
        printErrorsMessage($categoryFormValidation);
    else{
        try {

```



```

        require_once '../../config/connection.php';
        include '../functions.php';
        $checkCategoryName = checkCategoryName($name);
        if($checkCategoryName){
            echo json_encode("Kategorija sa takvim nazivom vec postoji");
            http_response_code(409);
        }else{
            storeNewCategory($name);
            echo json_encode([
                'categories' => getAllCategories(),
                'message' => "Nova kategorija je dodata"
            ]);
        }
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}

else http_response_code(404);

```

```

models/categories/update.php <?php

header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $name = $_POST['name'];
    $id = $_POST['id'];
    include '../validations.php';
    $categoryFormValidation = categoryFormValidation($name);
    if(count($categoryFormValidation) > 0)
        printErrorsMessage($categoryFormValidation);
    else{
        try {
            require_once '../../config/connection.php';
            include '../functions.php';
            $checkCategoryName = checkCategoryName($name);
            if($checkCategoryName && $checkCategoryName->name == $name &&
            $checkCategoryName->id != $id){
                echo json_encode("Kategorija sa takvim nazivom vec postoji");
                http_response_code(409);
            }else{
                updateCategory($id,$name);
                echo json_encode(getOneCategoryFullRow($id));
            }
        }
    }
}

```

```

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

else http_response_code(404);

```

models/dates/delete.php

```

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

    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        changeStatus('tour_dates', $status, $id);
        echo json_encode(getOneDateFullRow($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}

else http_response_code(404);

```

models/dates/store.php

```

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

    include '../validations.php';
    $dateFormValidation = dateFormValidation($date);
    if(count($dateFormValidation)> 0)
        printErrorsMessage($dateFormValidation);
    else{
        try {

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

```

```

        $checkDate = checkTourDate($date, $tour_id);
        if($checkDate){
            echo json_encode("Tura koja je planirana da se odrzi na taj datum
vec postoji");
            http_response_code(409);
        }else{
            insertTourDate($date, $tour_id);
            echo json_encode([
                'tours' => getTourDates($tour_id),
                'message' => 'Novi termin je dodat'
            ]);
        }
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
}
else http_response_code(404);

```

<models/dates/edit.php>

```

<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'GET'){
    $id = $_GET['id'];
    try {
        require_once '../config/connection.php';
        include '../functions.php';
        echo json_encode(getOneDate($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
else{
    http_response_code(404);
}

```

<models/dates/update.php>

```

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

    $id = $_POST['id'];
    $tour_id = $_POST['tour_id'];
    $date = $_POST['date'];

    include '../validations.php';
    $dateFormValidation = dateFormValidation($date);
    if(count($dateFormValidation) > 0)
        printErrorsMessage($dateFormValidation);
    else{
        try {
            require_once '../../config/connection.php';
            include '../functions.php';
            $checkDate = checkTourDate($date, $tour_id);
            if($checkDate && $checkDate->date == $date && $checkDate->tour_id
== $tour_id && $checkDate->id != $id){
                echo json_encode("Tura koja je planirana da se odrzi na taj
datum vec postoji");
                http_response_code(409);
            }else{
                updateTourDate($date, $id);
                echo json_encode(getOneDateFullRow($id));
            }

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

```

[models/messages/getOne.php](#)

```

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

```

```

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

```

models/messages/pagination.php

```

<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == "GET"){
    $limit = $_GET["limit"];
    try {
        require_once '../config/connection.php';
        include '../functions.php';
        echo json_encode([
            'messages' => getAllMessages($limit),
            'pages' => messagePagination(),
            'limit' => $limit,
            'offset' => ADMIN_OFFSET
        ]);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
else http_response_code(404);

```

models/messages/store.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'];
    $message = $_POST['message'];

    include '../validations.php';
    $contactValidation =
contactFormValidation($first_name,$last_name,$email,$message);
    if(count($contactValidation)> 0)
        printErrorsMessage($contactValidation);
    else{

```

```

        try {
            require_once '../../config/connection.php';
            include '../functions.php';
            storeNewMessage($first_name, $last_name, $email, $message);
            echo json_encode("Hvala sto ste nas kontaktirali");
        } catch (PDOException $th) {
            echo json_encode($th->getMessage());
            http_response_code(500);
        }
    }
}

```

else http_response_code(404);

[models/questions/delete.php](#)

```

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

```

[models/questions/getOne.php](#)

```

<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'GET'){
    $id = $_GET['id'];
    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        echo json_encode(getOneQuestion($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}

```

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

models/question/store.php

```
<?php  
header("Content-type:application/json");  
if($_SERVER["REQUEST_METHOD"] == "POST"){  
    $name = $_POST['name'];  
    include '../validations.php';  
  
    $questionValidation = questionFormValidation($name);  
    if(count($questionValidation)>0)  
        printErrorsMessage($questionValidation);  
    else{  
        try {  
            require_once '../../config/connection.php';  
            include '../functions.php';  
            $checkQuestion = checkQuestion($name);  
            if($checkQuestion){  
                echo json_encode("Takvo pitanje vec postoji");  
                http_response_code(409);  
            }else{  
                insertQuestion($name);  
                echo json_encode([  
                    'questions' => getAllQuestions(),  
                    'message' => 'Novo pitanje je kreirano'  
                ]);  
            }  
        } catch (PDOException $th) {  
            echo json_encode($th->getMessage());  
            http_response_code(500);  
        }  
    }  
}  
}  
else http_response_code(404);
```

models/questions/update.php

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

```

include '../validations.php';
$questionValidation = questionFormValidation($name);
if(count($questionValidation) > 0)
    printErrorMessage($questionValidation);
else{
    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        $checkQuestion = checkQuestion($name);
        if($checkQuestion && $checkQuestion->name == $name && $checkQuestion->id != $id){
            echo json_encode("Takvo pitanje vec postoji");
            http_response_code(409);
        }else{
            updateQuestion($name, $id);
            echo json_encode(getQuestionFullRow($id));
        }
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
}
else http_response_code(404);

```

[models/reservations/store.php](#)

```

<?php
session_start();
header("Content-type:Application/json");
if($_SERVER['REQUEST_METHOD'] == "POST"){
    $user_id = $_SESSION['user']->id;
    $tour_id = $_POST['tour_id'];
    $date = $_POST['date'];

    include '../validations.php';
    $tourReservationFormValidation = tourFormResValidation($date);
    if(count($tourReservationFormValidation)>0)
        printErrorMessage($tourReservationFormValidation);
    else {
        try {
            require_once '../../config/connection.php';
            include '../functions.php';
            storeNewReservation($tour_id, $user_id);

```



```

        echo json_encode("Hvala vam sto ste rezervisali");
        http_response_code(201);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
}
else http_response_code(404);

```

[models/survey/delete.php](#)

```

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

    try {
        require_once '../../config/connection.php';
        include '../functions.php';

        if($status == 0){
            if(checkIfIsAnyActive()->numberOfSurveys == 0){
                changeStatusSurvey($id, $status);
                echo json_encode(getSurveyOneFullRow($id));
            }else{
                echo json_encode('We allready have active survey!');
                http_response_code(409);
            }
        }else{
            changeStatusSurvey($id, $status);
            echo json_encode(getSurveyOneFullRow($id));
        }

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

```

models/survey/store.php

```
<?php
header("Content-type:application/json");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $question = $_POST['question'];
    $date= $_POST['date'];
    $answers = explode(',', $_POST['answers']);

    include '../validations.php';
    $surveyValidation = surveyFormValidation($date, $question, $answers);
    if(count($surveyValidation) > 0)
        printErrorMessage($surveyValidation);
    else{
        try {

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

            $connection->beginTransaction();

            insertSurvey($date, $question, $answers);
            $connection->commit();
            echo json_encode([
                'message' => "Nova anketa je postavljena",
                'surveys' => getAllSurveys()
            ]);
        } catch (PDOException $ex) {
            $connection->rollBack();
            echo json_encode($ex->getMessage());
            http_response_code(500);
        }
    }
}
else http_response_code(404);
```

models/survey/update.php

```
<?php
header("Content-type:application/json");
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $id = $_POST['id'];
    $question = $_POST['question'];
    $date= $_POST['date'];
    $answers = explode(",", $_POST['answers']);
```

```

include '../validations.php';
$surveyValidation = surveyFormValidation($date, $question, $answers );
if(count($surveyValidation) > 0)
    printErrorsMessage($surveyValidation);
else{
    try {
        include '../../config/connection.php';
        include '../functions.php';

        $connection->beginTransaction();
        updateSurvey($id, $date, $question, $answers);
        echo json_encode(getSurveyOneFullRow($id));
        $connection->commit();

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

```

[models/survey/getOne.php](#)

```

<?php
header("Content-type:application/json");
if($_SERVER["REQUEST_METHOD"] == "GET"){
    $id = $_GET['id'];
    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        echo json_encode(getOneSurvey($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
else http_response_code(404);

```

[models/survey/vote.php](#)

```

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

```

```

$survey_id = $_POST['survey_id'];
$option_id = $_POST['radio_value'];
$user_id = $_SESSION['user']->id;

include '../validations.php';
$surveyVoteValidation = surveyVoteValidation($option_id);
if(count($surveyVoteValidation)>0)
    printErrorsMessage($surveyVoteValidation);
else{
    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        insertVote($survey_id, $option_id, $user_id);
        echo json_encode("Hvala sto se glasali");
        http_response_code(201);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
}

```

models/tours/delete.php

```

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

    try {
        require_once '../../config/connection.php';
        include '../functions.php';

        changeStatus('tours', $status, $id);
        echo json_encode(getOneTourFullRow($id));

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

```

models/tours/edit.php

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

    try {
        require_once '../../config/connection.php';
        include '../functions.php';
        echo json_encode(getOneTour($id));
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }
}
else{
    http_response_code(404);
}
```

models/tours/store.php

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

    $name = $_POST['name'];
    $price = $_POST['price'];
    $image = $_FILES['image'];
    $duration = $_POST['duration'];
    $categories = explode(",",$_POST['categories']);
    $short_description = $_POST['short_description'];
    $long_description= $_POST['long_description'];
    $limit = $_POST['limit'];

    include '../validations.php';

    $tourFormValidation = tourFormValidation($name, $price, $duration,
    $categories, $short_description, $long_description, $image);
    if(count($tourFormValidation) > 0)
        printErrorsMessage($tourFormValidation);
    else{
        try {
            require_once '../../config/connection.php';
```



```

header("Content-type:Application/json");
if($_SERVER['REQUEST_METHOD'] == 'GET'){
    $text = isset($_GET['text']) ? $_GET['text'] : '';
    $categories = isset($_GET['selectedCheckboxes']) ? implode(",",
$_GET['selectedCheckboxes']) : '';
    $limit = isset($_GET['limit']) ? $_GET['limit'] : 0;

    try {
        require_once '../../config/connection.php';
        include '../functions.php';

        $tours = '';
        $pages = '';
        $offset = '';

        if(!isset($_SESSION['user']) || isset($_SESSION['user']) &&
$_SESSION['user']->role_id == 2){
            $tours = getAvailableTours($limit, $text, $categories);
            $pages = getAvaliableToursPagination($text, $categories);
        }else if(isset($_SESSION['user']) && $_SESSION['user']->role_id == 1){
            $tours = getAllTours($limit);
            $pages = getAllToursPagination();
            $offset = ADMIN_OFFSET;
        }

        echo json_encode([
            'tours' => $tours,
            'pages' => $pages,
            'offset' => $offset
        ]);
    } catch (PDOException $th) {
        echo json_encode($th->getMessage());
        http_response_code(500);
    }

}else http_response_code(404);

```

assets/js/auth.js

```

$(document).ready(function () {
    $('#btnRegister').click(function (e) {
        e.preventDefault();
    });

```



```

const first_name = $('#first_name').val()
const last_name = $('#last_name').val()
const email = $('#email').val()
const password = $('#password').val()

if(registerValidationForm(first_name, last_name, email, password).length
== 0){
    $.ajax({
        type: "post",
        url: "models/auth/register.php",
        data: {first_name, last_name, email, password},
        dataType: "json",
        success: function (response) {
            window.location.href='index.php?page=login'
        },error:function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'register_response_message')
        }
    });
}

});

$('#btnLogin').click(function (e) {
    e.preventDefault();

    const email = $('#email').val()
    const password = $('#password').val()

    if(loginFormValidation(email, password).length == 0){
        $.ajax({
            type: "post",
            url: "models/auth/login.php",
            data: {email, password},
            dataType: "json",
            success: function (response) {

                response == 1 ? window.location.href='admin.php' :
window.location.href='index.php'
            },error:function(jqXHR, statusTxt, xhr){
                createResponseMessages('danger', jqXHR.responseJSON,
'login_response_message')
            }
        });
    }
}

```

```
});  
  
});
```

[assets/js/utilities.js](#)

```
// validation functions  
  
function inputValidation(input, regInput, errorArray, elementId, message){  
    if(!regInput.test(input)){  
        errorArray.push(message)  
        createValidationMessage(elementId, message)  
    }  
    else  
        removeValidationMessage(elementId)  
}  
  
function validateCheckBox(checkboxArray ,errorArray, errorElement, errorMessage){  
    if(checkboxArray.length == 0){  
        errorArray.push(errorMessage)  
        createValidationMessage(errorElement, errorMessage)  
    }else  
        removeValidationMessage(errorElement)  
}  
  
function selectFormValidation(select, errorArray, errorElement, errorMessage){  
    if(select == 0 || select == 'default'){  
        errorArray.push(errorMessage)  
        createValidationMessage(errorElement, errorMessage)  
    }else  
        removeValidationMessage(errorElement)  
}  
  
function inputFileValidation(image, errors, errorElement, errorMessage){  
    const [imageEmptyError, imageFormatError, imageSizeError] = errorMessage  
  
    if(image.length == 0){  
        errors.push(imageEmptyError)  
        createValidationMessage(errorElement, imageEmptyError)  
    }else{  
  
        const allowedType = ['image/png','image/jpeg','image/png'];  
  
        const imageType = image[0]['type']  
        const imageSize = image[0]['size']
```

```

        if(imageSize > 3 * 1024 * 1024){
            errors.push(imageSizeError)
            createValidationMessage(errorElement, imageSizeError)
        }else if(!allowedType.includes(imageType)){
            errors.push(imageFormatError)
            createValidationMessage(errorElement, imageFormatError)
        }else {
            removeValidationMessage(errorElement)
        }
    }
}

function extractValuesFromCheckboxes(checkBoxArray, valuesArray){
    checkBoxArray.forEach(checkbox => {
        valuesArray.push(checkbox.value)
    })
    return valuesArray
}

function validateInputDate(date,errorArray, errorElement, errorMessage){

    const [emptyMessage, invalidDate] = errorMessage

    const currentDate = new Date()
    currentDate.setHours(0,0,0,0)
    const currentTimeStamp = currentDate.getTime();
    const myDate= new Date(date)
    myDate.setHours(0,0,0,0)
    const myDateTimeStamp = myDate.getTime()
    if(date == ""){
        errorArray.push(emptyMessage)
        createValidationMessage(errorElement, emptyMessage)
    }else if(currentTimeStamp > myDateTimeStamp) {
        errorArray.push(invalidDate)
        createValidationMessage(errorElement, invalidDate)
    }else{
        removeValidationMessage(errorElement)
    }
}

// messages functions

```

```

const validationClasses = ['text-danger'];
function createValidationMessage(elementId, message){
    const element = $(`#${elementId}`)
    element.text(message)
    element[0].classList.add(...validationClasses)
}

function removeValidationMessage(elementId){
    const element = $(`#${elementId}`)
    element.text('')
    element[0].classList.remove(...validationClasses)
    element.removeAttr('class')
}

function createResponseMessages(color, text, whereToPlace){
    let content = `<div class="alert alert-${color} text-center alert-dismissible
fade show" role="alert">
    ${text}
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
</div>`
    $(`#${whereToPlace}`).html(content)
}

function formatDate(dateD){
    console.log(dateD)
    const date = dateD.split(" ")[0].split("-")
    const time = dateD.split(" ")[1] != null ? dateD.split(" ")[1] : ''

    return date[2] + '/' + date[1] + '/' + date[0] + " " + time
}

function printPagination(pages, limit, whereToPlace, cls){
    let content= ''

    if(pages > 0){
        for(let i = 0; i<pages; i++){

            let activeClass = parseInt(i) === parseInt(limit) ? 'active' : ''
            content+= `<li class="page-item"><a class="${cls} page-link
${activeClass}" href="#" data-limit = "${i}">${parseInt(i)+1}</a></li>`
        }
    }
}

```

```

    $('#${whereToPlace}`).html(content)
}

function radioInputValidation(radio,errorArray, errorElement, errorMessage){
    if(radio == null){
        errorArray.push(errorMessage)
        createValidationMessage(errorElement, errorMessage)
    }else{
        removeValidationMessage(errorElement)
    }
}
}

```

[assets/js/validation.js](#)

```

// validation functions

function inputValidation(input, regInput, errorArray, elementId, message){
    if(!regInput.test(input)){
        errorArray.push(message)
        createValidationMessage(elementId, message)
    }
    else
        removeValidationMessage(elementId)
}

function validateCheckBox(checkboxArray ,errorArray, errorElement, errorMessage){
    if(checkboxArray.length == 0){
        errorArray.push(errorMessage)
        createValidationMessage(errorElement, errorMessage)
    }else
        removeValidationMessage(errorElement)
}

function selectFormValidation(select, errorArray, errorElement, errorMessage){
    if(select == 0 || select == 'default'){
        errorArray.push(errorMessage)
        createValidationMessage(errorElement, errorMessage)
    }else
        removeValidationMessage(errorElement)
}

function inputFileValidation(image, errors, errorElement, errorMessage){
    const [imageEmptyError, imageFormatError, imageSizeError] = errorMessage

    if(image.length == 0){
        errors.push(imageEmptyError)
    }
}

```

```

        createValidationMessage(errorElement, imageEmptyError)
    }else{

        const allowedType = ['image/png','image/jpeg','image/png'];

        const imageType = image[0]['type']
        const imageSize = image[0]['size']

        if(imageSize > 3 * 1024 * 1024){
            errors.push(imageSizeError)
            createValidationMessage(errorElement, imageSizeError)
        }else if(!allowedType.includes(imageType)){
            errors.push(imageFormatError)
            createValidationMessage(errorElement, imageFormatError)
        }else {
            removeValidationMessage(errorElement)
        }

    }
}

function extractValuesFromCheckboxes(checkBoxArray, valuesArray){
    checkBoxArray.forEach(checkbox => {
        valuesArray.push(checkbox.value)
    })
    return valuesArray
}

function validateInputDate(date,errorArray, errorElement, errorMessage){

    const [emptyMessage, invalidDate] = errorMessage

    const currentDate = new Date()
    currentDate.setHours(0,0,0,0)
    const currentTimeStamp = currentDate.getTime();
    const myDate= new Date(date)
    myDate.setHours(0,0,0,0)
    const myDateTimeStamp = myDate.getTime()
    if(date == ""){
        errorArray.push(emptyMessage)
        createValidationMessage(errorElement, emptyMessage)
    }else if(currentTimeStamp > myDateTimeStamp) {
        errorArray.push(invalidDate)
        createValidationMessage(errorElement, invalidDate)
    }
}

```

```

    }else{
        removeValidationMessage(errorElement)
    }
}

// messages functions

const validationClasses = ['text-danger'];
function createValidationMessage(elementId, message){
    const element = $('#${elementId}`)
    element.text(message)
    element[0].classList.add(...validationClasses)
}

function removeValidationMessage(elementId){
    const element = $('#${elementId}`)
    element.text('')
    element[0].classList.remove(...validationClasses)
    element.removeAttr('class')
}

function createResponseMessages(color, text, whereToPlace){
    let content = `<div class="alert alert-${color}" text-center alert-dismissible
fade show" role="alert">
    ${text}
    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
</div>`
    $('#${whereToPlace}`).html(content)
}

function formatDate(dateD){
    console.log(dateD)
    const date = dateD.split(" ")[0].split("-")
    const time = dateD.split(" ")[1] != null ? dateD.split(" ")[1] : ''

    return date[2] + '/' + date[1] + '/' + date[0] + " " + time
}

function printPagination(pages, limit, whereToPlace, cls){
    let content= ''

    if(pages > 0){

```

```

        for(let i = 0; i<pages; i++){

            let activeClass = parseInt(i) === parseInt(limit) ? 'active' : ''
            content+= `<li class="page-item"><a class="${cls} page-link
${activeClass}" href="#" data-limit = "${i}">${parseInt(i)+1}</a></li>`
            }
        }
        $(`#${whereToPlace}`).html(content)
    }

function radioInputValidation(radio,errorArray, errorElement, errorMessage){
    if(radio == null){
        errorArray.push(errorMessage)
        createValidationMessage(errorElement, errorMessage)
    }else{
        removeValidationMessage(errorElement)
    }
}

```

assets/js/admin.js

```

$(document).ready(function () {

    let editor
    ClassicEditor
        .create( document.contains(document.querySelector( '#tour_description')) ?
document.querySelector( '#tour_description' ) :'' )
        .then( newEditor => {
            editor = newEditor;
        } )
        .catch( error => {
            console.error( error );
        } );

    let page = window.location.href.split("/")
    let activePage = page[page.length - 1].split("?")

    if(activePage.length == 1 || activePage[1] == "page=home"){
        getData()
    }else if(activePage.length == 2 && activePage[1] == 'page=destinacije'){
        filterTourBy()
    }else if (activePage[1].split('&')[0] != 'page=dates'){
        localStorage.clear()
    }
}

```



```

// categories
$(document).on('click', '.btn-edit-category', function (e) {
    e.preventDefault();

    const id = $(this).data("id")
    const index = $(this).data('index')

    $.ajax({
        type: "get",
        url: "models/categories/getOne.php",
        data: {id},
        dataType: "json",
        success: function (response) {
            fillCategoryForm(response, index)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger',
jqXHR.responseText, 'category_response_message')
        }
    });
});

$(document).on('click', '.btn-delete-category', function(e){
    e.preventDefault();
    const id = $(this).data('id')
    const index = $(this).data('index')
    const status = $(this).data('status')
    const whereToPlace = `category_${index}`;

    $.ajax({
        type: "post",
        url: "models/categories/delete.php",
        data: {status, id},
        dataType: "json",
        success: function (response) {
            printCategoryRow(response, index, whereToPlace)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger', jqXHR.responseText,
'category_response_message')
        }
    });
});

$('#btnSaveCategory').click(function (e) {
    e.preventDefault();

    const name = $('#name').val()

```

```

const id = $('#category_id').val()
const index = $('#category_index').val()
const whereToPlace = `category_${index}`
let url = ''
let formData = new FormData()
formData.append('name', name)

if(id != ""){
    url= "models/categories/update.php"
    formData.append('id', id)
}else url ='models/categories/store.php'

if(categoryFormValidation(name).length == 0){
    $.ajax({
        type: "POST",
        url: url,
        data: formData,
        processData:false,
        contentType:false,
        dataType: "json",
        success: function (response) {
            if(id == ""){
                printAllCategories(response.categories)
                createResponseMessages('success',
response.message, 'category_response_message')
            }else
                printCategoryRow(response, index, whereToPlace)
                clearCategoryForm()
            },error:function(jqXHR, statusTxt, xhr){
                createResponseMessages('danger', jqXHR.responseJSON,
'category_response_message')
            }
        });
    }

});

$('#btnResetCategory').click(function(e){
    e.preventDefault()
    clearCategoryForm()
})

function printAllCategories(categories){
    let content = '', index = 0
    categories.forEach(category => {
        content+= printCategoryRow(category,index)
    })
}

```

```

        index++
    })
    $('#categories').html(content)
}
function printCategoryRow(category, index, whereToPlace='') {
    const {id, name, created_at, updated_at, is_deleted} = category
    let content = `
    <tr id="category_${index}">
        <th scope="row">${parseInt(index) + 1}</th>
        <td>${name}</td>
        <td>${formatDate(created_at)}</td>
        <td>${updated_at != null ? formatDate(updated_at) : "-"}</td>
        <td><button class="btn btn-sm btn-success btn-edit-category"
type="button" data-id="${id}" data-index="${index}">Edit</button></td>
        <td><button class="btn btn-sm btn-danger btn-delete-category"
type="button" data-id="${id}" data-index="${index}" data-
status="${is_deleted}">${is_deleted == 0 ? "Delete" : "Activate"}</button></td>
    </tr>
    `

    if(whereToPlace != "") {
        document.querySelector(`#${whereToPlace}`).innerHTML = content
    }
    return content
}

function clearCategoryForm() {
    $('#category_id').val('').removeAttr('value')
    $('#category_index').val('').removeAttr("value")
    $('#name').val('').removeAttr('value')
}

function fillCategoryForm(category, index) {
    const {id, name} = category
    $('#category_id').val(id)
    $('#category_index').val(index)
    $('#name').val(name)
}

// posts
$(document).on('click', '.btn-edit-tour', function(e) {
    e.preventDefault()

    const id = $(this).data('id')
    const index = $(this).data('index')

    $.ajax({

```

```

        type: "get",
        url: "models/tours/edit.php",
        data: {id},
        dataType: "json",
        success: function (response) {
            fillTourForm(response, index)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger', jqXHR.responseText,
'tour_response_message')
        }
    });

})

$(document).on('click', '.btn-delete-tour', function(e){
    e.preventDefault();
    const id = $(this).data('id')
    const index = $(this).data('index')
    const status = $(this).data('status')

    const whereToPlace = `tour_${index}`

    $.ajax({
        type: "post",
        url: "models/tours/delete.php",
        data: {id, status},
        dataType: "json",
        success: function (response) {
            printTourFullRow(response, index, whereToPlace)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger', jqXHR.responseText,
'tour_response_message')
        }
    });

})

$(document).on('click', '#btnResetTour', function(e){
    e.preventDefault()
    clearTourForm()
})

$(document).on('click', '#btnSaveTour', function(e){
    e.preventDefault()
    const id = $('#tour_id').val()
    const index = $('#tour_index').val()
    const name = $('#name').val()
    const short_description = $('#short_description').val()
    const image = $('#image').prop('files')

```

```

const price = $('#price').val()
const duration = $('#duration').val()
const long_description = editor.getData()
const categories =
document.querySelectorAll('input[name="categories"]:checked')

let url = '', formData = new FormData(), selectedCheckboxes = []
formData.append('name', name)
formData.append('price', price)
image.length > 0 ? formData.append("image", image[0]) : ''
console.log(image)

formData.append('duration', duration)
formData.append('short_description', short_description)
formData.append('long_description', long_description)
formData.append('categories', extractValuesFromCheckboxes(categories,
selectedCheckboxes))

let currentIndex = localStorage.getItem('page') != null ?
localStorage.getItem('page') : 0

const whereToPlace = `tour_${index}`

if(id != ""){
    url = "models/tours/update.php"
    formData.append("id", id)
    let currentImage = document.querySelector('#tour_img').src.split('/')
    let image_name = currentImage[currentImage.length - 1]
    formData.append("currentImage", image_name)
}else{
    url = "models/tours/store.php"
    formData.append('limit', currentIndex)
}

if(tourFormValidation(name, price, duration, categories,
short_description, long_description, image,id).length == 0){
    $.ajax({
        type: "post",
        url: url,
        data: formData,
        processData:false,
        contentType:false,
        dataType: "json",

```

```

        success: function (response) {
            clearTourForm()
            if(id == ""){
                createResponseMessages('success', response.message,
'tour_response_message')
                printAllTours(response.tours, currentIndex,
response.offset)
                printPagination(response.pages, currentIndex, 'tour-
pages', 'tour-page')
            }else{
                printTourFullRow(response, index, whereToPlace)
            }
        }, error: function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'tour_response_message')
        }
    });
}

})
$(document).on('click', '.tour-page', function(e){
    e.preventDefault()
    let limit = $(this).data("limit")
    localStorage.setItem('page', limit)
    filterTourBy()
})
function filterTourBy(){
    let limit = localStorage.getItem('page') !== null ?
localStorage.getItem('page') : 0
    $.ajax({
        type: "get",
        url: "models/tours/filter.php",
        data: {limit},
        dataType: "json",
        success: function (response) {
            printAllTours(response.tours, limit, response.offset)
            printPagination(response.pages, limit, 'tour-pages', 'tour-page')
        }, error: function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'tour_response_message')
        }
    });
}

function clearTourForm(){
    $('#tour_id').val('').removeAttr('value')
}

```

```

    $('#tour_index').val('').removeAttr('value')
    $('#name').val('').removeAttr('value')
    $('#price').val('').removeAttr('value')
    $('#duration').val('').removeAttr('value')
    $('#short_description').val('')
    editor.setData("")
    let categories =
document.querySelectorAll('input[name="categories"]:checked')
    categories.forEach(cat => {
        let cat_id= cat.id
        $('#${cat_id}`).prop('checked', false)
    })
    $('#image').val('')
    let image = document.querySelector('#tour_img')
    image.src = ''
    image.classList.remove('img-fluid')
    image.classList.add('d-none')
}
function fillTourForm(tour, index){
    const {id, name,short_description, long_description,image_path, days,
price, categories} = tour
    $('#tour_id').val(id)
    $('#tour_index').val(index)
    $('#name').val(name)
    $('#price').val(price)
    $('#short_description').val(short_description)
    editor.setData(long_description)
    $('#duration').val(days)

    const cat_front = document.querySelectorAll('input[name="categories"]')
    cat_front.forEach(cat_f => {
        let id = cat_f.id
        let value = parseInt(cat_f.value)
        $('#${id}`).prop('checked',false)
        categories.forEach(cat => {
            let cat_id = parseInt(cat.category_id)
            cat_id === value ? $('#${id}`).prop("checked", true) : ''
        })
    })
    let image = document.querySelector('#tour_img')
    image.src = `assets/img/${image_path}`
    image.classList.remove('d-none')
    image.classList.add("img-fluid")
}
function printAllTours(tours, limit = 0, offset = 0){

```

```

        console.log(limit, offset)
        let content = '', index = parseInt(limit) * parseInt(offset)
        tours.forEach(tour => {
            content+= printTourFullRow(tour, index)
            index++
        })
        document.contains(document.querySelector('#tours')) ?
document.querySelector('#tours').innerHTML = content : ''
    }
    function printTourFullRow(tour, index, whereToPlace = ''){
        const {id, name, days, price, created_at, updated_at, is_deleted,
categories} = tour
        let content = `<tr id='tour_${index}'>
        <th scope="row">${parseInt(index)+ 1}</th>
        <td>${name}</td>
        <td>${price}</td>
        <td>${days}</td>
        <td>`
        categories.forEach((category, index, array) => {
            if( index !== array.length - 1)
                content+= ` ${category.name},<br/>`
            else
                content+= ` ${category.name}`
        })
        content+=` </td><td><a href="admin.php?page=dates&id=${id}" class="btn
btn-sm btn-primary">Dodaj</a></td>
        <td>${formatDate(created_at)}</td>
        <td>${updated_at != null ? formatDate(updated_at):"-"}</td>
        <td><button class="btn btn-sm btn-success btn-edit-tour" data-id="${id}"
data-index="${index}">Edit</button></td>
        <td><button class="btn btn-sm btn-danger btn-delete-tour" type="button"
data-id="${id}"
        data-index="${index}" data-status="${is_deleted}">${is_deleted == 0 ?
"Delete":"Activate"}</button></td>
        </tr>`
        if(whereToPlace != "")
            document.querySelector(`#${whereToPlace}`).innerHTML =content
        return content
    }
    // dates

$(document).on('click', '.btn-edit-date',function(e){
    e.preventDefault()

    const id = $(this).data('id')

```



```

const index = $(this).data('index')

$.ajax({
  type: "get",
  url: "models/dates/edit.php",
  data: {id},
  dataType: "json",
  success: function (response) {
    fillDateForm(response, index)
  }, error: function (jqXHR, statusTxt, xhr) {
    createResponseMessages('danger',
jqXHR.responseText, 'date_response_message')
  }
});
})
$(document).on('click', '.btn-delete-date', function(e){
  e.preventDefault()
  const id = $(this).data("id")
  const index = $(this).data("index")
  const status = $(this).data('status')
  const whereToPlace = `date_${index}`
  $.ajax({
    type: "post",
    url: "models/dates/delete.php",
    data: {id, status},
    dataType: "json",
    success: function (response) {
      printDateFullRow(response, index, whereToPlace)
    }, error: function (jqXHR, statusTxt, xhr) {
      createResponseMessages('danger', jqXHR.responseText,
'date_response_message')
    }
  });
})
$(document).on('click', '#btnResetDate', function(e){
  e.preventDefault()
  $('#date_id').val('').removeAttr('value')
  $('#date_index').val('').removeAttr('value')
  $('#date').val('')
})
$(document).on('click', '#btnSaveDate', function(e){
  e.preventDefault()

```

```

const id = $('#date_id').val()
const index = $('#date_index').val()
const tour_id = $('#tour_id').val()
const date = $('#date').val()
const whereToPlace = `date_${index}`
let url = '', formData = new FormData()

formData.append("tour_id", tour_id)
formData.append('date', date)

if(id != ""){
    formData.append('id', id)
    url = 'models/dates/update.php'
}else
    url= 'models/dates/store.php'

if(dateFormValidation(date).length == 0){
    $.ajax({
        type: "post",
        url: url,
        data: formData,
        dataType: "json",
        processData:false,
        contentType:false,
        success: function (response) {
            clearDateForm()

            if(id == ""){
                createResponseMessages('success', response.message,
'date_response_message')
                printAllTourDates(response.tours)
            }
            printDateFullRow(response, index, whereToPlace)
        },error:function(jqXHR,responseJSON,xhr){
            createResponseMessages('danger',
jqXHR.responseJSON, 'date_response_message')
        }
    });
}
})

```

```

function clearDateForm(){
    $("#date_id").val("").removeAttr('value')
    $("#date_index").val("").removeAttr('value')
    $('#date').val('').removeAttr('value')
}
function printAllTourDates(tour_data){
    let content = '', index = 0
    tour_data.forEach(tour => {
        content+= printDateFullRow (tour, index)
        index++
    })
    $('#dates').html(content)
}
function printDateFullRow(dateD, index, whereToPlace = ""){
    const {id, date, tour_id, created_at, updated_at, is_deleted} = dateD
    let content = `<tr id="date_${index}">
    <th scope="row">${parseInt(index)+1}</th>
    <td>${formatDate(date)}</td>
    <td>${formatDate(created_at)}</td>
    <td>${updated_at != null ? formatDate(updated_at) : "-"}</td>
    <td><button class="btn btn-sm btn-success btn-edit-date" type="button"
data-id="${id}" data-index="${index}">Edit</button></td>
    <td><button class="btn btn-sm btn-danger btn-delete-date" type='button'
data-id="${id}" data-index="${index}" data-status="${is_deleted}">${is_deleted ==
0 ? "Delete" : "Activate"}</button></td>
    </tr>`
    if(whereToPlace != "")
        document.querySelector(`#${whereToPlace}`).innerHTML = content
    return content
}
function fillDateForm(date_data, index){
    const {id, date} = date_data
    $('#date_id').val(id)
    $('#date_index').val(index)
    $('#date').val(date)
}

// messages

$(document).on("click", '.btn-read-message', function(e){
    e.preventDefault()
    const id = $(this).data('id')

    $.ajax({
        type: "get",

```

```

        url: "models/messages/getOne.php",
        data:{id},
        dataType: "json",
        success: function (response) {
            const {first_name, last_name, email, message,created_at} =
response
            const full_name = `${first_name} ${last_name}`
            $('#user-from').text(full_name)
            $('#email-from').text(email)
            $('#arrived_at').text(formatDate(created_at))
            $('#message').text(message)
        },error:function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'message_response_message')
        }
    });
})
$(document).on('click','.message-page', function(e){
    e.preventDefault()

    const limit = $(this).data('limit')
    $.ajax({
        type: "get",
        url: "models/messages/pagination.php",
        data: {limit},
        dataType: "json",
        success: function (response) {
            printAllMessages(response.messages, limit, response.offset)
            printPagination(response.pages, limit,'message-pages','message-
page' )
        },error:function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'message_response_message')
        }
    });
})
function printAllMessages(messages, limit, offset){

    let content = '', index = limit * offset

    messages.forEach(message => {
        content+= printMessage(message, index)
        index++
    })
    $('#messages').html(content)

```

```

}
function printMessage(messageD, index){
    const {id, first_name, last_name, email, message, created_at} = messageD

    let content = `<tr id="date_${index}">
    <th scope="row">${parseInt(index)+1}</th>
    <td>${first_name+' '+last_name}</td>
    <td>${email}</td>
    <td>${formatDate(created_at)}</td>
    <td> <button type="button" class="btn btn-primary btn-sm btn-read-message"
data-bs-toggle="modal" data-bs-target="#exampleModal" data-
id="${id}">Read</button>
    </td>
</tr>`
    return content
}

// questions
$(document).on('click', '.btn-edit-question', function(e){
    e.preventDefault()
    const id = $(this).data('id')
    const index = $(this).data("index")

    $.ajax({
        type: "get",
        url: "models/questions/getOne.php",
        data: {id},
        dataType: "json",
        success: function (response) {
            fillQuestionForm(response, index)
        }, error: function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseText, 'question_resp
onse_message')
        }
    });
})

$(document).on('click', '.btn-delete-question', function(e){
    e.preventDefault()
    const id = $(this).data('id')
    const index = $(this).data('index')
    const status = $(this).data('status')
    const whereToPlace = `question_${index}`
    $.ajax({
        type: "post",

```

```

        url: "models/questions/delete.php",
        data: {id, status},
        dataType: "json",
        success: function (response) {
            printQuestionFullRow(response, index, whereToPlace)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger', jqXHR.responseJSON,
'question_response_message')
        }
    });
})
$(document).on('click', '#btnSaveQuestion', function(e){
    e.preventDefault()

    const id = $('#question_id').val()
    const index = $('#question_index').val()
    const name = $('#question').val()

    let url = '', data = new FormData()
    const whereToPlace = `question_${index}`

    data.append("name", name)

    if(id != ""){
        url = "models/questions/update.php"
        data.append('id', id)
    }else
        url = "models/questions/store.php"

    if(questionFormValidation(name).length == 0){
        $.ajax({
            type: "post",
            url: url,
            data: data,
            contentType: false,
            processData: false,
            dataType: "json",
            success: function (response) {
                clearQuestionForm()
                if(id == ""){
                    createResponseMessages('success', response.message,
'question_response_message')
                    printAllQuestions(response.questions)

```

```

        }
        else
            printQuestionFullRow(response, index, whereToPlace)
    },error:function(jqXHR,statusTxt,xhr){
        createResponseMessages('danger', jqXHR.responseJSON,
'question_response_message')
    }
    });
}

})
$(document).on('click','#btnResetQuestion',function (e) {
    e.preventDefault()
    clearQuestionForm()
})

function fillQuestionForm(question, index){
    const {id, name} = question
    $('#question_id').val(id)
    $('#question_index').val(index)
    $('#question').val(name)
}

function clearQuestionForm(){
    $('#question_id').val('').removeAttr('value')
    $('#question_index').val('').removeAttr('value')
    $('#question').val('').removeAttr('value')
}

function printAllQuestions(questions){
    let content = '', index = 0
    questions.forEach(question => {
        content+= printQuestionFullRow(question, index)
    })
    $('#questions').html(content)
}

function printQuestionFullRow(question, index, whereToPlace = ''){
    const {id, name, created_at, updated_at, is_deleted} = question
    let content = `
<tr id="question_${index}">
    <th scope="row">${parseInt(index)+1}</th>
    <td>${name}</td>
    <td>${formatDate(created_at)}</td>
    <td>${updated_at != null ? formatDate(updated_at) : '-'}</td>
    <td><button class="btn btn-sm btn-success btn-edit-question"
type="button" data-id="${id}" data-index="${index}">">Edit</button></td>
    <td><button class="btn btn-sm btn-danger btn-delete-question"

```

```

        type="button" data-index="${index}" data-status = "${is_deleted}"
data-id="${id}"
        >${is_deleted == 0 ? "Delete" : "Active"}</button></td>
    </tr>
    `
    if(whereToPlace != ''){
        document.querySelector(`#${whereToPlace}`).innerHTML = content
    }
    return content
}

// answers

$(document).on('click', '.btn-edit-answer', function(e){

    e.preventDefault()

    const id = $(this).data('id')
    const index = $(this).data('index')

    $.ajax({
        type: "get",
        url: "models/answers/getOne.php",
        data: {id},
        dataType: "json",
        success: function (response) {
            fillAnswerForm(response, index)
        }, error: function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
"answer_response_message")
        }
    });
})

$(document).on("click", '.btn-delete-answer', function(e){

    e.preventDefault()

    const id = $(this).data('id')
    const index = $(this).data('index')
    const status = $(this).data('status')

    const whereToPlace = `answer_${index}`
    $.ajax({
        type: "post",
        url: "models/answers/delete.php",

```



```

        data: {id, status},
        dataType: "json",
        success: function (response) {
            printAnswerFullRow(response, index, whereToPlace)
        }, error: function (jqXHR, statusTxt, xhr) {
            createResponseMessages('danger', jqXHR.responseJSON,
'answer_response_message')
        }
    });
})
$(document).on('click', '#btnSaveAnswer', function(e){
    e.preventDefault()

    const id = $('#answer_id').val()
    const index = $('#answer_index').val()
    const option = $('#answer').val()

    const whereToPlace = `answer_${index}`

    let url = '', formData = new FormData()
    formData.append('option', option)
    if(id != ""){
        url = "models/answers/update.php"
        formData.append('id', id)
    }else url = 'models/answers/store.php'

    if(answerFormValidation(option).length == 0){
        $.ajax({
            type: "post",
            url: url,
            data: formData,
            contentType: false,
            processData: false,
            dataType: "json",
            success: function (response) {
                clearAnswerForm()
                if(id == ""){
                    createResponseMessages('success', response.message,
'anser_response_message')
                    printAllAnswers(response.options)
                }else
                    printAnswerFullRow(response, index, whereToPlace)
            }, error: function (jqXHR, statusTxt, xhr) {
                createResponseMessages('danger', jqXHR.responseJSON,
'answer_response_message')
            }
        });
    }
});

```

```

    }
  });
}
})
function fillAnswerForm(answer, index){
  const {id, name} = answer
  $('#answer_id').val(id)
  $('#answer_index').val(index)
  $('#answer').val(name)
}
function clearAnswerForm(){
  $('#answer_id').val('').removeAttr('value')
  $('#answer_index').val('').removeAttr('value')
  $('#answer').val('').removeAttr('value')
}

function printAllAnswers(answers){
  let content = '', index = 0
  answers.forEach(answer => {
    content+= printAnswerFullRow(answer, index)
    index++
  })
  $('#answers').html(content)
}

function printAnswerFullRow(answer, index, whereToPlace = ''){

  const {id, name, created_at, updated_at, is_deleted} = answer
  let content = `
    <tr id="answer_${index}">
      <th scope="row">${parseInt(index)+1}</th>
      <td>${name}</td>
      <td>${formatDate(created_at)}</td>
      <td>${updated_at != null ? formatDate(updated_at) : '-'}</td>
      <td><button class="btn btn-sm btn-success btn-edit-answer"
type="button" data-id="${id}" data-index="${index}">Edit</button></td>
      <td><button class="btn btn-sm btn-danger btn-delete-answer"
type="button" data-id = "${id}">" data-index="${index}" data-
status="${is_deleted}">${is_deleted == 0 ? "Delete" : "Activate"}</button></td>
    </tr>`

  if(whereToPlace != ""){
    document.querySelector(`#${whereToPlace}`).innerHTML = content
  }
  return content
}

```

```

// survey

$(document).on('click', '#btnSaveSurvey', function(e){
    e.preventDefault()
    const id = $('#survey_id').val()
    const index = $('#survey_index').val()
    const date = $('#date').val()
    const question = $('#question').val()
    const answers =
document.querySelectorAll('input[name="answers"]:checked')
    let selectedCheckboxes = []

    let url = '', data = new FormData()
    const whereToPlace = `survey_${index}`

    if(id != ""){
        url = "models/survey/update.php"
        data.append('id', id)
    }else
        url = "models/survey/store.php"

    data.append('date', date)
    data.append('question', question)
    data.append("answers", extractValuesFromCheckboxes(answers,
selectedCheckboxes))

    if(surveyFormValidation(date, question, selectedCheckboxes).length == 0){
        $.ajax({
            type: "post",
            url: url,
            data: data,
            processData:false,
            contentType:false,
            dataType: "json",
            success: function (response) {
                clearSurveyForm()
                if(id == ""){
                    createResponseMessages('success', response.message,
'survey_message')
                    printAllSurveys(response.surveys)
                }
            }
        })
    }else{

```

```

        printSurveyFullRow(response, index, whereToPlace)
    }
    },error:function(jqXHR,statusTxt,xhr){
        createResponseMessages('danger',
jqXHR.responseJSON,'survey_message')
    }
    });
}
})
$(document).on('click','.btn-edit-survey', function(e){
    e.preventDefault()
    const id = $(this).data('id')
    const index = $(this).data('index')

    $.ajax({
        type: "get",
        url: "models/survey/getOne.php",
        data: {id},
        dataType: "json",
        success: function (response) {
            fillSurveyForm(response, index)
        },error:function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'survey_message')
        }
    });
})
$(document).on("click",'.btn-delete-survey',function(e){
    e.preventDefault()
    const status = $(this).data("active")
    const id = $(this).data('id')
    const index = $(this).data('index')
    const whereToPlace = `survey_${index}`
    $.ajax({
        type: "post",
        url: "models/survey/delete.php",
        data: {id, status},
        dataType: "json",
        success: function (response) {
            printSurveyFullRow(response, index, whereToPlace)
        },error:function(jqXHR, statusTxt,xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'survey_message')
        }
    });
});

```

```

})

function fillSurveyForm(survey, index){
    const {id, question_id, expire_date, answers} = survey
    $('#survey_id').val(id)
    $('#survey_index').val(index)
    $('#question').val(question_id)
    $('#date').val(expire_date)

    const answersFront = document.querySelectorAll('input[name="answers"]')
    answersFront.forEach(answerF => {
        let answer_value = answerF.value
        let answerId = answerF.id

        $('#${answerId}`).prop('checked', true)

        answers.forEach(answer => {
            let answer_id = answer.answer_id
            parseInt(answer_value) === parseInt(answer_id) ?
$('#${answerId}`).prop('checked', true) : ''
        })
    })
}

function clearSurveyForm(){
    $('#survey_id').val('').removeAttr('value')
    $('#survey_index').val('').removeAttr('value')
    $('#date').val('')
    $('#question').val(0)
    let answers = document.querySelectorAll('input[name="answers"]:checked')
    answers.forEach(answer => {
        let answer_id = answer.id
        $('#${answer_id}`).prop('checked', false)
    })
}

function printAllSurveys(surveys){
    let content= '', index = 0
    surveys.forEach(survey => {
        content+=printSurveyFullRow(survey, index)
    })
    $('#surveys').html(content)
}

function printSurveyFullRow(survey, index, whereToPlace = ''){

```

```

    const {id,expire_date,question_id, created_at, updated_at,is_active,
questionName} = survey

    let content = ` <tr id="survey_${index}">
    <th scope="row">${parseInt(index)+1}</th>
    <td>${questionName}</td>
    <td>${formatDate(expire_date)}</td>
    <td>${formatDate(created_at)}</td>
    <td>${updated_at != null ? formatDate(updated_at) : ''}</td>
    <td><button class="btn btn-sm btn-success btn-edit-survey"
    type="button" data-id="${id}" data-index="${index}">Edit</button></td>
    <td>
        <button class="btn btn-sm btn-danger btn-delete-survey"
        type="button" data-id="${id}" data-index="${index}"
        data-active="${is_active}">${is_active == 1 ? "Delete" :
"Activate"}</button>
    </td>
    </tr>`
    if(whereToPlace != ""){
        document.querySelector(`#${whereToPlace}`).innerHTML = content
    }
    return content
}

function getData(){
$.ajax({
    type: "get",
    url: "models/getData.php",
    dataType: "json",
    success: function (response) {
        document.contains(document.querySelector('#numberOfUsers')) ?
document.querySelector('#numberOfUsers')
        .innerHTML = response.numberOfUsers.numberOfUsers : ''
        document.contains(document.querySelector('#numberOfTours')) ?
document.querySelector('#numberOfTours')
        .innerHTML = response.numberOfTours.numberOfTours : ''
        printMostVisitedTours(response.mostVisitedTours)
    },error:function(jqXHR,statusTxt,xhr){
        createResponseMessages('danger', jqXHR.responseJSON, 'home_message')
    }
});
}

function printMostVisitedTours(mostVisitedTours){
    let content = '',index = 0

```

```

    mostVisitedTours.forEach(tour => {
        content+= printVisitedTour(tour, index)
        index++
    })
    $('#reservations').html(content)
}
function printVisitedTour(tour, index){
    const {mostVisitedTours, name, date} = tour
    return `
        <tr>
            <th scope='row'>${parseInt(index) + 1}</th>
            <td>${name}</td>
            <td>${formatDate(date)}</td>
            <td>${mostVisitedTours}</td>
        </tr>
    `
}
});

```

assets/js/user.js

```

$(document).ready(function () {

    let page = window.location.href.split("?")[1];
    if(page == 'page=tours'){
        filterData()
    }else{
        localStorage.clear()
    }

    $(document).on('click', '#btnContactUs', function(e){
        e.preventDefault()

        const first_name = $('#first_name').val()
        const last_name = $('#last_name').val()
        const email = $('#email').val()
        const message = $('#message').val()

        if(contactFormValidation(first_name, last_name, email, message).length ==
0){
            $.ajax({
                type: "post",
                url: "models/messages/store.php",
                data: {first_name, last_name, email, message},
                dataType: "json",
            })
        }
    })

```

```

        success: function (response) {
            createResponseMessages('success', response,
'contact_response_message')
            $('#first_name').val('').removeAttr('value')
            $('#last_name').val('').removeAttr('value')
            $('#email').val('').removeAttr('value')
            $('#message').val('').removeAttr('value')
        },error:function(jqXHR, statusTxt, xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'contact_response_message')
        }
    });
}

})

$(document).on("keyup", '#search_text', function(e){
    e.preventDefault()

    let keyword = $("#search_text").val()
    localStorage.setItem('keyword', keyword)
    filterData()
})

$(document).on("click", '.page-tours', function(e){
    e.preventDefault()
    let limit = $(this).data('limit')
    localStorage.setItem('tour-page', limit)
    filterData()
})

$(document).on('change', 'input[name="categories"]', function(e){
    e.preventDefault()
    let checks = document.querySelectorAll('input[name="categories"]:checked')

    let selctedCheckes = [];
    filterData()
    localStorage.setItem('checkboxes', extractValuesFromCheckboxes(checks,
selctedCheckes));

})

$(document).on('click', '#btnReserveTour',function(e){
    e.preventDefault()
    const tour_id = $("#tour_id").val()
    const date = $('#date').val()
    if(reservationDateTour(date).length == 0){
        $.ajax({
            type: "post",

```



```

        url: "models/reservations/store.php",
        data: {tour_id, date},
        dataType: "json",
        success: function (response) {
            createResponseMessages('success', response,
'tour_reservation_response_message')
            $('#date').val(0)
        },error:function(jqXHR,statusTxt,xhr){
            createResponseMessages('danger', jqXHR.responseJSON,
'tour_reservation_response_message')
        }
    });
}

})

$(document).on("click", '#btnVote',function(e){
    e.preventDefault()
    let radioInput = document.querySelector('input[name="answers"]:checked')
    let survey_id = $('#survey_id').val()
    if(anketaFormValidation(radioInput).length == 0){
        const radio_value = radioInput.value
        $.ajax({
            type: "post",
            url: "models/survey/vote.php",
            data: {radio_value, survey_id},
            dataType: "json",
            success: function (response) {
                createResponseMessages('success', response,
'tour_reservation_response_message')
                $("#survey")[0].classList.add("d-none")
            },error:function(jqXHR,statusTxt,xhr){
                createResponseMessages('danger', jqXHR.responseJSON,
'tour_reservation_response_message')
            }
        });
    }
})

function filterData(){
    let text= $('#search_text').val()
    let checkboxes =
document.querySelectorAll('input[name="categories"]:checked')
    let selectedCheckboxes = []
    extractValuesFromCheckboxes(checkboxes, selectedCheckboxes)
}

```

```

        let limit = localStorage.getItem('tour-page') != null ?
localStorage.getItem('tour-page') : 0;
        console.log(limit)
        $.ajax({
            type: "get",
            url: "models/tours/filter.php",
            data: {text, selectedCheckboxes, limit},
            dataType: "json",
            success: function (response) {
                printAllTours(response.tours)
                printPagination(response.pages, limit, 'pages-tours','page-tours')
            },error:function(jqXHR,statusTxt,xhr){
                createResponseMessages('danger', jqXHR.responseJSON,
'tours_response_message')
            }
        });
    }

    function printAllTours(tours){
        let content = ``
        if(tours.length > 0){
            tours.forEach(tour => {
                content+= printTour(tour)
            })
        }else {
            content+= `<h1 class='text-center'>Trenutno nemamo u ponudi ni jednu
turu</h1>`
            $('#tours-pages').html('')
        }
        $('#tours').html(content)
    }

    function printTour(tour){
        const {id, name, image_path} = tour
        return ` <div class="col-lg-3 mb-2">
<div class="card h-100">
            
            <div class="card-body text-center">
                <h2 class="fs-5"> <a href="index.php?page=tour&id=${id}" class="nav-
link">${name}</a></h2>
            </div>
        </div>
</div>`
    }
});

```