

DOKUMENTACIJA SAJTA

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



PRAKTIKUM IZ PHP-a

Dijana Lazić 19/18

Sadržaj

Uvod	3
<i>Korišćeni programski jezici</i>	2
<i>Opis funkcionalnosti</i>	3
<i>Template</i>	4
Organizacija	5
<i>Organizaciona šema</i>	6
<i>Mapa sajta</i>	7
<i>Organizacija foldera</i>	7
<i>Slike stranica i opisi funkcionalnosti</i>	8
Kodovi	9
<i>Index.php</i>	10
<i>Views</i>	11
<i>Fixed</i>	11
<i>head.php</i>	11
<i>nav.php</i>	11
<i>footer.php</i>	11
<i>Admin</i>	11
<i>menu.php</i>	11
<i>list.php</i>	11
<i>category.php</i>	11
<i>panel.php</i>	11
<i>user.php</i>	11
<i>User</i>	11
<i>bucket.php</i>	11
<i>create.php</i>	11
<i>list.php</i>	11
<i>steps.php</i>	11
<i>home.php</i>	11

<i>register.php</i>	11
<i>login.php</i>	11
<i>contact.php</i>	11
<i>author.php</i>	11
Models	11
Bucket	11
<i>change.php</i>	11
<i>insert.php</i>	11
<i>functions.php</i>	11
Category	11
<i>delete.php</i>	11
<i>insert.php</i>	11
<i>filterNsort.php</i>	11
<i>functions.php</i>	11
File	11
<i>excel.php</i>	11
<i>functions.php</i>	11
<i>word.php</i>	11
List	11
<i>insert.php</i>	11
<i>delete.php</i>	11
<i>edit.php</i>	11
<i>functions.php</i>	11
<i>pagination.php</i>	11
<i>update.php</i>	11
PHPMailer	11
User	11
<i>contact.php</i>	11
<i>delete.php</i>	11
<i>edit.php</i>	11
<i>functions.php</i>	11

<i>login.php</i>	11
<i>logout.php</i>	11
<i>register.php</i>	11
<i>update.php</i>	11
<i>Visit</i>	11
<i>functions.php</i>	11
<i>Config</i>	11
<i>.env</i>	11
<i>config.php</i>	11
<i>connection.php</i>	11
<i>Data</i>	11
<i>errors.txt</i>	11
<i>log.txt</i>	11
<i>Assets</i>	11
<i>fontawesome</i>	11
<i>css</i>	11
<i>style.css</i>	11
<i>img</i>	12
<i>js</i>	13
<i>main.js</i>	11

Uvod

Korišćeni programski jezici

Prilikom izrade ovog sajta, ušli su u obzir sledeći jezici:

- *HTML5*
- *XML*
- *CSS3*
- *SASS*
- *JavaScript*
- *jQuery*
- *PHP*
- *SQL*

**Pisano u Visual Studio Code-u*

Opis funkcionalnosti

*Pre samog početka, treba naglasiti da se ceo sadržaj realizuje kroz index.php stranicu koja, u zavisnosti od prosleđenog parametra (poput **page-a**), menja svoju centralnu esenciju. Naime, svakom ko pristupa sajtu se prikazuje ceo skup ideja (goals/lists), premda korisnik nije ulogovan. Ergo, može da vidi sve, šta više i da pošalje poruku. Ukoliko ipak naumi da postane registrovani korisnik, mora prvo da napravi nalog, potom da se uloguje (sve je ovo praćeno proverom podataka sa klijentske kao i sa serverske strane!) i samim tim dobije pristup bucket-u. Pak, ako administrator nastupi na scenu, dobiće admin panel i moći menjanja ideja (goals/lists), dodavanja novih i brisanja (isto važi i za ostala svojstva poput usera,category-ja i menu-ja).*

***Ukoliko nema pristup nekoj od stranica, osoba koja je to pokušala da uradi, se prosto redirektuje na neku drugu, njemu prikladnu.*

Tempate

Preuzeto ovog puta sa https://templatemo.com/live/templatemo_550_diagoona

**uz prikladne izmene autora

warning!!!prilikom nekih dešavanja, desiće se da se dijagonalna desnog main div-a pomera

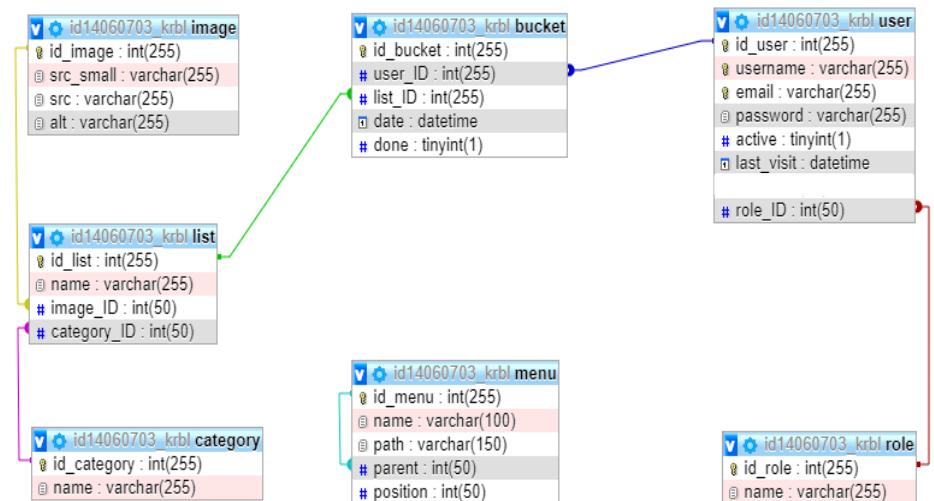
Organizacija

Organizaciona šema:

sajta

baze

INDEX.HTML



Mapa sajta

```

<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

<url>
    <loc>https://bucketlistnkeanureeves.000webhostapp.com/index.php</loc>
    <lastmod>2020-06-15</lastmod>
    <changefreq>yearly</changefreq>
    <priority>1.00</priority>
</url>
</urlset>
    
```

Organizacija foldera:

```

▲ assets
  ▲ css
    # bootstrap.min.css
    # style.css
    # templatemo-diagoona.css
  ▶ fontawesome
  ▶ img
  ▲ js
    JS bootstrap.min.js
    JS jquery-3.4.1.min.js
    JS jquery.backstretch.min.js
    JS main.js
    JS templatemo-script.js
  ▲ config
    ⚙ .env
    ❁ config.php
    ❁ connection.php

  ▲ views
    ▲ admin
      ❁ category.php
      ❁ list.php
      ❁ menu.php
      ❁ panel.php
      ❁ user.php
    ▲ fixed
      ❁ footer.php
      ❁ head.php
      ❁ nav.php
    ▲ user
      ❁ bucket.php
      ❁ create.php
      ❁ author.php
      ❁ contact.php
      ❁ home.php
      ❁ list.php
      ❁ login.php
      ❁ register.php
      ❁ steps.php

```

```

▲ data
  ❁ errors.txt
  ❁ log.txt
  ▲ models
    ▶ bucket
    ▶ category
    ▶ file
    ▶ list
    ▶ PHPMailer
    ▶ user
    ▶ visit
  ▲ views
    ▶ admin
    ▶ fixed
    ▶ user
    ❁ author.php
    ❁ contact.php
    ❁ home.php
    ❁ list.php
    ❁ login.php
    ❁ register.php
    ❁ steps.php
    ❁ ~$cumentation.docx
    ❁ documentation.docx
    ❁ index.php

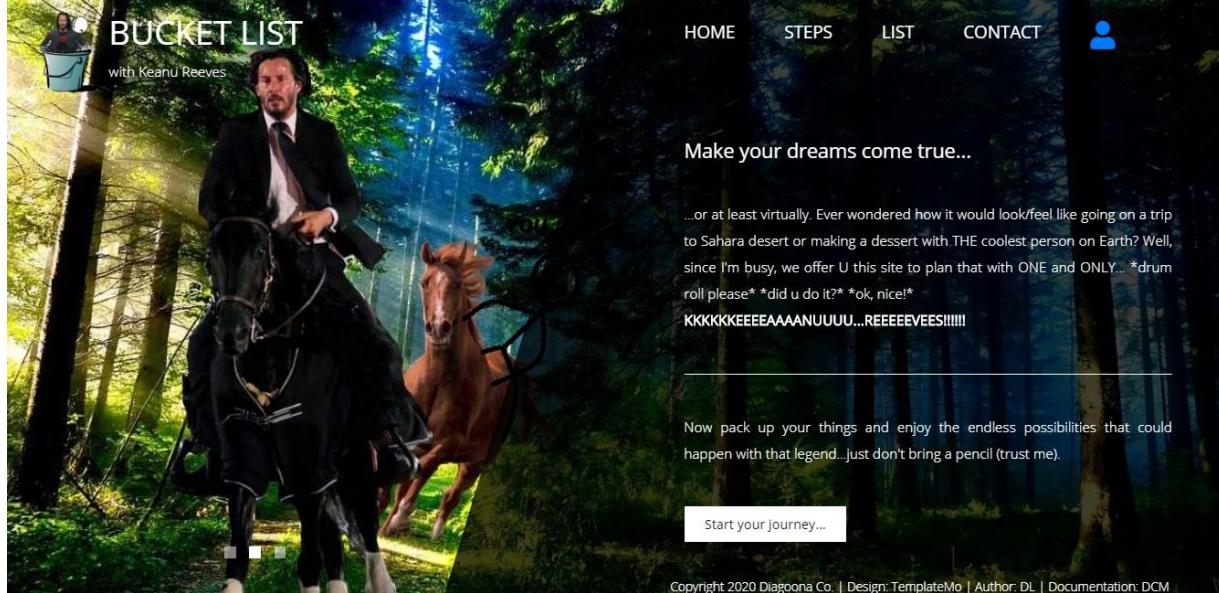
```

```

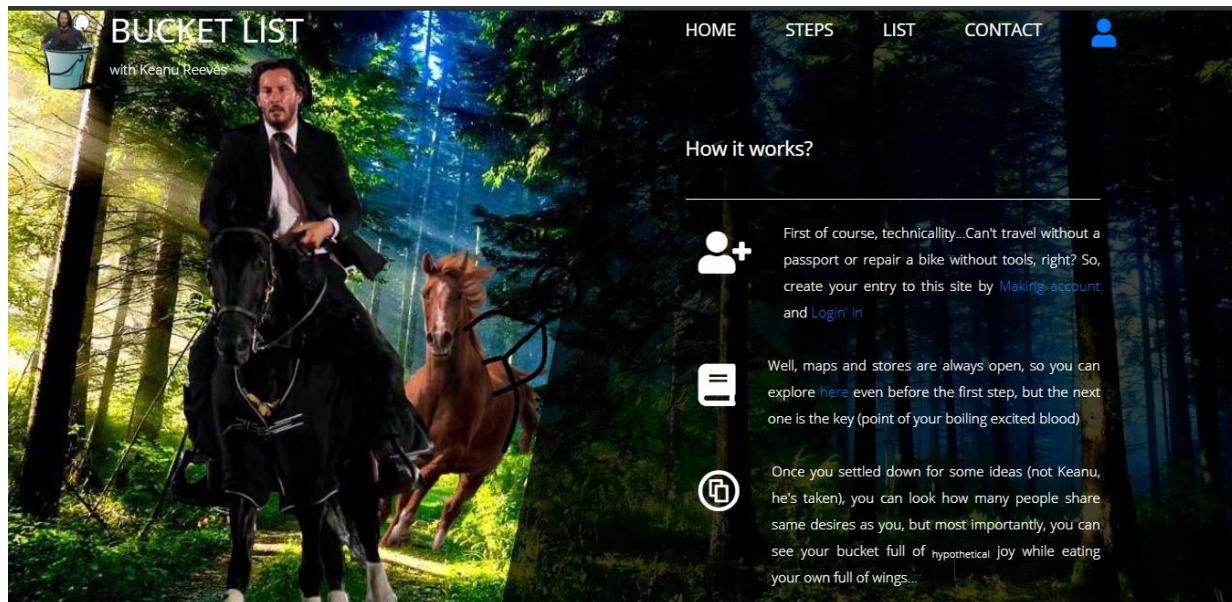
  ▶ assets
  ▶ config
  ▶ data
  ▲ models
    ▲ bucket
      ❁ change.php
      ❁ functions.php
      ❁ insert.php
    ▲ category
      ❁ delete.php
      ❁ filterNsort.php
      ❁ functions.php
      ❁ insert.php
    ▲ file
      ❁ excel.php
    ❁ word.php
  ▲ list
    ❁ delete.php
    ❁ edit.php
    ❁ functions.php
    ❁ insert.php
    ❁ pagination.php
    ❁ update.php
  ▶ PHPMailer
  ▲ user
    ❁ contact.php
    ❁ delete.php
    ❁ edit.php
    ❁ functions.php
    ❁ login.php
    ❁ logout.php
    ❁ register.php
    ❁ update.php
  ▲ visit
    ❁ functions.php
  ▶ views

```

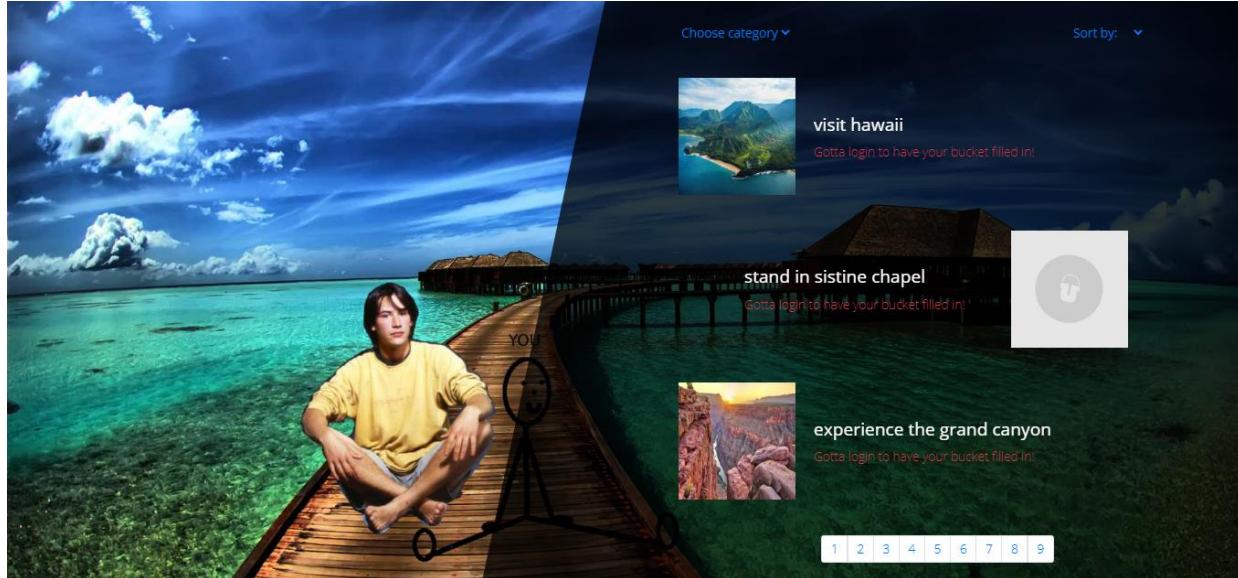
Slike stranica i opis funkcionalnosti:



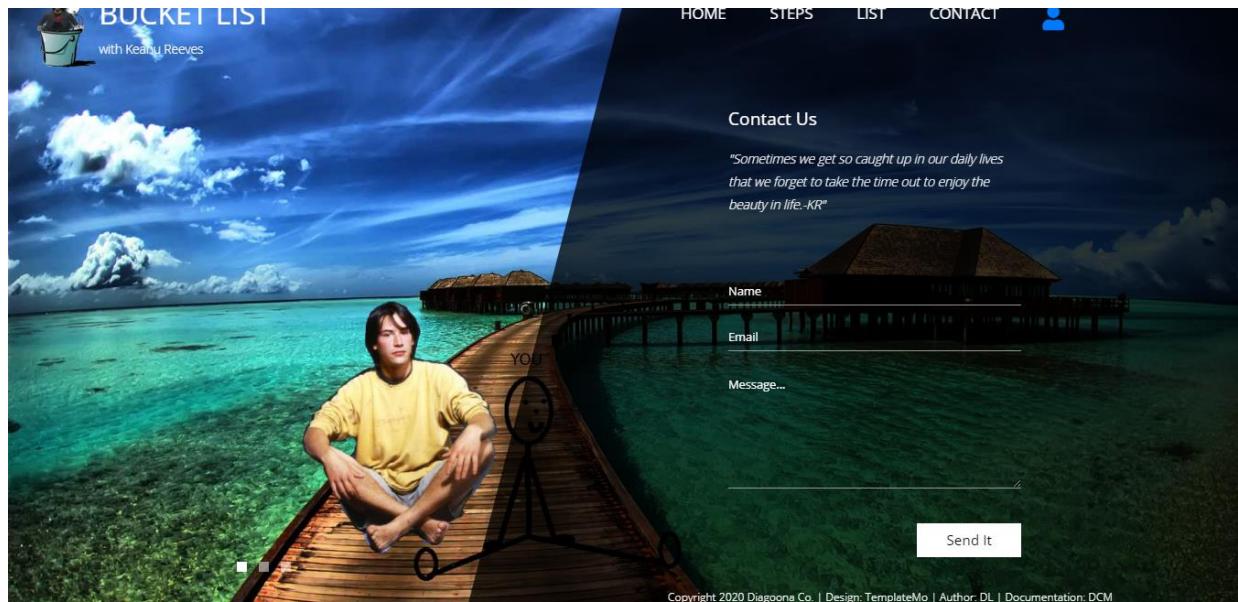
Početna stranica čiji sadržaj objašnjava korisniku o čemu je ovde konkretno reč.



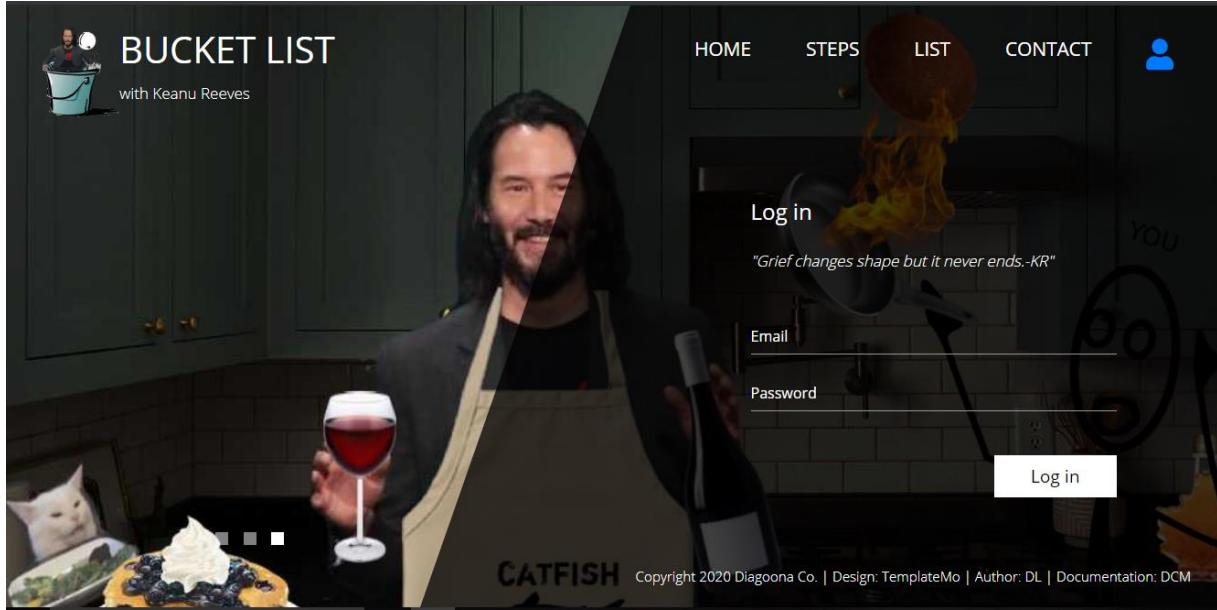
Postupci(=uputstvo) korišćenja ovoga sajta.



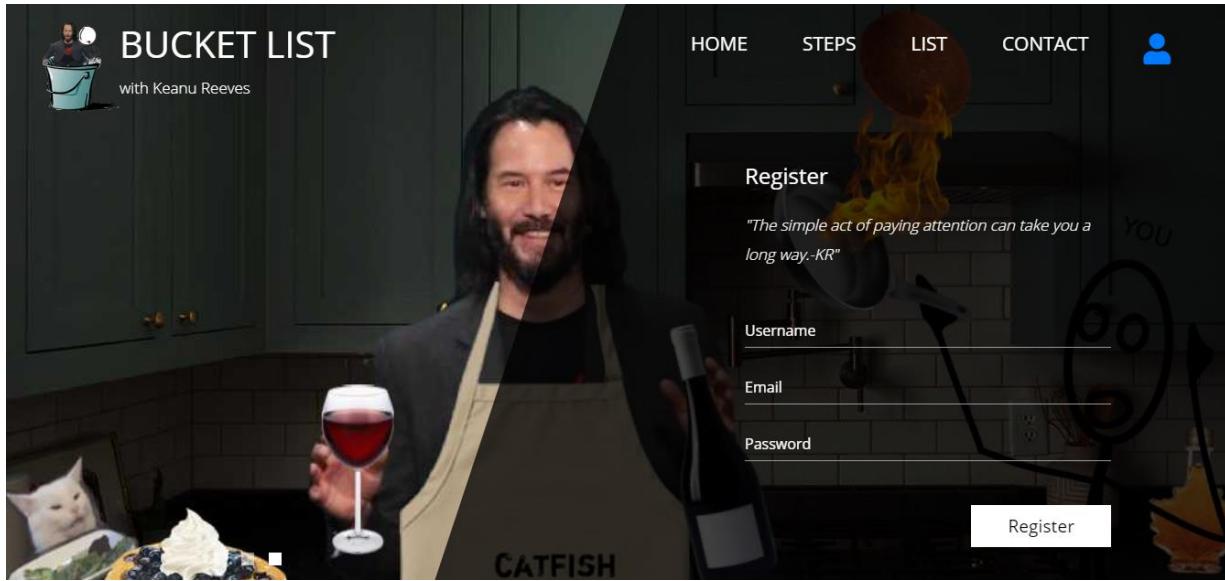
Prikaz sadržaja neautorizovanom korisniku. Mogućnost pregleda svih sa liste pomoću paginacije koja prikazuje maximum 3 na stranici. Odabirom kategorije tj abecednog redosleda prikaza iz padajuće liste, sadržaj se filtrira odnosno sortira u skladu sa tim. Dinamički:ispisane liste.AJAX-om: ukoliko se klikne na paginaciju, filter ili sort listu.



Kontaktiranje administratora slanjem mejla čiji je sadržaj upravo poslata poruka(praćeno klijentskom i serverskom proverom vrednosti).

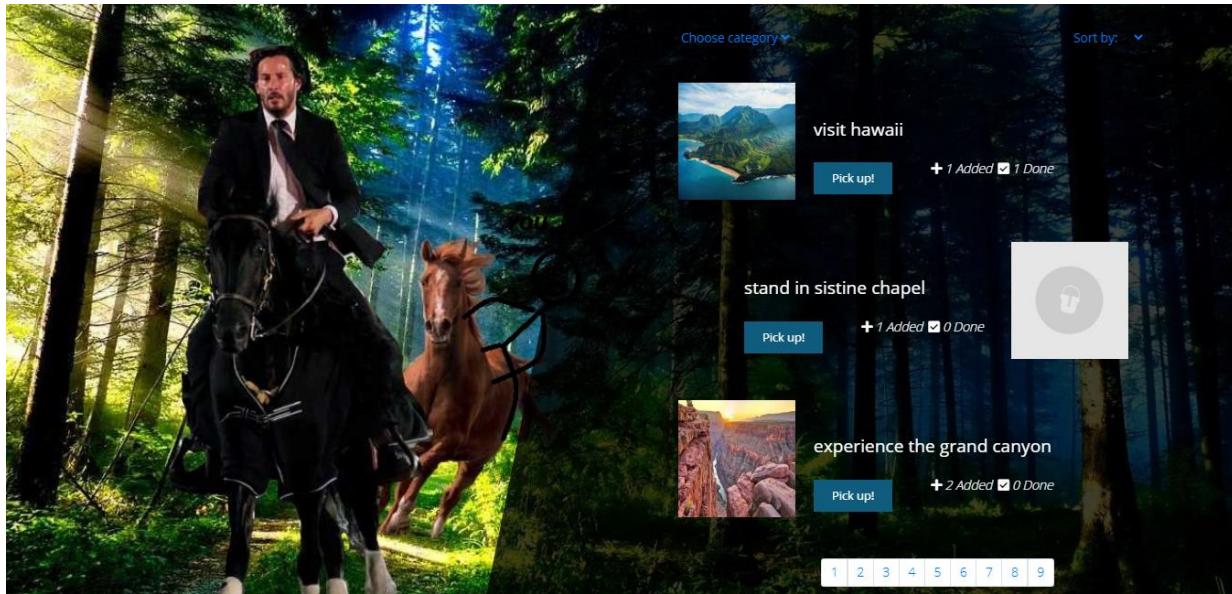


Logovanje ukoliko je korisnik prethodno registrovan. U slučaju nepostojećeg mejla, prikazuje se uzrok problema, međutim ukoliko korisnik pogreši šifru, šalje se mejl upozorenja korisniku sa zadatim mejlom(praćeno klijentskom i serverskom proverom vrednosti).

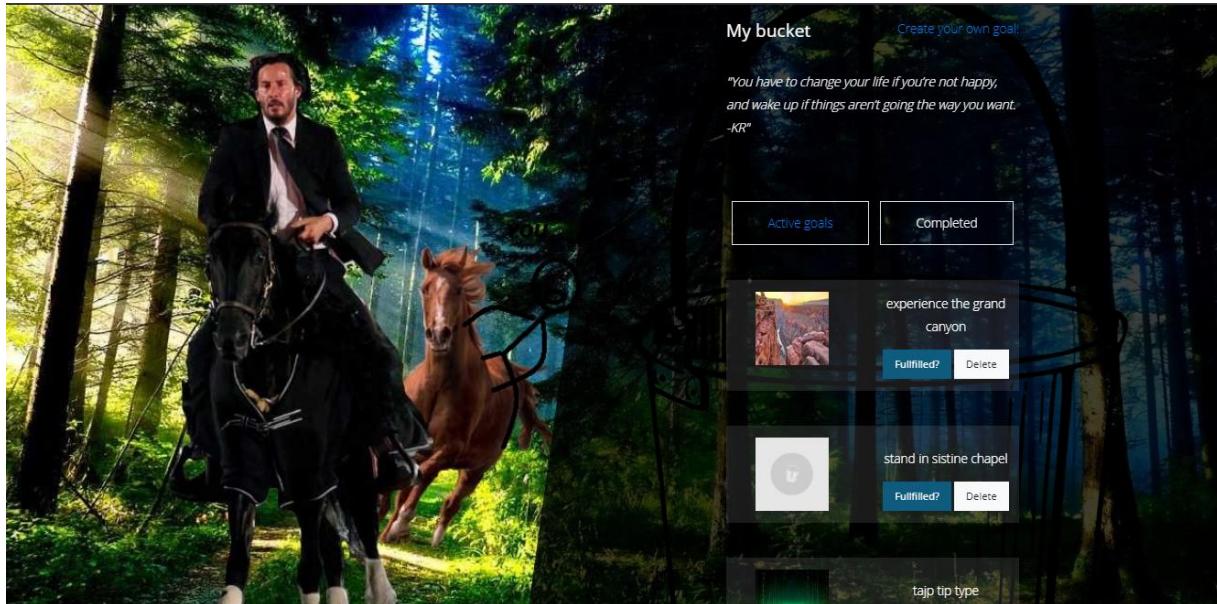


Provera -kako formata tako i stanja u bazi.Ukoliko postoji već username ili email-izbacuje grešku (I sve to praćeno klijentskom i serverskom proverom vrednosti).

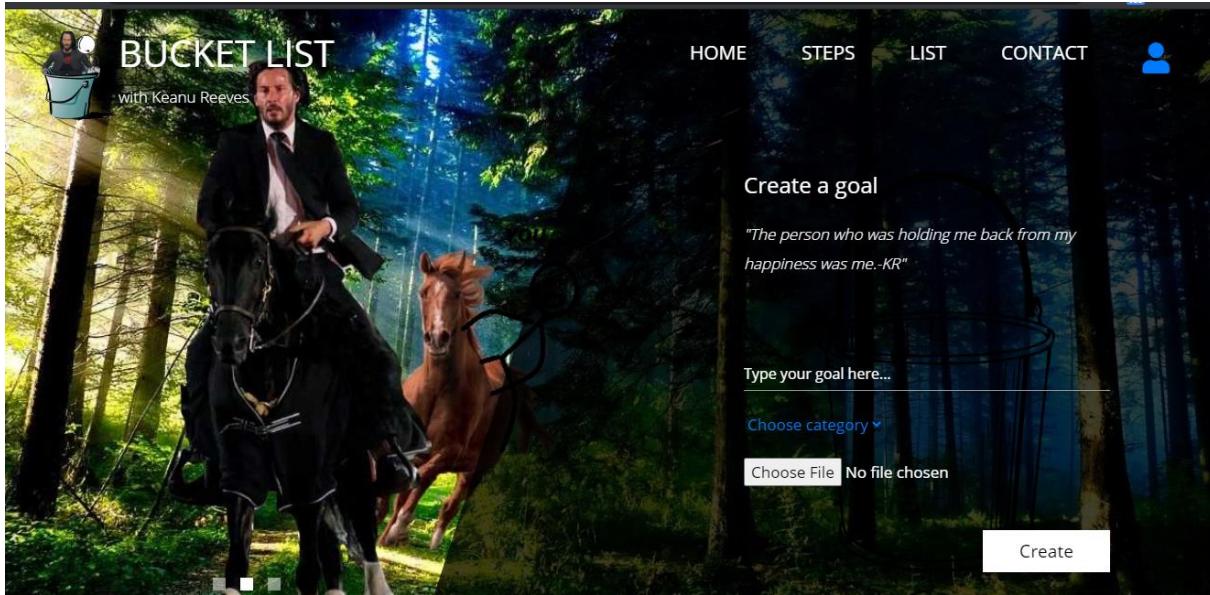
Logged in user:



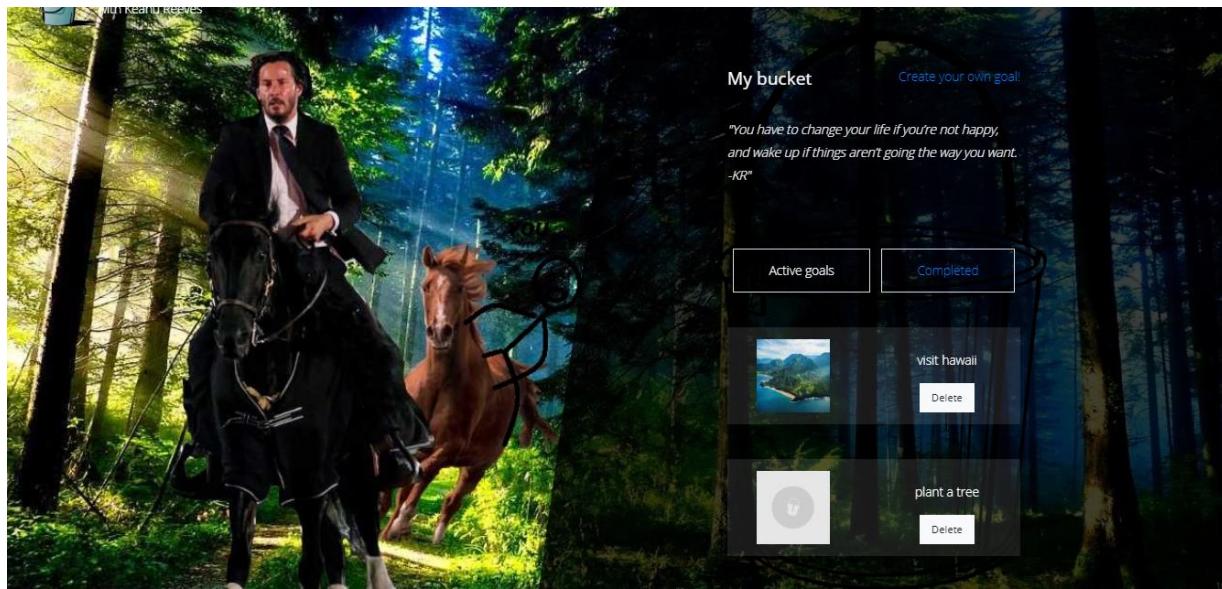
Ukoliko se korisnik ulogovao, prikazuje mu se lista na drugačiji način, s tim da ima: mogućnost dodavanja u kofu, pregled ukupnog dodavanja i završetka istog od strane drugih korisnika.Ukoliko je već izabrao tj dodao to u kofu, izbacuje obaveštenjce.



Kafa dinamički popunjena iz baze ulogovanog korisnika sa opcijom čekiranja kao urađenih ciljeva i brisanja. Pri čekiranju, se prebacuje u completed listu.



Ukoliko korisnik nije zadovoljan ponudom, može sam da popuni listu i inspiriše druge dodavajući novu! (praćeno klijentskom i serverskom proverom vrednosti).

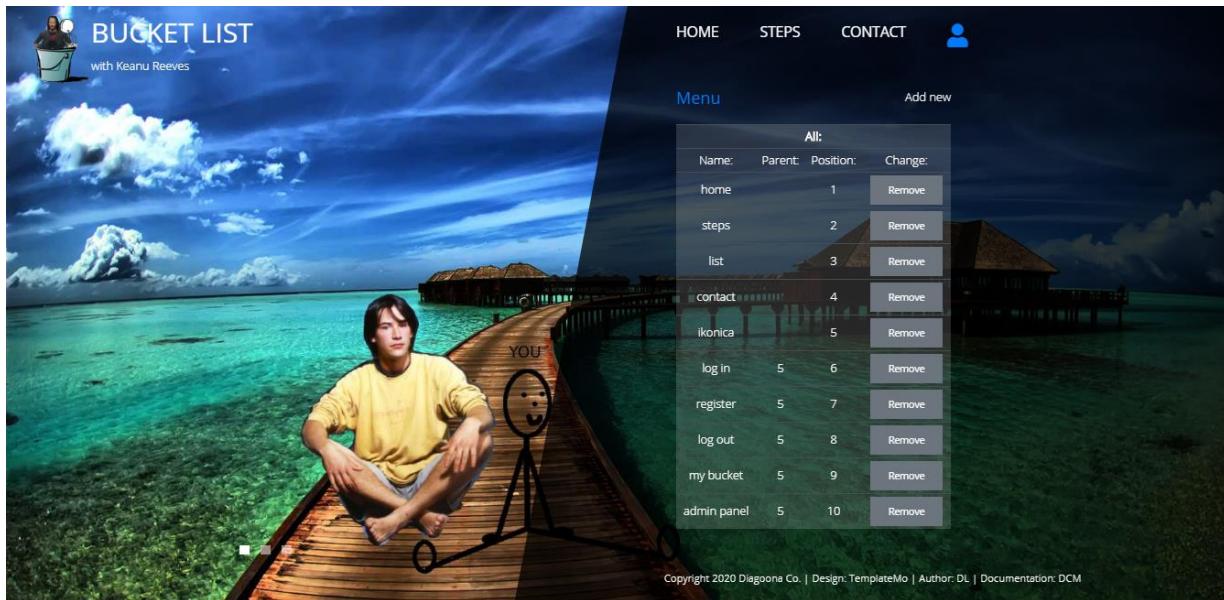


Urađeni ciljevi mogu samo da se izbrišu.

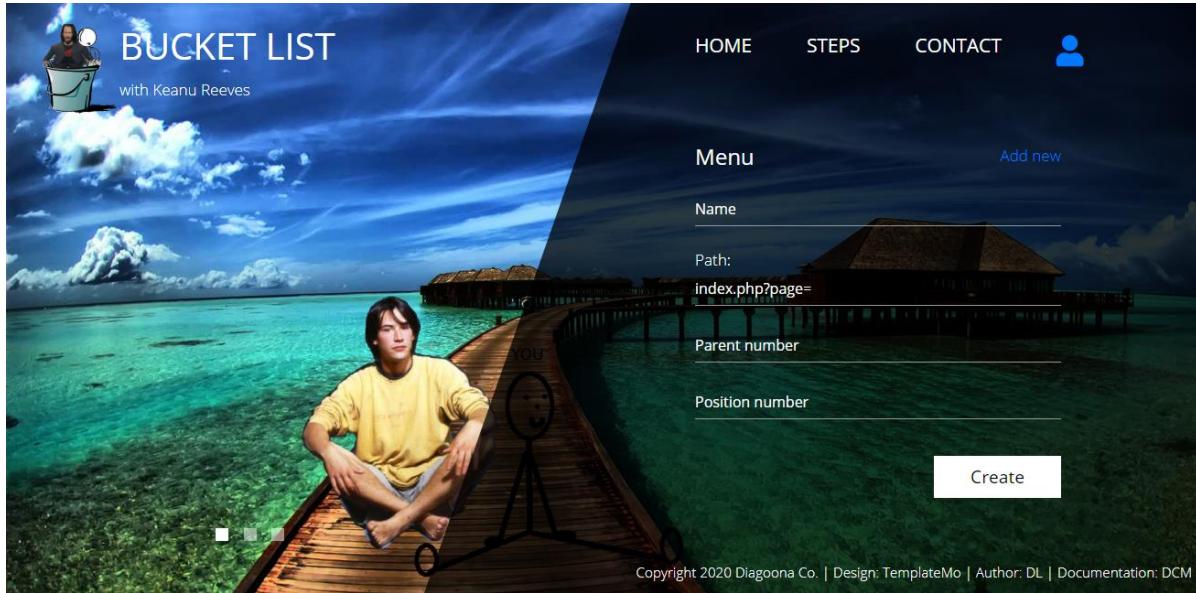
Logged in admin:



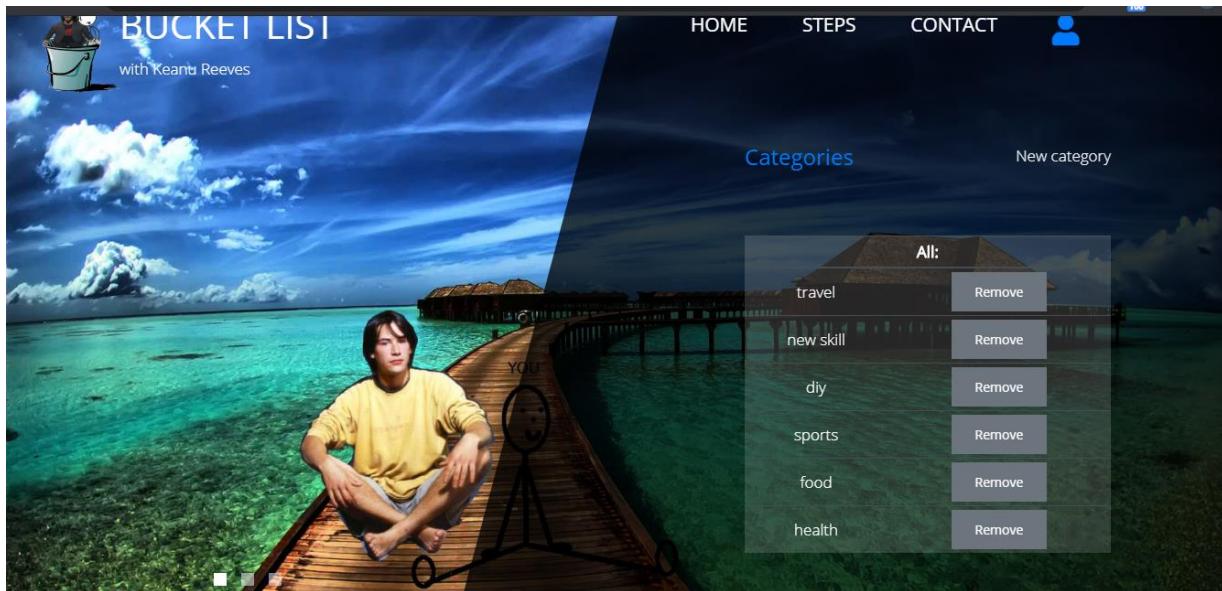
Prikaz tabela kojima može da se upravlja, procenat pristupa stranicama I trenutan broj ulogovanih usera.*Sve dinamički dovedeno iz baze u celom panelu



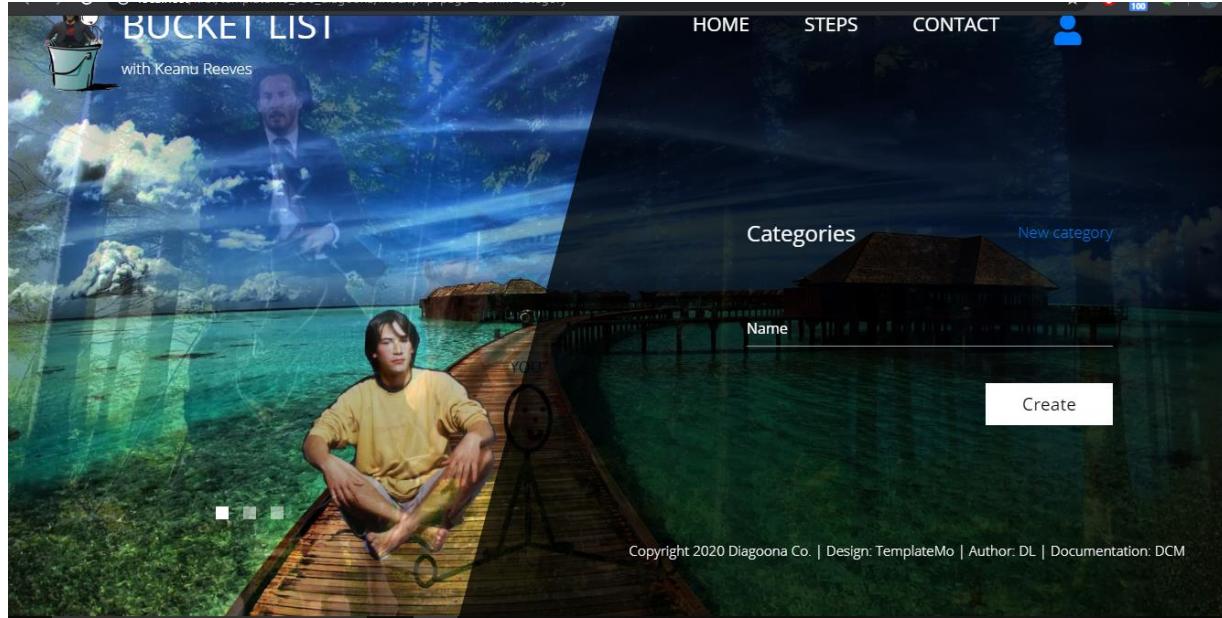
Prikaz svih elemenata menija, mogućnost brisanja I dodavanja novih.



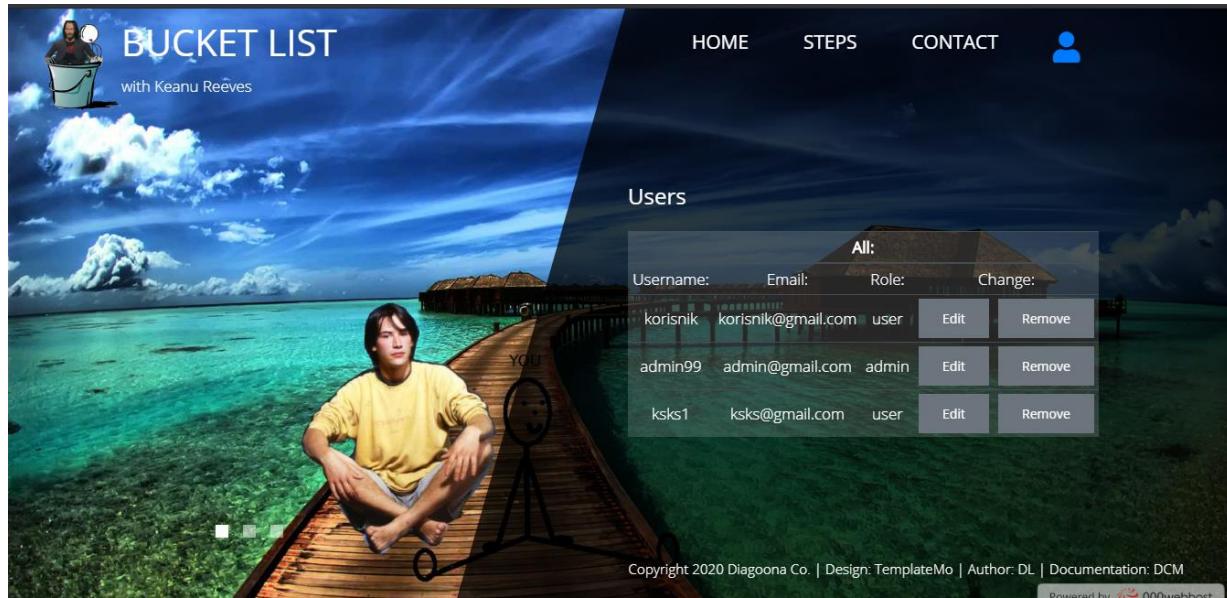
Kreiranje novog člana menija. (praćeno klijentskom I serverskom proverom vrednosti).



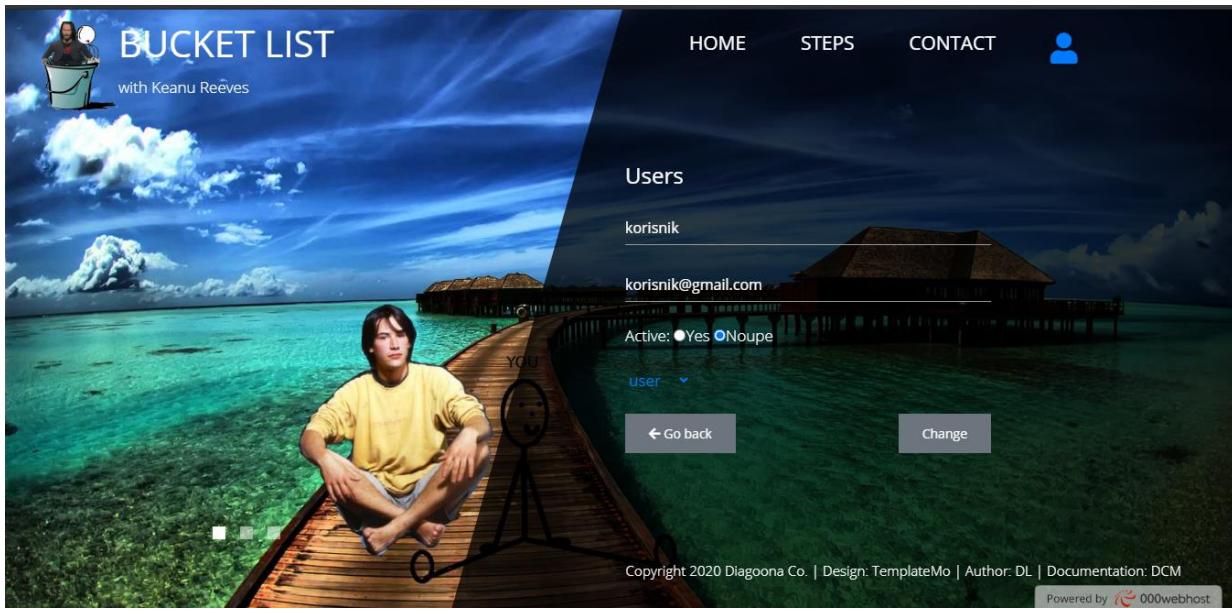
Brisanje I ubacivanje novih kategorija.



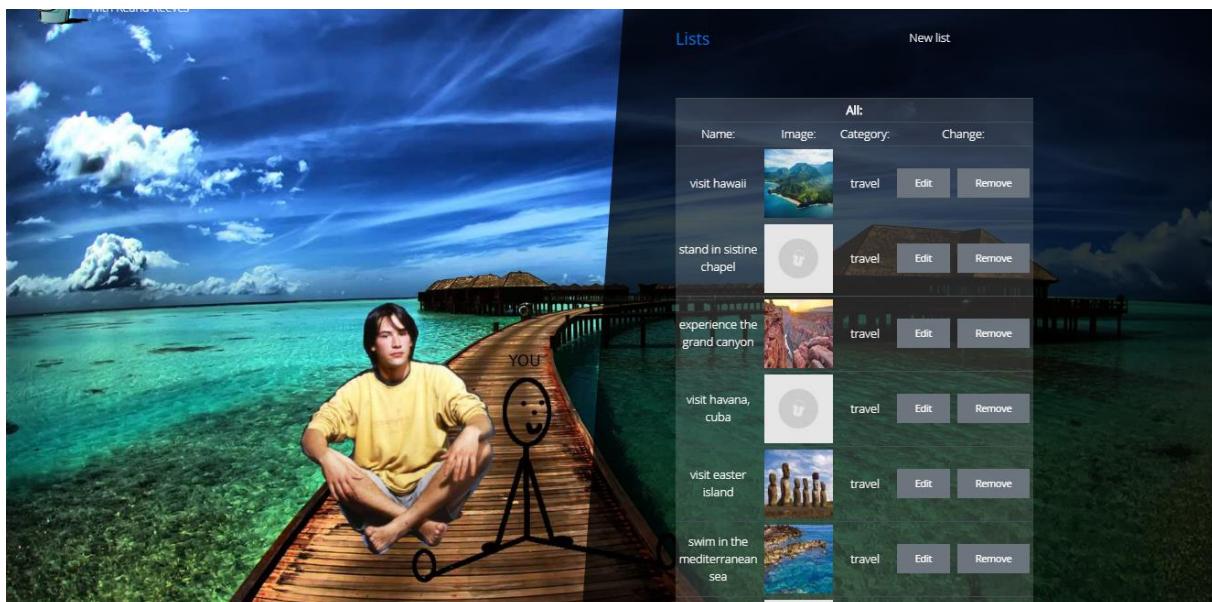
Dodavanje nove kategorije. (praćeno klijentskom I serverskom proverom vrednosti).



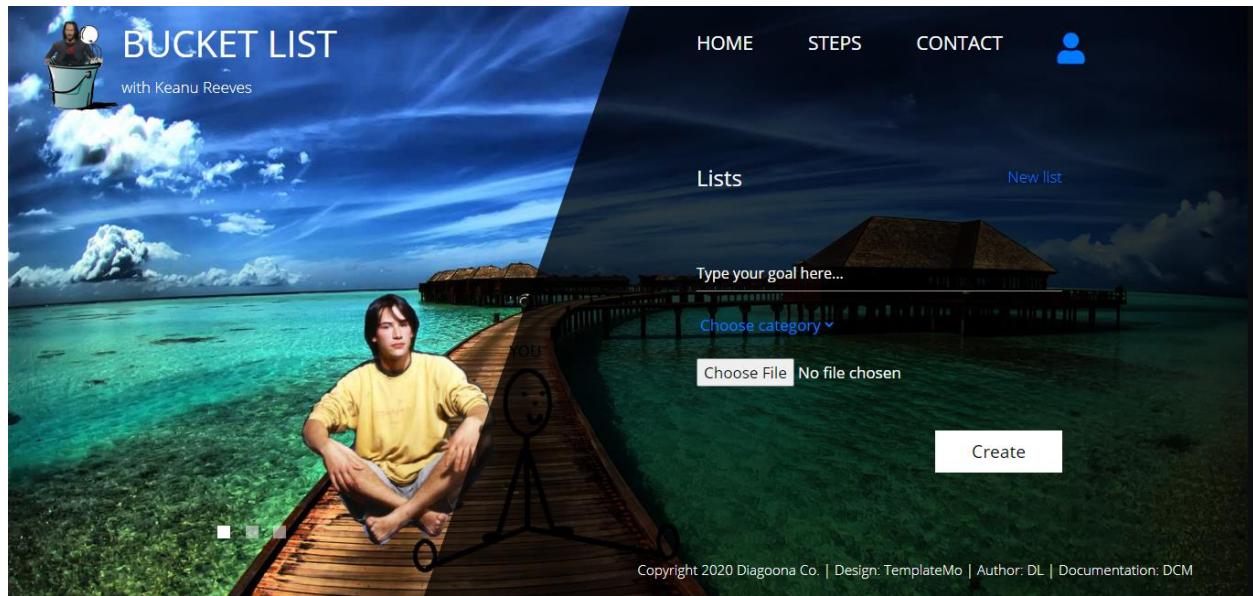
Prikaz , izmena I brisanje postojećih korisnika.



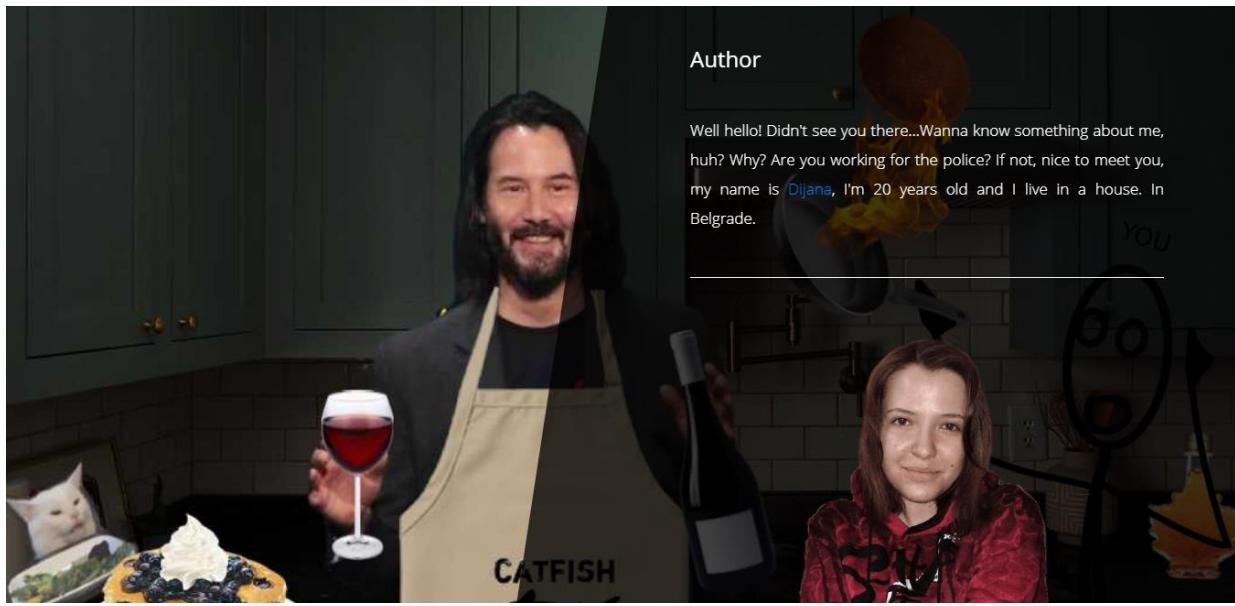
Izmena izabranog korisnika (praćeno klijentskom I serverskom proverom vrednosti).



Prikaz lista, mogućnost izmene, brisanja I dodavanje nove.



Dodavanje novog cilja u listu. (praćeno klijentskom I serverskom proverom vrednosti).



Prikaz autora I opcije za preuzimanje word verzije.

Kodovi

1) INDEX.PHP-HOME PAGE

```
<?php
session_start();
ob_start();
require_once("config/connection.php");
require_once("views/fixed/head.php");
?>

<!--sadrzaj -->
<?php
require_once("views/fixed/nav.php");

if (isset($_GET['page'])) {
    $page = $_GET['page'];
    switch ($page) {
        case 'home':
            require_once("views/home.php");
            break;
        case 'steps':
            require_once("views/steps.php");
            break;
        case 'list':
            require_once("views/list.php");
            break;
        case 'contact':
            require_once("views/contact.php");
            break;
        case 'register':
            require_once("views/register.php");
            break;
        case 'login':
            require_once("views/login.php");
            break;
        case 'logout':
            require_once("models/user/logout.php");
            break;
        case 'bucket':
            require_once("views/user/bucket.php");
    }
}
```

```
        break;
    case 'create-goal':
        require_once("views/user/create.php");
        break;
    case 'admin-panel':
        require_once("views/admin/panel.php");
        break;
    case 'admin-menu':
        require_once("views/admin/menu.php");
        break;
    case 'admin-category':
        require_once("views/admin/category.php");
        break;
    case 'admin-user':
        require_once("views/admin/user.php");
        break;
    case 'admin-list':
        require_once("views/admin/list.php");
        break;
    case 'author':
        require_once("views/author.php");
        break;

    default:
        require_once("views/home.php");
        break;
    }
} else {
    require_once("views/home.php");
}
?>
<!-- futer -->
<?php require_once("views/fixed/footer.php"); ?>
```

VIEWS**FIXED****1)HEAD.PHP**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="keywords" content="Keanu, Reeves, bucket, list, goals"/>
    <meta name="author" content="Dijana Lazic" target="_blank"/>
    <meta name="description" content="Try crossing things from your list but with
a twist! Add fun, smile and imagination with you and one and only - Keanu Reeves!
"/>
    <title>Bucket List with Keanu Reeves</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400"
rel="stylesheet" /> <!-- https://fonts.googleapis.com/ -->
    <link href="assets/css/bootstrap.min.css" rel="stylesheet" /> <!--
https://getbootstrap.com/ -->
    <link href="assets/fontawesome/css/all.min.css" rel="stylesheet" /> <!--
https://fontawesome.com/ -->
    <link href="assets/css/templatemo-diagoona.css" rel="stylesheet" />
    <link href="assets/css/style.css" rel="stylesheet" />
    <link rel="shortcut icon" href="assets/img/lists/icon.ico"/>

</head>
<body>
```

2)NAV.PHP

```
<div class="tm-container">
    <div class="tm-row pt-4">
        <div class="tm-col-left">
            <div class="tm-site-header media">
                <i class="fas fa-umbrella-beach fa-3x mt-1 tm-logo"></i>
                <div class="media-body">
                    <h1 class="tm-sitename text-uppercase">Bucket List</h1>
                    <p class="tm-slogan">with Keanu Reeves</p>
                </div>
            </div>
        </div>
    </div>
```

```

<div class="tm-col-right">
    <nav class="navbar navbar-expand-lg" id="tm-main-nav">
        <button class="navbar-toggler toggler-example mr-0 ml-auto"
type="button" data-toggle="collapse" data-target="#navbar-nav" aria-
controls="navbar-nav" aria-expanded="false" aria-label="Toggle navigation">
            <span><i class="fas fa-bars"></i></span>
        </button>
        <div class="collapse navbar-collapse tm-nav" id="navbar-nav">
            <ul class="navbar-nav text-uppercase">
                <?php
                    require_once("config/connection.php");
                    require_once("models/category/functions.php");
                    $meni = queryExecute(getMenu());
                    foreach ($meni as $m) : ?>
                        <?php if ($m->name == "ikonica") : ?>
                            <li class="nav-item text-center">
                                <a href="<?= $m->path ?>" id="juzer">
                                    <i class="fas fa-user fa-2x mt-1"></i>
                                </a>

                                <div class="mt-2">
                                    <?php
                                        $submenu = get_submenu($m-
>id_menu);
                                        foreach ($submenu as $sm) {
                                            if ($sm->name == "log in") {
                                                if
                                            }
                                            if ($sm->name == "register")
                                            if
                                        }
                                        if ($sm->name == "log out") {
                                            if
                                        }
                                        if ($sm->name == "my bucket")
                                            if
                                        elseif
                                            if
                                        }
                                    </?php
                                </div>
                            </li>
                        <?php if ($m->name == "ikonica") : ?>
                    </foreach>
                </?php
            </ul>
        </div>
    </nav>
</div>

```

```
if ($sm->name == "admin
panel") {
    if
(!isset($_SESSION['user'])) continue;
    elseif
($_SESSION['user']->role == 'user') continue;
}
echo "<div><a href='\$sm-
>path'> $sm->name</a></div>";
}
?>
</div>
</li>
<?php continue ?>
<?php endif ?>
<?php
    if ($m->name == "list") {
        if
(isset($_SESSION["user"])&&($_SESSION['user']->role == 'admin')) continue;
    } ?>
    <li class="nav-item">
        <a class="nav-link tm-nav-link" href="<?= $m-
>path; ?>"><?= $m->name; ?> <span class="sr-only">(current)</span></a>
    </li>
    <?php endforeach ?>
</ul>
</div>
</nav>
</div>
<input type="hidden" name="" id="idUser" value="<?php if
(isset($_SESSION["user"])) echo $_SESSION["user"]->uid;else echo "0"; ?>">
</div>
</div>
```

1)FOOTER.PHP

```
<div class="tm-row">
    <div class="tm-col-left text-center">
        <ul class="tm-bg-controls-wrapper">
            <li class="tm-bg-control active" data-id="0"></li>
            <li class="tm-bg-control" data-id="1"></li>
            <li class="tm-bg-control" data-id="2"></li>
        </ul>
    </div>
    <div class="tm-col-right tm-col-footer">
        <footer class="tm-site-footer text-right">
            <p class="mb-0">Copyright 2020 Diagoona Co.

                | Design: <a rel="nofollow" target="_parent"
                href="https://templatemo.com" class="tm-text-link">TemplateMo</a>
                | Author: <a href="index.php?page=author" class="tm-text-link"
                >DL</a>
                | Documentation: <a href="documentation.pdf" class="tm-text-link"
                target="_blank" >DCM</a>
            </p>
        </footer>
    </div>
</div>

<!-- Diagonal background design --&gt;
&lt;div class="tm-bg"&gt;
    &lt;div class="tm-bg-left"&gt;&lt;/div&gt;
    &lt;div class="tm-bg-right"&gt;&lt;/div&gt;
&lt;/div&gt;

&lt;/div&gt;
<!-- skripte --&gt;

&lt;script src="assets/js/jquery-3.4.1.min.js"&gt;&lt;/script&gt;
&lt;script src="assets/js/bootstrap.min.js"&gt;&lt;/script&gt;
&lt;script src="assets/js/jquery.backstretch.min.js"&gt;&lt;/script&gt;
&lt;script src="assets/js/templatemo-script.js"&gt;&lt;/script&gt;
&lt;script src="assets/js/main.js"&gt;&lt;/script&gt;
&lt;/body&gt;

&lt;/html&gt;</pre>
```

ADMIN**1)LIST.PHP**

```
<?php if ((isset($_SESSION['user']) && $_SESSION['user']->role == 'admin')) : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact m-0">
            <div class="tm-row d-flex justify-content-between mb-5">
                <a href="#alllists" class="mb-4 tm-content-title h2 aktiviraj">Lists</a>
                <a href="#newlist" class="text-light aktiviraj">New list</a>
            </div>
            <div>
                <div id="alllists" class="table-responsive-sm table-responsive-md table-responsive-lg">
                    <table class="table table-sm table-dark table-hover text-center">
                        <thead>
                            <tr>
                                <th scope="col" colspan="6">All:</th>
                            </tr>
                        </thead>
                        <tbody>
                            <tr>
                                <td>Name:</td>
                                <td>Image:</td>
                                <td>Category:</td>
                                <td colspan="2">Change:</td>
                            </tr>
                            <?php require_once("models/list/functions.php");
                            $lists = queryExecute(getListsForAdmin());
                            foreach ($lists as $l) :
                                ?>
                                <tr>
                                    <td><?= $l->name ?></td>
                                    <td>
                                        alt ?>">
                                    </td>
                                    <td><?= $l->catName ?></td>
                                    <td>
```

```

                <button type="button" class="btn btn-sm
my-auto btn-secondary mt-4 dugmeEditList" data-idlist=<?= $l->id_list ?>
name="">Edit</button>
            </td>
            <td>
                <button type="button" class="btn btn-sm
my-auto btn-secondary mt-4 dugmeRemoveList" data-idlist=<?= $l->id_list ?>
name="">Remove</button>
            </td>
        </tr>
    <?php endforeach ?>
</tbody>
</table>
</div>
<div id="newlist">
    <form action="models/list/insert.php" method="post"
enctype="multipart/form-data" onsubmit="return proveraCreate()">
        <div class="form-group mb-4">
            <input type="text" name="name" id="name" class="form-
control" placeholder="Type your goal here...">
        </div>
        <div class="form-group mb-4">
            <select name="category" id="category" class="text-
primary">
                <option value="0">Choose category</option>
                <?php
                    require_once("models/category/functions.php");
                    $categories = queryExecute(getCategories());
                    foreach ($categories as $c) :
                ?>
                <option value=<?= $c->id_category ?>><?=
$c->name ?></option>
            <?php endforeach ?>
            </select>
        </div>
        <div class="form-group mb-5">
            <input type="file" name="picture" id="picture"
class=""><span></span>
        </div>
        <div class="text-right">
            <button type="submit" id="dugmeCreate"
name="dugmeCreate" class="btn btn-big btn-primary">Create</button>
        </div>
    </form>
</div>

```

```

<div id="editlist">
    <div class="d-flex justify-content-center mb-5" >
        <img src="" alt="" id="editImage">
    </div>
    <form action="models/list/update.php" id="editForm"
enctype="multipart/form-data" method="post" onsubmit="return changeList()">
        <input type="hidden" name="idList" id="idList">
        <div class="form-group mb-4">
            <input type="file" name="editPicture" id="">
        </div>
        <div class="form-group mb-4">
            <input type="text" name="editName" id="editName"
class="form-control" placeholder="Name" required="" /><span></span>
        </div>
        <div class="form-group mb-4">
            <select name="editCategory" id="editCategory"
class="text-primary">
                <?php
                    $categories = queryExecute(getCategories());
                    foreach ($categories as $c) :
                ?>
                <option value=<?= $c->id_category ?>><?=
$c->name ?></option>
            <?php endforeach ?>
        </select>
    </div>
    <div class="form-group mb-4 mt-3 d-flex justify-content-between">
        <button type="button" class="btn btn-sm my-auto btn-secondary mt-4 backArrow" data-back="list" name="">
            <i class="fas fa-arrow-left mr-1"></i>Go
back</button>
        <button type="submit" class="btn btn-sm my-auto btn-secondary mt-4" id="changeList" name="changeList">Change</button>
    </div>
    </form>
</div>
<div id="poruka"></div>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>
```

1)CATEGORY.PHP

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role == 'admin') : ?>
    <div class="tm-row">
        <div class="tm-col-left"></div>
        <main class="tm-col-right tm-contact-main">
            <!-- Content -->
            <section class="tm-content tm-contact">
                <div class="tm-row d-flex justify-content-between">
                    <a href="#allcat" class="mb-4 tm-content-title h2 aktiviraj">Categories</a>
                    <a href="#newcat" class="text-light aktiviraj">New category</a>
                </div>
                <div class="mt-5">
                    <div id="allcat" class="table-responsive-sm table-responsive-md table-responsive-lg">
                        <table class="table table-sm table-dark table-hover text-center">
                            <thead>
                                <tr>
                                    <th scope="col" colspan="2">All:</th>
                                </tr>
                            </thead>
                            <tbody>
                                <?php
require_once("models/category/functions.php");
                                $categories = queryExecute(getCategories());
                                foreach ($categories as $c) :
                                    ?>
                                <tr>
                                    <td><?= $c->name ?></td>
                                    <td>
                                        <button type="button" class="btn btn-sm my-auto btn-secondary mt-4 dugmeRemoveCat" data-idcat="<?= $c->id_category ?>" name="">Remove</button>
                                    </td>
                                </tr>
                                <?php endforeach ?>
                            </tbody>
                        </table>
                    </div>
                    <div id="newcat">
                        <form id="" action="models/category/insert.php" method="POST" onsubmit="return proveraNewCat()">

```

```

<div class="form-group mb-4">
    <input type="text" name="name" id="name"
class="form-control" placeholder="Name" required="" /><span></span>
</div>
<div class="text-right">
    <button type="submit" id="dugmeCreateCat"
name="dugmeCreateCat" class="btn btn-big btn-primary mt-3 ">Create</button>
</div>
<?php
    if (isset($_SESSION["errors"])) {
        print_r($_SESSION["errors"]);
        unset($_SESSION["errors"]);
    }
    if (isset($_SESSION["success"])) {
        print_r($_SESSION["success"]);
    }
    ?>
</form>
</div>
</div>
<div id="poruka"></div>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>
```

1)PANEL.PHP

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role == 'admin')) : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact table-responsive-sm table-
responsive-md table-responsive-lg">
            <h2 class="mb-5 tm-content-title">Admin panel</h2>
            <table class="table table-sm table-dark table-hover text-center">
                <thead>
                    <tr>
                        <!-- <th scope="col">#</th> -->
                        <th scope="col">Manage</th>
                    </tr>
                </thead>
```

```
<tbody>
<tr>
    <!-- <th scope="row">3</th> -->
    <td><a href="index.php?page=admin-menu">menu</a></td>
</tr>
<tr>
    <!-- <th scope="row">1</th> -->
    <td><a href="index.php?page=admin-
category">categories</a></td>
</tr>
<tr>
    <!-- <th scope="row">2</th> -->
    <td><a href="index.php?page=admin-user">users</a></td>
</tr>
<tr>
    <!-- <th scope="row">3</th> -->
    <td><a href="index.php?page=admin-list">lists</a></td>
</tr>
</tbody>
</table>
<table class="table table-sm table-dark table-hover text-center my-
5">
    <thead>
        <tr>
            <!-- <th scope="col">#</th> -->
            <th scope="col" colspan="2">Visits</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Page:</td>
            <td>Per%:</td>
        </tr>
        <?php
            require_once("models/visit/functions.php");
            $visita = count_visits();
            // var_dump($visita);
            foreach ($visita as $i => $v) :
                ?>
                <tr>
                    <td><?= $i ?></td>
                    <td><?= $v ?></td>
                </tr>
            <?php endforeach ?>
    </tbody>
```

```

        </table>
    <table class="table table-sm table-dark table-hover text-center">
        <thead>
            <tr>
                <!-- <th scope="col">#</th> -->
                <th scope="col" colspan="2">Current number of logged
in:</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td><?= count_current_logged_in()->total; ?></td>
            </tr>
        </tbody>
    </table>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>
```

1)USER.PHP

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role == 'admin')) : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main p-0">
        <!-- Content -->
        <section class="tm-content tm-contact m-0 table-responsive-sm table-
responsive-md table-responsive-lg">
            <h2 class="mb-4 tm-content-title">Users</h2>
            <table class="table table-sm table-dark table-hover text-center"
id="user-table">
                <thead>
                    <tr>
                        <th scope="col" colspan="6">All:</th>
                    </tr>
                </thead>
                <tbody>
                    <td>Username:</td>
                    <td>Email:</td>
                    <td>Role:</td>
                    <td colspan="2">Change:</td>
                    <?php require_once("models/user/functions.php");
```

```

$users = queryExecute(getUsers());
foreach ($users as $u) :
    ?>
    <tr>
        <td><?= $u->username ?></td>
        <td><?= $u->email ?></td>
        <td><?= $u->role ?></td>
        <td>
            <button type="button" class="btn btn-sm my-auto
btn-secondary mt-4 dugmeEditUser" data-iduser="<?= $u->id_user ?>" name="">Edit</button>
        </td>
        <td>
            <button type="button" class="btn btn-sm my-auto
btn-secondary mt-4 dugmeRemoveUser" data-iduser="<?= $u->id_user ?>" name="">Remove</button>
        </td>
    </tr>
<?php endforeach ?>
</tbody>
</table>
<form action="" id="user-form">
    <input type="hidden" name="idUser" id="idUser" >
    <div class="form-group mb-4">
        <input type="text" name="username" id="username" class="form-
control" placeholder="Username" required="" /><span></span>
    </div>
    <div class="form-group mb-4">
        <input type="text" name="email" id="email" class="form-
control" placeholder="Email" required="" /><span></span>
    </div>
    <div class="form-group mb-4">
        Active:
        <input type="radio" name="active" id="active" required="" value="1" />Yes<span></span>
        <input type="radio" name="active" id="non-active" required="" value="0" />Noupe<span></span>
    </div>
    <div class="form-group mb-4">
        <select name="role" id="role" class="text-primary">
            <?php
                require_once("models/user/functions.php");
                $roles = queryExecute(getRoles());
                foreach ($roles as $r) :
            ?>

```

```

        <option value="= $r-&gt;id_role ?&gt;&lt;?= $r-&gt;name
?&gt;&lt;/option&gt;
            &lt;?php endforeach ?&gt;
        &lt;/select&gt;
    &lt;/div&gt;
    &lt;div class="form-group mb-4 d-flex justify-content-between"&gt;
        &lt;button type="button" class="btn btn-sm my-auto btn-secondary
mt-4 backArrow" data-back="user" name=""&gt;
            &lt;i class="fas fa-arrow-left mr-1"&gt;&lt;/i&gt;Go back&lt;/button&gt;
        &lt;button type="button" class="btn btn-sm my-auto btn-secondary
mt-4" id="changeUser" name=""&gt;Change&lt;/button&gt;
    &lt;/div&gt;
&lt;/form&gt;
&lt;div id="poruka"&gt;&lt;/div&gt;
&lt;/section&gt;
&lt;/main&gt;
&lt;/div&gt;
&lt;?php else : header("Location:index.php?page=home"); ?&gt;
&lt;?php endif ?&gt;</pre

```

5)MENU.PHP

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role == 'admin')) : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact m-0 table-responsive-sm table-
responsive-md table-responsive-lg">
            <div class="tm-row d-flex justify-content-between">
                <a href="#allmenu" class="mb-4 tm-content-title h2
aktiviraj">Menu</a>
                <a href="#newmenu" class="text-light aktiviraj">Add new</a>
            </div>
            <div id="allmenu">
                <table class="table table-sm table-dark table-hover text-center"
id="all-menu">
                    <thead>
                        <tr>
                            <th scope="col" colspan="6">All:</th>
                        </tr>
                    </thead>
                    <tbody>
                        <td>Name:</td>
```

```

<td>Parent:</td>
<td>Position:</td>
<td>Change:</td>
<?php require_once("models/list/functions.php");
$menu = queryExecute(getWholeMenu());
foreach ($menu as $m) :
    ?>
    <tr>
        <td><?= $m->name ?></td>
        <td><?= $m->parent ?></td>
        <td><?= $m->position ?></td>
        <td>
            <button type="button" class="btn btn-sm my-
auto btn-secondary mt-4 dugmeRemoveMenu" data-idmenu="<?= $m->id_menu ?>" name="">Remove</button>
        </td>
    </tr>
    <?php endforeach ?>
</tbody>
</table>
</div>
<div id="newmenu">
    <form id="" action="models/list/insert.php" method="POST"
onsubmit="return proveraNewMenu()">
        <div class="form-group mb-4">
            <input type="text" name="nameMenu" id="nameMenu"
class="form-control" placeholder="Name" /><span></span>
        </div>
        <div class="form-group mb-4">
            Path:
            <input type="text" name="namePath" id="namePath"
class="form-control" placeholder="Path" value="index.php?page=" /><span></span>
        </div>
        <div class="form-group mb-4">
            <input type="text" name="numParent" id="numParent"
class="form-control" placeholder="Parent number" /><span></span>
        </div>
        <div class="form-group mb-4">
            <input type="text" name="numPosition" id="numPosition"
class="form-control" placeholder="Position number" /><span></span>
        </div>
        <div class="text-right">
            <button type="submit" id="dugmeCreateMenu"
name="dugmeCreateMenu" class="btn btn-big btn-primary mt-3 ">Create</button>
        </div>
    </form>
</div>

```

```

<?php
if (isset($_SESSION["errors"])) {
    print_r($_SESSION["errors"]);
    unset($_SESSION["errors"]);
}
if (isset($_SESSION["success"])) {
    print_r($_SESSION["success"]);
}
?>
</form>
</div>
<div id="poruka"></div>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>

```

USER**1)BUCKET.PHP**

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role =='user') : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main" id="izaBucket">
        <!-- Content -->
        <section class="tm-content tm-contact">
            <div class="tm-row d-flex justify-content-between">
                <h2 class="mb-4 tm-content-title">My bucket</h2>
                <a href="index.php?page=create-goal">Create your own
goal!</a>
            </div>
            <p class="mb-85 mt-3 font-italic">"You have to change your life
if you're not happy, and wake up if things aren't going the way you want. - 
KR"</p>
            <div class="d-flex justify-content-around">
                <a href="#todo" class="border px-5 py-3 aktiviraj
todo">Active goals</a>
                <a href="#done" class="border px-5 py-3 text-light aktiviraj
done">Completed</a>
            </div>
            <div>
                <div class="" id="todo">
                    <?php

```

```

        require_once("models/bucket/functions.php");
        //active goals
        $agoals = getGoals($_SESSION["user"]->uid, 0);
        ispisiGoals($agoals);
    ?>
</div>
<div class="" id="done">
<?php
        require_once("models/bucket/functions.php");
        //completed goals
        $cgoals = getGoals($_SESSION["user"]->uid, 1);
        ispisiGoals($cgoals); ?>
</div>
</div>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>
```

1)CREATE.PHP

```

<?php if ((isset($_SESSION['user'])) && $_SESSION['user']->role =='user') : ?>
<div class="tm-row">
<div class="tm-col-left"></div>
<main class="tm-col-right tm-contact-main">
    <!-- Content -->
    <section class="tm-content tm-contact" id="izaBucket" >
        <h2 class="mb-4 tm-content-title">Create a goal</h2>
        <p class="mb-85 font-italic">"The person who was holding me back
from my happiness was me.-KR"</p>
        <form action="models/list/insert.php" method="post"
enctype="multipart/form-data" onsubmit="return proveraCreate()">
            <div class="form-group mb-4">
                <input type="text" name="name" id="name" class="form-
control" placeholder="Type your goal here...">
            </div>
            <div class="form-group mb-4">
                <select name="category" id="category" class="text-
primary" >
                    <option value="0">Choose category</option>
                    <?php
                        require_once("models/category/functions.php");
                        $categories = queryExecute(getCategories());
```

```

        foreach ($categories as $c) :
            ?
        <option value=<?= $c->id_category ?>><?= $c-
>name ?></option>
            <?php endforeach ?>
        </select>
    </div>
    <div class="form-group mb-5">
        <input type="file" name="picture" id="picture"
class=""><span></span>
    </div>
    <div class="text-right">
        <button type="submit" id="dugmeCreate" name="dugmeCreate"
class="btn btn-big btn-primary">Create</button>
    </div>
</form>
</section>
</main>
</div>
<?php else : header("Location:index.php?page=home"); ?>
<?php endif ?>
```

1)AUTHOR.PHP

```

<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right">
        <section class="tm-content">
            <h2 class="mb-5 tm-content-title">Author</h2>
            <p class="mb-5 text-justify">Well hello! Didn't see you there...Wanna
know something about me, huh? Why? Are you working for the police? If not, nice
to meet you, my name is <span><a href="https://znaciporfolio.netlify.com"
target="_blank" title="Portfolio" class="crveno">Dijana</a></span>, I'm 20 years
old and I live in a house. In Belgrade.</p>
            <hr class="mb-5">
            
            <hr class="mb-5">
            <p class="mb-5 text-justify">
                After successfully finishing both kindergarten and elementary
school, I decided to continue my education at the Philological High School
because I really liked learning about languages. I was so good that I managed to
master six of them ( English, French, Latin, Serbian, Bosnian and Montenegrin),
but I realized I needed a new challenge...I always wanted to talk to the machines
        </section>
    </main>
</div>
```

and the best way to do that is by learning computer languages. I will be able to speak to my laptop, right?

```
</p>
<div class="text-right">
    <form action="models/file/word.php" method="post">
        <button type="submit" id="dugmeWord" name="dugmeWord"
class="btn btn-big btn-primary">Get this in word!</button>
    </form>
</div>
</section>
</main>
</div>
```

1)CONTACT.PHP

```
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact">
            <h2 class="mb-4 tm-content-title">Contact Us</h2>
            <p class="mb-85 font-italic">"Sometimes we get so caught up in our
daily lives that we forget to take the time out to enjoy the beauty in life.-  
KR"</p>
            <form id="" action="" method="POST">
                <div class="form-group mb-4">
                    <input type="text" name="name" id="name" class="form-control"
placeholder="Name" required="" />
                    <span></span>
                </div>
                <div class="form-group mb-4">
                    <input type="email" name="email" id="email" class="form-
control" placeholder="Email" required="" /><span></span>
                </div>
                <div class="form-group mb-5">
                    <textarea rows="6" name="message" id="message" class="form-
control" placeholder="Message..." required="" /></textarea><span></span>
                </div>
                <div class="text-right">
                    <button type="button" id="dugmeContact" name="dugmeContact"
class="btn btn-big btn-primary">Send It</button>
                </div>
                <div id="rispone"></div>
            </form></section></main></div>
```

1)HOME.PHP

```
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right">
        <section class="tm-content text-justify">
            <h2 class="mb-5 tm-content-title">Make your dreams come true...</h2>
            <p class="mb-5">...or at least virtually. Ever wondered how it would look/feel like going on a trip to Sahara desert or making a dessert with THE coolest person on Earth? Well, since I'm busy, we offer U this site to plan that with ONE and ONLY... *drum roll please* *did u do it?* *ok, nice!* <span class="font-weight-bold"> <br> KKKKKKEEEEAAAANUUUU...REEEEEVEES!!!!!
</span></p>
            <hr class="mb-5">
            <p class="mb-5">Now pack up your things and enjoy the endless possibilities that could happen with that legend...just don't bring a pencil (trust me).</p>
            <a href="index.php?page=steps" class="btn btn-primary">Start your journey...</a>
        </section>
    </main>
</div>
```

1)LIST.PHP

```
<?php if (isset($_SESSION["user"])) && $_SESSION['user']->role == 'admin') : ?>
<?php header("Location:../index.php?page=home");?>
<?php else : ?>
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right">
        <div id="filterisorteri" class="d-flex justify-content-between tm-content">
            <select name="" id="filter" class="text-primary">
                <option value="0">Choose category</option>
                <?php
                    require_once("models/category/functions.php");
                    $categories = queryExecute(getCategories());
                    foreach ($categories as $c) :
                ?>
                <option value="<?= $c->id_category ?>"><?= $c->name ?>
            ?>
            <?php endforeach ?>
        </select>
    </main>
</div>
```

```

<select name="" id="sort" class="text-primary">
    <option value="0">Sort by:</option>
    <option value="1">Name A-Z</option>
    <option value="2">Name Z-A</option>
</select>
</div>
<section class="tm-content" id="liste">
    <?php
        require_once("models/list/functions.php");
        //zukupno
        $lists = getLists(0);
        foreach ($lists as $l) :
            ?>
            <div class="media my-3 mb-5 tm-service-media okreci">
                alt ?>" class="tm-service-img">
                    <div class="media-body tm-service-text">
                        <h2 class="mt-5 tm-content-title"><?= $l->listName ?></h2>
                        <!-- ovde ce biti ukuono, uradjeno i dodaj -->
                        <?php if (!isset($_SESSION['user'])) : ?>
                            <p class="text-danger">Gotta login to have your
bucket filled in!</p>
                        <?php else : ?>
                            <p>
                                <button type="button" class="btn btn-secondary
mt-4 plavo dugmeBucket" data-idlist="<?= $l->id_list ?>" name="">Pick
up!</button>
                                <i class="fas fa-plus fa-1x ml-5"></i> <span
class="font-italic" ><?php $result=getAddedStatistics($l->id_list); if($result)
echo $result->total; else echo 0;?> Added</span>
                                <i class="fas fa-check-square fa-1x ml-1 "></i>
<span class="font-italic" ><?php $result=getDoneStatistics($l->id_list);
if($result) echo $result->total; else echo 0;?> Done</span>
                            </p>
                        <?php endif ?>
                    </div>
                </div>
            <?php endforeach ?>
        </section>
        <nav aria-label="Page navigation example" class="d-flex justify-content-
center">
            <ul class="pagination" id="paginacija">

```

```

        </ul>
    </nav>
    <div class="text-center mt-5">
        <form action="models/file/excel.php" method="post">
            <button type="submit" id="dugmeExcel" name="dugmeExcel"
class="btn btn-big btn-primary">Get this in excel!</button>
        </form>
    </div>
</main>
</div>
<?php endif ?>
```

1)LOGIN.PHP

```

<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact">
            <h2 class="mb-4 tm-content-title">Log in</h2>
            <p class="mb-5 font-italic">"Grief changes shape but it never ends.-KR"</p>
            <form id="" action="models/user/login.php" method="POST"
onsubmit="return proveraLogin()">
                <div class="form-group mb-4">
                    <input type="email" name="email" id="email" class="form-control" placeholder="Email" required="" /><span></span>
                </div>
                <div class="form-group mb-5">
                    <input type="password" name="password" id="password" class="form-control" id="" placeholder="Password" required="" /><span></span>
                </div>
                <div class="text-right">
                    <button type="submit" id="dugmeLogin" name="dugmeLogin"
class="btn btn-big btn-primary">Log in</button>
                </div>
            <?php
                if(isset($_SESSION["errors"])){
                    print_r($_SESSION["errors"]);
                    unset($_SESSION["errors"]);
                }
                if(isset($_SESSION["user"])){
                    print_r($_SESSION["user"]);
                }
            
```

```
?>
</form>
</section>
</main>
</div>
```

1)REGISTER.PHP

```
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right tm-contact-main">
        <!-- Content -->
        <section class="tm-content tm-contact">
            <h2 class="mb-4 tm-content-title">Register</h2>
            <p class="mb-5 font-italic">"The simple act of paying attention can
take you a long way.-KR"</p>
            <form id="contact-form" action="" method="POST">
                <div class="form-group mb-4">
                    <input type="text" name="username" id="username" class="form-
control" placeholder="Username" required="" /><span></span>
                </div>
                <div class="form-group mb-4">
                    <input type="email" name="email" id="email" class="form-
control" placeholder="Email" required="" /><span></span>
                </div>
                <div class="form-group mb-5">
                    <input type="password" name="password" id="password"
class="form-control" id="" placeholder="Password" required="" /><span></span>
                </div>
                <div class="text-right">
                    <button type="button" id="dugmeRegister" class="btn btn-big
btn-primary">Register</button>
                </div>
                <div id="message"></div>
            </form>
        </section>
    </main>
</div>
```

1)STEPS.PHP

```
<div class="tm-row">
<div class="tm-row">
    <div class="tm-col-left"></div>
    <main class="tm-col-right">
        <section class="tm-content tm-about text-justify">
            <h2 class="mb-5 tm-content-title">How it works?</h2>
            <hr class="mb-4">
            <div class="media my-3">
                <i class="fas fa-user-plus fa-3x p-3 mr-4"></i>
                <div class="media-body">
                    <p>First of course, technicality...Can't travel without a passport or repair a bike without tools, right? So, create your entry to this site by <a href="">Making account</a> and <a href="">Login' in</a></p>
                </div>
            </div>
            <div class="media my-3">
                <i class="fas fa-book fa-3x p-3 mr-4"></i>
                <div class="media-body">
                    <p>Well, maps and stores are always open, so you can explore <a href="">here</a> even before the first step, but the next one is the key (point of your boiling excited blood) </p>
                </div>
            </div>
            <div class="media my-3">
                <i class="fab fa-creative-commons-share fa-3x p-3 mr-4"></i>
                <div class="media-body">
                    <p>Once you settled down for some ideas (not Keanu, he's taken), you can look how many people share same desires as you, but most importantly, you can see your bucket full of <sub>hypothetical</sub> joy while eating your own full of wings... </p>
                </div>
            </div>
            <div class="media my-3">
                <i class="fas fa-bookmark fa-3x p-3 mr-4"></i>
                <div class="media-body">
                    <p>And last but not least, i'm going to write this (det im a rapper?) No, gotta stay humble while picking a jungle to visit with Keanu and his dog too! Where were I? Right, well, once you added things from list to your bucket, you can keep track of them and see what you've done or what is still left to cross off the <sub>bucket</sub> list... And of course, enjoy thy journey called life! <i class="fa fa-heart"></i></p>
                </div>
            </div></section></main></div>
```

MODELS**BUCKET****1)CHANGE.PHP**

```
<?php
ob_start();

require_once("../config/connection.php");
require_once("functions.php");

//manage bucket

if(isset($_POST["dugmeDone"])){
    $id=$_POST["idBucketDo"];

    try {
        markAsDone($id);
        header("Location:../../index.php?page=bucket");
    } catch (PDOException $e) {
        catchErrors($e->getMessage());
        $_SESSION["errors"] = "Something wrong with the database, we're
sorry...";
        header("Location:../../index.php?page=bucket");
    }
}
else{
    header("Location:../../index.php?page=bucket");
}

if(isset($_POST["dugmeDelete"])){
    $id=$_POST["idBucketDelete"];

    try {
        deleteIzBucket($id);
        header("Location:../../index.php?page=bucket");
    } catch (PDOException $e) {
        catchErrors($e->getMessage());
        $_SESSION["errors"] = "Something wrong with the database, we're
sorry...";
        header("Location:../../index.php?page=bucket");
    }
}
else{
    header("Location:../../index.php?page=bucket");
}?>
```

1)FUNCTIONS.PHP

```

<?php
    function getGoals($id_user,$done){
        global $conn;
        $agoal = $conn -> prepare("SELECT *, l.name as listName FROM bucket b
INNER JOIN list l on b.list_id=l.id_list INNER JOIN image i on
l.image_ID=i.id_image WHERE b.user_ID=? and b.done=?");
        $agoal -> execute([$id_user,$done]);
        if ($agoal->rowCount() > 0) {
            $agoal = $agoal->fetchAll();
            return $agoal;
        }
    }

    function ispisiGoals($data){
        if ($data) {
            foreach ($data as $d) {
                echo "
                    <div class='tm-row my-5 transparentno py-3'>
                    <div class='tm-col-left'>
                        <img src='assets/img/lists/$d->src_small' alt='$d->alt'>
                    </div>
                    <div class='tm-col-right px-1 mt-auto text-center'>
                        <p>$d->listName</p>
                        <div class='d-flex justify-content-center'>
                            <form method='post'
action='models/bucket/change.php'>
                                ";
                if($d->done==0){
                    echo "<input type='hidden' value='".$d->id_bucket'
name='idBucketDo' />
                    <button type='submit' data-idbaket='' name='dugmeDone'
class='btn btn-success btn-sm px-3 plavo '>Fullfilled?</button>";
                }
                echo "
                    <input type='hidden' value='".$d->id_bucket'
name='idBucketDelete' />
                    <button type='submit' name='dugmeDelete'
class='btn btn-light border-0 btn-sm px-3'>Delete</button>
                </form>
            </div>
        </div>
    ";
}

```

```
        }
    }
else{
    echo "<h5>None taken</h5>";
}
}

function proveraUbucket($id_list,$id_user){
    global $conn;
    $pub = $conn->prepare("SELECT * FROM bucket WHERE list_ID=? and user_ID=?");
    $pub->execute([$id_list,$id_user]);
    if ($pub->rowCount() > 0) return false;
    return true;
}

function insertUbucket($id_list,$id_user){
    global $conn;
    $date=date("Y-m-d H:i:s",time());
    $ubaket = $conn -> prepare("INSERT into bucket(user_ID,list_ID,date,done) VALUES(?, ?, ?, 0)");
    $ubaket -> execute([$id_user,$id_list,$date]);
}

function markAsDone($id_bucket){
    global $conn;
    $done = $conn -> prepare("UPDATE bucket SET done=1 WHERE id_bucket=?");
    $done -> execute([$id_bucket]);
}

function deleteIzBucket($id_bucket){
    global $conn;
    $delete = $conn -> prepare("DELETE FROM bucket WHERE id_bucket=?");
    $delete -> execute([$id_bucket]);
}
?>
```

1)INSERT.PHP

```
<?php
    ob_start();
if (isset($_POST['send'])) {

    require_once("../config/connection.php");
    require_once("functions.php");
    header("Content-type:application/json");
    $data=[];
    $idList=$_POST["idList"];
    $idUser=$_POST["idUser"];
    try {
        if(proveraUbucket($idList,$idUser)){
            insertUbucket($idList,$idUser);
            // $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
            $data=[ "message"=>"Successfully inserted in bucket!"];
            $code=201;
        }
        else{
            $data=[ "message"=>"Already chosen..."];
            $code=409;
        }
    }

    } catch (PDOException $e) {
        $code = 500;
        catchErrors($e->getMessage());
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
    }

    http_response_code($code);
    echo json_encode($data);
}
else{
    header("Location:../../index.php?page=bucket");
}
?>
```

CATEGORY**1)DELETE.PHP**

```
<?php
ob_start();
if (isset($_POST['send'])) {

    require_once "../../config/connection.php";
    require_once "functions.php";
    header("Content-type:application/json");

    $id = $_POST['idCat'];

    $data=[ ];

    try {
        deleteCat($id);
        $data=[ "message"=>"Successfully deleted user!"];
        $code = 201;
    } catch (PDOException $e) {
        $code = 500;
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
        catchErrors($e->getMessage());
    }

    http_response_code($code);
    echo json_encode($data);
}
else{
    header("Location:../../index.php?page=admin-category");
}

?>
```

1)FILTERnSORT.PHP

```
<?php
ob_start();
if (isset($_POST['send'])) {

    require_once("../../config/connection.php");
    require_once("functions.php");
    require_once("../../models/list/functions.php");
    header("Content-type:application/json");
```

```
$data=[ ];  
  
try {  
    $page = ($_POST['page'] - 1) * 3;  
    $idFilterKat = $_POST['idFilterKat'];  
    $idSort = $_POST['idSort'];  
    // $data = filterCat($id);  
    $query= "SELECT *, l.name as listName FROM list l INNER JOIN image i on  
l.image_ID=i.id_image";  
    //filtriranje  
    if($idFilterKat!=0){  
        $query.= " WHERE category_ID=? ORDER BY ";  
    }  
    else{  
        $query.=" ORDER BY ";  
    }  
    //sortiranje  
    switch ($idSort) {  
        case '0':  
            $query.=" id_list ";  
            break;  
        case '1':  
            $query.=" listName ";  
            break;  
        case '2':  
            $query.=" listName DESC ";  
            break;  
  
        default:  
            $query.=" id_list ";  
            break;  
    }  
    $code=201;  
    $query.= " LIMIT $page,3";  
    $priprema=$conn->prepare($query);  
    $priprema->execute([$idFilterKat]);  
    $all=$priprema->fetchAll();  
    foreach($all as $al){  
        $added[] = getAddedStatistics($al->id_list);  
        $done[] = getDoneStatistics($al->id_list);  
    }  
    array_push($data,$all);  
    array_push($data,$added);  
    array_push($data,$done);  
}
```

```

} catch (PDOException $e) {
    $code = 500;
    $data=[ "error"=>"Something wrong with the database, we're sorry..."];
    catchErrors($e->getMessage());
}

http_response_code($code);
echo json_encode($data);
}
else{
    header("Location: ../../index.php?page=list");
}
?>

```

1)FUNCTIONS.PHP

```

<?php
    function getMenu(){
        return ("select * from menu where parent is null order by position");
    }

    function getSubMenu($id_menu){
        global $conn;
        $submenu = $conn->prepare("SELECT * FROM menu WHERE parent=? ORDER BY
position");
        $submenu->execute([$id_menu]);
        if ($submenu->rowCount() > 0) {
            $submenu = $submenu->fetchAll();
            return $submenu;
        }
    }

    function getWholeMenu(){
        return ("select * from menu order by position");
    }

    function getCategories(){
        return ("select * from category");
    }

    function insertNewCat($name){
        global $conn;
        $new_cat = $conn->prepare("INSERT INTO category(name) VALUES (?)");
        $new_cat->execute([$name]);
    }

```

```
}

function deleteCat($id){
    global $conn;
    $del_cat = $conn->prepare("DELETE FROM category WHERE id_category=?");
    $del_cat->execute([$id]);
}
?>
```

1)INSERT.PHP

```
<?php

if(isset($_POST['dugmeCreateCat'])){

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

    $name=$_POST["name"];

    $errors = [];

    try {
        $result=insertNewCat($name);
        $_SESSION["success"] = "Successfully inserted!";
        header("Location: ../../index.php?page=admin-category");
    } catch (PDOException $e) {
        //sve ovo uvek u txt error fajl
        catchErrors($e->getMessage());
        $_SESSION["errors"] = "Something wrong with the database, we're sorry...";
        header("Location: ../../index.php?page=admin-category");
    }
}
else{
    header("Location: ../../index.php?page=admin-category");
}
```

FILE

*Uprkos nesradnji hostinga sa clasom COM, problem preuzimanja datoteka se resio na sledeci nacin:

1)EXCEL.PHP

```
<?php
    require_once ("../../config/connection.php");
    if(isset($_POST["dugmeExcel"])){
        $lists=queryExecute("SELECT l.id_list as id,l.name as listName,c.name as
catName,i.src as imgPath FROM list l INNER JOIN category c on
l.category_ID=c.id_category INNER JOIN image i on l.image_ID=i.id_image");

        // $excel_file = new COM("excel.Application");
        // $excel_file -> Visible = true;
        // $excel_file -> DisplayAlerts = 1;
        // $workbook = $excel_file -> Workbooks -> Add();
        // $worksheet = $excel_file -> Worksheets("Sheet1");
        // $worksheet -> activate;

        // $niz=[ "Id"=>"A1","Name"=>"B1","Category"=>"C1","Image"=>"D1"];
        // foreach($niz as $index=>$n){
        //     $do = $worksheet -> Range("$n");
        //     $do -> activate;
        //     $do -> Value = "$index";
        // }

        // $brojac = 2;
        // foreach($lists as $l)
        // {
        //     $cell_id = $worksheet ->Range("A{$brojac}");
        //     $cell_id -> activate;
        //     $cell_id -> Value = $l -> id;
        //     $cell_name = $worksheet -> Range("B{$brojac}");
        //     $cell_name -> activate;
        //     $cell_name -> Value = $l -> listName;
        //     $cell_category = $worksheet ->Range("C{$brojac}");
        //     $cell_category -> activate;
        //     $cell_category -> Value = $l -> catName;
        //     $cell_img = $worksheet -> Range("D{$brojac}");
        //     $cell_img -> activate;
        //     $cell_img -> Value = $l -> imgPath;
        //     $brojac++;
        // }

        // $sum = $worksheet -> Range("E1");
    }
}
```

```
// $sum -> activate;
// $sum -> Value = "Sum";

// $count_list = $worksheet -> Range("E2");
// $count_list -> activate;
// $count_list -> Value = count($lists);

// // $workbook -> _SaveAs("list");
// $workbook -> Saved = true;
// $workbook -> Save();
// // $workbook -> Close;
// // $excel_file -> Workbooks -> Close();
// // $excel_file -> Quit();
// unset($workbook);
// unset($worksheet);
// unset($excel_file);
// header("Location: ../../index.php");

$ispis = '';
$ispis .= "<table>
<tr>
<td>Id</td>
<td>Name</td>
<td>Category</td>
<td>Image</td>
</tr>";
foreach($lists as $l){
$ispis .= "<tr>
<td>".$l->id."</td>
<td>".$l->listName."</td>
<td>".$l->catName."</td>
<td>".$l->imgPath."</td>
</tr>";
}
header("Content-type: application/xls");
header("Content-Disposition: attachment;Filename=lists.xls");
echo $ispis;
}
else{
    header("Location: ../../index.php?page=list");
}
?>
```

1)WORD.PHP

```
<?php
    if(isset($_POST["dugmeWord"])){
        // $word_file = new COM("word.application") or die("Cannot create Word
object");
        // $word_file -> Visible = true;
        // $word_file -> Documents -> Add();
        // $word_file -> Selection -> TypeText("Hi! It's me again... \n");
        // $word_file -> Selection -> TypeText("Hi! My name is...Hi! My name
is...*chicka,chicka* Dixy Lazic \n");
        // $word_file -> Selection -> TypeText("My index numero:19/18");
        // $word_file -> Documents[1] -> SaveAs("me.docx");
        // $word_file -> Quit();
        // $word_file = null;
        // header("Location: ../../index.php");

        header("Content-type: application/vnd.ms-word");
        header("Content-Disposition: attachment;Filename=autor.doc");
        echo "<html>";
        echo "<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=Windows1252\">";
        echo "<body>";
        echo "<b>The author</b>";
        echo "<p>Well hello! Didn't see you there...Wanna know something about
me, huh? Why? Are you working for the police? If not, nice to meet you, my name
is Dijana, I'm 20 years old and I live in a house. In Belgrade.</p>";
        echo "<p>After successfully finishing both kindergarten and elementary
school, I decided to continue my education at the Philological High School
because I really liked learning about languages. I was so good that I managed to
master six of them ( English, French, Latin, Serbian, Bosnian and Montenegrin),
but I realized I needed a new challenge...I always wanted to talk to the machines
and the best way to do that is by learning computer languages. I will be able to
speak to my laptop, right?</p>";
        echo "</body>";
        echo "</html>";

    }
    else{
        header("Location: ../../index.php?page=author");
    }
?>
```

LIST**1)DELETE.PHP**

```
<?php
ob_start();
require_once "../../config/connection.php";
require_once "functions.php";

if (isset($_POST['send'])) {

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

    $id = $_POST['idList'];

    $data=[ ];

    try {
        deleteList($id);
        // $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
        $data=[ "message"=>"Successfully deleted list!"];
        $code = 201;
    } catch (PDOException $e) {
        $code = 500;
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
        catchErrors($e->getMessage());
    }

    http_response_code($code);
    echo json_encode($data);
}

else if (isset($_POST['sendMenu'])) {

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

    $id = $_POST['idMenu'];

    $data=[ ];

    try {
        deleteMenu($id);
        // $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
        $data=[ "message"=>"Successfully deleted menu item!"];
        $code = 201;
    }
```

```

} catch (PDOException $e) {
    $code = 500;
    $data=[ "error"=>"Something wrong with the database, we're sorry..."];
    catchErrors($e->getMessage());
}

http_response_code($code);
echo json_encode($data);
}
else{
    header("Location:../../index.php?page=admin-category");
}

?>

```

1)EDIT.PHP

```

<?php

if(isset($_POST['send'])){

    require_once "../../config/connection.php";
    require_once "functions.php";
    header("Content-type:application/json");

    $id=$_POST["idList"];

    $errors = [];

    try {
        $user=findListById($id);
        $code=201;
        $data=$user;

    } catch (PDOException $e) {
        $code = 500;
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
        catchErrors($e->getMessage());
    }

    http_response_code($code);
    echo json_encode($data);
} else{ header("Location:../../index.php?page=admin-list");}

```

1)FUNCTIONS.PHP

```

<?php
    function getLists($page)
    {
        global $conn;
        $perPage = $conn -> prepare("SELECT *, l.name as listName FROM list l
INNER JOIN image i on l.image_ID=i.id_image ORDER BY l.id_list LIMIT $page,3");
        $perPage -> execute([$page]);
        return $perPage -> fetchAll();
    }

    function countAllLists(){
        global $conn;
        $countAll=$conn->query("SELECT count(*) as allLists FROM list");
        return $countAll->fetch();
    }

    function countSpecificList($perpage){
        global $conn;
        $countAll=$conn->prepare("SELECT count(*) as allLists FROM list WHERE
category_ID=?");
        $countAll->execute([$perpage]);
        return $countAll->fetch();
    }

    function insertUimages($src_small, $src,$alt){
        global $conn;
        $insert = $conn->prepare("INSERT INTO image(src_small,src,alt) VALUES(?,?,
?,?)");
        $isInserted = $insert->execute([$src_small, $src,$alt]);
        return $isInserted;
    }

    function insertUlists($name, $category,$id_of_image){
        global $conn;
        $insert = $conn->prepare("INSERT INTO list(name,category_ID,image_ID)
VALUES(?, ?, ?)");
        $isInserted = $insert->execute([$name, $category,$id_of_image]);
        return $isInserted;
    }

    function insertMenu($name,$path,$parent,$position){
        global $conn;
    }

```

```

$insert = $conn->prepare("INSERT INTO menu(name,path,parent,position)
VALUES(?, ?, ?, ?)");
$insert->execute([$name,$path,$parent,$position]);
}

function deleteMenu($id_menu){
    global $conn;
    $demenu = $conn -> prepare("DELETE FROM menu WHERE id_menu = ?");
    $demenu -> execute([$id_menu]);
}

function getAddedStatistics($id_list){
    global $conn;
    $gets = $conn->prepare("SELECT count(*) as total FROM bucket WHERE
list_ID=? GROUP BY list_ID order by total");
    $gets->execute([$id_list]);
    if ($gets->rowCount() > 0) return $gets->fetch();
}

function getDoneStatistics($id_list){
    global $conn;
    $gets = $conn->prepare("SELECT count(*) as total FROM bucket WHERE
list_ID=? and done=1 GROUP BY list_ID order by total");
    $gets->execute([$id_list]);
    if ($gets->rowCount() > 0) return $gets->fetch();
}

function getListsForAdmin(){
    return ("SELECT l.*,i.*,c.name as catName FROM list l INNER JOIN category
c ON l.category_ID=c.id_category INNER JOIN image i on l.image_ID=i.id_image
ORDER BY l.id_list");
}

function findListById($id_list){
    global $conn;
    $find=$conn->prepare("SELECT l.*,i.*,c.name as catName FROM list l INNER
JOIN category c on l.category_ID=c.id_category INNER JOIN image i on
l.image_ID=i.id_image WHERE id_list=?");
    $find->execute([$id_list]);
    return $find->fetch();
}

function updateList($id_list,$name,$category,$id_of_image=null){
    global $conn;
    $upit="UPDATE list SET name = ?, category_ID=?";
}

```

```

if($id_of_image!=null) $upit.=",image_ID=?";
$upit.=" WHERE id_list = ?";
$visita = $conn -> prepare($upit);
if($id_of_image!=null) $visita ->
execute([$name,$category,$id_of_image,$id_list]);
else $visita -> execute([$name,$category,$id_list]);
}

function deleteList($id_list){
global $conn;
$delist = $conn -> prepare("DELETE FROM list WHERE id_list = ?");
$delist -> execute([$id_list]);
}

function createPicture($picture_name,$picture_tmp_name,$picture_type){
list($sirina, $visina) = getimagesize($picture_tmp_name);

$picture = null;
switch($picture_type){
    case 'image/jpeg':
        $picture = imagecreatefromjpeg($picture_tmp_name);
        break;
    case 'image/png':
        $picture = imagecreatefrompng($picture_tmp_name);
        break;
}

$smallSirinaNvisina = 100;
$bigSirinaNvisina = 160;
// $bigVisina = ($bigSirina/$sirina) * $visina; // bigVisina : visina =
bigSirina : sirina
// $smallVisina = ($bigSirina/$sirina) * $visina;    al treba mi
kvadrat...

// NEW EMPTY
$new_big_picture = imagecreatetruecolor($bigSirinaNvisina,
$bigSirinaNvisina);
$new_small_picture = imagecreatetruecolor($smallSirinaNvisina,
$smallSirinaNvisina);

// RESIZE

imagecopyresampled($new_big_picture, $picture, 0, 0, 0, 0,
$bigSirinaNvisina, $bigSirinaNvisina, $sirina, $visina);

```

```

imagecopyresampled($new_small_picture, $picture, 0, 0, 0, 0,
$smallSirinaNvisina, $smallSirinaNvisina, $sirina, $visina);

// UPLOAD
$src = "big_".time().$picture_name;
$src_small = "small_".time().$picture_name;
$alt= $picture_name;
$path= 'assets/img/lists/';
$big_path = $path.$src;
$small_path = $path.$src_small;

switch($picture_type){
    case 'image/jpeg':
        imagejpeg($new_big_picture, '../../'.$big_path, 75);
        imagejpeg($new_small_picture, '../../'.$small_path, 75);
        break;
    case 'image/png':
        imagepng($new_big_picture, '../../'.$big_path);
        imagepng($new_small_picture, '../../'.$small_path);
        break;
}
if((move_uploaded_file($picture_tmp_name, '../../'.$path.$picture_name)))
return array($src_small,$src,$alt);
else return false;

}
?>

```

1)INSERT.PHP

```

<?php

if(isset($_POST['dugmeCreate'])){

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

    //za listu

    $name=$_POST["name"];
    $category=$_POST["category"];

    //za im(i)dz

```

```

$picture_name = $_FILES['picture']['name'];
$picture_tmp_name = $_FILES['picture']['tmp_name'];
$picture_type = $_FILES['picture']['type'];
$picture_size = $_FILES['picture']['size'];

$errors = [];

$allowed = ['image/jpg', 'image/jpeg', 'image/png'];

if(!in_array($picture_type, $allowed)){
    array_push($errors, "Oh no! Only jpg/jpeg/png formats allowed!");
}
if($picture_size > 3000000){
    array_push($errors, "Too big! Gotta be less than 3MB!");
}

if(count($errors) > 0){
    $_SESSION["errors"]=$errors;
    if($_SESSION["user"]->role=="user")header("Location: ../../index.php?page=list");
    else header("Location: ../../index.php?page=admin-list");
    //basic
}
else{

list($src_small,$src,$alt)=createPicture($picture_name,$picture_tmp_name,$picture_type);

try {
    $conn->beginTransaction();

    $isInserted = insertUimages($src_small, $src,$alt);
    $id_of_image=$conn->lastInsertId();
    $isInserted2 = insertUlists($name, $category,$id_of_image);

    $conn->commit();

    if(($isInserted)&&($isInserted2)){
        if ((isset($_SESSION['user'])) && $_SESSION['user']->role =='user')) header("Location: ../../index.php?page=list");
        else header("Location: ../../index.php?page=admin-list");
    }
} catch(PDOException $ex){
}
}

```

```

        $_SESSION["errors"]=[ "Something wrong with the database, we're
sorry..."];
        catchErrors($ex->getMessage());
        if($_SESSION["user"]->role=="user")header("Location:
../../../../index.php?page=list");
        else header("Location: ../../index.php?page=admin-list");
    }

}

//menu

else if(isset($_POST['dugmeCreateMenu'])){

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

//za listu

$name=$_POST["nameMenu"];
$path=$_POST["namePath"];
$parent=$_POST["numParent"]==0?null:$_POST["numParent"];
$position=$_POST["numPosition"];

$regP="/^[\d-]+$/";
$errors=[];

if (($name == "")||($path == "")||($position=="")){
    $errors[]="Must not be empty values!";
}
else if(!preg_match($regP,$position)){
    $errors[]="Must be only number value!";
}
if(($parent!=null)&&(!preg_match($regP,$parent))){
    $errors[]="Must be only number value!";
}

if(count($errors) > 0){
    $_SESSION["errors"]=$errors;
    header("Location: ../../index.php?page=admin-menu");
}
else{
    try {

```

```

        insertMenu($name,$path,$parent,$position);
        header("Location: ../../index.php?page=admin-menu");

    } catch(PDOException $ex){
        $_SESSION["errors"]=[ "You can't add to non-existent parent!"];
        catchErrors($ex->getMessage());
        header("Location: ../../index.php?page=admin-menu");
    }

}

else{
    header("Location: ../../index.php?page=admin-panel");
}

```

1)PAGINATION.PHP

```

<?php
ob_start();
//na kliktanje brojeva
//filter, sort i paginacija sve idu na jednu filter-sort stranicu
//ova je za dohvatanje paginacije, tj liste sve sto idu za paginaciju samo
if (isset($_POST['send'])) {

    require_once("../config/connection.php");
    require_once("functions.php");
    header("Content-type:application/json");

    try {
        $perpage=$_POST["perpage"];//0 na pocetku, zustale prosledjen ID za
filter/sort
        if($perpage==0){
            $data=countAllLists();
        }
        else{
            $data=countSpecificList($perpage);
        }
        $code=201;

    } catch (PDOException $e) {
        $code = 500;
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
    }
}

```

```

        catchErrors($e->getMessage());
    }

    http_response_code($code);
    echo json_encode($data);
}
else{
    header("Location:../../index.php?page=register");
}
?>

```

1)UPDATE.PHP

```

<?php
ob_start();
if (isset($_POST['changeList'])) {

    require_once("../config/connection.php");
    require_once("functions.php");

    $id = $_POST['idList'];
    $name = $_POST['editName'];
    $category = $_POST['editCategory'];

    $errors = [];
    $allowed = ['image/jpg', 'image/jpeg', 'image/png'];

    $jelPoslataSLika=strlen($_FILES["editPicture"]["tmp_name"]);

    if($jelPoslataSLika!=0){
        $picture_name = $_FILES['editPicture']['name'];
        $picture_tmp_name = $_FILES['editPicture']['tmp_name'];
        $picture_type = $_FILES['editPicture']['type'];
        $picture_size = $_FILES['editPicture']['size'];

        if(!in_array($picture_type, $allowed)){
            array_push($errors, "Oh no! Only jpg/jpeg/png formats allowed!");
        }
        if($picture_size > 3000000){
            array_push($errors, "Too big! Gotta be less than 3MB!");
        }
    }

    if($name==""){

```

```
        array_push($errors, "Name is required!");
    }

    if (count($errors)>0) {
        $_SESSION["errors"]=$errors;
        header("Location:../../index.php?page=admin-list");
    }
    else {
        if($jelPoslataSLika!=0){

list($src_small,$src,$alt)=createPicture($picture_name,$picture_tmp_name,$picture_type);
        }
        try {

            $conn->beginTransaction();

            if($jelPoslataSLika!=0){
                $isInserted = insertUimages($src_small, $src,$alt);
                $id_of_image=$conn->lastInsertId();
            }

            updateList($id,$name,$category,$id_of_image);

            $conn->commit();

            header("Location:../../index.php?page=admin-list");

        } catch (PDOException $e) {

            header("Location:../../index.php?page=admin-list");
            catchErrors($e->getMessage());
        }
    }
}
else{
    header("Location:../../index.php?page=admin-panel");
}
?>
```

USER

1)CONTACT.PHP

```
<?php
ob_start();
if (isset($_POST['send'])) {

    require_once("../config/connection.php");
    require_once("functions.php");
    header("Content-type:application/json");

    $name = $_POST['name'];
    $email = $_POST['email'];
    $message = $_POST['message'];

    $errors = [];
    $code = 404;
    $data = null;

    $reName = "/^([A-Z][a-z]+(\s[A-Z][a-z]+)*$/";

    if($name==""){
        array_push($errors, "Name is required! <br/>");
    }
    else if (!preg_match($reName, $name)) {
        array_push($errors, "Name in bad format!-Start with an uppercase every word! <br/>");
    }

    if($email==""){
        array_push($errors, "Email is required!");
    }
    else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        array_push($errors, "Email in bad format!Gotta have @ in it!Like this: try.again@gmail.com <br/>");
    }

    if($message==""){
        array_push($errors, "Write something...!");
    }

    if (count($errors)>0) {
        $data = $errors;
        $code = 422;
    }
}
```

```

    }
    else {
        try {
            //slanje mejla adminu
            $to= "bucket.list.with.k.r@gmail.com";
            $recipient_name=~K.R. Bucket List";
            createMail($email,$name,$message,$to,$recipient_name);
            $code = 201;
            $data=[ "success"=>"Successfully sent to admin!"];
        } catch (PDOException $e) {
            $code = 500;
            catchErrors($e->getMessage());
            $data=[ "error"=>"Something wrong with the database, we're sorry..."];
        }
    }
    http_response_code($code);
    echo json_encode($data);
}
else{
    header("Location:../index.php?page=contact");
}

?>

```

1)DELETE.PHP

```

<?php
ob_start();
if (isset($_POST['send'])) {

    require_once "../config/connection.php";
    require_once "functions.php";
    header("Content-type:application/json");

    $id = $_POST['idUser'];

    $data=[ ];

    try {
        deleteUser($id);
        // $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
        $data=[ "message"=>"Successfully deleted user!"];
        $code = 201;
    } catch (PDOException $e) {

```

```

$code = 500;
$data=[ "error"=>"Something wrong with the database, we're sorry..."];
catchErrors($e->getMessage());
}

http_response_code($code);
echo json_encode($data);
}
else{
    header("Location:../../index.php?page=admin-category");
}

?>

```

1)EDIT.PHP

```

<?php

if(isset($_POST['send'])){

    require_once "../../config/connection.php";
    require_once "functions.php";
    header("Content-type:application/json");

    $id=$_POST["idUser"];

    $errors = [];

    try {
        $user=findUserById($id);
        $code=201;
        $data=$user;

    } catch (PDOException $e) {
        $code = 500;
        $data=[ "error"=>"Something wrong with the database, we're sorry..."];
        catchErrors($e->getMessage());
    }

    http_response_code($code);
    echo json_encode($data);

}
else{ header("Location:../../index.php?page=admin-category");}

```

1)FUNCTIONS.PHP

```
<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
use PHPMailer\PHPMailer\SMTP;

require_once ABSOLUTE_PATH.'/models/PHPMailer/src/Exception.php';
require_once ABSOLUTE_PATH.'/models/PHPMailer/src/PHPMailer.php';
require_once ABSOLUTE_PATH.'/models/PHPMailer/src/SMTP.php';

function checkUser($email,$username){
    global $conn;
    $check = $conn->prepare("SELECT * FROM user WHERE email = ? OR username = ?");
    $check->execute([$email,$username]);
    $check=$check->rowCount();
    return $check;
}

function insertUser($username,$email, $password){
    global $conn;
    $date=date("Y-m-d H:i:s",time());
    $insertuj=$conn->prepare("INSERT INTO
user(username,email,password,active,last_visit,role_ID) values(?,?,?,?,?,2)");
    $insertuj->execute([$username,$email,$password,$date]);
}

function findUser($email,$password){
    global $conn;
    $find=$conn->prepare("SELECT u.id_user as uid,u.username,r.name as role
FROM user u inner join role r on u.role_ID=r.id_role WHERE u.email=? and
u.password=?");
    $find->execute([$email,$password]);
    return $find;
}

function findUserByEmail($email){
    global $conn;
    $find=$conn->prepare("SELECT * FROM user WHERE email=?");
    $find->execute([$email]);
    return $find->fetch();
```

```

}

function findUserById($id_user){
    global $conn;
    $find=$conn->prepare("SELECT * FROM user WHERE id_user=?");
    $find->execute([$id_user]);
    return $find->fetch();
}

function updateVisit($id_user){
    global $conn;
    $date=date("Y-m-d H:i:s",time());
    $visita = $conn -> prepare("UPDATE user SET last_visit = ?, active=1
WHERE id_user = ?");
    $visita -> execute([$date,$id_user]);
}

function deleteUser($id_user){
    global $conn;
    $delus = $conn -> prepare("DELETE FROM user WHERE id_user = ?");
    $delus -> execute([$id_user]);
}

function updateUser($id_user,$username,$email,$active,$role){
    global $conn;
    $upus = $conn -> prepare("UPDATE user SET username = ?, email=?, active=?, role_ID=? WHERE id_user = ?");
    $upus -> execute([$username,$email,$active,$role,$id_user]);
}

function loginError($id_user){
    global $conn;
    $errorUp = $conn -> prepare("UPDATE user SET error = error + 1 WHERE
id_user = ?");
    $errorUp -> execute([$id_user]);
}

function logout($id_user){
    global $conn;
    $errorUp = $conn -> prepare("UPDATE user SET active = 0 WHERE id_user =
?");
    $errorUp -> execute([$id_user]);
}

function countErrors($id_user){
}

```

```

global $conn;
$conn = new PDO("mysql:host=localhost;dbname=your_database_name", "username", "password");
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

function getAllErrors($id_user) {
    global $conn;
    $allErrors = $conn -> prepare("SELECT error FROM user WHERE id_user = ?");
    $allErrors -> execute([$id_user]);
    return $allErrors->fetch();
}

function createMail($from,$name,$message,$to,$recipient_name){
    $mail = new PHPMailer(true);

    try {
        //Server settings
        // $mail->SMTPDebug = SMTP::DEBUG_SERVER;
        $mail->isSMTP();
        $mail->Host      = 'smtp.gmail.com';
        $mail->SMTPAuth  = true;
        $mail->Username   = 'bucket.list.with.k.r@gmail.com';
        $mail->Password   = 'youareallbreathtaking';
        $mail->SMTPSecure = 'tls';
        $mail->Port       = 587;

        //Recipients
        $mail->setFrom($from,$name);
        $mail->addAddress($to, $recipient_name);
        $mail->addReplyTo('bucket.list.with.k.r@gmail.com', 'For more
info:');

        // Content
        $mail->isHTML(true);
        $mail->Subject = 'Read!';
        $mail->Body    = $message;

        $mail->send();
    } catch (Exception $e) {
        catchErrors($mail->ErrorInfo);
        catchErrors($e->getMessage());
    }
}

function getUsers(){
    return ("SELECT u.*,r.name as role FROM user u INNER JOIN role r on
u.role_ID=r.id_role");
}
function getRoles(){
    return ("SELECT * FROM role");}?>
```

1)LOGIN.PHP

```

<?php
    ob_start();
    session_start();
    if (isset($_POST['dugmeLogin'])) {

        require_once("../config/connection.php");
        require_once("functions.php");
        require_once("../file/functions.php");

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

        $errors = [];

        $rePass = "/^[\w\W]{7,50}$/";

        if($email==""){
            array_push($errors, "Email is required!");
        }
        else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            array_push($errors, "Email in bad format!Gotta have @ in it!And be followed by . Like this: try.again@gmail.com <br/>");
        }

        if($password==""){
            array_push($errors, "Password is required!");
        }
        else if (!preg_match($rePass, $password)) {
            array_push($errors, "Password in bad format! At least 7 characters!**Only letters and numbers! <br/>");
        }

        if (count($errors)>0) {
            $_SESSION["errors"]=$errors;
            header("Location:../../index.php?page=login");
        } else {
            $password = md5($password);
            try {
                $user=findUserByEmail($email);
                var_dump($user);
                $usersPassword=$user->password;
                $usersUsername=$user->username;
                $usersId=$user->id_user;
            }
        }
    }
}

```

```

        if(!$user){
            array_push($errors, "Email doesn't exist!");
            $_SESSION["errors"]=$errors;
            header("Location:../../index.php?page=login");
        }
        elseif($usersPassword!=$password){
            $mismatch="Password doesn't match!";
            loginError($usersId);
            $sumErrors=countErrors($usersId);
            $sumErrors=$sumErrors->error;
            if($sumErrors==3){
                // poslati mejl upozorenja
                $from= "bucket.list.with.k.r@gmail.com";
                $name=~K.R. Bucket List ";
                $message="Ooops, your account is in danger! Please change
your password immediately!";
                createMail($from,$name,$message,$email,$usersUsername);
                array_push($errors, "Intruder!Intruder!");
                $_SESSION["errors"]=$errors;
                header("Location:../../index.php?page=login");
            }
            $_SESSION["errors"]=$errors;
            header("Location:../../index.php?page=login");
        }
        $found=findUser($email,$password);
        if($found->rowCount()==1){
            $result=$found->fetch();
            updateVisit($result->uid);
            $_SESSION['user']=$result;
            header("Location:../../index.php?page=home");
        }

    } catch (PDOException $e) {
        catchErrors($e->getMessage());
        $_SESSION["errors"] = "Something wrong with the database, we're
sorry...";
        header("Location:../../index.php?page=login");
    }
}
else{
    header("Location:../../index.php?page=login");
}
?>
```

1)LOGOUT .PHP

```
<?php
    session_start();
    ob_start();
    require_once("functions.php");
    if(isset($_SESSION['user'])){
        logout($_SESSION["user"]->uid);
        unset($_SESSION['user']);
        header("Location:index.php?page=login");
    }
    else{
        header("Location:../../index.php?page=home");
    }
?>
```

1)REGISTER.PHP

```
<?php
    ob_start();
if (isset($_POST['send'])) {

    require_once("../config/connection.php");
    require_once("functions.php");
    header("Content-type:application/json");

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

    $errors = [];
    $code = 404;
    $data = null;

    $rePass = "/^[\A-z0-9]{7,50}$/";
    $reUser = "/^[\d\w\_\-\.\@]{5,30}$/";

    if($username==""){
        array_push($errors, "Username is required!");
    }
    else if (!preg_match($reUser, $username)) {
        array_push($errors, "Username in bad format!-At least 5 characters!
<br/>");
```

```

    }

    if($email==""){
        array_push($errors, "Email is required!");
    }
    else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        array_push($errors, "Email in bad format!Gotta have @ in it and
domains!Like this: try.again@gmail.com <br/>");
    }

    if($password==""){
        array_push($errors, "Password is required!");
    }
    else if (!preg_match($rePass, $password)) {
        array_push($errors, "Password in bad format! At least 7 characters!**Only
letters and numbers! <br/>");
    }

    if (count($errors)>0) {
        $data = $errors;
        $code = 422;
    }
    else {
        $password = md5($password);
        try {
            $check=checkUser($email,$password);

            if ($check>=1) {
                $code=409;
            }else{
                insertUser($username,$email, $password);
                // $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
                $code = 201;
                $data=[ "message"=>"Successfully registered!<br/>Almost
welcome(when you login)];
            }
        } catch (PDOException $e) {
            $code = 500;
            $data=[ "error"=>"Something wrong with the database, we're sorry..."];
            catchErrors($e->getMessage());
        }
    }
    http_response_code($code);
    echo json_encode($data);
}

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

```

1)UPDATE.PHP

```

<?php
ob_start();
if (isset($_POST['send'])) {

    require_once("../config/connection.php");
    require_once("functions.php");
    header("Content-type:application/json");

    $id = $_POST['idUser'];
    $username = $_POST['username'];
    $email = $_POST['email'];
    $active = $_POST['active'];
    $role = $_POST['role'];

    $errors = [];
    $code = 404;
    $data = [];

    $reUser = "/^[\d\w\_\-\.\@]{5,30}$/";

    if($username==""){
        array_push($errors, "Username is required!");
    }
    else if (!preg_match($reUser, $username)) {
        array_push($errors, "Username in bad format!-At least 5 characters!
<br/>");
    }

    if($email==""){
        array_push($errors, "Email is required!");
    }
    else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        array_push($errors, "Email in bad format!Gotta have @ in it!Like this:
try.again@gmail.com <br/>");
    }

    if (count($errors)>0) {
        $data = $errors;
        $code = 422;
    }
    else {
        try {
            updateUser($id,$username,$email,$active,$role);
        }
    }
}

```

```

// $data=[ "message"=>"Uspeh!Uspeh!Uspeli smo jej!"];
$data=[ "message"=>"Successfully updated user!"];
$code = 201;

} catch (PDOException $e) {
    $code = 500;
    $data=[ "error"=>"Something wrong with the database, we're sorry..."];
    catchErrors($e->getMessage());
}
}

http_response_code($code);
echo json_encode($data);
}

else{
    header("Location:../../index.php?page=register");
}
?>
```

[VISIT](#)

1)FUNCTIONS.PHP

```

<?php
function count_visits(){
    $array=[];
    $count_home = 0;
    $count_steps = 0;
    $count_list = 0;
    $count_contact = 0;
    $count_register = 0;
    $count_login = 0;
    $count_bucket = 0;
    $count_new_goal = 0;
    $count_admin_panel = 0;
    $count_author = 0;
    $sum=0;
    $yesterday=strtotime("-1 day");

    @$file=file(LOG_FAJL);
    if(count($file)){
        foreach($file as $f){
            $part=explode("\t",$f);
            list($url, $rest_of_url) = array_pad(explode('.php', $part[0], 2), 2, null);
            if(!empty($rest_of_url)) {
```

```

        $page=explode("=", $rest_of_url);
    }

    if(strtotime($part[1])>=$yesterday){
        if($page!=null){
            switch($page[1]){
                case "":$count_home++;$sum++;break;
                case "home":$count_home++;$sum++;break;
                case "steps":$count_steps++;$sum++;break;
                case "list":$count_list++;$sum++;break;
                case "contact":$count_contact++;$sum++;break;
                case "register":$count_register++;$sum++;break;
                case "login":$count_login++;$sum++;break;
                case "bucket":$count_bucket++;$sum++;break;
                case "create-goal":$count_new_goal++;$sum++;break;
                case "admin-panel":$count_admin_panel++;$sum++;break;
                case "author":$count_author++;$sum++;break;

                default:$count_home++;$sum++;break;
            }
        }
    }

    if($sum>0){
        $array[ "home" ]=round($count_home*100/$sum,2);
        $array[ "steps" ]=round($count_steps*100/$sum,2);
        $array[ "list" ]=round($count_list*100/$sum,2);
        $array[ "contact" ]=round($count_contact*100/$sum,2);
        $array[ "register" ]=round($count_register*100/$sum,2);
        $array[ "login" ]=round($count_login*100/$sum,2);
        $array[ "bucket" ]=round($count_bucket*100/$sum,2);
        $array[ "goal" ]=round($count_new_goal*100/$sum,2);
        $array[ "panel" ]=round($count_admin_panel*100/$sum,2);
        $array[ "author" ]=round($count_author*100/$sum,2);
    }
}

return $array;
}

function count_current_logged_in(){
    global $conn;
    $curr=$conn->query("SELECT COUNT(*) as total FROM user WHERE active=1");
    return $curr->fetch();}?
}

```

CONFIG

1).ENV

DBNAME=d

SERVER=i

USERNAME=x

PASSWORD=y

1)CONFIG.PHP

```
<?php

//putanja
define("ABSOLUTE_PATH", $_SERVER["DOCUMENT_ROOT"]);
// /storage/ssd3/703/14060703/public_html

//fajlici
define("ENV_FAJL", ABSOLUTE_PATH."/config/.env");
define("LOG_FAJL", ABSOLUTE_PATH."/data/log.txt");
define("SEPARATOR", "&");

//baza
define("SERVER", env("SERVER"));
define("DATABASE", env("DBNAME"));
define("USERNAME", env("USERNAME"));
define("PASSWORD", env("PASSWORD"));

function env($naziv){
    $open = fopen(ENV_FAJL, "r");
    $podaci = file(ENV_FAJL);
    $vrednost = "";
    foreach($podaci as $key=>$value){
        $konfig = explode("=", $value);
        if($konfig[0]==$naziv){
            $vrednost = trim($konfig[1]);
        }
    }
    return $vrednost;
}
```

1) CONNECTION.PHP

```
<?php

require_once "config.php";

pageAccess();

try {
    $conn = new PDO("mysql:host=".SERVER.";dbname=".DATABASE.";charset=utf8",
USERNAME, PASSWORD);
    $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $ex){
    echo $ex->getMessage();
}

function queryExecute($query){
    global $conn;
    return $conn->query($query)->fetchAll();
}

function pageAccess(){
    $open = fopen(LOG_FAJL, "a");
    $date = date('Y-m-d H:i:s');
    if($open){
        fwrite($open,
"{$_SERVER['REQUEST_URI']}\t{$date}\t{$_SERVER['REMOTE_ADDR']}\n");
        fclose($open);
    }
}

function catchErrors($error){
    $file = @ fopen ("../data/errors.txt", "a");
    if($file)
    {
        $date = date("Y-m-d H:i:s");
        $data = $error . "\t" . $date . "\t\n";
        fwrite($file, $data);
        fclose($file);
        return true;
    }
    return false;
}
```

DATA

FIXED

1)ERRORS.TXT

SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'SELEC * FROM user WHER email='korisnik@gmail.com" at line 1 2020-06-12 20:41:00

1)LOG.TXT

/index.php?page=admin-list 2020-06-15 19:27:59 109.245.95.777

ASSETS

CSS

1)STYLE.CSS

```
.okreci:nth-child(even){  
    display: flex;  
    flex-direction: row-reverse;  
}  
#spinner{  
    width: 5rem;  
    height: 5rem;  
}  
.sakrij{  
    display: none !important;  
}  
select{  
    background-color: transparent;  
    border: none;  
}  
table tr td{  
    vertical-align: middle !important;  
}  
.plavo{  
    /* color: #006994 !important; */  
    background-color: #0f5c7e !important;  
    border: 0px !important;  
}  
.transparentno{  
    background-color: rgba(255, 255, 255,0.1) !important;  
}  
table{  
    background-color: rgba(255, 255, 255,0.15) !important;  
}  
#izaBucket{  
    background-image: url("../img/buc.png");  
    background-position: center;  
    background-repeat: no-repeat;  
    background-attachment: center;  
    background-size: contain;  
}
```

JS

1) MAIN.JS

```

window.onload = function () {
    let url = location.href;
    if (url.indexOf("page=list") != -1) {
        dohvatanjePagination(0);
    }
    $("#spinner").addClass("sakrij");
    $("#done").hide();
    $("#newmenu").hide();
    $("#newcat").hide();
    $("#newlist").hide();
    $("#editlist").hide();
    $("#user-form").hide();
    $("#juzer").next().hide();
    $("#juzer").click(function (e) {
        e.preventDefault();
        $("#juzer").next().toggle("slow", function () { });
    })
    $("#dugmeRegister").click(proveraRegister);
    $("#dugmeLogin").click(proveraLogin);
    $("#dugmeContact").click(proveraContact);
    $(".dugmeBucket").click(izaberizaBucket);
    $(".dugmeRemoveCat").click(deleteCategory);
    $(".dugmeRemoveMenu").click(deleteMenu);
    $(".dugmeEditUser").click(editUser);
    $(".dugmeRemoveUser").click(deleteUser);
    $(".dugmeEditList").click(editList);
    $(".dugmeRemoveList").click(deleteList);
    $(".aktiviraj").click(flipLinks);
    $(".backArrow").click(goBack);
    $("#changeUser").click(changeUser);
    $("#filter").change(filterNsortList);
    $("#sort").change(filterNsortList);
}
//provere
function proveraRegister() {
    let validno = true;
    let username = $("#username").val().trim();
    let email = $("#email").val().trim();
    let password = $("#password").val().trim();

```

```

let remail = /^[A-z0-9\.\\_\\-]+@[A-z0-9\\.\\_\\-]+(\\.[A-z]{2,})+$/;
let rePass = /^[A-z0-9]{7,50}$/;
let reUser = /^[\\w\\d\\s\\.\\-\\@\\_]{5,50}$/;

if (username == "") {
    validno = false;
    $("#username").css("border-bottom", "2px solid red");
}
else if (!reUser.test(username)) {
    validno = false;
    $("#username").css("border-bottom", "2px solid red");
    $("#username").next().html("At least 5 characters!");
}
else {
    $("#username").css("border-bottom", "2px solid green");
    $("#username").next().html("");
}

if (email == "") {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
}
else if (!remail.test(email)) {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
    $("#email").next().html("Gotta have @ in it and 2 domains after!<br/>
Like this: try.again@gmail.com");
}
else {
    $("#email").css("border-bottom", "2px solid green");
    $("#email").next().html("");
}

if (password == "") {
    validno = false;
    $("#password").css("border-bottom", "2px solid red");
}
else if (!rePass.test(password)) {
    validno = false;
    $("#password").css("border-bottom", "2px solid red");
    $("#password").next().html("At least 7 characters!<br/>**Only letters and
numbers!");
}
else {
    $("#password").css("border-bottom", "2px solid green");
}

```

```
        $("#password").next().html("");
    }

    if (validno) {
        var obj = {
            username: username,
            email: email,
            password: password,
            send: true
        };

        $.ajax({
            url: "models/user/register.php",
            method: "POST",
            data: obj,
            success: function (data) {
                $("#message").html(data.message);
                console.log(data)
            },
            error: function (xhr, status, error) {
                var poruka = "Oops, an error occurred...";
                console.log(xhr)
                console.log(xhr.responseJSON)
                var i;
                switch (xhr.status) {
                    case 404: poruka = "Page not found!"; break;
                    case 409: poruka = "Username or email already exists!";
break;
                    case 422: poruka = "Values are not valid!"; break;
                    case 500: poruka = "Error"; break;
                }
                // $("#poruka").css("border","2px solid red");
                $("#message").html(poruka + "<br/>");
                for (i of xhr.responseJSON) {
                    $("#message").append(i);
                }
            }
        })
    }

function proveraLogin() {
    let validno = true;

    let email = $("#email").val().trim();
```

```

let password = $("#password").val().trim();

let remail = /^[A-z0-9\.\_\-\-]+@[A-z0-9\.\_\-\-]+\([\.][A-z]{2,}\)+$/;

let rePass = /^[A-z0-9]{7,50}\$/;

if (email == "") {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
}
else if (!remail.test(email)) {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
    $("#email").next().html("Gotta have @ in it!And be followed by 2
domains.<br/> Like this: try.again@gmail.com");
}
else {
    $("#email").css("border-bottom", "2px solid green");
    $("#email").next().html("");
}
if (password == "") {
    validno = false;
    $("#password").css("border-bottom", "2px solid red");
}
else if (!rePass.test(password)) {
    validno = false;
    $("#password").css("border-bottom", "2px solid red");
    $("#password").next().html("At least 7 characters!<br/>**Only letters and
numbers!");
}
else {
    $("#password").css("border-bottom", "2px solid green");
    $("#password").next().html("");
}

if (validno) {
    return true;
}
else {
    return false;
}
}

function proveraContact() {
    let validno = true;

```

```

let name = $("#name").val().trim();
let email = $("#email").val().trim();
let message = $("#message").val().trim();

let reName = /^[A-Z][a-z]+(\s[A-Z][a-z]+)*$/;
let remail = /^[A-z0-9\._\-\-]+@[A-z0-9\._\-\-]+\(\.[A-z]\{2,\}\)+$/;

if (name == "") {
    validno = false;
    $("#name").css("border-bottom", "2px solid red");
    $("#name").next().html("");
}
else if (!reName.test(name)) {
    validno = false;
    $("#name").css("border-bottom", "2px solid red");
    $("#name").next().html("Gotta all start with an uppercase letter!");
}
else {
    $("#name").css("border-bottom", "2px solid green");
    $("#name").next().html("");
}

if (email == "") {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
}
else if (!remail.test(email)) {
    validno = false;
    $("#email").css("border-bottom", "2px solid red");
    $("#email").next().html("Gotta have @ in it and 2 domains after!<br/>
Like this: try.again@gmail.com");
}
else {
    $("#email").css("border-bottom", "2px solid green");
    $("#email").next().html("");
}

if (message == "") {
    validno = false;
    $("#message").css("border-bottom", "2px solid red");
    $("#message").next().html("Write something");
}
else {
    $("#message").css("border-bottom", "2px solid green");
}

```

```
$("#message").next().html("");
}

if (validno) {
    var obj = {
        name: name,
        email: email,
        message: message,
        send: true
    };

    $.ajax({
        url: "models/user/contact.php",
        method: "POST",
        data: obj,
        success: function (data) {
            $("#risponce").html(data.success);
            console.log(data)
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
            console.log(xhr.responseJSON)
            var i;
            switch (xhr.status) {
                case 404: poruka = "Page not found!"; break;
                case 422: poruka = "Values are not valid!"; break;
                case 500: poruka = "Error"; break;
            }
            $("#risponce").css("color", "red");
            $("#risponce").html(poruka + "<br/>");
            for (i of xhr.responseJSON) {
                $("#risponce").append(i);
            }
        }
    })
}

function proveraCreate() {
    let validno = true;

    let name = $("#name").val().trim();
    let category = $("#category").val();
    let picture = $("#picture").val();
```

```
console.log(name, category, picture)

let path = picture.split(".");
let format = path[path.length - 1];

let reFormat = /^(jpg|jpeg|png)$/;

if (name == "") {
    validno = false;
    $("#name").css("border-bottom", "2px solid red");
}
else {
    $("#name").css("border-bottom", "2px solid green");
}

if (category == "0") {
    validno = false;
    $("#category").addClass("text-danger");
}
else {
    $("#category").addClass("text-success");
}

if (picture == "") {
    validno = false;
    $("#picture").css("border", "2px solid red");
}
else if (!reFormat.test(format)) {
    validno = false;
    $("#picture").css("border", "2px solid red");
    $("#picture").next().html("Must be a picture in jpg/jpeg/png format!");
}
else {
    $("#picture").css("border", "2px solid green");
    $("#picture").next().html("");
}

if (validno) return true;
else return false;
}

function proveraNewCat(){
    let validno = true;

    let name = $("#name").val().trim();
```

```
if (name == "") {
    validno = false;
    $("#name").css("border-bottom", "2px solid red");
}
else {
    $("#name").css("border-bottom", "2px solid green");
}

if (validno) return true;
else return false;
}

function proveraNewMenu(){
    let validno = true;

    let name = $("#nameMenu").val().trim();
    let path = $("#namePath").val().trim();
    let parent = $("#numParent").val().trim();
    let position = $("#numPosition").val().trim();

    let regP=/^[\d]+$/;

    if (name == "") {
        validno = false;
        $("#nameMenu").css("border-bottom", "2px solid red");
    }
    else {
        $("#nameMenu").css("border-bottom", "2px solid green");
    }

    if (path == "") {
        validno = false;
        $("#namePath").css("border-bottom", "2px solid red");
    }
    else if(path.indexOf("index.php?page=")==-1){
        validno = false;
        $("#namePath").css("border-bottom", "2px solid red");
        $("#namePath").val("index.php?page=");
    }
    else {
        $("#namePath").css("border-bottom", "2px solid green");
    }

    if (parent == "") {
```

```
validno = false;
$("#numParent").css("border-bottom", "2px solid red");
$("#numParent").attr("placeholder","Put 0 if it doesn't have a parent!");
}
else if(!regP.test(parent)){
    validno = false;
    $("#numParent").css("border-bottom", "2px solid red");
    $("#numParent").next().html("Only numbers!");
}
else {
    $("#numParent").css("border-bottom", "2px solid green");
}

if (position == "") {
    validno = false;
    $("#numPosition").css("border-bottom", "2px solid red");
}
else if(!regP.test(position)){
    validno = false;
    $("#numPosition").css("border-bottom", "2px solid red");
    $("#numPosition").next().html("Only numbers!");
}
else {
    $("#numPosition").css("border-bottom", "2px solid green");
}

if (validno) return true;
else return false;
}
//paginacija
function dohvatanjePagination(perpage) {
$.ajax({
    url: "models/list/pagination.php", //dohvata brojeve, tj kolko ih ima
    method: "POST",
    data: {
        perpage: perpage,
        send: true
    },
    success: function (data) {
        ispisiPagination(data);
        console.log(data)
    },
    error: function (xhr, status, error) {
        var poruka = "Oops, an error occurred...";
        console.log(xhr)
    }
});
```

```
        console.log(xhr.responseJSON)
        var i;
        switch (xhr.status) {
            case 500: poruka = "Error"; break;
        }
        // $("#poruka").css("border","2px solid red");
        $("#poruka").html(poruka + "<br/>");
        for (i of xhr.responseJSON) {
            $("#poruka").append(i);
        }
    })
}

function ispisPagination(data) {
    let ispis = "";
    let sum = data.allLists;
    let links = Math.ceil(sum / 3);
    for (let i = 1; i <= links; i++) {
        ispis += `<li class="page-item"><a class="page-link paginatedLinks" data-perpage="${i}" href="#">${i}</a></li>`;
    }

    $("#paginacija").html(ispis);
    $(".paginatedLinks").click(pagination);
}

function pagination(e) {
    e.preventDefault();
    let page = $(this).data("perpage");
    let idSort = $("#sort").val();
    let idFilterKat = $("#filter").val();
    console.log(page, idSort, idFilterKat)

    $.ajax({
        url: "models/category/filterNsort.php",
        method: "POST",
        data: {
            page: page,
            idSort: idSort,
            idFilterKat: idFilterKat,
            send: true
        },
        success: function (data) {
            ispisLista(data);
        }
    });
}
```

```

        console.log(data)
    },
    error: function (xhr, status, error) {
        var poruka = "Oops, an error occurred...";
        console.log(xhr)
        console.log(xhr.responseJSON)
        var i;
        switch (xhr.status) {
            case 500: poruka = "Error"; break;
        }
        // $("#poruka").css("border", "2px solid red");
        $("#poruka").html(poruka + "<br/>");
        for (i of xhr.responseJSON) {
            $("#poruka").append(i);
        }
    }
}

function ispisLista(data) {
    console.log("gleđaj ovo +" + data)
    let added = []
    for (let d of data[1]) {
        if (d == null) added.push(0);
        else added.push(d.total)
    }
    let done = []
    for (let d of data[2]) {
        if (d == null) done.push(0);
        else done.push(d.total)
    }
    let ispis = "";
    for (let [index, element] of data[0].entries()) {
        ispis += `
            <div class="media my-3 mb-5 tm-service-media okreci">
                
                    <div class="media-body tm-service-text">
                        <h2 class="mt-5 tm-content-title">${element.listName}</h2>
                        <!-- ovde će biti ukuono, uradjeno i dodaj -->
                    `;
        if ($("#idUser").val() == 0) {
            ispis += `
                <p class="text-danger">Gotta login to have your bucket filled
in!</p>
            `;
        }
    }
}

```

```

        `;
    }
    else {
        ispis += `
            <button type="button" class="btn btn-secondary plavo mt-4
dugmeBucket" data-idlist="${element.id_list}" name="">Pick up!</button>
            <i class="fas fa-plus fa-1x ml-5"></i> <span>${added[index]}</span>
Added</span>
            <i class="fas fa-check-square fa-1x ml-1 "></i>
<span>${done[index]} Done</span>
        `;
    }
    ispis += `
        <p></p>
        </div>
    </div>
`;
}

$("#liste").html(ispis);
$(".dugmeBucket").click(izaberizaBucket);
}

//filter i sort
function filterNsort() {
    let idFilterKat = $("#filter").val();
    let idSort = $("#sort").val();
    // $("#sort").val(0);
    dohvatanjePagination(idFilterKat);

    $.ajax({
        url: "models/category/filterNsort.php",
        method: "POST",
        data: {
            idFilterKat: idFilterKat,
            idSort: idSort,
            page: 1,
            send: true
        },
        success: function (data) {
            ispisLista(data);
            console.log(data)
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
        }
    });
}

```

```

        console.log(xhr.responseJSON)
        var i;
        switch (xhr.status) {
            case 500: poruka = "Error"; break;
        }
        $("#poruka").html(poruka);
    }
})
}
//listNbucket
function izaberizaBucket() {
    let idList = $(this).data("idlist");
    let idUser = $("#idUser").val();
    let dugme = $(this).html("Picked up!");
    console.log(idList, idUser)

    $.ajax({
        url: "models/bucket/insert.php",
        method: "POST",
        data: {
            idList: idList,
            idUser: idUser,
            send: true
        },
        success: function (data) {
            console.log(data)
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
            console.log(xhr.responseJSON)
            var i;
            switch (xhr.status) {
                case 409: poruka = "Already chosen..."; break;
                case 500: poruka = "Error"; break;
            }
            $(dugme).html(poruka);
        }
    })
}

function flipLinks(e) {
    e.preventDefault();
    $(this).removeClass("text-light");
    $(this).siblings(".aktiviraj").addClass("text-light");
}

```

```

let link = $(this).attr("href")
console.log(link)
if (link == "#done") {
    $("#done").show();
    $("#todo").hide();
}
else if (link == "#todo") {
    $("#done").hide();
    $("#todo").show();
}
else if (link == "#allcat") {
    $("#newcat").hide();
    $("#allcat").show();
}
else if (link == "#newcat") {
    $("#newcat").show();
    $("#allcat").hide();
}
else if (link == "#alllists") {
    $("#alllists").show();
    $("#newlist").hide();
    $("#editlist").hide();
}
else if (link == "#newlist") {
    $("#newlist").show();
    $("#alllists").hide();
    $("#editlist").hide();
}
else if (link == "#allmenu") {
    $("#newmenu").hide();
    $("#allmenu").show();
}
else if (link == "#newmenu") {
    $("#newmenu").show();
    $("#allmenu").hide();
}

}

//category
function deleteCategory() {
    let idCat = $(this).data("idcat");
    console.log(idCat)

    $.ajax({
        url: "models/category/delete.php",

```

```
method: "POST",
data: {
    idCat: idCat,
    send: true
},
success: function (data) {
    console.log(data)
    location.reload();
},
error: function (xhr, status, error) {
    var poruka = "Oops, an error occurred...";
    console.log(xhr)
    console.log(xhr.responseJSON)
    var i;
    switch (xhr.status) {
        case 500: poruka = "Error"; break;
    }
    $("#poruka").html(poruka);
}
}

function deleteMenu(){
let idMenu = $(this).data("idmenu");
console.log(idMenu)

$.ajax({
    url: "models/list/delete.php",
    method: "POST",
    data: {
        idMenu: idMenu,
        sendMenu: true
    },
    success: function (data) {
        console.log(data)
        location.reload();
    },
    error: function (xhr, status, error) {
        var poruka = "Oops, an error occurred...";
        console.log(xhr)
        console.log(xhr.responseJSON)
        var i;
        switch (xhr.status) {
            case 500: poruka = "Error"; break;
        }
    }
})
```

```
        $("#poruka").html(poruka);
    }
})
}
//user
function editUser() {
    let idUser = $(this).data("iduser");
    console.log(idUser)

    $.ajax({
        url: "models/user/edit.php",
        method: "POST",
        data: {
            idUser: idUser,
            send: true
        },
        success: function (data) {
            console.log(data)
            $("#user-form").show();
            $("#user-table").hide();
            ispisiUserVrednosti(data)
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
            console.log(xhr.responseJSON)
            var i;
            switch (xhr.status) {
                case 500: poruka = "Error"; break;
            }
            $("#poruka").html(poruka);
        }
    })
}

function ispisiUserVrednosti(data) {
    $("#idUser").val(data.id_user);
    $("#username").val(data.username);
    $("#email").val(data.email);
    if (data.active == 1) $("#active").prop("checked", true)
    else $("#non-active").prop("checked", true)
    $("#role > option").each(function () {
        if (data.role_ID == this.value) $(this).attr('selected', 'selected');
    });
}
```

```

function goBack() {
    if ($(this).data("back") == "user") {
        $("#user-form").hide();
        $("#user-table").show();
    }
    else if ($(this).data("back") == "list") {
        $("#editlist").hide();
        $("#alllists").show();
    }
}

function changeUser() {
    let validno = true;

    let idUser = $("#idUser").val();
    let username = $("#username").val().trim();
    let email = $("#email").val().trim();
    let active = $("input[type=radio]:checked").val();
    let role = $("#role").val();
    console.log(active, role)

    let reUser = /^[^\w\d\.\-\.@_\_]{7,50}$/;
    let remail = /^[A-z0-9\.\_\-\-]+@[A-z0-9\.\_\-\-]+\(\.[A-z]\{2,\}\)+$/;

    if (username == "") {
        validno = false;
        $("#username").css("border-bottom", "2px solid red");
    }
    else if (!reUser.test(username)) {
        validno = false;
        $("#username").css("border-bottom", "2px solid red");
        $("#username").next().html("At least 7 characters!");
    }
    else {
        $("#username").css("border-bottom", "2px solid green");
        $("#username").next().html("");
    }

    if (email == "") {
        validno = false;
        $("#email").css("border-bottom", "2px solid red");
    }
    else if (!remail.test(email)) {
        validno = false;
    }
}

```

```
$("#email").css("border-bottom", "2px solid red");
$("#email").next().html("Gotta have @ in it!<br/> Like this:
try.again@gmail.com");
}

else {
    $("#email").css("border-bottom", "2px solid green");
    $("#email").next().html("");
}

if (validno) {
    var obj = {
        idUser: idUser,
        username: username,
        email: email,
        active: active,
        role: role,
        send: true
    };

    $.ajax({
        url: "models/user/update.php",
        method: "POST",
        data: obj,
        success: function (data) {
            console.log(data)
            location.reload()
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
            console.log(xhr.responseJSON)
            var i;
            switch (xhr.status) {
                case 404: poruka = "Page not found!"; break;
                case 422: poruka = "Values are not valid!"; break;
                case 500: poruka = "Error"; break;
            }
            $("#poruka").html(poruka + "<br/>");
            for (i of xhr.responseJSON) {
                $("#poruka").append(i);
            }
        }
    })
}
}
```

```
function deleteUser() {
    let idUser = $(this).data("iduser");

    $.ajax({
        url: "models/user/delete.php",
        method: "POST",
        data: {
            idUser: idUser,
            send: true
        },
        success: function (data) {
            console.log(data)
            location.reload();
        },
        error: function (xhr, status, error) {
            var poruka = "Oops, an error occurred...";
            console.log(xhr)
            console.log(xhr.responseJSON)
            var i;
            switch (xhr.status) {
                case 500: poruka = "Error"; break;
            }
            $("#poruka").html(poruka);
        }
    })
}

//list
function editList() {
    let idList = $(this).data("idlist");
    console.log(idList)

    $.ajax({
        url: "models/list/edit.php",
        method: "POST",
        data: {
            idList: idList,
            send: true
        },
        success: function (data) {
            console.log(data)
            $("#editlist").show();
            $("#alllists").hide();
            ispisiListVrednosti(data)
        },
    })
}
```

```
error: function (xhr, status, error) {
    var poruka = "Oops, an error occurred...";
    console.log(xhr)
    console.log(xhr.responseJSON)
    var i;
    switch (xhr.status) {
        case 500: poruka = "Error"; break;
    }
    $("#poruka").html(poruka);
}
})

function ispisiListVrednosti(data) {
    $("#idList").val(data.id_list);
    $("#editImage").attr("src","assets/img/lists/"+data.src);
    $("#editName").val(data.name);
    $("#editCategory > option").each(function () {
        if (data.category_ID == this.value) $(this).attr('selected', 'selected');
    });
}

function changeList() {
    let validno = true;

    let name = $("#editName").val().trim();
    // console.log(active, role)

    if (name == "") {
        validno = false;
        $("#editName").css("border-bottom", "2px solid red");
    }
    else {
        $("#editName").css("border-bottom", "2px solid green");
        $("#editName").next().html("");
    }

    if(validno) return true;
    else return false;
}

function deleteList() {
    let idList = $(this).data("idlist");

    $.ajax({
```

```
url: "models/list/delete.php",
method: "POST",
data: {
    idList: idList,
    send: true
},
success: function (data) {
    console.log(data)
    location.reload();
},
error: function (xhr, status, error) {
    var poruka = "Oops, an error occurred...";
    console.log(xhr)
    console.log(xhr.responseJSON)
    var i;
    switch (xhr.status) {
        case 500: poruka = "Error"; break;
    }
    $("#poruka").html(poruka);
})
}
}
```

TEMPLATE

CSS

1)STYLE.CSS

```
/*
TemplateMo 550 Diagoona
https://templatemo.com/tm-550-diagoona

*/
body {
    font-family: 'Open Sans', Arial, sans-serif;
    font-size: 17px;
    font-weight: 300;
    overflow-x: hidden;
    color: white;
    background-color: #2C353C;
}

a {
    transition: all 0.3s ease;
    text-decoration: none;
}

ul { padding: 0; }
a:hover { text-decoration: none; }
button:focus { outline: none; }
p { line-height: 1.9; }
.tm-logo { margin-right: 20px; }
.tm-slogan { font-size: 0.8rem; }

/* Navigation */
.navbar-toggler {
    padding: 8px 15px;
    border: 0;
    border-radius: 0;
    transform: skewX(-15deg);
    background-color: rgba(0,0,0,0.5);
    color: white;
}
```

```
.navbar-toggler i { transform: skewX(15deg); }

.navbar-expand-lg {
  padding-left: 0;
  padding-right: 0;
}

.navbar-expand-lg .navbar-nav .nav-link { padding: 0; }

.navbar-expand-lg .navbar-nav .tm-nav-link {
  font-size: 1.3rem;
  font-weight: 400;
  color: white;
  padding-bottom: 30px;
}

.nav-item { margin-right: 60px; }

.nav-item:last-child { margin-right: 0; }

.nav-item.active .tm-nav-link,
.nav-item:hover .tm-nav-link {
  color: #9CC;
}

/* Page Background */
.tm-bg {
  position: absolute;
  left: 0;
  right: 0;
  width: 100%;
  height: 100%;
  display: flex;
  z-index: -1000;
}

.tm-bg-left,
.tm-bg-right {
  position: relative;
  width: 50%;
  height: 100%;
}

.tm-bg-right { background-color: rgba(0,0,0,0.5); }
```

```
.tm-bg-left {
    border-right: 230px solid rgba(0,0,0,0.5);
    border-top: 100vh solid transparent;
}

.tm-bg-controls-wrapper {
    display: inline-block;
    margin-left: -150px;
}

.tm-bg-control {
    display: inline-block;
    width: 14px;
    height: 14px;
    background-color: rgba(255,255,255,0.5);
    margin: 6px;
    cursor: pointer;
}

.tm-bg-control.active,
.tm-bg-control:hover {
    background-color: white;
}

.tm-container {
    width: 100%;
    height: 100%;
    min-height: 100vh;
    position: relative;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}

.tm-row { display: flex; }

.tm-col-left {
    width: 50%;
    height: 100%;
    padding-left: 40px;
    padding-right: 40px;
}

.tm-col-right {
```

```
width: 50%;  
height: 100%;  
padding-left: 70px;  
padding-right: 70px;  
}  
  
.tm-content {  
    max-width: 660px;  
    margin-top: 50px;  
    padding-right: 20px;  
}  
  
.tm-about { max-width: 525px; }  
  
::-webkit-scrollbar {  
    -webkit-appearance: none;  
    width: 10px;  
}  
  
::-webkit-scrollbar-thumb {  
    border-radius: 5px;  
    background-color: rgba(255,255,255,.8);  
    box-shadow: 0 0 1px rgba(255,255,255,.8);  
}  
  
.tm-content-title { font-size: 1.5rem; }  
hr { border-top: 1px solid white; }  
  
.btn {  
    padding: 10px 25px;  
    border-radius: 0;  
}  
  
.btn-big {  
    padding: 9px 40px;  
    font-size: 1.2rem;;  
}  
  
.btn-primary {  
    background-color: white;  
    color: black;  
    border: 0;  
}  
  
.btn-primary:hover {
```

```
background-color: #9CC;
color: black;
}

.tm-site-footer {
padding: 40px 0 20px;
max-width: 660px;
font-size: 0.95rem;
}

.tm-col-footer { padding-left: 0; }
.tm-text-link { color: white; }

.tm-text-link:hover,
.tm-text-link:focus {
color: #9CC;
}

/* Service */
.tm-service-text { max-width: 365px; }
.tm-service-img { margin-right: 25px; }
.tm-service-img-r { margin-left: 25px; }

/* Contact */
.form-control,
textarea {
color: white;
background-color: transparent;
background-clip: padding-box;
border: none;
border-bottom: 1px solid #999A9B;
border-radius: 0;
transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
padding: 10px 0;
font-family: 'Open Sans', Arial, sans-serif;
font-size: 1rem;
font-weight: 400;
line-height: 1.5;
}

.form-control:focus {
color: white;
background-color: transparent;
box-shadow: none;
border-color: #9CC;
```

```
outline: none;
}

.mb-85 { margin-bottom: 85px; }
.tm-contact-main { max-width: 690px; }

.tm-contact {
    max-width: 420px;
    margin-left: auto;
    margin-right: auto;
}

.form-control::-webkit-input-placeholder { color: white; } /* Edge */
.form-control:-ms-input-placeholder { color: white; } /* Internet Explorer 10-11 */
*/
.form-control::placeholder { color: white; }

@media (max-width: 1200px) {
    .nav-item { margin-right: 30px; }

    .tm-page-right {
        padding-left: 30px;
        padding-right: 30px;
    }
}

@media (max-width: 992px) {
    .tm-nav {
        position: absolute;
        background: rgba(0, 0, 0, 0.8);
        color: white;
        top: 44px;
        right: 4px;
    }

    .navbar-expand-lg .navbar-nav .tm-nav-link {
        color: white;
        font-size: 1.1rem;
        padding: 13px 20px;
    }
}

.nav-item { margin-right: 0; }

.nav-item.active .tm-nav-link,
.nav-item:hover .tm-nav-link {
```

```
        color: #9CC;
    }

.tm-content { margin-top: 20px; }
.tm-site-footer { padding-top: 40px; }
.tm-col-left { width: 40%; }

.tm-col-right {
    width: 60%;
    padding-left: 100px;
    padding-right: 30px;
}

.tm-col-footer { padding-left: 0; }
}

@media (max-width: 768px) {
    .tm-row { flex-direction: column; }

    .tm-col-left,
    .tm-col-right,
    .tm-bg-left,
    .tm-bg-right {
        width: 100%;
    }

    #tm-main-nav {
        position: fixed;
        top: 50px;
        right: 40px;
        padding-bottom: 0;
    }

    .tm-bg { flex-direction: column; }

    .tm-bg-left {
        height: 200px;
        border-bottom: 50px solid rgba(0,0,0,0.5);
        border-right: 0;
        border-top: 0;
    }

    .tm-bg-right { height: calc(100% - 200px); }

    .tm-bg-controls-wrapper {
```

```
    margin-left: 0;
    margin-top: 50px;
}

.tm-content {
    max-height: none;
    margin-top: 40px;
}

.tm-col-right {
    padding-top: 40px;
    padding-left: 30px;
}

.tm-col-footer {
    padding-left: 10px;
    padding-right: 10px;
}
}

@media (max-width: 460px) {
    .tm-logo { margin-right: 15px; }
    .tm-sitename { font-size: 1.8rem; }
    .tm-slogan { font-size: 0.8rem; }
}

@media (max-width: 450px), (min-width: 768px) and (max-width: 830px), (min-width: 992px) and (max-width: 1040px) {
    .tm-service-media { flex-direction: column; }
    .tm-service-media-img-l { flex-direction: column-reverse; }
    .tm-service-text { margin-top: 20px; }
    .tm-service-img { margin-right: 0; }
    .tm-service-img-r { margin-left: 0; }
}
```

JS**1) MAINjihov.JS-**

```
const initBg = (autoplay = true) => {
    const bgImgNames = ['diagoona-bg-1.jpg', 'bg-4.jpg', 'diagoona-bg-2.jpg'];
    const bgImg = bgImgNames.map(img => "assets/img/" + img);

    $.backstretch(bgImg, {duration: 4000, fade: 500});

    if(!autoplay) {
        $.backstretch('pause');
    }
}

const setBg = id => {
    $.backstretch('show', id);
}

const setBgOverlay = () => {
    const windowWidth = window.innerWidth;
    const bgHeight = $('body').height();
    console.log(bgHeight);
    const tmBgLeft = $('.tm-bg-left');

    $('.tm-bg').height(bgHeight);

    if(windowWidth > 768) {
        tmBgLeft.css('border-left', `0`)
            .css('border-top', `${bgHeight}px solid transparent`);
    } else {
        tmBgLeft.css('border-left', `${windowWidth}px solid transparent`)
            .css('border-top', `0`);
    }
}

$(document).ready(function () {
    const autoplayBg = true;      // set Auto Play for Background Images
    initBg(autoplayBg);
    setBgOverlay();

    const bgControl = $('.tm-bg-control');
    bgControl.click(function() {
        bgControl.removeClass('active');
        $(this).addClass('active');
    })
})
```

```
const id = $(this).data('id');
setBg(id);
});

$(window).on("backstretch.after", function (e, instance, index) {
    const bgControl = $('.tm-bg-control');
    bgControl.removeClass('active');
    const current = $(".tm-bg-controls-wrapper").find(`[data-id=${index}]`);
    current.addClass('active');
});

$(window).resize(function() {
    setBgOverlay();
});
});
```