Visoka škola strukovnih studija za informacione i komunikacione tehnologije

Dokumentacija

http://25h-kitchen.epizy.com/

Nikola Nedeljković 2/17

Sadržaj

1. UVOD

1.1 Korišćeni programski / markap jezici

- Za izradu ovog sajta korišćene su sledeće tehnologije:

  - Mark-up jezici: HTML 5, XML
  - CSS3
  - Skripting-programski jezici: JavaScript, PHP
  - JQuery – JavaScript framework
  - JSON, AJAX
  - PHP biblioteke: PHPmailer, PHPWord i PHPStylesheet

- Sajt je napisan u Atom-u

1.2 Opis funkcionalnosti

- **index.php**
  - ✓ Sadržaj svih stranica i dinamičko otvaranje istih preko GET metoda.

- **main_template.php**
  - ✓ Landing page sa gif animacijom

- **food_post_template.php**
  - ✓ Sadržaj svih stranica i dinamičko otvaranje istih preko GET metoda.
  - ✓ Sadržaj menija sa strane

- **login_template.php**
  - ✓ Sadržaj svih stranica i dinamičko otvaranje istih preko GET metoda.
  - ✓ Login, Logout, Register, Forgot Password

- **header_template.php**
  - ✓ Navigacija i odabir jezika

- **footer_template.php**
  - ✓ Navigacija, najnovije objave i tagovi

- **contact_template.php**
  - ✓ Sadržaj kontakt stranice

- ✓ O autoru
- ✓ Download word o autoru

- ○ **admin_template.php**
  - ✓ Forma za admina
  - ✓ Download dokumentacije
  - ✓ Export to excel
  - ✓ Broj logovanih korisnika
  - ✓ Broj poseta posebnim stranicama

- ○ **add_post_template.php**
  - ✓ Forma za dodavanje nove objave

- ○ **404.php**
  - ✓ Stranica za grešku

- ○ **add_post.php**
  - ✓ Dodavnje nove objave
  - ✓ Korišćenje transakcija
  - ✓ Rad sa slikama
  - ✓ Upis u srpski i engleski RSS
  - ✓ Provera podataka u PHP-u i vraćanje statusnih kodova

- ○ **comment.php**
  - ✓ Dodavanje komentara na neku objavu
  - ✓ Provera podataka u PHP-u i vraćanje statusnih kodova

- ○ **comments.php**
  - ✓ Dohvatanje svih komentara za određenu objavu

- ○ **contact.php**
  - ✓ Slanje email-a administratoru korišćenjem PHP mailer-a

- ○ **export_excel.php**
  - ✓ Prebacivanje podataka u Excel i automatsko čuvanje na Desktop korišćenjem PHPSpreadsheets

- ○ **forgot_password.php**
  - ✓ Slanje mail-a administratorima za zaboravljenu šifru

- **logged_users.php**
    - ✓ Dohvatanje svih logovanih korisnika
- **login.php**
    - ✓ Logovanje korisnika
    - ✓ Slanje mail-a u slučaju neuspešnog logovanja
- **logout.php**
    - ✓ Odjava korisnika
- **navigation.php**
    - ✓ Dohvatanje navigacije
- **voting.php**
    - ✓ Provera da li je logovani korisnik već glasao
- **page_access_info.php**
    - ✓ Dodavanje informacija o logovanju korisnika
- **add_navigation.php**
    - ✓ Dohvatanje svih objava
- **random_quote.php**
    - ✓ Dohvatanje nasumičnog statusa
- **add_user.php**
    - ✓ Dodavanje novog korisnika
- **register.php**
    - ✓ Registracija korisnika
    - ✓ Šifrovanje podataka
    - ✓ Provera podataka u PHP-u i vraćanje statusnih kodova
- **save_word.php**
    - ✓ Sačuvljavanje podataka o autoru u Word-u na korisnički desktop putem COM klase

- **single_post.php**
  - ✓ Dohvatanje pojedinačne objave

- **subscribe.php**
  - ✓ Prijava korisnika na bilten
  - ✓ Provera podataka u PHP-u i vraćanje statusnih kodova

- **tags.php**
  - ✓ Dohvatanje svih objava korisnika

- **top_posts.php**
  - ✓ Dohvatanje najnovijih objava

- **types.php**
  - ✓ Dohvatanje svih tipova objava

- **write_error.php**
  - ✓ Upisivanje grešaka u poseban fajl

- **contact_mail.php**
  - ✓ Funkcije za slanje mejlova korisnicima i administratoru

- **comments.php**
  - ✓ Dohvatanje svih komentara

- **delete_comment.php**
  - ✓ Brisanje komentara

- **delete_newsletter.php**
  - ✓ Brisanje pretplate

- **delete_picture.php**
  - ✓ Brisanje slike i veze sa objavom

- **delete_post_tag.php**
  - ✓ Brisanje veze između objave i taga

- **delete_post.php**
  - ✓ Brisanje objave i uklanjanje slika i veza za objavu
  - ✓ Uklanjanje iz RSS-a sve podatke o objavi

- **delete_tag.php**
  - ✓ Brisanje taga

- **delete_user.php**
  - ✓ Brisanje korisnika

- **edit_post.php**
  - ✓ Izmena objave i izmena RSS-a vezanog za tu objavu

- **edit_tag.php**
  - ✓ Izmena taga

- **edit_user.php**
  - ✓ Izmena korisnika

- **insert_picture.php**
  - ✓ Dodavanje slike

- **insert_post_tag.php**
  - ✓ Dodavanje veze između post-a i taga

- **insert_tag.php**
  - ✓ Dodavanje taga

- **insert_user.php**
  - ✓ Dodavanje korisnika

- **newsletter.php**
  - ✓ Dohvatanje svih korisnika projavljenih na bilten

- **pictures.php**
  - ✓ Dohvatanje svih slika

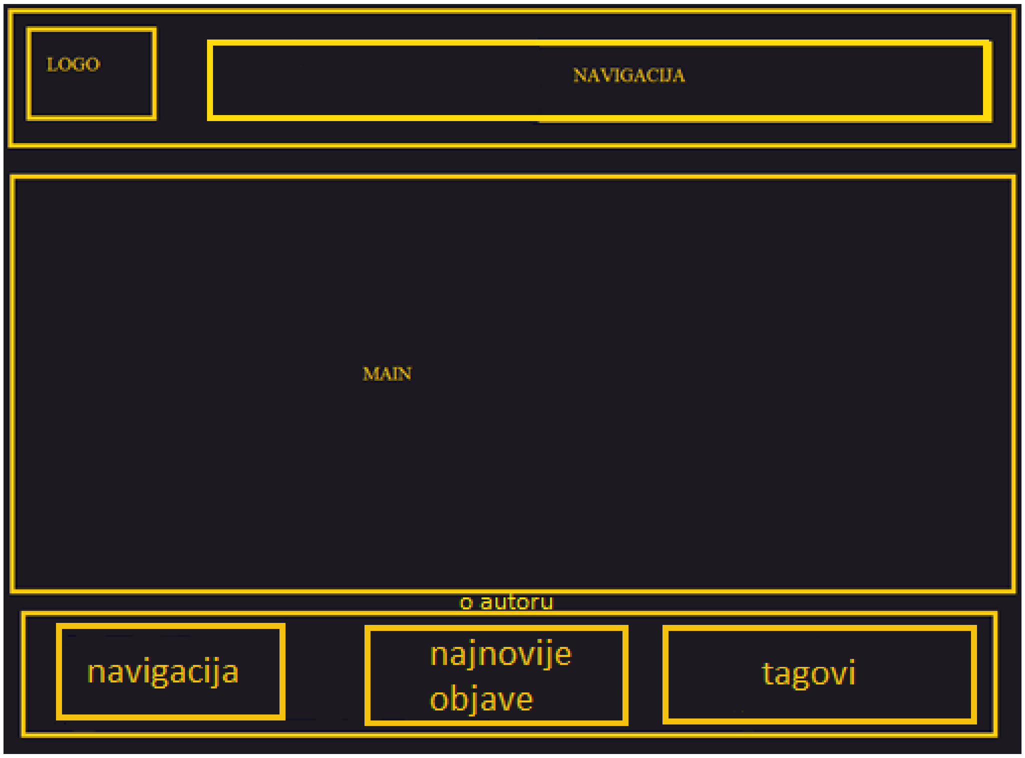
- **post_tag.php**
  - ✓ Dohvatanje svih veza između objave i tagova

- **posts.php**
  - ✓ Dohvatanje svih objava

- **tags.php**
  - ✓ Dohvatanje svih tagova

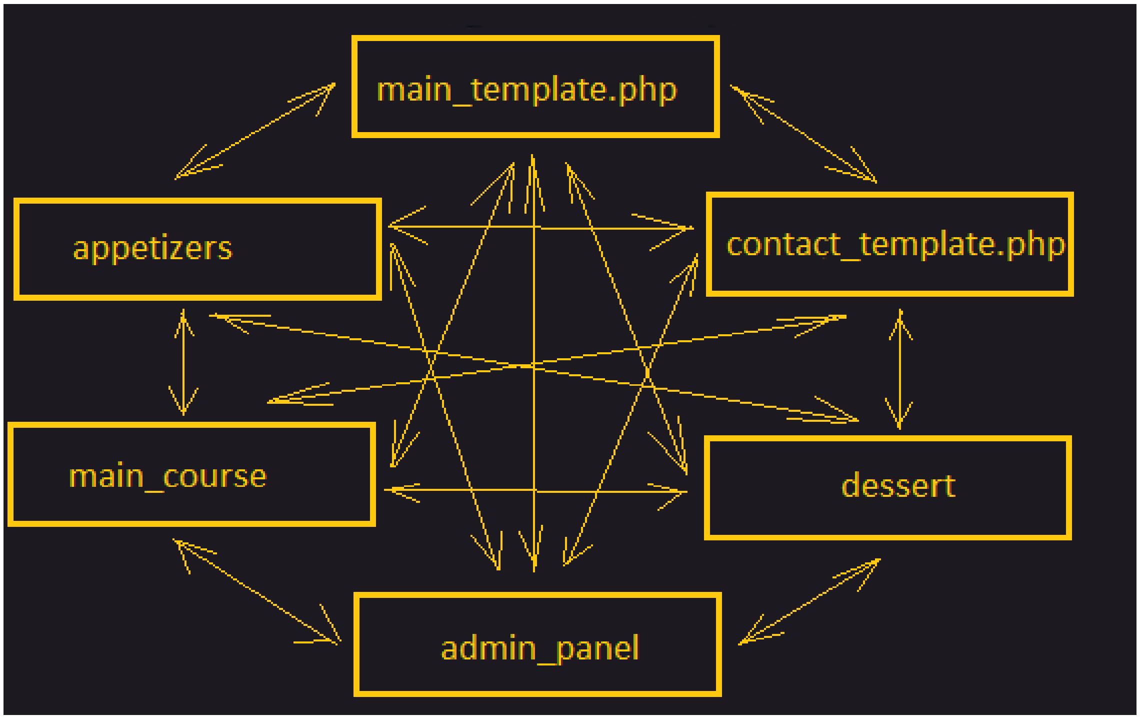
- **users.php**
  - ✓ Dohvatanje svih korisnika

1.3 Izgled stranica

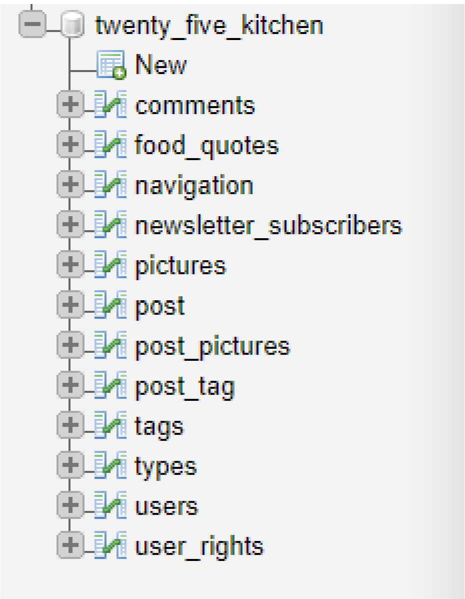


2. ORGANIZACIJA

2.1 Organizacija sajta



2.2 Organizacija baze

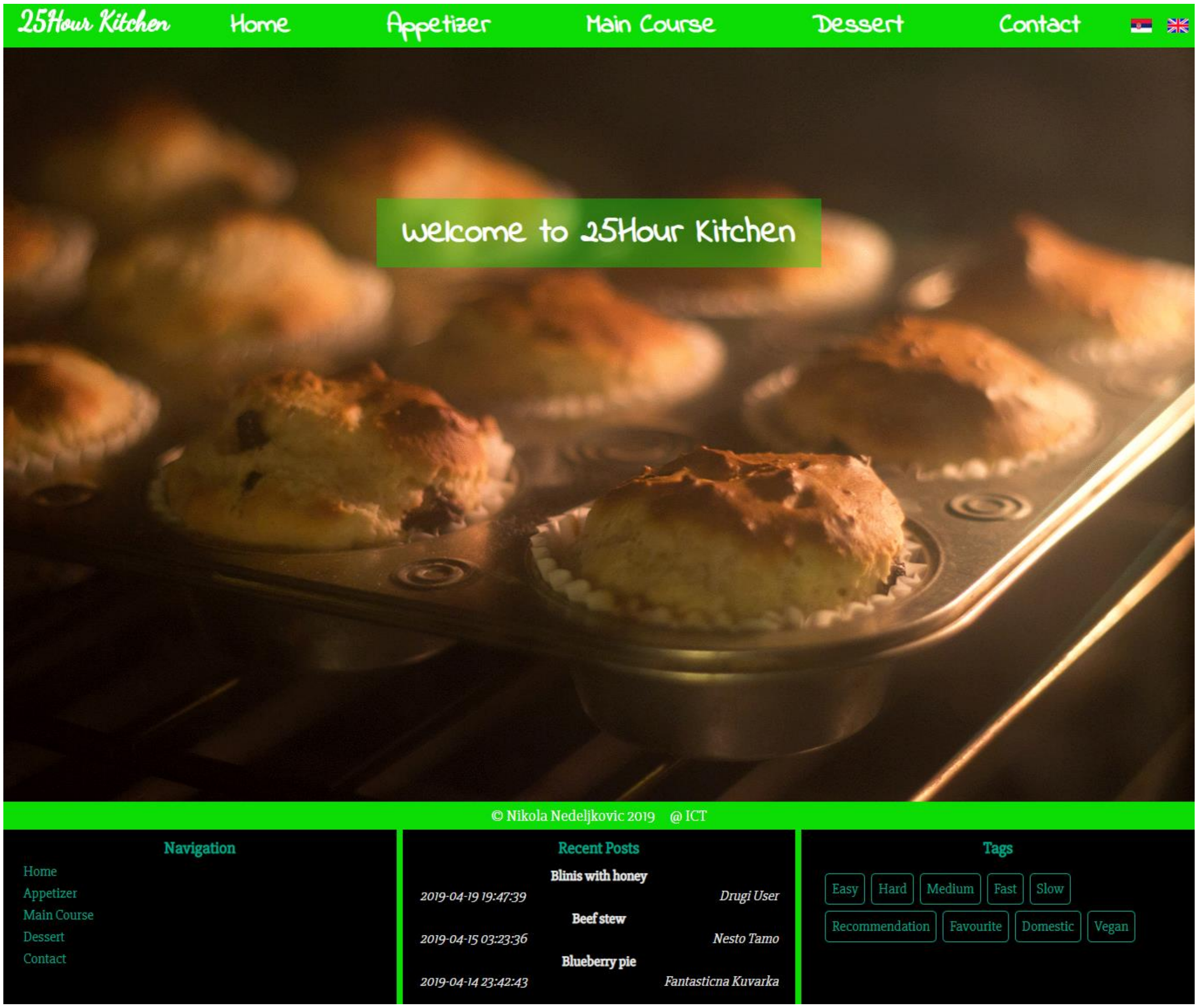


twenty_five_kitchen
- New
- comments
- food_quotes
- navigation
- newsletter_subscribers
- pictures
- post
- post_pictures
- post_tag
- tags
- types
- users
- user_rights

2.3 Sitemap.xml

```
<?xml version="1.0"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
        <loc>http://25h-kitchen.epizy.com/index.php</loc>
        <lastmod>2019-11-06</lastmod>
        <changefreq>daily</changefreq>
        <priority>1</priority>
</url>
</urlset>
```

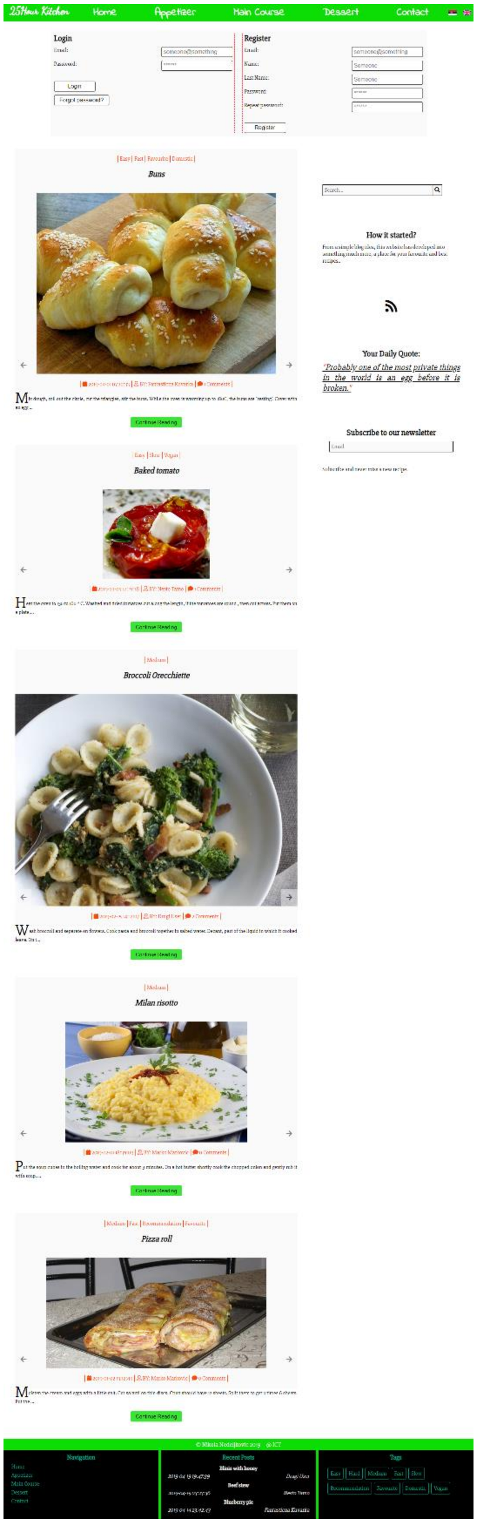
2.4 Slike stranica sa funkcionalnostima



Blinis with honey
2019-04-19 19:47:39 Drugi User
Beef stew
2019-04-15 03:23:36 Nesto Tamo
Blueberry pie
2019-04-14 23:42:43 Fantasticna Kuvarka

main_template.php

- Animirani gif, navigacija, najnovije objave, tagovi, promena jezika



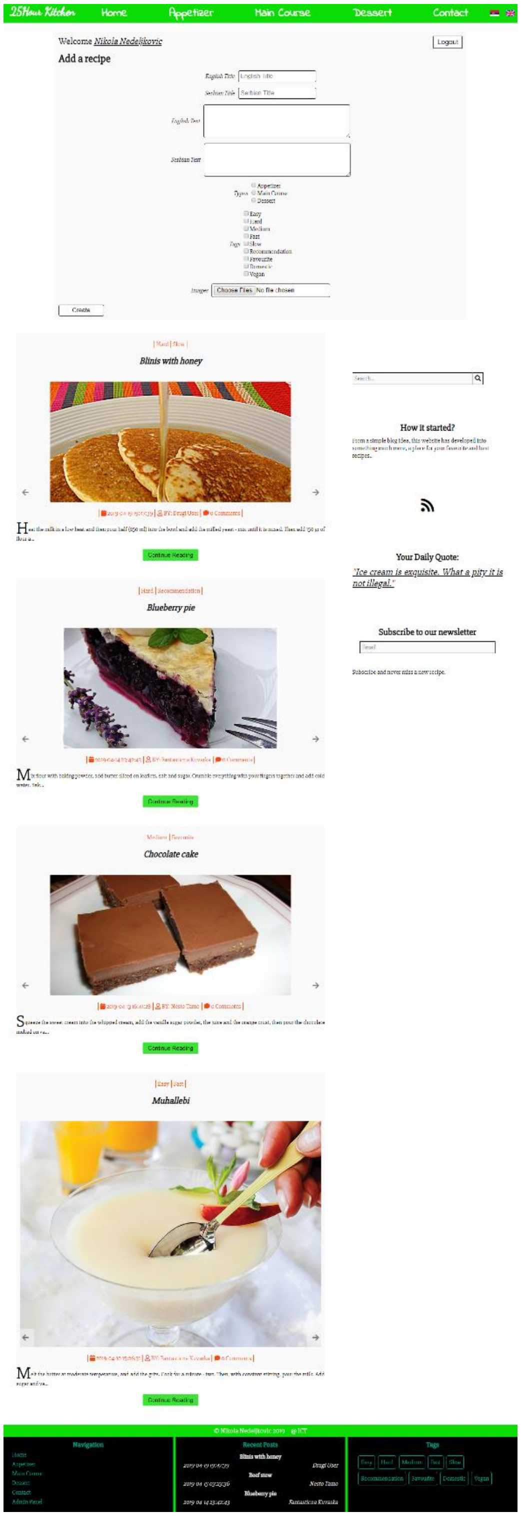
From a simple blog idea, this website has developed into something much more, a place for your favourite and best recipes..

"Probably one of the most private things in the world is an egg before it is broken."

Subscribe and never miss a new recipe.

food_post_template.php

- Prikazivanje svih objava za tu odabranu objavu, vidi više, login, logout, register, forgot password, pretraga, prijava na bilten, kopiranje RSS-a **U ZAVISNOSTI** od izabranog jezika



Welcome *Nikola Nedeljkovic*

Logout

Add a recipe

English Title English Title

Serbian Title Serbian Title

English Text

Serbian Text

Types ○ Appetizer ○ Main Course ○ Dessert

Tags □ Easy □ Hard □ Medium □ Fast □ Slow □ Recommendation □ Favourite □ Domestic □ Vegan

Images Choose Files No file chosen

Create

| Hard | Slow |

Blinis with honey

| 2019-04-15 15:07:59 | BY: Drugi User | 0 Comments |

Heat the milk in a low heat and then pour half (250 ml) into the bowl and add the milled yeast - mix until it is mixed. Then add 750 gr of flour a...

Continue Reading

| Hard | Recommendation |

Blueberry pie

| 2019-04-14 23:42:43 | BY: Fantasticna Kovanka | 0 Comments |

Mix flour with baking powder, add butter sliced on leaflets, salt and sugar. Crumble everything with your fingers together and add cold water. Bak...

Continue Reading

Medium | Favourite

Chocolate cake

| 2019-04-13 16:41:28 | BY: Nesto Tamo | 0 Comments |

Squeeze the sweet cream into the whipped cream, add the vanilla sugar powder, the juice and the orange crust, then pour the chocolate melted on va...

Continue Reading

| Easy | Fast |

Muhallebi

| 2019-04-10 15:06:51 | BY: Fantasticna Kovanka | 0 Comments |

Melt the butter at moderate temperature, and add the grits. Cook for a minute - two. Then, with constant stirring, pour the milk. Add sugar and va...

Continue Reading

Search...

How it started?

From a simple blog idea, this website has developed into something much more, a place for your favourite and best recipes.

Your Daily Quote:

"Ice cream is exquisite. What a pity it is not illegal."

Subscribe to our newsletter

Email

Subscribe and never miss a new recipe.

© Nikola Nedeljkovic 2019 @ ICT

Navigation

Home
Appetizer
Main Course
Dessert
Contact
Admin Panel

Recent Posts

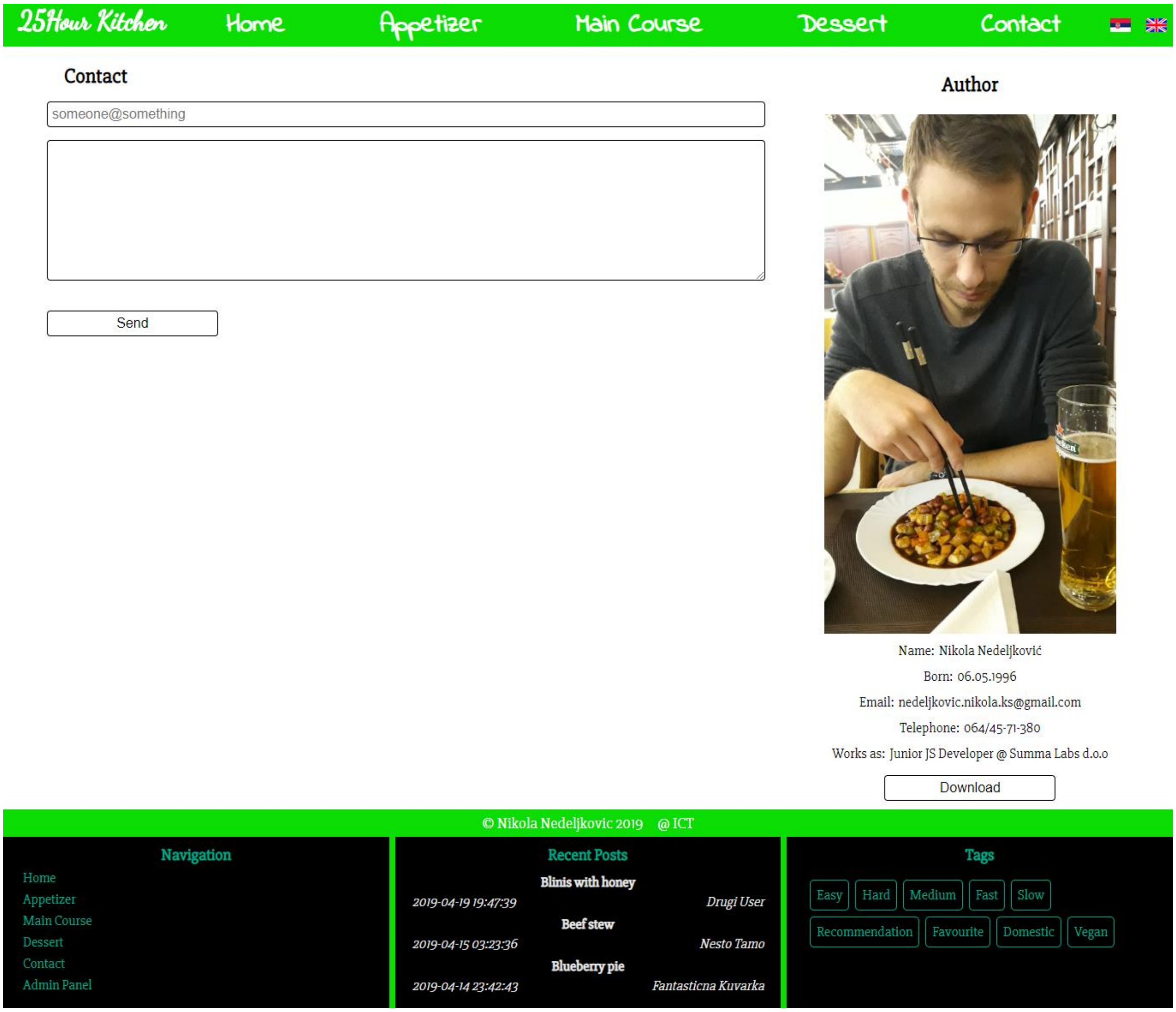
Blinis with honey
2019 04 15 15:07:59 *Drugi User*
Beef stew
2019 04 15 07:23:36 *Nesto Tamo*
Blueberry pie
2019 04 14 23:42:43 *Fantasticna Kovanka*

Tags

Easy Hard Medium Fast Slow Recommendation Favourite Domestic Vegan

food_post_template.php

- Dodavanje novih objava, i odjavljivanje uz ono prethodno



Name: Nikola Nedeljković

Born: 06.05.1996

Email: nedeljkovic.nikola.ks@gmail.com

Telephone: 064/45-71-380

Works as: Junior JS Developer @ Summa Labs d.o.o

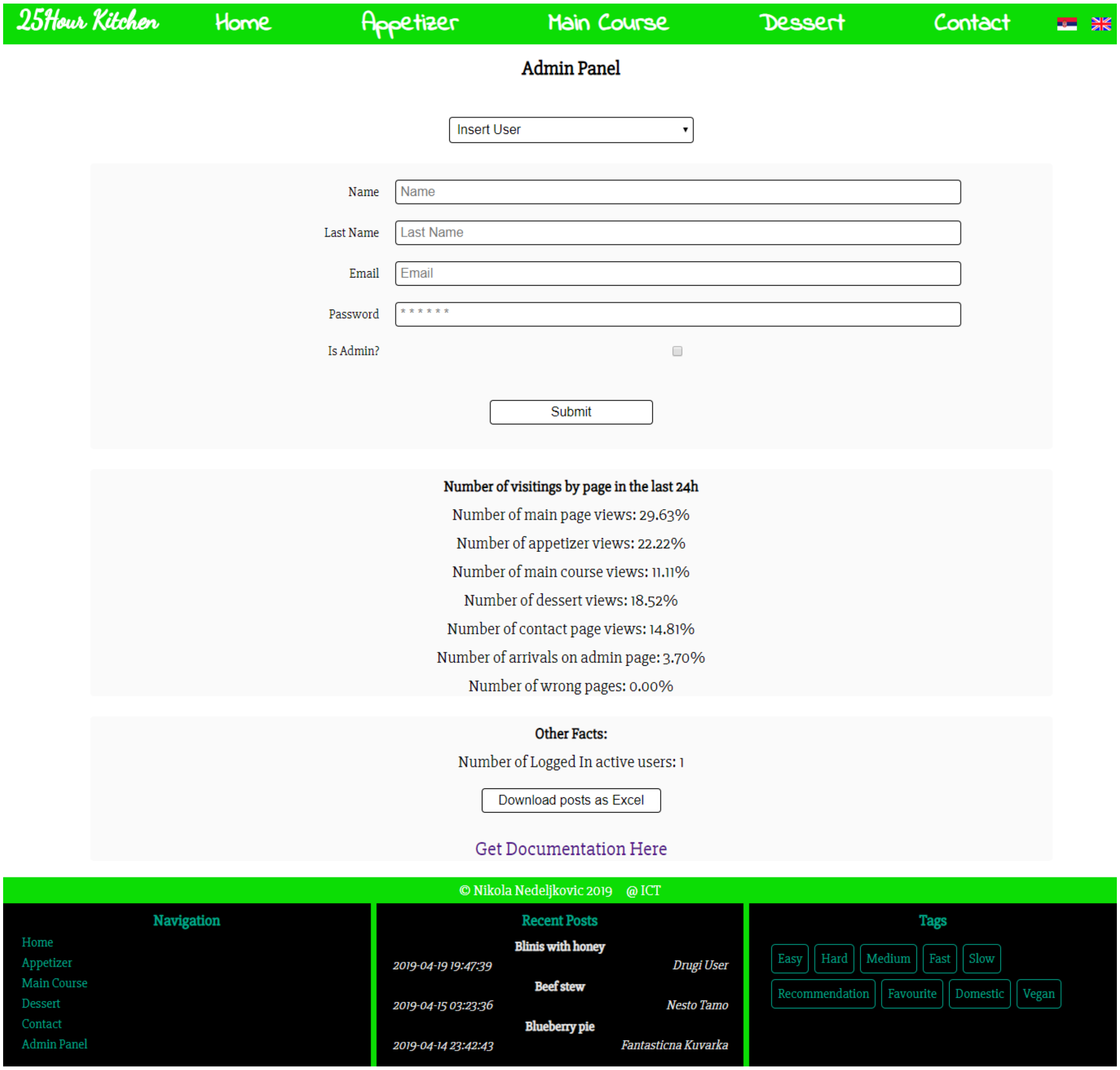
© Nikola Nedeljkovic 2019 @ ICT

Home
Appetizer
Main Course
Dessert
Contact
Admin Panel

Blinis with honey	
2019-04-19 19:47:39	*Drugi User*
Beef stew	
2019-04-15 03:23:36	*Nesto Tamo*
Blueberry pie	
2019-04-14 23:42:43	*Fantasticna Kuvarka*

contact_template.php

- Kontakt forma za administratore, o autoru, download u word



Admin Panel

Name: Name
Last Name: Last Name
Email: Email
Password: ******
Is Admin? □

Number of visitings by page in the last 24h

Number of main page views: 29.63%

Number of appetizer views: 22.22%

Number of main course views: 11.11%

Number of dessert views: 18.52%

Number of contact page views: 14.81%

Number of arrivals on admin page: 3.70%

Number of wrong pages: 0.00%

Other Facts:

Number of Logged In active users: 1

© Nikola Nedeljkovic 2019 @ICT

Navigation

Home
Appetizer
Main Course
Dessert
Contact
Admin Panel

Recent Posts

Blinis with honey
2019-04-19 19:47:39 *Drugi User*

Beef stew
2019-04-15 03:23:36 *Nesto Tamo*

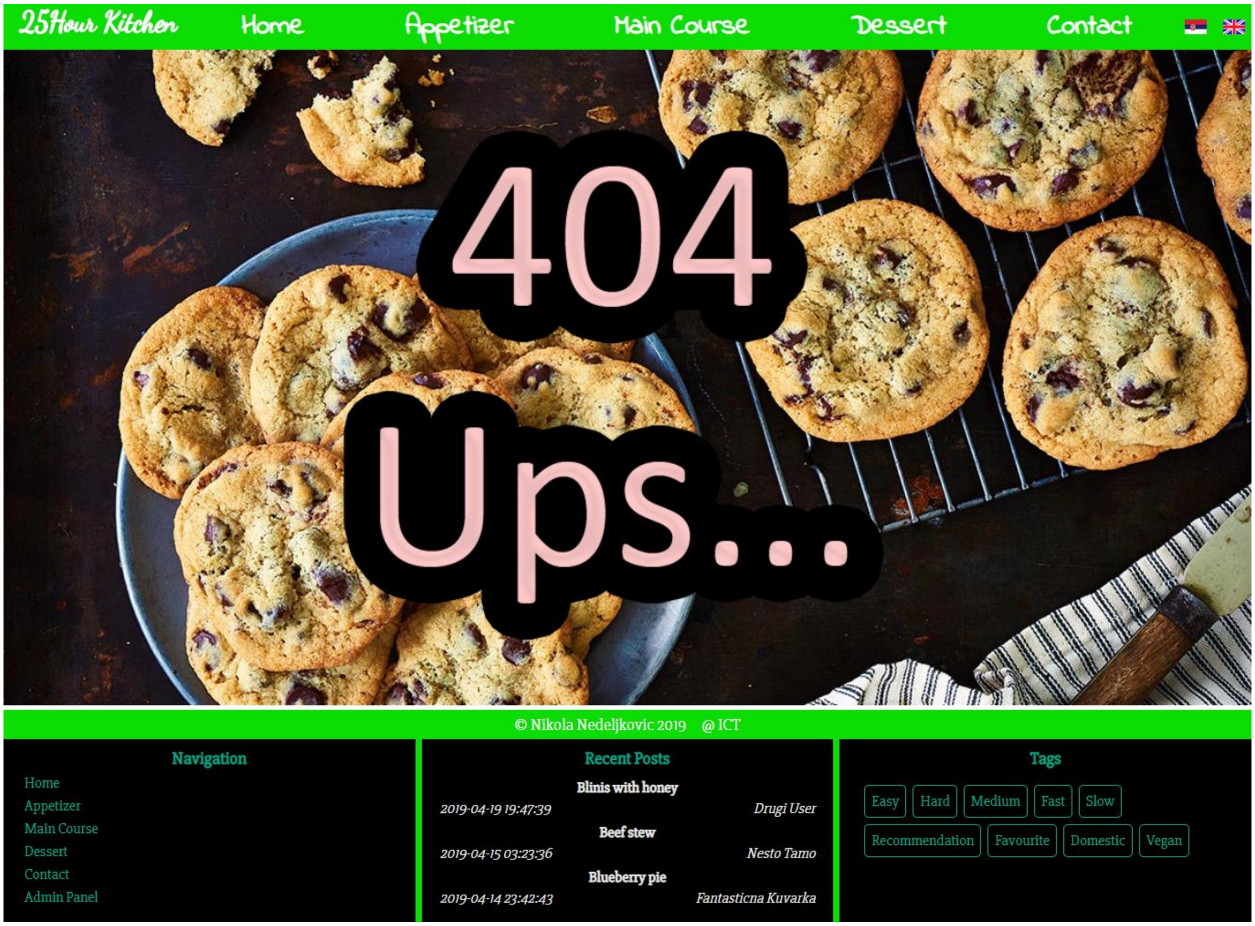
Blueberry pie
2019-04-14 23:42:43 *Fantasticna Kuvarka*

Tags

Easy Hard Medium Fast Slow Recommendation Favourite Domestic Vegan

admin_template.php

- Radnja svih admin funkcija, pregld posete stranicama, broj logovanih korisnika, export u excel



Blinis with honey
2019-04-19 19:47:39 Drugi User

Beef stew
2019-04-15 03:23:36 Nesto Tamo

Blueberry pie
2019-04-14 23:42:43 Fantasticna Kuvarka

page_404.php

- Stranica greške

3. KODOVI

3.1 index.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <meta name="author" content="Nikola Nedeljkovic"/>
    <meta name="copyright" content="RarOks 1008 Entertainment @ 2019"/>
    <meta name="robots" content="index,follow"/>
    <link rel="shortcut icon" href="favicon.ico"/>
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css"
integrity="sha384-50oBUHEmvpQ+1lW4y57PTFmhCaXp0ML5d60M1M7uH2+nqUivzIebhndOJK28anvf"
crossorigin="anonymous"/>

    <link rel="stylesheet" href="assets/css/style.css"/>
    <link rel="stylesheet" href="assets/css/main_template.css"/>
    <link rel="stylesheet" href="assets/css/header_template.css"/>
    <link rel="stylesheet" href="assets/css/footer_template.css"/>
    <link rel="stylesheet" href="assets/css/food_post_template.css"/>
    <link rel="stylesheet" href="assets/css/add_post_template.css"/>
    <link rel="stylesheet" href="assets/css/login_template.css"/>
    <link rel="stylesheet" href="assets/css/contact_template.css"/>
    <link rel="stylesheet" href="assets/css/admin_template.css"/>
    <link rel="stylesheet" href="assets/css/page_404.css"/>
  </head>
  <body>
    <?php
      require_once "config/connection.php";
      session_start();
      include("views/header_template.php");
      echo "<div id=\"main_panel\">";
      if (isset($_GET["page"])) {
        switch($_GET["page"]) {
          case "main_template":
            zabeleziPristupStranici($_GET["page"]);
            include "views\main_template.php";
            break;
          case "appetizer":
          case "main_course":
          case "dessert":
            zabeleziPristupStranici($_GET["page"]);
            include "views/food_post_template.php";
            break;
          case "contact_template":
            zabeleziPristupStranici($_GET["page"]);
            include "views/contact_template.php";
            break;
          case "admin_template":
            zabeleziPristupStranici($_GET["page"]);
            include "views/admin_template.php";
            break;
          default:
            zabeleziPristupStranici("page_404");
            include "views/404.php";
            break;
        }
      } else {
        include "views\main_template.php";
      }
      echo "</div>";
      include("views/footer_template.php");
    ?>
    <div class="notification">
      <span></span>
      <span></span>
    </div>
    <h1></h1>
  </body>

  <script type="text/javascript" src="assets/js/3rd/jquery.min.js"></script>

  <script type="text/javascript" src="assets/js/main.js"></script>

  <script type="text/javascript" src="assets/js/header_template.js"></script>
  <script type="text/javascript" src="assets/js/footer_template.js"></script>
  <script type="text/javascript" src="assets/js/main_template.js"></script>
  <script type="text/javascript" src="assets/js/food_post_template.js"></script>
  <script type="text/javascript" src="assets/js/add_post_template.js"></script>
  <script type="text/javascript" src="assets/js/login_template.js"></script>
  <script type="text/javascript" src="assets/js/contact_template.js"></script>
  <script type="text/javascript" src="assets/js/admin_template.js"></script>
</html>
```

3.2. views/login_template.php

```
<div class="landing_screen">
  <h2></h2>
  <img src="assets/images/baking.gif" alt=""/>
</div>
```

3.3. views/header_template.php

```
<div class="header">
  <div class="logo_and_nav">
    <div class="logo"><img src="assets/images/logo.png" alt="Logo"/></div>
    <span class="fas fa-bars" id="menu_click"></span>
  </div>
  <ul id="header_content"></ul>
  <div class="language">
    <img src="assets/images/serbia_flag.png" alt="" data-img="srb"/>
    <img src="assets/images/UK_flag.jpg" alt="" data-img="eng"/>
  </div>
</div>
```

3.4 views/footer_template.php

```
<div class="footer">
  <p class="copyright">&copy; Nikola Nedeljkovic 2019 &nbsp;&nbsp;&nbsp; @ ICT</p>
  <div class="footer_holder">
    <div class="footer_part footer_navigation">
      <h3>Navigation</h3>
      <ul id="navigation_footer">
        <?php
          if (isset($_SESSION["user"])) {
            if ($_SESSION["user"]->ID == 1) {
              echo '<li><a href="index.php?page=admin_template">Admin Panel</a></li>';
            }
          }
        ?>
      </ul>
    </div>
    <div class="footer_part footer_recents">
      <h3>Recent Posts</h3>
      <div id="recent_posts_footer"></div>
    </div>
    <div class="footer_part footer_tags">
      <h3>Tags</h3>
      <div id="tags_footer"></div>
    </div>
  </div>
</div>
```

3.5. views/food_post_template.php

```
<div id="new_post_area">
  <?php
    if (isset($_SESSION['user'])) {
      echo "<div class='add_post_area'>";
      echo "<div class='welcome_user'><div><span></span><span>".$_SESSION['user']->name." ".
$_SESSION['user']->last_name."</span></div><button id='logout'></button></div>";
      echo "<div id='add_post'>";
      include("add_post_template.php");
      echo "</div>";
      echo "</div>";
    } else {
      include("login_template.php");
    }
  ?>
</div>
<div id="main_panel_load_food"></div>
<div id="main_panel_load_edit">
  <div class="search_field">
    <input type="text" name="search" id="search" placeholder=""/>
    <button><span class="fa fa-search" aria-hidden="true"></span></button>
  </div>
  <div class="how_it_started">
    <h2></h2>
    <p></p>
  </div>
  <div class="rss_feed">
    <button><span class="fas fa-rss"></span></button>
  </div>
  <div class="quote_field">
    <h2></h2>
    <p></p>
  </div>
  <div class="subscribe_newsletter">
    <h2></h2>
    <input type="text" name="subscribe" id="subscribe" placeholder=""/>
    <p id="subscribe_error"></p>
    <p id="subscribe_text"></p>
  </div>
</div>
```

3.6 views/contact_template.php

```
<div id="contact_template">
  <div class="contact_area">
    <h2></h2>
    <input type="email" name="message_mail" id="message_mail" placeholder=""/>
    <textarea rows="8" cols="40" name="send_message" id="send_message"></textarea>
    <p id="message_error"></p>
    <button name="send_contact" id="sent_contact"></button>
  </div>
  <div class="author">
    <h2></h2>
    <img src="assets/images/author.jpg" alt=""/>
    <div class="author_name author_data">
      <p></p>
      <p>Nikola Nedeljković</p>
    </div>
    <div class="author_birth author_data">
      <p></p>
      <p>06.05.1996</p>
    </div>
    <div class="author_mail author_data">
      <p></p>
      <p>nedeljkovic.nikola.ks&#64;gmail.com</p>
    </div>
    <div class="author_number author_data">
      <p></p>
      <p>064/45-71-380</p>
    </div>
    <div class="author_work author_data">
      <p></p>
      <p>Junior JS Developer @ Summa Labs d.o.o</p>
    </div>
    <button name="download_author_data" id="download_author_data"></button>
  </div>
</div>
```

3.7 views/admin_template.php

```
<div id="admin_panel">
  <h2></h2>
  <select name="admin_option" id="admin_option"></select>
  <div class="admin_controls">
    <select name="admin_extra_option" id="admin_extra_option"></select>
    <div id="admin_extra_space"></div>
  </div>
  <div class="admin_controls file_controls">
    <h3></h3>
    <div class="admin_file_holder visiting_main">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_appetizer">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_main_course">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_dessert">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_contact">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_admin">
      <span></span>&nbsp;<span></span>
    </div>
    <div class="admin_file_holder visiting_404">
      <span></span>&nbsp;<span></span>
    </div>
  </div>
  <div class="admin_controls other_controls">
    <h3></h3>
    <div class="admin_file_holder number_logged_users">
      <span></span>&nbsp;<span></span>
    </div>
    <button name="export_to_excel" id="export_to_excel"></button>
    <a href="documentation/documentation.pdf"></a>
  </div>
</div>
```

3.8. views/add_post_template.php

```
<form onsubmit="return add_post_param_check()" method="POST" action="models/add_post.php"
class="add_post_form" name="add_post_form" enctype="multipart/form-data">
  <h2></h2>
  <div class="add_post_param">
    <label for="add_post_title"></label>
    <input type="text" name="add_post_title" id="add_post_title" placeholder=""/>
  </div>
  <div class="add_post_param">
    <label for="add_post_srb_title"></label>
    <input type="text" name="add_post_srb_title" id="add_post_srb_title" placeholder=""/>
  </div>
  <div class="add_post_param">
    <label for="add_post_text"></label>
    <textarea rows="4" cols="40" name="add_post_text" id="add_post_text"></textarea>
  </div>
  <div class="add_post_param">
    <label for="add_post_srb_text"></label>
    <textarea rows="4" cols="40" name="add_post_srb_text" id="add_post_srb_text"></textarea>
  </div>
  <div class="add_post_param">
    <label for="add_post_type"></label>
    <div class="add_post_param_holder add_post_type_holder"></div>
  </div>
  <div class="add_post_param">
    <label for="add_post_tag"></label>
    <div class="add_post_param_holder add_post_tag_holder"></div>
  </div>
  <div class="add_post_param">
    <label for="add_post_image"></label>
    <input type="file" name="add_post_image[]" id="add_post_image" multiple accept=".gif, .jpeg,
.jpg, .png"/>
  </div>
  <p id="add_post_error"></p>
  <input type="submit" name="add_post_button" id="add_post_button"/>
</form>
```

3.9 views/404.php

```
<div class="page_404">
  <img src="assets/images/404.png" alt="404"/>
</div>
```

3.10 models/add_post.php

```
<?php
  include("../config/connection.php");
  include("composer/contact_mail.php");
  session_start();
  $error = "";
  if (isset($_POST['add_post_button'])) {
    $title = $_POST['add_post_title'];
    $srbTitle = $_POST['add_post_srb_title'];
    $text = $_POST['add_post_text'];
    $srbText = $_POST['add_post_srb_text'];
    $type = $_POST['add_post_type'];
    $tags = $_POST['add_post_tag'];
    $images = $_FILES['add_post_image'];
    $title_patern = "/^[A-ZŠĐŽČĆ][a-z0-9ščćđž]+$/";
    $date = date("Y-m-d H:i:s", time());
    $user_id = $_SESSION["user"]->ID;
    $number_of_images = count($images["name"]);
    if ($title == "" || $srbTitle == "" || $text == "" || $srbText == "") { $error = "Fields are empty"; }
    if (!$type) { $error = "Type not selected"; }
    if (!preg_match($title_patern, $title) || !preg_match($title_patern, $srbTitle)) { $error = "Titles are
not valid"; }
    if (count($tags) <= 0 || $number_of_images <= 0) { $error = "Tags or Images not selected"; }
    try {
      $conn->beginTransaction();
      $query = "INSERT INTO post (type_id, user_id, title, SRBtitle, text, SRBtext, date) VALUES
(:type_id, :user_id, :title, :SRBtitle, :text, :SRBtext, :date)";
      $priprema = $conn -> prepare($query);
      $priprema->bindParam(":type_id", $type);
      $priprema->bindParam(":user_id", $user_id);
      $priprema->bindParam(":title", $title);
      $priprema->bindParam(":SRBtitle", $srbTitle);
      $priprema->bindParam(":text", $text);
      $priprema->bindParam(":SRBtext", $srbText);
      $priprema->bindParam(":date", $date);
      $rezultat = $priprema->execute();
      if (!$rezultat) { $error = "Error adding post"; }
      $last_post_id = $conn->lastInsertId();
      $tag_query = "INSERT INTO post_tag (ID_post, ID_tag) VALUES";
      foreach ($tags as $tag) {
        $tag_query .= " (".$last_post_id.",".$tag."),";
      }
      $tag_query = rtrim($tag_query,',');
      $tag_priprema = $conn -> prepare($tag_query);
      $tag_rezultat = $tag_priprema->execute();
      if (!$tag_rezultat) { $error = "Error adding tags"; }
      for ($i = 0; $i < $number_of_images; $i++) {
        $image_name = $images["name"][$i];
        $image_size = $images["size"][$i];
        $image_tmp = $images["tmp_name"][$i];

        $imageFileType = strtolower(pathinfo($image_name,PATHINFO_EXTENSION));
        if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" &&
$imageFileType != "gif" ) { $error = "Not valid image"; }

        $alt = $title . " " . ($i + 1);
        $srbAlt = $srbTitle . " " . ($i + 1);
        $upload_dir = '../assets/images/recipes/';
        $src = 'assets/images/recipes/'.$image_name;

        $add_picture = "INSERT INTO pictures (src, alt, SRBalt) VALUES (:src, :alt, :srbAlt)";
        $priprema_pic = $conn -> prepare($add_picture);
        $priprema_pic->bindParam(":src", $src);
        $priprema_pic->bindParam(":alt", $alt);
        $priprema_pic->bindParam(":srbAlt", $srbAlt);
        $rezultat_pic = $priprema_pic->execute();
        if (!$tag_rezultat) { $error = "Error adding image"; }
        $last_picture_id = $conn->lastInsertId();
        $image_path = $upload_dir."food_".$last_picture_id.".".$imageFileType;

        list($sirina, $visina) = getimagesize($image_tmp);
        $novaVisina = 200;
        $novaSirina = ($novaVisina/$visina) * $sirina;
        $novaSlika = imagecreatetruecolor($novaSirina, $novaVisina);
        $postojecaSlika = null;
        switch ($imageFileType) {
          case 'jpeg':
          case 'jpg':
            $postojecaSlika = imagecreatefromjpeg($image_tmp);
            break;
          case 'png':
            $postojecaSlika = imagecreatefrompng($image_tmp);
            break;
          case 'gif':
            $postojecaSlika = imagecreatefromgif($image_tmp);
            break;
        }
        imagecopyresampled($novaSlika, $postojecaSlika, 0, 0, 0, 0, $novaSirina, $novaVisina, $sirina,
$visina);
        switch ($imageFileType) {
          case 'jpeg':
          case 'jpg':
            imagejpeg($novaSlika, $image_path, 75);
            break;
          case 'png':
            imagepng($novaSlika, $image_path);
            break;
          case 'gif':
            imagegif($novaSlika, $image_path);
            break;
        }

        $new_src = 'assets/images/recipes/food_'.$last_picture_id.".".$imageFileType;
        $change_pic = "UPDATE pictures SET src = :src WHERE ID = :id";
        $priprema_change = $conn -> prepare($change_pic);
        $priprema_change->bindParam(":src", $new_src);
        $priprema_change->bindParam(":id", $last_picture_id);
        $rezultat_change = $priprema_change->execute();
        if (!$rezultat_change) { $error = "Error changing name in database"; }

        $post_picture_query = "INSERT INTO post_pictures (ID_post, ID_picture) VALUES (:post_id,
:picture_id)";
        $priprema_pp = $conn -> prepare($post_picture_query);
        $priprema_pp->bindParam(":post_id", $last_post_id);
        $priprema_pp->bindParam(":picture_id", $last_picture_id);
        $rezultat_pp = $priprema_pp->execute();
        if (!$rezultat_pp) { $error = "Error adding to table"; }

        $srb_xml = simplexml_load_file("../data/rss-srb.xml") or die("Error: Cannot create object");
        $srb_channel = $srb_xml->channel;
        $item = $srb_channel->addChild('item');
        $item->addChild('title', $srbTitle);
        $item->addChild('link', 'http://25h-kitchen.epizy.com/index.php');
        $item->addChild('description', $_SESSION["user"]->name.' '.$_SESSION["user"]->last_name);
        $srb_xml->asXML("../data/rss-srb.xml");

        $eng_xml = simplexml_load_file("../data/rss-eng.xml") or die("Error: Cannot create object");
        $eng_channel = $eng_xml->channel;
        $item = $eng_channel->addChild('item');
        $item->addChild('title', $title);
        $item->addChild('link', 'http://25h-kitchen.epizy.com/index.php');
        $item->addChild('description', $_SESSION["user"]->name.' '.$_SESSION["user"]->last_name);
        $eng_xml->asXML("../data/rss-eng.xml");
      }
      if ($error != "") {
        echo $error;
        $conn->rollback();
      } else {
        $conn->commit();
        $query = "SELECT * FROM newsletter_subscribers";
        $subscribers = executeQuery($query);
        foreach ($subscribers as $subscriber) {
               $result = sendOthersMail($subscriber->email, "New post: ".$title);
            }
            header("Location: ../index.php");
            die();
         }
      }catch (PDOException $e) {
         echo $e->getMessage();
         $conn->rollback();
      }
   } else {
      $error = "Must send by form!";
      echo $error;
   }
?>
```

3.11. models/comment.php

```
<?php
   header("Content-Type: application/json");
   session_start();
   include("../config/connection.php");
   $code = 500;
   $error = "";
   $post_id = $_POST["post_id"];
   $comment_text = $_POST["comment_text"];

   if (strlen($comment_text) < 5) { $error = "empty_comment"; }
   if (!isset($_SESSION["user"])) { $error = "log_to_comment"; }

   if ($error == "") {
      $user_id = $_SESSION["user"]->ID;
      $date = date("Y-m-d H:i:s", time());
      $query = "INSERT INTO comments (user_id, post_id, text, date) VALUES (:user_id, :post_id, :text,
:date)";
      $priprema = $conn -> prepare($query);
      $priprema->bindParam(":user_id", $user_id);
      $priprema->bindParam(":post_id", $post_id);
      $priprema->bindParam(":text", $comment_text);
      $priprema->bindParam(":date", $date);
      $rezultat = $priprema->execute();
      if ($rezultat) {
         $code = 200;
         echo json_encode("Successful");
      } else {
         $code = 503;
         echo json_encode("database_error");
      }
```

```
  } else {
    $code = 503;
    echo json_encode($error);
  }
  http_response_code($code);
?>
```

3.12 models/comments.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $code = 500;
  $postId = $_GET["postId"];
  $query = "SELECT c.text AS commentText, c.date AS commentDate, u.name AS userName,
u.last_name AS userLastName FROM comments c INNER JOIN users u ON c.user_id = u.ID WHERE
c.post_id = :postId";
  $priprema = $conn->prepare($query);
  $priprema->bindParam(":postId", $postId);
  $rezultat = $priprema->execute();
  if ($rezultat) {
    $posts = $priprema->FetchAll();
    $code = 200;
    echo json_encode($posts);
  } else {
    $code = 504;
    echo json_encode("Update failed");
  }
  http_response_code($code);
?>
```

3.13. models/contact.php

```
<?php
  header("Content-Type: application/json");
  include("composer/contact_mail.php");
  $mail = $_POST["mail"];
  $text = $_POST["text"];
  $code = 500;
  $error = "";
  $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
  if ($mail == "" || $text == "") { $error = "empty_field"; }
  if (!preg_match($email_pattern, $mail)) { $error = "mail_invalid"; }
  if ($error == "") {
    $result = sendContactMail($mail, $text);
    if ($result) {
      $code = 201;
      echo json_encode("Successful");
    } else {
          $code = 503;
          echo json_encode("database_error");
       }
   } else {
       $code = 503;
       echo json_encode($error);
   }
   http_response_code($code);
?>
```

3.14. models/export/excel.php

```
<?php
http_response_code(200);
use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

require_once($_SERVER["DOCUMENT_ROOT"]."/config/connection.php");
require_once ABSOLUTE_PATH.'/models/composer/vendor/autoload.php';
$save_loc = exec('echo %SystemDrive%') . '\\Users\\' . get_current_user() . '\\Desktop';

   $posts = executeQuery("SELECT p.ID AS id, title, SRBtitle, text, SRBtext, name, last_name FROM post p
INNER JOIN users u ON p.user_id = u.ID");
   $spreadsheet = new Spreadsheet();
   $sheet = $spreadsheet->getActiveSheet();
   $sheet->setCellValue('A1', 'Post ID:');
   $sheet->setCellValue('B1', 'Title:');
   $sheet->setCellValue('C1', 'Naslov:');
   $sheet->setCellValue('D1', 'Text:');
   $sheet->setCellValue('E1', 'Tekst:');
   $sheet->setCellValue('F1', 'Author/Autor:');
   $i = 0;
   foreach ($posts as $post) {
       $sheet->setCellValue('A'.($i + 2), $post->id);
       $sheet->setCellValue('B'.($i + 2), $post->title);
       $sheet->setCellValue('C'.($i + 2), $post->SRBtitle);
       $sheet->setCellValue('D'.($i + 2), $post->text);
       $sheet->setCellValue('E'.($i + 2), $post->SRBtext);
       $sheet->setCellValue('F'.($i + 2), $post->name." ".$post->last_name);
       $i += 1;
   }
   $writer = new Xlsx($spreadsheet);
   $writer->save($save_loc.'\\posts.xlsx');
   echo "OK";
?>
```

3.15. models/forgot_password.php

```
<?php
   header("Content-Type: application/json");
   include("composer/contact_mail.php");
   $mail = $_POST["email"];
   $code = 500;
   $error = "";
   $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
   if ($mail == "") { $error = "empty_field"; }
   if (!preg_match($email_pattern, $mail)) { $error = "mail_invalid"; }
   if ($error == "") {
      $result = sendContactMail($mail, "User: ".$mail. " requests password change.");
      if ($result) {
         $code = 201;
         echo json_encode("Successful");
      } else {
         $code = 503;
         echo json_encode("database_error");
      }
   } else {
      $code = 503;
      echo json_encode($error);
   }
   http_response_code($code);
?>
```

3.16. models/logged_users.php

```
<?php
   header("Content-Type: application/json");
   include("../config/connection.php");
   $logged = executeQuery("SELECT COUNT(*) AS logged_users FROM users WHERE LastTimeSeen >
DATE_SUB(NOW(), INTERVAL 5 MINUTE)");
   echo json_encode($logged);
?>
```

3.17. models/login.php

```
<?php
   header("Content-Type: application/json");
   session_start();
   include("../config/connection.php");
   include("composer/contact_mail.php");
   $code = 500;
   $error = "";
   $email = $_POST["email"];
   $password = $_POST["password"];
   $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
   if ($email == "" || $password == "") { $error = "empty_field"; }
```

```
    if (!preg_match($email_pattern, $email)) { $error = "not_valid_email"; }
    if (strlen($password) < 6) { $error = "password_short"; }

    if ($error == "") {
        $query = "SELECT * FROM users WHERE email=:email AND password=:password";
        $password = md5($password);
        $priprema = $conn -> prepare($query);
        $priprema->bindParam(":email", $email);
        $priprema->bindParam(":password", $password);
        $rezultat = $priprema->execute();
        if ($rezultat) {
            if ($priprema -> rowCount() == 1) {
                $user = $priprema -> Fetch();
                $_SESSION["user"] = $user;
                $code = 200;
                echo json_encode("Successful");
            } else if ($priprema -> rowCount() == 0) {
                $code = 203;
                $result = sendOthersMail($email, "Somebody tried to access your account on ".date('d-m-Y
H:i:s'));
                echo json_encode("wrong_params");
            } else {
                $code = 503;
                echo json_encode("database_error");
            }
        } else {
            $code = 503;
            echo json_encode("database_error");
        }
    } else {
        $code = 503;
        echo json_encode($error);
    }
    http_response_code($code);
?>
```

3.18. models/logout.php

```
<?php
    session_start();
    if(isset($_SESSION['user'])) {
        unset($_SESSION['user']);
        session_destroy();
    }
    echo true;
?>
```

3.19. models/navigation.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $query = "SELECT * FROM navigation";
  $navigation = executeQuery($query);
  echo json_encode($navigation);
?>
```

3.20. models/page_access_info.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $podaci = file(LOG_FAJL);
  $objekat = array("main_template" => 0,
            "appetizer" => 0,
            "main_course" => 0,
            "dessert" => 0,
            "contact_template" => 0,
            "admin_template" => 0,
            "page_404" => 0);

  foreach ($podaci as $key => $value) {
    $vrednost = explode("\t", $value);
    $date_from_file = strtotime($vrednost[1]);
    if (time() - $date_from_file <= 24 * 60 * 60) {
      $objekat[$vrednost[0]] += 1;
    }
  }
  echo json_encode($objekat);
?>
```

3.21. models/posts.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $code = 500;
  $type_id = $_GET["type_id"];
  $selected_tags = $_GET["selected_tags"];
  $tag_count = count($selected_tags);
  $query = "SELECT p.ID AS id, p.title AS title, p.SRBtitle AS SRBtitle, p.text AS postText, p.SRBtext AS SRBtext, p.date AS postDate, u.name AS name, u.last_name AS last_name, pic.src AS picsrc, pic.ID AS picid, pic.alt AS picalt, pic.SRBalt AS picSRBalt, t.name AS tagname, t.ID AS tagid, t.SRBname AS tagSRBname, (SELECT COUNT(*) FROM comments c WHERE c.post_id = p.ID) AS commentNumber FROM post p INNER JOIN users u ON p.user_id = u.ID INNER JOIN post_pictures pp ON p.ID = pp.ID_post
```

```
INNER JOIN pictures pic ON pp.ID_picture = pic.ID INNER JOIN post_tag pt ON p.ID = pt.ID_post INNER
JOIN tags t ON t.ID = pt.ID_tag WHERE type_id = :type_id";
  if ($tag_count >= 1 && $selected_tags[0] != "") {
    $query .= " AND pt.ID_tag IN (".implode(',',$selected_tags).")";
  }
  if (isset($_GET["text"])) {
    $text = $_GET["text"];
    $text = strtolower($text);
    $query .=  " AND LOWER(p.title) LIKE '%".$text."%'";
  }
  $query .= " ORDER BY p.date DESC";
  $priprema = $conn->prepare($query);
  $priprema->bindParam(":type_id", $type_id);
  $rezultat = $priprema->execute();
  if ($rezultat) {
    $posts = $priprema->FetchAll();
    $code = 200;
    echo json_encode($posts);
  } else {
    $code = 504;
    echo json_encode("Update failed");
  }
  http_response_code($code);
?>
```

3.22. models/random_quote.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $query = "SELECT COUNT(*) AS number FROM food_quotes";
  $max_number = executeQuery($query);
  $random_number = rand(0, $max_number[0]->number - 1);
  $query = "SELECT * FROM food_quotes";
  $food_quotes = executeQuery($query);
  echo json_encode($food_quotes[$random_number]);
?>
```

3.23. models/register.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $code = 500;
  $error = "";
  $email = $_POST["email"];
  $name = $_POST["name"];
  $lastname = $_POST["lastname"];
  $password = $_POST["password"];
  $password_repeat = $_POST["password_repeat"];
  $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
  $name_pat = "/^[A-Z][a-z]+$/";
  $lastname_pat = "/^([A-Z][a-z]+)+$/";
  if ($email == "" || $name == "" || $lastname == "" || $password == "" || $password_repeat == "") {
$error = "empty_field"; }
  if (strlen($password) < 6) { $error = "password_short"; }
  if ($password != $password_repeat) { $error = "passwords_not_match"; }
  if (!preg_match($email_pattern, $email)) { $error = "not_valid_email"; }
  if (!preg_match($name_pat, $name) || !preg_match($lastname_pat, $lastname)) { $error =
"name_invalid"; }
  $query = "SELECT COUNT(email) AS rowCount FROM users WHERE email = :email";
  $priprema = $conn -> prepare($query);
  $priprema->bindParam(":email", $email);
  $rezultat = $priprema->execute();
  $redovi = $priprema->FetchAll();
  $redovi = $redovi[0]->rowCount;
  if ($redovi == 1) { $error = "email_taken"; }
  if ($error == "") {
    $query = "INSERT INTO users (email, password, rights_id, name, last_name) VALUES (:email,
:password, 2, :name, :lastname)";
    $password = md5($password);
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":email", $email);
    $priprema->bindParam(":password", $password);
    $priprema->bindParam(":name", $name);
    $priprema->bindParam(":lastname", $lastname);
    $rezultat = $priprema->execute();
    if ($rezultat) {
      $code = 201;
      echo json_encode("Successful");
    } else {
      $code = 503;
      echo json_encode("database_error");
    }
  } else {
    $code = 503;
    echo json_encode($error);
  }
  http_response_code($code);
?>
```

3.24. models/save_word.php

```
<?php
  @ ob_start();
  header("Content-Type: application/octet-stream");
  header("Content-Disposition: attachment; filename=nikola_nedeljkovic.doc");
  http_response_code(200);
  $save_loc = exec('echo %SystemDrive%') . '\\Users\\' . get_current_user() . '\\Desktop';

  $word = new COM("word.application") or die("Unable to instantiate Word");
  $word->Visible = 1;
  $word->Documents->Add();
  $word->Selection->TypeText($_POST['author_name']);
  $word->Selection->TypeText($_POST['author_name_val']);
  $word->Selection->TypeText("\n");
  $word->Selection->TypeText($_POST['author_birth']);
  $word->Selection->TypeText($_POST['author_birth_val']);
  $word->Selection->TypeText("\n");
  $word->Selection->TypeText($_POST['author_mail']);
  $word->Selection->TypeText($_POST['author_mail_val']);
  $word->Selection->TypeText("\n");
  $word->Selection->TypeText($_POST['author_number']);
  $word->Selection->TypeText($_POST['author_number_val']);
  $word->Selection->TypeText("\n");
  $word->Selection->TypeText($_POST['author_work']);
  $word->Selection->TypeText($_POST['author_work_val']);

  $word->Documents[1]->SaveAs($save_loc.'\\nikola_nedeljkovic.doc');
  $word->Quit();
  $word = null;

  echo "OK";
?>
```

3.25. models/single_post.php

```
<?php
  header("Content-Type: application/json");
  include("../config/connection.php");
  $code = 500;
  $postId = $_POST["postId"];
  $query = "SELECT p.ID AS id, p.title AS title, p.SRBtitle AS SRBtitle, p.text AS postText, p.SRBtext AS SRBtext, p.date AS postDate, u.name AS name, u.last_name AS last_name, pic.src AS picsrc, pic.ID AS picid, pic.alt AS picalt, pic.SRBalt AS picSRBalt, t.name AS tagname, t.ID AS tagid, t.SRBname AS tagSRBname, (SELECT COUNT(*) FROM comments c WHERE c.post_id = p.ID) AS commentNumber FROM post p INNER JOIN users u ON p.user_id = u.ID INNER JOIN post_pictures pp ON p.ID = pp.ID_post INNER JOIN pictures pic ON pp.ID_picture = pic.ID INNER JOIN post_tag pt ON p.ID = pt.ID_post INNER JOIN tags t ON t.ID = pt.ID_tag WHERE p.ID = :postId";
  $priprema = $conn->prepare($query);
```

```
   $priprema->bindParam(":postId", $postId);
   $rezultat = $priprema->execute();
   if ($rezultat) {
      $posts = $priprema->FetchAll();
      $code = 200;
      echo json_encode($posts);
   } else {
      $code = 504;
      echo json_encode("Update failed");
   }
   http_response_code($code);
?>
```

3.26. models/subscribe.php

```
<?php
   header("Content-Type: application/json");
   include("../config/connection.php");
   $code = 500;
   $error = "";
   $subscription_mail = $_POST["subscription_mail"];
   $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
   if ($subscription_mail == "") { $error = "subscribe_empty_field"; }
   if (!preg_match($email_pattern, $subscription_mail)) { $error = "not_valid_email"; }
   $query = "SELECT COUNT(email) AS rowCount FROM newsletter_subscribers WHERE email = 
:subscription_mail";
   $priprema = $conn -> prepare($query);
   $priprema->bindParam(":subscription_mail", $subscription_mail);
   $rezultat = $priprema->execute();
   $redovi = $priprema->FetchAll();
   $redovi = $redovi[0]->rowCount;
   if ($redovi == 1) { $error = "subscription_mail_exists"; }
   if ($error == "") {
      $query = "INSERT INTO newsletter_subscribers (email) VALUES (:subscription_mail)";
      $priprema = $conn -> prepare($query);
      $priprema->bindParam(":subscription_mail", $subscription_mail);
      $rezultat = $priprema->execute();
      if ($rezultat) {
         $code = 201;
         echo json_encode("Successful");
      } else {
         $code = 503;
         echo json_encode("database_error");
      }
   } else {
      $code = 503;
      echo json_encode($error);
   }
```

```
    http_response_code($code);
?>
```

3.27. models/tags.php

```
<?php
    header("Content-Type: application/json");
    include("../config/connection.php");
    $query = "SELECT * FROM tags";
    $tags = executeQuery($query);
    echo json_encode($tags);
?>
```

3.28. models/top_posts.php

```
<?php
    header("Content-Type: application/json");
    include("../config/connection.php");
    $query = "SELECT p.title AS title, p.SRBtitle AS SRBtitle, p.date AS postDate, u.name AS name,
u.last_name AS lastname FROM post p INNER JOIN users u ON p.user_id = u.ID ORDER BY p.date DESC
LIMIT 3";
    $top_posts = executeQuery($query);
    echo json_encode($top_posts);
?>
```

3.29. views/types.php

```
<?php
    header("Content-Type: application/json");
    include("../config/connection.php");
    $query = "SELECT * FROM types";
    $types = executeQuery($query);
    echo json_encode($types);
?>
```

3.30. views/write_error.php

```
<?php
    include("../config/connection.php");
    $text = $_POST["text"];
    $open = fopen(ERROR_FAJL, "a");
    if ($open) {
        $date = date('d-m-Y H:i:s');
        fwrite($open, "{$content}\t{$date}\t\n");
        fclose($open);
    }
    echo "OK";
?>
```

3.31. models/composer/contact_mail.php

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require_once($_SERVER["DOCUMENT_ROOT"]."/config/connection.php");
require_once ABSOLUTE_PATH.'/models/composer/vendor/autoload.php';

function sendContactMail ($to, $content) {
  $mail = new PHPMailer(true);
  try {
    $mail->SMTPDebug = 0;
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'twentyfivekitchen@gmail.com';
    $mail->Password = '25Kitchen';
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    $mail->setFrom('twentyfivekitchen@gmail.com', '25Hour Kitchen');
    $mail->addAddress('twentyfivekitchen@gmail.com');
    $mail->addReplyTo($to, 'Information: ');

    $mail->isHTML(true);
    $mail->Subject = '25Hours Kitchen Contact Form';
    $mail->Body = $content;

    $mail->send();

    return true;
  } catch (Exception $e) {
    echo $e;
    return false;
  }
}

function sendOthersMail ($to, $content) {
  $mail = new PHPMailer(true);
  try {
    $mail->SMTPDebug = 0;
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'twentyfivekitchen@gmail.com';
    $mail->Password = '25Kitchen';
    $mail->SMTPSecure = 'tls';
```

```
        $mail->Port = 587;

        $mail->setFrom($to, '25Hour Kitchen');
        $mail->addAddress($to);
        $mail->addReplyTo($to, 'Information: ');

        $mail->isHTML(true);
        $mail->Subject = '25Hours Kitchen Contact Form';
        $mail->Body = $content;

        $mail->send();

        return true;
    } catch (Exception $e) {
        echo $e;
        return false;
    }
}

?>
```

3.32. models/admin/comments.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $query = "SELECT text, ID FROM comments";
    $comments = executeQuery($query);
    echo json_encode($comments);
?>
```

3.33. models/admin/delete_comment.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $code = 500;
    $comment = $_POST["comment"];
    $query = "DELETE FROM comments WHERE ID = :comment";
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":comment", $comment);
    $rezultat = $priprema->execute();
    if ($rezultat) {
        $code = 202;
        echo json_encode("Successful");
    } else {
        $code = 503;
        echo json_encode("database_error");
    }
```

```
   http_response_code($code);
?>
```

3.34. models/admin/delete_newsletter.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $code = 500;
   $newsletter = $_POST["newsletter"];
   $query = "DELETE FROM newsletter_subscribers WHERE ID = :newsletter";
   $priprema = $conn -> prepare($query);
   $priprema->bindParam(":newsletter", $newsletter);
   $rezultat = $priprema->execute();
   if ($rezultat) {
      $code = 202;
      echo json_encode("Successful");
   } else {
      $code = 503;
      echo json_encode("database_error");
   }
   http_response_code($code);
?>
```

3.35. models/admin/delete_picture.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $code = 500;
   $picture_id = $_POST["picture"];
   $query_select = "SELECT * FROM pictures WHERE ID = :picture";
   $priprema_select = $conn -> prepare($query_select);
   $priprema_select->bindParam(":picture", $picture_id);
   $rezultat_select = $priprema_select->execute();
   if ($rezultat_select) {
      $pictures = $priprema_select->FetchAll();
      foreach ($pictures as $picture) {
         $where = '../../'.$picture->src;
         unlink($where);
      }
      $query = "DELETE FROM pictures WHERE ID = :picture";
      $priprema = $conn -> prepare($query);
      $priprema->bindParam(":picture", $picture_id);
      $rezultat = $priprema->execute();
      if ($rezultat) {
         $code = 202;
         echo json_encode("Successful");
      } else {
            $code = 503;
            echo json_encode("database_error");
        }
    } else {
        $code = 503;
        echo json_encode("database_error");
    }

    http_response_code($code);
?>
```

3.36. models/admin/delete_post_tag.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $code = 500;
    $post_tag = $_POST["post_tag"];
    $ids = explode("-", $post_tag);
    $query = "DELETE FROM post_tag WHERE ID_post = :post_id AND ID_tag = :tag_id";
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":post_id", $ids[0]);
    $priprema->bindParam(":tag_id", $ids[1]);
    $rezultat = $priprema->execute();
    if ($rezultat) {
        $code = 202;
        echo json_encode("Successful");
    } else {
        $code = 503;
        echo json_encode("database_error");
    }
    http_response_code($code);
?>
```

3.37. models/admin/delete_post.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $code = 500;
    $error = "";
    $post = $_POST["post"];
    $conn->beginTransaction();
    $query_pictures = "SELECT * FROM post_pictures WHERE ID_post = :post";
    $priprema_pictures = $conn -> prepare($query_pictures);
    $priprema_pictures->bindParam(":post", $post);
    $rezultat_pictures = $priprema_pictures->execute();
    if ($rezultat_pictures) {
        $picture_ids = $priprema_pictures->FetchAll();
```

```
foreach ($picture_ids as $picture_id) {
   $query_pict = "SELECT * FROM pictures WHERE ID = :pic_id";
   $priprema_pict = $conn -> prepare($query_pict);
   $priprema_pict->bindParam(":pic_id", $picture_id->ID_picture);
   $rezultat_pict = $priprema_pict->execute();
   if ($rezultat_pict) {
      $pictures = $priprema_pict->FetchAll();
      foreach($pictures as $picture) {
         $where = '../../'.$picture->src;
         unlink($where);
         $query_pic = "DELETE FROM pictures WHERE ID = :picture";
         $priprema_pic = $conn -> prepare($query_pic);
         $priprema_pic->bindParam(":picture", $picture->ID);
         $rezultat_pic = $priprema_pic->execute();
         if (!$rezultat_pic) {
            $error = "database_error";
         }
      }
   } else {
      $error = "database_error";
   }
}
$query_post = "SELECT * FROM post WHERE ID = :post";
$priprema_post = $conn ->prepare($query_post);
$priprema_post->bindParam(":post", $post);
$rezultat_post = $priprema_post->execute();
if ($rezultat_post) {
   $post_rez = $priprema_post -> Fetch();
   $srb_xml = simplexml_load_file("../../data/rss-srb.xml") or die("Error: Cannot create object");
   $srb_channel = $srb_xml->channel;
   $srb_nodes = $srb_xml->xpath('//channel/item/title[.="'.$post_rez->SRBtitle.'"]/parent::*');
   $srb_trazeni_node = $srb_nodes[0];
   unset($srb_trazeni_node);
   $srb_xml->asXML("../../data/rss-srb.xml");

   $eng_xml = simplexml_load_file("../../data/rss-eng.xml") or die("Error: Cannot create object");
   $eng_channel = $eng_xml->channel;
   $eng_nodes = $eng_xml->xpath('//channel/item/title[.="'.$post_rez->title.'"]/parent::*');
   $eng_trazeni_node = $eng_nodes[0];
   unset($eng_trazeni_node);
   $eng_xml->asXML("../../data/rss-eng.xml");
} else {
   $error = "database_error";
}


$query = "DELETE FROM post WHERE ID = :post";
$priprema = $conn -> prepare($query);
```

```
        $priprema->bindParam(":post", $post);
        $rezultat = $priprema->execute();
        if ($rezultat) {
            $code = 202;
            if ($error == "") {
                echo json_encode("Successful");
                $conn->commit();
            } else {
                $conn->rollback();
                $code = 503;
                echo $error;
            }
        } else {
            $code = 503;
            $conn->rollback();
            echo json_encode("database_error");
        }
    } else {
        $code = 503;
        $conn->rollback();
        echo json_encode("database_error");
    }
    http_response_code($code);
?>
```

3.38. models/admin/delete_tag.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $code = 500;
    $tag = $_POST["tag"];
    $query = "DELETE FROM tags WHERE ID = :tag";
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":tag", $tag);
    $rezultat = $priprema->execute();
    if ($rezultat) {
        $code = 202;
        echo json_encode("Successful");
    } else {
        $code = 503;
        echo json_encode("database_error");
    }
    http_response_code($code);
?>
```

3.39. models/admin/delete_user.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $code = 500;
  $user = $_POST["user"];
  $query = "DELETE FROM users WHERE ID = :user";
  $priprema = $conn -> prepare($query);
  $priprema->bindParam(":user", $user);
  $rezultat = $priprema->execute();
  if ($rezultat) {
    $code = 202;
    echo json_encode("Successful");
  } else {
    $code = 503;
    echo json_encode("database_error");
  }
  http_response_code($code);
?>
```

3.40. models/admin/edit_post.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $code = 500;
  $error = "";
  $title = $_POST["title"];
  $srb_title = $_POST["srb_title"];
  $text = $_POST["text"];
  $srb_text = $_POST["srb_text"];
  $id = $_POST["id"];
  $title_pat = "/^([A-Z][a-z]+)+$/";
  if ($title == "" || $srb_title == "" || $text == "" || $srb_text == "") { $error = "empty_field"; }
  if (!preg_match($title_pat, $title) || !preg_match($title_pat, $srb_title)) { $error =
"title_name_invalid"; }
  if ($error == "") {
    $query_post = "SELECT * FROM post WHERE ID = :id";
    $priprema_post = $conn ->prepare($query_post);
    $priprema_post->bindParam(":id", $id);
    $rezultat_post = $priprema_post->execute();
    if ($rezultat_post) {
      $post_rez = $priprema_post -> Fetch();
      $srb_xml = simplexml_load_file("../../data/rss-srb.xml") or die("Error: Cannot create object");
      $srb_nodes = $srb_xml->xpath('//channel/item/title[.="'.$post_rez->SRBtitle.'"]/parent::*');
      $srb_trazeni_node = $srb_nodes[0];
      $srb_trazeni_node->title = $srb_title;
      $srb_xml->asXML("../../data/rss-srb.xml");
```

```
         $eng_xml = simplexml_load_file("../../data/rss-eng.xml") or die("Error: Cannot create object");
         $eng_nodes = $eng_xml->xpath('//channel/item/title[.="'.$post_rez->title.'"]/parent::*');
         $eng_trazeni_node = $eng_nodes[0];
         $eng_trazeni_node->title = $title;
         $eng_xml->asXML("../../data/rss-eng.xml");

         $query = "UPDATE post SET title = :title, SRBtitle = :srb_title, text = :text, SRBtext = :srb_text
WHERE ID = :id";
         $priprema = $conn -> prepare($query);
         $priprema->bindParam(":title", $title);
         $priprema->bindParam(":srb_title", $srb_title);
         $priprema->bindParam(":text", $text);
         $priprema->bindParam(":srb_text", $srb_text);
         $priprema->bindParam(":id", $id);
         $rezultat = $priprema->execute();
         if ($rezultat) {
           $code = 201;
           echo json_encode("Successful");
         } else {
           $code = 503;
           echo json_encode("database_error");
         }
       } else {
         $code = 503;
         echo json_encode("database_error");
       }
   } else {
     $code = 503;
     echo json_encode($error);
   }
   http_response_code($code);
?>
```

3.41. models/admin/edit_tag.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $code = 500;
   $error = "";
   $name = $_POST["name"];
   $srb_name = $_POST["srb_name"];
   $id = $_POST["id"];
   $name_pat = "/^[A-Z][a-z]+$/";
   if ($name == "" || $srb_name == "") { $error = "empty_field"; }
   if (!preg_match($name_pat, $name) || !preg_match($name_pat, $srb_name)) { $error =
"tag_name_invalid"; }
```

```
  $query = "SELECT COUNT(name) AS rowCount FROM tags WHERE (name = :name OR SRBname =
:srb_name) AND ID != :id";
  $priprema = $conn -> prepare($query);
  $priprema->bindParam(":name", $name);
  $priprema->bindParam(":srb_name", $srb_name);
  $priprema->bindParam(":id", $id);
  $rezultat = $priprema->execute();
  $redovi = $priprema->FetchAll();
  $redovi = $redovi[0]->rowCount;
  if ($redovi == 1) { $error = "tag_exists"; }
  if ($error == "") {
    $query = "UPDATE tags SET name = :name, SRBname = :srb_name WHERE ID = :id";
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":name", $name);
    $priprema->bindParam(":srb_name", $srb_name);
    $priprema->bindParam(":id", $id);
    $rezultat = $priprema->execute();
    if ($rezultat) {
      $code = 201;
      echo json_encode("Successful");
    } else {
      $code = 503;
      echo json_encode("database_error");
    }
  } else {
    $code = 503;
    echo json_encode($error);
  }
  http_response_code($code);
?>
```

3.42. models/admin/edit_user.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $code = 500;
  $error = "";
  $email = $_POST["email"];
  $name = $_POST["name"];
  $lastname = $_POST["lastname"];
  $id = $_POST["id"];
  $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
  $name_pat = "/^[A-Z][a-z]+$/";
  $lastname_pat = "/^([A-Z][a-z]+)+$/";
  if ($email == "" || $name == "" || $lastname == "") { $error = "empty_field"; }
  if (!preg_match($email_pattern, $email)) { $error = "not_valid_email"; }
```

```
   if (!preg_match($name_pat, $name) || !preg_match($lastname_pat, $lastname)) { $error =
"name_invalid"; }
   $query = "SELECT COUNT(email) AS rowCount FROM users WHERE email = :email AND id != :id";
   $priprema = $conn -> prepare($query);
   $priprema->bindParam(":email", $email);
   $priprema->bindParam(":id", $id);
   $rezultat = $priprema->execute();
   $redovi = $priprema->FetchAll();
   $redovi = $redovi[0]->rowCount;
   if ($redovi == 1) { $error = "email_taken"; }
   if ($error == "") {
      $query = "UPDATE users SET email = :email, name = :name, last_name = :lastname WHERE ID = :id";
      $priprema = $conn -> prepare($query);
      $priprema->bindParam(":email", $email);
      $priprema->bindParam(":name", $name);
      $priprema->bindParam(":lastname", $lastname);
      $priprema->bindParam(":id", $id);
      $rezultat = $priprema->execute();
      if ($rezultat) {
         $code = 201;
         echo json_encode("Successful");
      } else {
         $code = 503;
         echo json_encode("database_error");
      }
   } else {
      $code = 503;
      echo json_encode($error);
   }
   http_response_code($code);
?>
```

3.43. models/admin/insert_picture.php

```
<?php
   include("../../config/connection.php");
   $error = "";
   if (isset($_POST['admin_insert_picture_submit'])) {
      $alt = $_POST['admin_insert_picture_alt'];
      $srb_alt = $_POST['admin_insert_picture_srbalt'];
      $post_id = $_POST['admin_insert_picture_options'];

      $image = $_FILES['admin_insert_picture_picture'];
      $name_pat = "/^[A-Z][a-z]+$/";
      if ($alt == "" || $srb_alt == "") { $error = "Fields are empty"; }
      if (!preg_match($name_pat, $alt) || !preg_match($name_pat, $srb_alt)) { $error = "Titles are not
valid"; }
      try {
        $conn->beginTransaction();
        $image_name = $image["name"];
        $image_size = $image["size"];
        $image_tmp = $image["tmp_name"];

        $imageFileType = strtolower(pathinfo($image_name,PATHINFO_EXTENSION));
        if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" &&
$imageFileType != "gif" ) { $error = "Not valid image"; }

        $alt = $alt . " 1";
        $srb_alt = $srb_alt . " 1";
        $upload_dir = '../../assets/images/recipes/';
        $src = 'assets/images/recipes/'.$image_name;

        $add_picture = "INSERT INTO pictures (src, alt, SRBalt) VALUES (:src, :alt, :srb_alt)";
        $priprema_pic = $conn -> prepare($add_picture);
        $priprema_pic->bindParam(":src", $src);
        $priprema_pic->bindParam(":alt", $alt);
        $priprema_pic->bindParam(":srb_alt", $srb_alt);
        $rezultat_pic = $priprema_pic->execute();
        $last_picture_id = $conn->lastInsertId();
        $image_path = $upload_dir."food_".$last_picture_id.".".$imageFileType;

        list($sirina, $visina) = getimagesize($image_tmp);
        $novaVisina = 200;
        $novaSirina = ($novaVisina/$visina) * $sirina;
        $novaSlika = imagecreatetruecolor($novaSirina, $novaVisina);
        $postojecaSlika = null;
        switch ($imageFileType) {
          case 'jpeg':
          case 'jpg':
            $postojecaSlika = imagecreatefromjpeg($image_tmp);
            break;
          case 'png':
            $postojecaSlika = imagecreatefrompng($image_tmp);
            break;
          case 'gif':
            $postojecaSlika = imagecreatefromgif($image_tmp);
            break;
        }
        imagecopyresampled($novaSlika, $postojecaSlika, 0, 0, 0, 0, $novaSirina, $novaVisina, $sirina,
$visina);
        switch ($imageFileType) {
          case 'jpeg':
          case 'jpg':
            imagejpeg($novaSlika, $image_path, 75);
            break;
          case 'png':
```

```
          imagepng($novaSlika, $image_path);
          break;
        case 'gif':
          imagegif($novaSlika, $image_path);
          break;
      }

      $new_src = 'assets/images/recipes/food_'.$last_picture_id.".".$imageFileType;
      $change_pic = "UPDATE pictures SET src = :src WHERE ID = :id";
      $priprema_change = $conn -> prepare($change_pic);
      $priprema_change->bindParam(":src", $new_src);
      $priprema_change->bindParam(":id", $last_picture_id);
      $rezultat_change = $priprema_change->execute();
      if (!$rezultat_change) { $error = "Error changing name in database"; }

      $post_picture_query = "INSERT INTO post_pictures (ID_post, ID_picture) VALUES (:post_id,
:picture_id)";
      $priprema_pp = $conn -> prepare($post_picture_query);
      $priprema_pp->bindParam(":post_id", $post_id);
      $priprema_pp->bindParam(":picture_id", $last_picture_id);
      $rezultat_pp = $priprema_pp->execute();
      if (!$rezultat_pp) { $error = "Error adding to table"; }
      if ($error != "") {
        echo $error;
        $conn->rollback();
      } else {
        $conn->commit();
        header("Location: ../../index.php");
        die();
      }
    }catch (PDOException $e) {
      echo $e->getMessage();
      $conn->rollback();
    }
  } else {
    $error = "Must send by form!";
    echo $error;
  }
?>
```

3.44. models/admin/insert_post_tag.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $code = 500;
  $error = "";
  $post = $_POST["post"];
  $tag = $_POST["tag"];
  $query = "SELECT COUNT(ID_post) AS rowCount FROM post_tag WHERE ID_post = :post AND ID_tag =
:tag";
  $priprema = $conn -> prepare($query);
  $priprema->bindParam(":post", $post);
  $priprema->bindParam(":tag", $tag);
  $rezultat = $priprema->execute();
  $redovi = $priprema->FetchAll();
  $redovi = $redovi[0]->rowCount;
  if ($redovi == 1) { $error = "post_tag_exists"; }
  if ($error == "") {
    $query = "INSERT INTO post_tag (ID_post, ID_tag) VALUES (:post, :tag)";
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":post", $post);
    $priprema->bindParam(":tag", $tag);
    $rezultat = $priprema->execute();
    if ($rezultat) {
      $code = 200;
      echo json_encode("Successful");
    } else {
      $code = 503;
      echo json_encode("database_error");
    }
  } else {
    $code = 503;
    echo json_encode($error);
  }
  http_response_code($code);
?>
```

3.45. models/admin/insert_tag.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $code = 500;
  $error = "";
  $name = $_POST["name"];
  $srb_name = $_POST["srb_name"];
  $name_pat = "/^[A-Z][a-z]+$/";
  if ($name == "" || $srb_name == "") { $error = "empty_field"; }
  if (!preg_match($name_pat, $name) || !preg_match($name_pat, $srb_name)) { $error =
"tag_name_invalid"; }
  $query = "SELECT COUNT(name) AS rowCount FROM tags WHERE name = :name OR SRBname =
:srb_name";
  $priprema = $conn -> prepare($query);
  $priprema->bindParam(":name", $name);
  $priprema->bindParam(":srb_name", $srb_name);
```

```
    $rezultat = $priprema->execute();
    $redovi = $priprema->FetchAll();
    $redovi = $redovi[0]->rowCount;
    if ($redovi == 1) { $error = "tag_exists"; }
    if ($error == "") {
        $query = "INSERT INTO tags (name, SRBname) VALUES (:name, :srb_name)";
        $priprema = $conn -> prepare($query);
        $priprema->bindParam(":name", $name);
        $priprema->bindParam(":srb_name", $srb_name);
        $rezultat = $priprema->execute();
        if ($rezultat) {
            $code = 201;
            echo json_encode("Successful");
        } else {
            $code = 503;
            echo json_encode("database_error");
        }
    } else {
        $code = 503;
        echo json_encode($error);
    }
    http_response_code($code);
?>
```

3.46. models/admin/insert_user.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $code = 500;
    $error = "";
    $email = $_POST["email"];
    $name = $_POST["name"];
    $lastname = $_POST["lastname"];
    $password = $_POST["password"];
    $is_admin = $_POST["is_admin"];
    $rights_id = 2;
    iF ($is_admin == "true") { $rights_id = 1; }
    $email_pattern = "^[a-zA-Z0-9.!#$%&’*+/=?_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$^";
    $name_pat = "/^[A-Z][a-z]+$/";
    $lastname_pat = "/^([A-Z][a-z]+)+$/";
    if ($email == "" || $name == "" || $lastname == "" || $password == "") { $error = "empty_field"; }
    if (strlen($password) < 6) { $error = "password_short"; }
    if (!preg_match($email_pattern, $email)) { $error = "not_valid_email"; }
    if (!preg_match($name_pat, $name) || !preg_match($lastname_pat, $lastname)) { $error =
"name_invalid"; }
    $query = "SELECT COUNT(email) AS rowCount FROM users WHERE email = :email";
    $priprema = $conn -> prepare($query);
```

```
  $priprema->bindParam(":email", $email);
  $rezultat = $priprema->execute();
  $redovi = $priprema->FetchAll();
  $redovi = $redovi[0]->rowCount;
  if ($redovi == 1) { $error = "email_taken"; }
  if ($error == "") {
    $query = "INSERT INTO users (email, password, rights_id, name, last_name) VALUES (:email,
:password, :rights_id, :name, :lastname)";
    $password = md5($password);
    $priprema = $conn -> prepare($query);
    $priprema->bindParam(":email", $email);
    $priprema->bindParam(":password", $password);
    $priprema->bindParam(":name", $name);
    $priprema->bindParam(":lastname", $lastname);
    $priprema->bindParam(":rights_id", $rights_id);
    $rezultat = $priprema->execute();
    if ($rezultat) {
      $code = 201;
      echo json_encode("Successful");
    } else {
      $code = 503;
      echo json_encode("database_error");
    }
  } else {
    $code = 503;
    echo json_encode($error);
  }
  http_response_code($code);
?>
```

3.47. models/admin/newsletter.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $query = "SELECT email, ID FROM newsletter_subscribers";
  $subscribers = executeQuery($query);
  echo json_encode($subscribers);
?>
```

3.48. models/admin/pictures.php

```
<?php
  header("Content-Type: application/json");
  include("../../config/connection.php");
  $query = "SELECT src, ID, alt, SRBalt FROM pictures";
  $pictures = executeQuery($query);
  echo json_encode($pictures);
?>
```

3.49. models/admin/post_tag.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $query = "SELECT p.title AS postTitle, p.SRBtitle AS postSRBTitle, p.ID AS postID, t.name AS tagName,
t.SRBname AS tagSRBName, t.ID AS tagID FROM post_tag pt INNER JOIN post p ON pt.ID_post = p.ID
INNER JOIN tags t ON pt.ID_tag = t.ID";
   $posttags = executeQuery($query);
   echo json_encode($posttags);
?>
```

3.50. models/admin/posts.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $query = "SELECT title, SRBtitle, ID, text, SRBtext FROM post";
   $posts = executeQuery($query);
   echo json_encode($posts);
?>
```

3.51. models/admin/tags.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $query = "SELECT name, SRBname, ID FROM tags";
   $tags = executeQuery($query);
   echo json_encode($tags);
?>
```

3.52. models/admin/users.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $query = "SELECT ID, email, name, last_name FROM users WHERE rights_id != 1";
   $users = executeQuery($query);
   echo json_encode($users);
?>
```

3.53. assets/lang/eng.json

```
{
   "page_title": "25Hour Kitchen",
   "header_template": {
      "uk_flag": "UK Flag",
      "serbian_flag": "Serbian Flag"
   },
   "footer_template": {
      "navigation": "Navigation",
      "recent_posts": "Recent Posts",
      "tags": "Tags"
   },
   "main_template": {
      "title": "Welcome to 25Hour Kitchen",
      "baking_image": "Baking Cookies"
   },
   "food_posts": {
      "by": "BY: ",
      "comments": "Comments",
      "continue_reading": "Continue Reading",
      "no_posts": "Currently there are no posts by these parameters...",
      "search": "Search...",
      "how_it_started_title": "How it started?",
      "how_it_started_text": "From a simple blog idea, this website has developed into something much more, a place for your favourite and best recipes..",
      "subscribe": "Email",
      "subscribe_newsletter_title": "Subscribe to our newsletter",
      "subscribe_newsletter_text": "Subscribe and never miss a new recipe.",
      "quote_title": "Your Daily Quote:",
      "no_comment_show": "No comments to show...",
      "add_comment_title": "Add a new comment",
      "add_comment_button": "Comment",
      "empty_comment": "Minimum comment size is 5 characters",
      "log_to_comment": "Must login to comment",
      "other_error": "Unknown error happened",
      "database_error": "Database Error",
      "subscribe_empty_field": "Email must be entered",
      "not_valid_email": "Email not valid",
      "subscription_mail_exists": "You have already subscribed"
   },
   "notification": {
      "rss_title": "RSS copied",
      "rss_text": "RSS was copied to ClipBoard",
      "login_title": "Login Successful",
      "login_text": "Login Successful, window will now reload",
      "register_title": "Register Successful",
      "register_text": "Register Successful, you can now login",
      "logout_title": "Logout Successful",
      "logout_text": "Logout Successful, window will now reload",
      "comment_title": "Commented successfully",
      "comment_text": "Commented successfully, window will now reload",
      "insert_user_title": "User inserted successfully",
      "insert_user_text": "User inserted successfully, you can now login as the new user",
      "insert_tag_title": "Tag inserted successfully",
      "insert_tag_text": "Tag inserted successfully, you can now use it",
      "insert_post_tag_title": "Connection inserted succesfully",
      "insert_post_tag_text": "Your post now has the new tag",
      "update_user_title": "User edited successfully",
      "update_user_text": "User edited successfully, you can now login",
      "update_post_title": "Post edited successfully",
      "update_post_text": "Post edited successfully",
      "update_tag_title": "Tag edited successfully",
      "update_tag_text": "Tag edited successfully",
      "delete_user_title": "User deleted successfully",
      "delete_user_text": "User deleted successfully",
      "delete_post_title": "Post deleted successfully",
      "delete_post_text": "Post deleted successfully",
      "delete_tag_title": "Tag deleted successfully",
      "delete_tag_text": "Tag deleted successfully",
      "delete_picture_title": "Picture deleted successfully",
      "delete_picture_text": "Picture deleted successfully",
      "delete_comment_title": "Comment deleted successfully",
      "delete_comment_text": "Comment deleted successfully",
      "delete_newsletter_title": "Newsletter subscription deleted successfully",
      "delete_newsletter_text": "Newsletter subscription deleted successfully",
      "delete_post_tag_title": "Relationship between post and tag deleted",
      "delete_post_tag_text": "Relationship between post and tag deleted",
      "subscription_title": "Subscription successfully",
      "subscription_text": "You have successfully subscribed to our news letter",
      "contact_mail_sent_title": "Contact mail sent",
      "contact_mail_sent_text": "We will contact you as soon as possible",
      "word_saved_title": "Author text saved",
      "word_saved_text": "Author text is saved to your desktop",
      "excel_saved_title": "Posts saved",
      "excel_saved_text": "Posts saved to your desktop",
      "forgot_password_title": "Mail reset successful",
      "forgot_password_text": "Mail reset successfully sent to admins"
   },
   "login_template": {
      "register_email": "Email:",
      "register_email_placeholder": "someone@something",
      "register_name": "Name:",
      "register_name_placeholder": "Someone",
      "register_lastname": "Last Name:",
      "register_lastname_placeholder": "Someone",
      "register_password": "Password:",
      "register_password_repeat": "Repeat password:",
      "register": "Register",
      "login": "Login",
      "forgot_password": "Forgot password?",
      "not_valid_email": "Email not valid",
      "empty_field": "None of the fields should be left empty",
      "password_short": "Password to short (min 6 characters)",
      "wrong_params": "Wrong parameters, try again",
      "passwords_not_match": "Passwords do not match",
      "name_invalid": "Name and last name must be capitalized and contain only letters",
      "other_error": "Unknown error happened",
      "database_error": "Database Error",
      "email_taken": "Email you have entered is already taken",
      "empty_email": "Email field must not be empty"
   },
   "add_post_template": {
      "welcome": "Welcome ",
      "logout": "Logout",
      "title": "Add a recipe",
      "create_button": "Create",
      "add_post_title": "English Title",
      "add_post_srb_title": "Serbian Title",
      "add_post_text": "English Text",
      "add_post_srb_text": "Serbian Text",
      "add_post_type": "Types",
      "add_post_image": "Images",
      "add_post_tag": "Tags",
      "empty_field": "None of the fields should be left empty",
      "type_not_selected": "You must select a type",
      "title_invalid": "Titles must be capitalized and contain only letters",
      "tag_unchecked": "You must check at least one tag",
      "no_image_selected": "At least one image must be chosen"
   },
   "contact_template": {
      "title": "Contact",
      "send_mail": "someone@something",
      "send_contact": "Send",
      "author": "Author",
      "author_name": "Name:",
      "author_birth": "Born:",
      "author_mail": "Email:",
      "author_number": "Telephone:",
      "author_work": "Works as:",
      "download_author_data": "Download",
      "empty_field": "None of the fields should be left empty",
      "mail_invalid": "Email is in not valid format",
      "database_error": "Database Error",
      "other_error": "Unknown error happened"
   },
   "admin_template": {
      "title": "Admin Panel",
      "visiting_title": "Number of visitings by page in the last 24h",
      "visiting_main": "Number of main page views: ",
      "visiting_appetizer": "Number of appetizer views: ",
      "visiting_main_course": "Number of main course views: ",
      "visiting_dessert": "Number of dessert views: ",
      "visiting_contact": "Number of contact page views: ",
      "visiting_admin": "Number of arrivals on admin page: ",
      "visiting_404": "Number of wrong pages: ",
      "other_controls_title": "Other Facts:",
      "number_logged_users": "Number of Logged In active users: ",
      "export_to_excel": "Download posts as Excel",
      "get_documentation": "Get Documentation Here",
      "admin_options": {
        "admin_insert_user": "Insert User",
        "admin_insert_tag": "Insert Tag",
        "admin_insert_picture": "Insert Picture",
        "admin_insert_post_tag": "Insert Post-Tag",
        "admin_edit_user": "Edit User",
        "admin_edit_post": "Edit Post",
        "admin_edit_tag": "Edit Tag",
        "admin_delete_user": "Delete User",
        "admin_delete_post": "Delete Post",
        "admin_delete_tag": "Delete Tag",
        "admin_delete_picture": "Delete Picture",
        "admin_delete_comment": "Delete Comment",
        "admin_delete_newsletter": "Delete Newsletter Subscription",
        "admin_delete_post_tag": "Delete Post-Tag"
      },
      "admin_controls": {
        "admin_insert_user_name": "Name",
        "admin_insert_user_lastname": "Last Name",
        "admin_insert_user_mail": "Email",
        "admin_insert_user_password": "Password",
        "admin_insert_user_admin": "Is Admin?",
        "admin_insert_user_submit": "Submit",
        "admin_insert_tag_name": "Tag Name",
        "admin_insert_tag_srbname": "Tag Name In Serbian",
        "admin_insert_tag_submit": "Submit",
        "admin_insert_picture_picture": "Picture",
        "admin_insert_picture_alt": "Description",
        "admin_insert_picture_srbalt": "Description In Serbian",
        "admin_insert_picture_options": "For Post",
        "admin_insert_picture_submit": "Submit",
        "admin_insert_post_tag_post_options": "Select Post",
        "admin_insert_post_tag_tag_options": "Select Tag",
        "admin_insert_post_tag_tag_submit": "Submit",
        "admin_edit_user_name": "Name",
        "admin_edit_user_lastname": "Last Name",
        "admin_edit_user_email": "Email",
        "admin_edit_user_submit": "Submit",
        "admin_edit_post_title": "Title",
        "admin_edit_post_srbtitle": "Title In Serbian",
        "admin_edit_post_text": "Text",
        "admin_edit_post_srbtext": "Text In Serbian",
        "admin_edit_post_submit": "Submit",
        "admin_edit_tag_name": "Title",
        "admin_edit_tag_srbname": "Title In Serbian",
        "admin_edit_tag_submit": "Submit",
        "admin_delete_user_submit": "Submit",
        "admin_delete_post_submit": "Submit",
        "admin_delete_tag_submit": "Submit",
        "admin_delete_picture_submit": "Submit",
        "admin_delete_comment_submit": "Submit",
        "admin_delete_newsletter_submit": "Submit",
        "admin_delete_post_tag_submit": "Submit"
      },
      "admin_errors": {
        "empty_field": "None of the fields should be left empty",
        "password_short": "Password to short (min 6 characters)",
        "wrong_params": "Wrong parameters, try again",
        "name_invalid": "Name and last name must be capitalized and contain only letters",
        "other_error": "Unknown error happened",
        "database_error": "Database Error",
        "email_taken": "Email you have entered is already taken",
        "tag_name_invalid": "Tag name must be capitalized and contain only letters",
        "tag_exists": "Tag name already exists",
        "image_name_invalid": "Image name must be capitalized and contain only letters",
        "no_image_selected": "Image must be selected",
        "post_tag_exists": "This post already has this tag",
        "title_name_invalid": "Title name must be capitalized and contain only letters"
      }
    }
  }
```

3.54. assets/lang/srb.json

```
{
  "page_title": "25Sati Kuhinja",
  "header_template": {
    "uk_flag": "Zastava Ujedinjenog Kraljevstva",
    "serbian_flag": "Srpska Zastava"
  },
  "footer_template": {
    "navigation": "Navigacija",
    "recent_posts": "Najnovije Objave",
    "tags": "Tagovi"
  },
  "main_template": {
    "title": "Dobrodošli na 25Sati Kuhinja",
    "baking_image": "Pečenje kolača"
  },
  "food_posts": {
    "by": "OD: ",
    "comments": "Komentara",
    "continue_reading": "Nastavite čitati",
    "no_posts": "Trenutno nema objava po ovim parametrima...",
    "search": "Pretraga...",
    "how_it_started_title": "Kako je sve počelo?",
    "how_it_started_text": "Od jedne jednostavne ideje o blogu, ovaj web sajt je izrastao u nešto
mnogo više, mesto za Vaše omiljene i najbolje recepte..",
    "subscribe": "Email",
    "subscribe_newsletter_title": "Pretplatite se na naš bilten",
    "subscribe_newsletter_text": "Pretplatite se i nikad ne propustite nov recept.",
    "quote_title": "Vaš Dnevni Citat:",
    "no_comment_show": "Nema komentara za prikaz...",
    "add_comment_title": "Dodajte komentar",
    "add_comment_button": "Komentarišite",
    "empty_comment": "Minimalna dužina komentara je 5 karaktera",
    "log_to_comment": "Morate biti prijavljeni za komentarisanje",
    "other_error": "Desila se nepoznata greška",
    "database_error": "Greška sa bazom podataka",
    "subscribe_empty_field": "Mora se uneti email",
    "not_valid_email": "Email nije ispravan",
    "subscription_mail_exists": "Već ste se prijavili"
  },
  "notification": {
    "rss_title": "RSS kopiran",
    "rss_text": "RSS je kopiran na Vašu privremenu memoriju",
    "login_title": "Prijavljivanje uspešno",
    "login_text": "Uspešno ste se prijavili, prozor će se osvežiti",
    "register_title": "Upisivanje uspešno",
    "register_text": "Upisivanje uspešno, sada se možete prijaviti",
    "logout_title": "Odjava uspešna",
    "logout_text": "Uspešno ste se odjavili, prozor će se osvežiti",
    "comment_title": "Komentarisanje uspešno",
    "comment_text": "Uspešno ste komentarisali, prozor će se osvežiti",
    "insert_user_title": "Korisnik napravljen uspešno",
    "insert_user_text": "Korisnik napravljen uspešno, sada se možete prijaviti kao nov korisnik",
    "insert_tag_title": "Tag napravljen uspešno",
    "insert_tag_text": "Tag napravljen uspešno, sada ga možete koristiti",
    "insert_post_tag_title": "Veza uspešno dodata",
    "insert_post_tag_text": "Vaša objava sadrži nov tag",
    "update_user_title": "Korisnik promenjen uspešno",
    "update_user_text": "Korisnik promenjen uspešno, sada se možete prijaviti",
    "update_post_title": "Objava promenjena uspešno",
    "update_post_text": "Objava promenjena uspešno",
    "update_tag_title": "Tag promenjen uspešno",
    "update_tag_text": "Tag promenjen uspešno",
    "delete_user_title": "Korisnik obrisan uspešno",
    "delete_user_text": "Korisnik obrisan uspešno",
    "delete_post_title": "Objava obrisana uspešno",
    "delete_post_text": "Objava obrisana uspešno",
    "delete_tag_title": "Tag obrisan uspešno",
    "delete_tag_text": "Tag obrisan uspešno",
    "delete_picture_title": "Slika obrisana uspešno",
    "delete_picture_text": "Slika obrisana uspešno",
    "delete_comment_title": "Komentar obrisan uspešno",
    "delete_comment_text": "Komentar obrisan uspešno",
    "delete_newsletter_title": "Pretplata na bilten obrisana uspešno",
    "delete_newsletter_text": "Pretplata na bilten obrisana uspešno",
    "delete_post_tag_title": "Veza između objave i taga obrisana",
    "delete_post_tag_text": "Veza između objave i taga obrisana",
    "subscription_title": "Pretplata uspešna",
    "subscription_text": "Uspešno ste se prijavili na naš bilten",
    "contact_mail_sent_title": "Kontakt poruka poslata",
    "contact_mail_sent_text": "Kontaktiraćemo vaš u najkraćem mogućem roku",
    "word_saved_title": "Tekst o autoru sačuvan",
    "word_saved_text": "Tekst o autoru je sačuvan na Vašem desktop-u",
    "excel_saved_title": "Objave sačuvane",
    "excel_saved_text": "Objave sačuvane na Vašem desktop-u",
    "forgot_password_title": "Resetovanje mail-a uspešno",
    "forgot_password_text": "Uspešno ste kontaktirali admine radi resetovanja email-a"
  },
  "login_template": {
    "register_email": "Email:",
    "register_email_placeholder": "neko@nesto",
    "register_name": "Ime:",
    "register_name_placeholder": "Neko",
    "register_lastname": "Prezime:",
    "register_lastname_placeholder": "Neko",
    "register_password": "Šifra:",
    "register_password_repeat": "Ponovite šifru:",
    "register": "Upisivanje",
    "login": "Prijaviti se",
    "forgot_password": "Zaboravio šifru?",
    "not_valid_email": "Email nije u ispravnom formatu",
    "empty_field": "Nijedno od polja ne sme biti prazno",
    "password_short": "Šifra je previše kratka (min 6 karaktera)",
    "wrong_params": "Pogrešni parametri, pobajte ponovo",
    "passwords_not_match": "Šifre se ne podudaraju",
    "name_invalid": "Ime i prezime moraju počinjati velikim slovom i da sadrže samo slova",
    "other_error": "Desila se nepoznata greška",
    "database_error": "Greška sa bazom podataka",
    "email_taken": "Email koji ste uneli je zauzet",
    "empty_email": "Polje za email ne sme biti prazno"
  },
  "add_post_template": {
    "welcome": "Dobrodošli ",
    "logout": "Odjavite se",
    "title": "Dodajte recept",
    "create_button": "Kreirajte",
    "add_post_title": "Engleski Naslov",
    "add_post_srb_title": "Srpski Naslov",
    "add_post_text": "Engleski Tekst",
    "add_post_srb_text": "Srpski Tekst",
    "add_post_type": "Tipovi",
    "add_post_image": "Slike",
    "add_post_tag": "Tagovi",
    "empty_field": "Nijedno od polja ne sme biti prazno",
    "type_not_selected": "Morate odabrati tip",
    "title_invalid": "Naslovi moraju počinjati velikim slovom i da sadrže samo slova",
    "tag_unchecked": "Barem jedan tag mora biti odabran",
    "no_image_selected": "Mora se odabrati barem jedna slika"
  },
  "contact_template": {
    "title": "Kontakt",
    "send_mail": "neko@nesto",
    "send_contact": "Pošaljite",
    "author": "Autor",
    "author_name": "Ime:",
    "author_birth": "Rođen:",
    "author_mail": "Email:",
    "author_number": "Telefon:",
    "author_work": "Radi kao:",
    "download_author_data": "Preuzmite",
    "empty_field": "Nijedno od polja ne sme biti prazno",
    "mail_invalid": "Email nije u ispravnom formatu",
    "database_error": "Greška sa bazom podataka",
    "other_error": "Desila se nepoznata greška"
  },
  "admin_template": {
    "title": "Admin Panel",
    "visiting_title": "Broj pregleda po strani u poslednjih 24h",
    "visiting_main": "Broj pregleda početne stranice: ",
    "visiting_appetizer": "Broj pregleda predjela: ",
    "visiting_main_course": "Broj pregleda glavnih jela: ",
    "visiting_dessert": "Broj pregleda dezerata: ",
    "visiting_contact": "Broj pregleda kontakt stranice: ",
    "visiting_admin": "Broj dolazaka na admin stranicu: ",
    "visiting_404": "Broj pogrešnih stranica: ",
    "other_controls_title": "Ostali podaci:",
    "number_logged_users": "Broj prijavljenih aktivnih korisnika: ",
    "export_to_excel": "Preuzmite objave kao Excel",
    "get_documentation": "Dohvatite Dokumentaciju Ovde",
    "admin_options": {
      "admin_insert_user": "Dodajte Korisnika",
      "admin_insert_tag": "Dodajte Tag",
      "admin_insert_picture": "Dodajte Sliku",
      "admin_insert_post_tag": "Dodajte Objavu-Tag",
      "admin_edit_user": "Promenite Korisnika",
      "admin_edit_post": "Promenite Objavu",
      "admin_edit_tag": "Promenite Tag",
      "admin_delete_user": "Obrišite Korisnika",
      "admin_delete_post": "Obrišite Objavu",
      "admin_delete_tag": "Obrišite Tag",
      "admin_delete_picture": "Obrišite Sliku",
      "admin_delete_comment": "Obrišite Komentar",
      "admin_delete_newsletter": "Obrišite Pretplatu",
      "admin_delete_post_tag": "Obrišite Objavu-Tag"
    },
    "admin_controls": {
      "admin_insert_user_name": "Ime",
      "admin_insert_user_lastname": "Prezime",
      "admin_insert_user_mail": "Email",
      "admin_insert_user_password": "Šifra",
      "admin_insert_user_admin": "Da li je admin?",
      "admin_insert_user_submit": "Pošaljite",
      "admin_insert_tag_name": "Ime Taga",
      "admin_insert_tag_srbname": "Ime Taga Na Srpskom",
      "admin_insert_tag_submit": "Pošaljite",
      "admin_insert_picture_picture": "Slika",
      "admin_insert_picture_alt": "Opis",
      "admin_insert_picture_srbalt": "Opis Na Srpskom",
      "admin_insert_picture_options": "Za Objavu",
      "admin_insert_picture_submit": "Pošaljite",
      "admin_insert_post_tag_post_options": "Izaberite Objavu",
      "admin_insert_post_tag_tag_options": "Izaberite Tag",
      "admin_insert_post_tag_tag_submit": "Pošaljite",
      "admin_edit_user_name": "Ime",
      "admin_edit_user_lastname": "Prezime",
      "admin_edit_user_email": "Email",
      "admin_edit_user_submit": "Pošaljite",
      "admin_edit_post_title": "Naslov",
      "admin_edit_post_srbtitle": "Naslov Na Srpskom",
      "admin_edit_post_text": "Tekst",
      "admin_edit_post_srbtext": "Tekst Na Srpskom",
      "admin_edit_post_submit": "Pošaljite",
      "admin_edit_tag_name": "Naslov",
      "admin_edit_tag_srbname": "Naslov Na Srpskom",
      "admin_edit_tag_submit": "Pošaljite",
      "admin_delete_user_submit": "Pošaljite",
      "admin_delete_post_submit": "Pošaljite",
      "admin_delete_tag_submit": "Pošaljite",
            "admin_delete_picture_submit": "Pošaljite",
            "admin_delete_comment_submit": "Pošaljite",
            "admin_delete_newsletter_submit": "Pošaljite",
            "admin_delete_post_tag_submit": "Pošaljite"
        },
        "admin_errors": {
            "not_valid_email": "Email nije u ispravnom formatu",
            "empty_field": "Nijedno od polja ne sme biti prazno",
            "password_short": "Šifra je previše kratka (min 6 karaktera)",
            "wrong_params": "Pogrešni parametri, pobajte ponovo",
            "name_invalid": "Ime i prezime moraju počinjati velikim slovom i da sadrže samo slova",
            "other_error": "Desila se nepoznata greška",
            "database_error": "Greška sa bazom podataka",
            "email_taken": "Email koji ste uneli je zauzet",
            "tag_name_invalid": "Ime taga mora počinjati velikim slovom i da sadrži samo slova",
            "tag_exists": "Ime taga već postoji",
            "image_name_invalid": "Ime slike mora počinjati velikim slovom i da sadrži samo slova",
            "no_image_selected": "Mora se odabrati slika",
            "post_tag_exists": "Ova objava već sadrži ovaj tag",
            "title_name_invalid": "Ime naslova mora počinjati velikim slovom i da sadrži samo slova"
        }
    }
}
```

3.55. assets/js/add_post_template.js

```
function start_add_post() {
    $(".welcome_user span:first-child").html(language_content.add_post_template.welcome);
    $(".welcome_user button").html(language_content.add_post_template.logout);
    $(".add_post_form h2").html(language_content.add_post_template.title);
    $(".add_post_form #add_post_button").val(language_content.add_post_template.create_button);

    $(".add_post_param
label[for='add_post_title']").html(language_content.add_post_template.add_post_title);
    $(".add_post_param #add_post_title").attr("placeholder",
language_content.add_post_template.add_post_title);

    $(".add_post_param
label[for='add_post_srb_title']").html(language_content.add_post_template.add_post_srb_title);
    $(".add_post_param #add_post_srb_title").attr("placeholder",
language_content.add_post_template.add_post_srb_title);

    $(".add_post_param
label[for='add_post_text']").html(language_content.add_post_template.add_post_text);
    $(".add_post_param
label[for='add_post_srb_text']").html(language_content.add_post_template.add_post_srb_text);
```

```
  $(".add_post_param
label[for='add_post_type']").html(language_content.add_post_template.add_post_type);
  $(".add_post_param
label[for='add_post_image']").html(language_content.add_post_template.add_post_image);
  $(".add_post_param
label[for='add_post_tag']").html(language_content.add_post_template.add_post_tag);

  post_get_types();
  post_get_tags();
}
function post_get_types() {
  $.ajax({
    url: "models/types.php",
    method: "GET",
    dataType: "json",
    success: function (data) {
      var text = "";
      data.forEach(function (t) {
        text += `<div><input type="radio" name="add_post_type" value="${t.ID}"/>
${lang=="eng"?t.text:(lang=="srb"?t.SRBtext:"Error")}</div>`;
      });
      $(".add_post_param .add_post_type_holder").html(text);
    },
    error: function (err) {
      write_error_to_file(err);
      $(".add_post_param .add_post_type_holder").html(err.responseText);
    }
  });
}
function post_get_tags() {
  $.ajax({
    url: "models/tags.php",
    method: "GET",
    dataType: "json",
    success: function (data) {
      var text = "";
      data.forEach(function (t) {
        text += `<div><input type="checkbox" name="add_post_tag[]" value="${t.ID}"/>
${lang=="eng"?t.name:(lang=="srb"?t.SRBname:"Error")}</div>`;
      });
      $(".add_post_param .add_post_tag_holder").html(text);
    },
    error: function (err) {
      write_error_to_file(err);
      $(".add_post_param .add_post_tag_holder").html(err.responseText);
    }
  });
}
```

```
function add_post_param_check() {
   var eng_title = document.getElementById("add_post_title").value,
      srb_title = document.getElementById("add_post_srb_title").value,
      eng_text = document.getElementById("add_post_text").value,
      srb_text = document.getElementById("add_post_srb_text").value,
      title_pattern = /^[A-ZŠĐŽČĆ][a-z0-9ščćđž]+$/,
      type_checked = $(".add_post_type_holder input:checked").val(),
      tags_checked = document.getElementsByName("add_post_tag[]"),
      has_checked = false,
      image_selected = document.getElementsByName("add_post_image[]").length;

   if (eng_title == "" || srb_title == "" || eng_text == "" || srb_text == "") {
      document.getElementById("add_post_error").innerHTML =
language_content.add_post_template.empty_field;
      return false;
   }
   if (!type_checked) {
      document.getElementById("add_post_error").innerHTML =
language_content.add_post_template.type_not_selected;
      return false;
   }
   if (!title_pattern.test(eng_title) || !title_pattern.test(srb_title)) {
      document.getElementById("add_post_error").innerHTML =
language_content.add_post_template.title_invalid;
      return false;
   }
   tags_checked.forEach(function (t) {
      if (t.checked) { has_checked = true; }
   });
   if (!has_checked) {
      document.getElementById("add_post_error").innerHTML =
language_content.add_post_template.tag_unchecked;
      return false;
   }
   if (image_selected <= 0) {
      document.getElementById("add_post_error").innerHTML =
language_content.add_post_template.no_image_selected;
      return false;
   }
   document.getElementById("add_post_error").innerHTML = "";
   return true;
}
```

3.56. assets/js/admin_template.js

```
function start_admin() {
   if (!$("#admin_panel:visible").length) { setTimeout(start_admin, 50); } else { load_admin_data(); }
}
```

```
function load_admin_data() {
   $("#admin_panel h2").html(language_content.admin_template.title);
   $(".admin_controls.file_controls h3").html(language_content.admin_template.visiting_title);
   $(".admin_controls .visiting_main span:first-child").html(language_content.admin_template.visiting_main);
   $(".admin_controls .visiting_appetizer span:first-child").html(language_content.admin_template.visiting_appetizer);
   $(".admin_controls .visiting_main_course span:first-child").html(language_content.admin_template.visiting_main_course);
   $(".admin_controls .visiting_dessert span:first-child").html(language_content.admin_template.visiting_dessert);
   $(".admin_controls .visiting_contact span:first-child").html(language_content.admin_template.visiting_contact);
   $(".admin_controls .visiting_admin span:first-child").html(language_content.admin_template.visiting_admin);
   $(".admin_controls .visiting_404 span:first-child").html(language_content.admin_template.visiting_404);

   $(".admin_controls.other_controls h3").html(language_content.admin_template.other_controls_title);
   $(".admin_controls.other_controls #export_to_excel").html(language_content.admin_template.export_to_excel);
   $(".admin_controls.other_controls a").html(language_content.admin_template.get_documentation);
   $(".admin_controls .number_logged_users span:first-child").html(language_content.admin_template.number_logged_users);

   var text = "",
      el,
      first_option = "";
   for (el in language_content.admin_template.admin_options) {
      if (!first_option) { first_option = el; }
      text += `<option value="${el}">${language_content.admin_template.admin_options[el]}</option>`;
   }
   $("#admin_panel #admin_option").html(text);
   $('#admin_panel #admin_option').on('change', function() {
      fill_admin_controls($(this).val());
   });
   $("#export_to_excel").on("click", export_to_excel);

   fill_admin_controls(first_option);
   get_page_access_info();
   get_logged_users();
}
function fill_admin_controls(option) {
   switch(option) {
      case "admin_insert_user":
         $("#admin_extra_option").css("display", "none");
         admin_insert_user_show();
         break;
      case "admin_insert_tag":
         $("#admin_extra_option").css("display", "none");
         admin_insert_tag_show();
         break;
      case "admin_insert_picture":
         $("#admin_extra_option").css("display", "none");
         admin_insert_picture_show();
         break;
      case "admin_insert_post_tag":
         $("#admin_extra_option").css("display", "none");
         admin_insert_post_tag_show();
         break;
      case "admin_edit_user":
         $("#admin_extra_option").css("display", "block");
         admin_edit_user_show();
         break;
      case "admin_edit_post":
         $("#admin_extra_option").css("display", "block");
         admin_edit_post_show();
         break;
      case "admin_edit_tag":
         $("#admin_extra_option").css("display", "block");
         admin_edit_tag_show();
         break;
      case "admin_delete_user":
         $("#admin_extra_option").css("display", "block");
         admin_delete_user_show();
         break;
      case "admin_delete_post":
         $("#admin_extra_option").css("display", "block");
         admin_delete_post_show();
         break;
      case "admin_delete_tag":
         $("#admin_extra_option").css("display", "block");
         admin_delete_tag_show();
         break;
      case "admin_delete_picture":
         $("#admin_extra_option").css("display", "block");
         admin_delete_picture_show();
         break;
      case "admin_delete_comment":
         $("#admin_extra_option").css("display", "block");
         admin_delete_comment_show();
         break;
      case "admin_delete_newsletter":
         $("#admin_extra_option").css("display", "block");
         admin_delete_newsletter_show();
```

```
            break;
        case "admin_delete_post_tag":
            $("#admin_extra_option").css("display", "block");
            admin_delete_post_tag_show();
            break;
    }
}
function admin_insert_user_show() {
    var text = `<div class="admin_option_holder"><label
for="admin_insert_user_name">${language_content.admin_template.admin_controls.admin_insert_us
er_name}</label><input type="text" name="admin_insert_user_name" id="admin_insert_user_name"
placeholder="${language_content.admin_template.admin_controls.admin_insert_user_name}"/></div>
`;
    text += `<div class="admin_option_holder"><label
for="admin_insert_user_lastname">${language_content.admin_template.admin_controls.admin_insert
_user_lastname}</label><input type="text" name="admin_insert_user_lastname"
id="admin_insert_user_lastname"
placeholder="${language_content.admin_template.admin_controls.admin_insert_user_lastname}"/></
div>`;
    text += `<div class="admin_option_holder"><label
for="admin_insert_user_mail">${language_content.admin_template.admin_controls.admin_insert_user
_mail}</label><input type="email" name="admin_insert_user_mail" id="admin_insert_user_mail"
placeholder="${language_content.admin_template.admin_controls.admin_insert_user_mail}"/></div>`;
    text += `<div class="admin_option_holder"><label
for="admin_insert_user_password">${language_content.admin_template.admin_controls.admin_insert
_user_password}</label><input type="password" name="admin_insert_user_password"
id="admin_insert_user_password" placeholder="* * * * * *"/></div>`;
    text += `<div class="admin_option_holder"><label
for="admin_insert_user_admin">${language_content.admin_template.admin_controls.admin_insert_us
er_admin}</label><input type="checkbox" id="admin_insert_user_admin"
name="admin_insert_user_admin"/></div>`;
    text += `<p id="admin_insert_user_error"></p>`;
    text += `<button
id="admin_insert_user_submit">${language_content.admin_template.admin_controls.admin_insert_us
er_submit}</button>`;
    $("#admin_extra_space").html(text);
    document.getElementById("admin_insert_user_name").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_insert_user_send(); } });
    document.getElementById("admin_insert_user_lastname").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_insert_user_send(); } });
    document.getElementById("admin_insert_user_mail").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_insert_user_send(); } });
    document.getElementById("admin_insert_user_password").addEventListener("keyup", function (e) {
if (e.key == "Enter") { admin_insert_user_send(); } });
    $("#admin_insert_user_submit").on("click", admin_insert_user_send);
}
function admin_insert_user_send() {
    var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
        name_pat = /^[A-Z][a-z]+$/,
        lastname_pat = /^([A-Z][a-z]+)+$/,
        email = document.getElementById("admin_insert_user_mail").value,
        name = document.getElementById("admin_insert_user_name").value,
        lastname = document.getElementById("admin_insert_user_lastname").value,
        password = document.getElementById("admin_insert_user_password").value,
        is_admin = document.getElementById("admin_insert_user_admin").checked;
    if (email == "" || name == "" || lastname == "" || password == "") {
        document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
        return;
    }
    if (password.length < 6) {
        document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.password_short;
        return;
    }
    if (!email_pat.test(email)) {
        document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.not_valid_email;
        return;
    }
    if (!name_pat.test(name) || !lastname_pat.test(lastname)) {
        document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.name_invalid;
        return;
    }
    document.getElementById("admin_insert_user_error").innerHTML = "";
    $.ajax({
        url: "models/admin/insert_user.php",
        method: "POST",
        dataType: "json",
        data: {
            email: email,
            name: name,
            lastname: lastname,
            password: password,
            is_admin: is_admin
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("insert_user");
                document.getElementById("admin_insert_user_mail").value = "";
                document.getElementById("admin_insert_user_name").value = "";
                document.getElementById("admin_insert_user_lastname").value = "";
                document.getElementById("admin_insert_user_password").value = "";
                document.getElementById("admin_insert_user_admin").checked = false;
            }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
          break;
        case "empty_field":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
          break;
        case "name_invalid":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.name_invalid;
          break;
        case "email_taken":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.email_taken;
          break;
        case "not_valid_email":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.not_valid_email;
          break;
        case "password_short":
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.password_short;
          break;
        default:
          document.getElementById("admin_insert_user_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
      }
    }
  });
}

function admin_insert_tag_show() {
  var text = `<div class="admin_option_holder"><label
for="admin_insert_tag_name">${language_content.admin_template.admin_controls.admin_insert_tag
_name}</label><input type="text" name="admin_insert_tag_name" id="admin_insert_tag_name"
placeholder="${language_content.admin_template.admin_controls.admin_insert_tag_name}"/></div>`;
  text += `<div class="admin_option_holder"><label
for="admin_insert_tag_srbname">${language_content.admin_template.admin_controls.admin_insert_t
ag_srbname}</label><input type="text" name="admin_insert_tag_srbname"
id="admin_insert_tag_srbname"
placeholder="${language_content.admin_template.admin_controls.admin_insert_tag_srbname}"/></di
v>`;
  text += `<p id="admin_insert_tag_error"></p>`;
```

```
    text += `<button id="admin_insert_tag_submit">${language_content.admin_template.admin_controls.admin_insert_tag_submit}</button>`;
    $("#admin_extra_space").html(text);
    document.getElementById("admin_insert_tag_name").addEventListener("keyup", function (e) { if (e.key == "Enter") { admin_insert_tag_send(); } });
    document.getElementById("admin_insert_tag_srbname").addEventListener("keyup", function (e) { if (e.key == "Enter") { admin_insert_tag_send(); } });
    $("#admin_insert_tag_submit").on("click", admin_insert_tag_send);
}
function admin_insert_tag_send() {
    var name_pat = /^[A-Z][a-z]+$/,
        name = document.getElementById("admin_insert_tag_name").value,
        srb_name = document.getElementById("admin_insert_tag_srbname").value;
    if (name == "" || srb_name == "") {
        document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.empty_field;
        return;
    }
    if (!name_pat.test(name) || !name_pat.test(srb_name)) {
        document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.tag_name_invalid;
        return;
    }
    document.getElementById("admin_insert_tag_error").innerHTML = "";
    $.ajax({
        url: "models/admin/insert_tag.php",
        method: "POST",
        dataType: "json",
        data: {
            name: name,
            srb_name: srb_name
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("insert_tag");
                document.getElementById("admin_insert_tag_name").value = "";
                document.getElementById("admin_insert_tag_srbname").value = "";
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.database_error;
                    break;
                case "empty_field":
                    document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.empty_field;
                    break;
                case "tag_name_invalid":
                    document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.tag_name_invalid;
                    break;
                case "tag_exists":
                    document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.tag_exists;
                    break;
                default:
                    document.getElementById("admin_insert_tag_error").innerHTML = language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_insert_picture_show() {
    $.ajax({
        url: "models/admin/posts.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = `<form onsubmit="return admin_insert_picture_check()" method="POST" action="models/admin/insert_picture.php" name="admin_insert_picture_form" enctype="multipart/form-data"><div class="admin_option_holder"><label for="admin_insert_picture_picture">${language_content.admin_template.admin_controls.admin_insert_picture_picture}</label><input type="file" name="admin_insert_picture_picture" id="admin_insert_picture_picture"/></div>`;
            text += `<div class="admin_option_holder"><label for="admin_insert_picture_alt">${language_content.admin_template.admin_controls.admin_insert_picture_alt}</label><input type="text" name="admin_insert_picture_alt" id="admin_insert_picture_alt" placeholder="${language_content.admin_template.admin_controls.admin_insert_picture_alt}"/></div>`;
            text += `<div class="admin_option_holder"><label for="admin_insert_picture_srbalt">${language_content.admin_template.admin_controls.admin_insert_picture_srbalt}</label><input type="text" name="admin_insert_picture_srbalt" id="admin_insert_picture_srbalt" placeholder="${language_content.admin_template.admin_controls.admin_insert_picture_srbalt}"/></div>`;
            text += `<div class="admin_option_holder"><label for="admin_insert_picture_options">${language_content.admin_template.admin_controls.admin_insert_picture_options}</label><select name="admin_insert_picture_options" id="admin_insert_picture_options">`;
            data.forEach(function (p) {
                text += `<option value="${p.ID}">${lang=="eng"?p.title:(lang=="srb"?p.SRBtitle:"Error")}</option>`;
            });
            text += `</select></div>`;
            text += `<p id="admin_insert_picture_error"></p>`;
            text += `<button id="admin_insert_picture_submit" name="admin_insert_picture_submit"
type="submit">${language_content.admin_template.admin_controls.admin_insert_picture_submit}</b
utton>`;
            text += "</form>";
            $("#admin_extra_space").html(text);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_insert_picture_check() {
    var name_pat = /^[A-Z][a-z]+$/,
        alt = document.getElementById("admin_insert_picture_alt").value,
        srb_alt = document.getElementById("admin_insert_picture_srbalt").value,
        image_selected = document.getElementsByName("admin_insert_picture_picture").length;

    if (alt == "" || srb_alt == "") {
        document.getElementById("admin_insert_picture_error").innerHTML =
language_content.admin_template.admin_controls.empty_field;
        return false;
    }
    if (!name_pat.test(alt) || !name_pat.test(srb_alt)) {
        document.getElementById("admin_insert_picture_error").innerHTML =
language_content.admin_template.admin_controls.image_name_invalid;
        return false;
    }
    if (image_selected <= 0) {
        document.getElementById("admin_insert_picture_error").innerHTML =
language_content.admin_template.admin_controls.no_image_selected;
        return false;
    }
    document.getElementById("admin_insert_picture_error").innerHTML = "";
    return true;
}
function admin_insert_post_tag_show() {
    var posts,
        tags;
    $.ajax({
        url: "models/admin/posts.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            posts = data;
            $.ajax({
                url: "models/admin/tags.php",
                method: "GET",
                dataType: "json",
                success: function (data) {
                    tags = data;
                    var text = `<div class="admin_option_holder"><label for="admin_insert_post_tag_post_options">${language_content.admin_template.admin_controls.admin_insert_post_tag_post_options}</label><select name="admin_insert_post_tag_post_options" id="admin_insert_post_tag_post_options">`;
                    posts.forEach(function (p) {
                        text += `<option value="${p.ID}">${lang=="eng"?p.title:(lang=="srb"?p.SRBtitle:"Error")}</option>`;
                    });
                    text += `</select></div>`;
                    text += `<div class="admin_option_holder"><label for="admin_insert_post_tag_tag_options">${language_content.admin_template.admin_controls.admin_insert_post_tag_tag_options}</label><select name="admin_insert_post_tag_tag_options" id="admin_insert_post_tag_tag_options">`;
                    tags.forEach(function (t) {
                        text += `<option value="${t.ID}">${lang=="eng"?t.name:(lang=="srb"?t.SRBname:"Error")}</option>`;
                    });
                    text += `</select></div>`;
                    text += `<p id="admin_insert_post_tag_tag_error"></p>`;
                    text += `<button id="admin_insert_post_tag_tag_submit">${language_content.admin_template.admin_controls.admin_insert_post_tag_tag_submit}</button>`;
                    $("#admin_extra_space").html(text);
                    $("#admin_insert_post_tag_tag_submit").on("click", admin_insert_post_tag_send);
                },
                error: function (err) {
                    write_error_to_file(err);
                    $("#admin_extra_space").html(err.responseText);
                }
            });
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_insert_post_tag_send() {
    var post = document.getElementById("admin_insert_post_tag_post_options").options[document.getElementById("admin_insert_post_tag_post_options").selectedIndex].value,
        tag =
document.getElementById("admin_insert_post_tag_tag_options").options[document.getElementById("
admin_insert_post_tag_tag_options").selectedIndex].value;

    document.getElementById("admin_insert_post_tag_tag_error").innerHTML = "";
    $.ajax({
        url: "models/admin/insert_post_tag.php",
        method: "POST",
        dataType: "json",
        data: {
            post: post,
            tag: tag
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("insert_post_tag");
            }
        },
        error: function (err) {
            write_error_to_file(err);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_insert_post_tag_tag_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                case "post_tag_exists":
                    document.getElementById("admin_insert_post_tag_tag_error").innerHTML =
language_content.admin_template.admin_errors.post_tag_exists;
                    break;
                default:
                    document.getElementById("admin_insert_post_tag_tag_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_edit_user_show() {
    var users;
    $.ajax({
        url: "models/admin/users.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            users = data;
            var text = "";
            users.forEach(function (u) {
                text += `<option value="${u.ID}">${u.name} ${u.last_name}</option>`;
            });
            $("#admin_extra_option").html(text);
            admin_edit_user_show_second(users, users[0].ID);
            $("#admin_extra_option").on('change', function() {
                admin_edit_user_show_second(users, $(this).val());
            });
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_edit_user_show_second(users, user_id) {
    users.forEach(function (u) {
        if (u.ID != user_id) { return; }
        var text = `<div class="admin_option_holder"><label
for="admin_edit_user_name">${language_content.admin_template.admin_controls.admin_edit_user_
name}</label><input type="text" name="admin_edit_user_name" id="admin_edit_user_name"
placeholder="${language_content.admin_template.admin_controls.admin_edit_user_name}"
value="${u.name}"/></div>`;
        text += `<div class="admin_option_holder"><label
for="admin_edit_user_lastname">${language_content.admin_template.admin_controls.admin_edit_us
er_lastname}</label><input type="text" name="admin_edit_user_lastname"
id="admin_edit_user_lastname"
placeholder="${language_content.admin_template.admin_controls.admin_edit_user_lastname}"
value="${u.last_name}"/></div>`;
        text += `<div class="admin_option_holder"><label
for="admin_edit_user_email">${language_content.admin_template.admin_controls.admin_edit_user_
email}</label><input type="email" name="admin_edit_user_email" id="admin_edit_user_email"
placeholder="${language_content.admin_template.admin_controls.admin_edit_user_email}"
value="${u.email}"/></div>`;
        text += `<input type="hidden" name="admin_edit_user_id" id="admin_edit_user_id"
value="${u.ID}"/></div>`;
        text += `<p id="admin_edit_user_error"></p>`;
        text += `<button
id="admin_edit_user_submit">${language_content.admin_template.admin_controls.admin_edit_user_
submit}</button>`;
        $("#admin_extra_space").html(text);
    });
    document.getElementById("admin_edit_user_name").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_user_send(); } });
    document.getElementById("admin_edit_user_lastname").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_user_send(); } });
    document.getElementById("admin_edit_user_email").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_user_send(); } });
    $("#admin_edit_user_submit").on("click", admin_edit_user_send);
}
function admin_edit_user_send() {
    var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
        name_pat = /^[A-Z][a-z]+$/,
        lastname_pat = /^([A-Z][a-z]+)+$/,
        email = document.getElementById("admin_edit_user_email").value,
        name = document.getElementById("admin_edit_user_name").value,
        lastname = document.getElementById("admin_edit_user_lastname").value,
        id = document.getElementById("admin_edit_user_id").value;
    if (email == "" || name == "" || lastname == "") {
        document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
        return;
    }
    if (!email_pat.test(email)) {
        document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.not_valid_email;
        return;
    }
    if (!name_pat.test(name) || !lastname_pat.test(lastname)) {
        document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.name_invalid;
        return;
    }
    document.getElementById("admin_edit_user_error").innerHTML = "";
    $.ajax({
        url: "models/admin/edit_user.php",
        method: "POST",
        dataType: "json",
        data: {
            email: email,
            name: name,
            lastname: lastname,
            id: id
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("update_user");
                document.getElementById("admin_edit_user_email").value = "";
                document.getElementById("admin_edit_user_name").value = "";
                document.getElementById("admin_edit_user_lastname").value = "";
                fill_admin_controls("admin_edit_user");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
```

```
                    break;
                case "empty_field":
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
                    break;
                case "name_invalid":
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.name_invalid;
                    break;
                case "email_taken":
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.email_taken;
                    break;
                case "not_valid_email":
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.not_valid_email;
                    break;
                default:
                    document.getElementById("admin_edit_user_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_edit_post_show() {
    var posts;
    $.ajax({
        url: "models/admin/posts.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            posts = data;
            var text = "";
            posts.forEach(function (p) {
                text += `<option
value="${p.ID}">${lang=="eng"?p.title:(lang=="srb"?p.SRBtitle:"Error")}</option>`;
            });
            $("#admin_extra_option").html(text);
            admin_edit_post_show_second(posts, posts[0].ID);
            $("#admin_extra_option").on('change', function() {
                admin_edit_post_show_second(posts, $(this).val());
            });
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
```

```
}
function admin_edit_post_show_second(posts, post_id) {
    posts.forEach(function (p) {
        if (p.ID != post_id) { return; }
        var text = `<div class="admin_option_holder"><label
for="admin_edit_post_title">${language_content.admin_template.admin_controls.admin_edit_post_tit
le}</label><input type="text" name="admin_edit_post_title" id="admin_edit_post_title"
placeholder="${language_content.admin_template.admin_controls.admin_edit_post_title}"
value="${p.title}"/></div>`;
        text += `<div class="admin_option_holder"><label
for="admin_edit_post_srbtitle">${language_content.admin_template.admin_controls.admin_edit_post
_srbtitle}</label><input type="text" name="admin_edit_post_srbtitle" id="admin_edit_post_srbtitle"
placeholder="${language_content.admin_template.admin_controls.admin_edit_post_srbtitle}"
value="${p.SRBtitle}"/></div>`;
        text += `<div class="admin_option_holder"><label
for="admin_edit_post_text">${language_content.admin_template.admin_controls.admin_edit_post_te
xt}</label><textarea rows="8" cols="40" name="admin_edit_post_text"
id="admin_edit_post_text">${p.text}</textarea></div>`;
        text += `<div class="admin_option_holder"><label
for="admin_edit_post_srbtext">${language_content.admin_template.admin_controls.admin_edit_post
_srbtext}</label><textarea rows="8" cols="40" name="admin_edit_post_srbtext"
id="admin_edit_post_srbtext">${p.SRBtext}</textarea></div>`;
        text += `<input type="hidden" name="admin_edit_post_id" id="admin_edit_post_id"
value="${p.ID}"/></div>`;
        text += `<p id="admin_edit_post_error"></p>`;
        text += `<button
id="admin_edit_post_submit">${language_content.admin_template.admin_controls.admin_edit_post_
submit}</button>`;
        $("#admin_extra_space").html(text);
    });
    document.getElementById("admin_edit_post_title").addEventListener("keyup", function (e) { if (e.key
== "Enter") { admin_edit_post_send(); } });
    document.getElementById("admin_edit_post_srbtitle").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_post_send(); } });
    document.getElementById("admin_edit_post_text").addEventListener("keyup", function (e) { if (e.key
== "Enter") { admin_edit_post_send(); } });
    document.getElementById("admin_edit_post_srbtext").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_post_send(); } });
    $("#admin_edit_post_submit").on("click", admin_edit_post_send);
}
function admin_edit_post_send() {
    var title_pat = /^([A-Z][a-z]+)+$/,
        title = document.getElementById("admin_edit_post_title").value,
        srb_title = document.getElementById("admin_edit_post_srbtitle").value,
        text = document.getElementById("admin_edit_post_text").value,
        srb_text = document.getElementById("admin_edit_post_srbtext").value,
        id = document.getElementById("admin_edit_post_id").value;
    if (title == "" || srb_title == "" || text == "" || srb_text == "") {
        document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
        return;
    }
    if (!title_pat.test(title) || !title_pat.test(srb_title)) {
        document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.title_name_invalid;
        return;
    }
    document.getElementById("admin_edit_post_error").innerHTML = "";
    $.ajax({
        url: "models/admin/edit_post.php",
        method: "POST",
        dataType: "json",
        data: {
            title: title,
            srb_title: srb_title,
            text: text,
            srb_text: srb_text,
            id: id
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("update_post");
                document.getElementById("admin_edit_post_title").value = "";
                document.getElementById("admin_edit_post_srbtitle").value = "";
                document.getElementById("admin_edit_post_text").value = "";
                document.getElementById("admin_edit_post_srbtext").value = "";
                fill_admin_controls("admin_edit_post");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                case "empty_field":
                    document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
                    break;
                case "title_name_invalid":
                    document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.title_name_invalid;
                    break;
                default:
```

```
                document.getElementById("admin_edit_post_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
          }
      }
   });
}
function admin_edit_tag_show() {
   var tags;
   $.ajax({
      url: "models/admin/tags.php",
      method: "GET",
      dataType: "json",
      success: function (data) {
         tags = data;
         var text = "";
         tags.forEach(function (t) {
            text += `<option
value="${t.ID}">${lang=="eng"?t.name:(lang=="srb"?t.SRBname:"Error")}</option>`;
         });
         $("#admin_extra_option").html(text);
         admin_edit_tag_show_second(tags, tags[0].ID);
         $("#admin_extra_option").on('change', function() {
            admin_edit_tag_show_second(tags, $(this).val());
         });
      },
      error: function (err) {
         write_error_to_file(err);
         $("#admin_extra_space").html(err.responseText);
      }
   });
}
function admin_edit_tag_show_second(tags, tag_id) {
   tags.forEach(function (t) {
      if (t.ID != tag_id) { return; }
      var text = `<div class="admin_option_holder"><label
for="admin_edit_tag_name">${language_content.admin_template.admin_controls.admin_edit_tag_na
me}</label><input type="text" name="admin_edit_tag_name" id="admin_edit_tag_name"
placeholder="${language_content.admin_template.admin_controls.admin_edit_tag_name}"
value="${t.name}"/></div>`;
      text += `<div class="admin_option_holder"><label
for="admin_edit_tag_srbname">${language_content.admin_template.admin_controls.admin_edit_tag_
srbname}</label><input type="text" name="admin_edit_tag_srbname" id="admin_edit_tag_srbname"
placeholder="${language_content.admin_template.admin_controls.admin_edit_tag_srbname}"
value="${t.SRBname}"/></div>`;
      text += `<input type="hidden" name="admin_edit_tag_id" id="admin_edit_tag_id"
value="${t.ID}"/></div>`;
      text += `<p id="admin_edit_tag_error"></p>`;
```

```
    text += `<button
id="admin_edit_tag_submit">${language_content.admin_template.admin_controls.admin_edit_tag_su
bmit}</button>`;
    $("#admin_extra_space").html(text);
  });
  document.getElementById("admin_edit_tag_name").addEventListener("keyup", function (e) { if (e.key
== "Enter") { admin_edit_tag_send(); } });
  document.getElementById("admin_edit_tag_srbname").addEventListener("keyup", function (e) { if
(e.key == "Enter") { admin_edit_tag_send(); } });
  $("#admin_edit_tag_submit").on("click", admin_edit_tag_send);
}
function admin_edit_tag_send() {
  var name_pat = /^[A-Z][a-z]+$/,
    name = document.getElementById("admin_edit_tag_name").value,
    srb_name = document.getElementById("admin_edit_tag_srbname").value,
    id = document.getElementById("admin_edit_tag_id").value;
  if (name == "" || srb_name == "") {
    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
    return;
  }
  if (!name_pat.test(name) || !name_pat.test(srb_name)) {
    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.tag_name_invalid;
    return;
  }
  document.getElementById("admin_edit_tag_error").innerHTML = "";
  $.ajax({
    url: "models/admin/edit_tag.php",
    method: "POST",
    dataType: "json",
    data: {
      name: name,
      srb_name: srb_name,
      id: id
    },
    success: function (data) {
      if (data == "Successful") {
        show_notification("update_tag");
        document.getElementById("admin_edit_tag_name").value = "";
        document.getElementById("admin_edit_tag_srbname").value = "";
        fill_admin_controls("admin_edit_tag");
      }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
```

```
                    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                case "empty_field":
                    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.empty_field;
                    break;
                case "tag_name_invalid":
                    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.tag_name_invalid;
                    break;
                case "tag_exists":
                    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.tag_exists;
                    break;
                default:
                    document.getElementById("admin_edit_tag_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_user_show() {
    $.ajax({
        url: "models/admin/users.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (u) {
                text += `<option value="${u.ID}">${u.name} ${u.last_name}</option>`;
            });
            $("#admin_extra_option").html(text);
            var button_text = `<p id="admin_delete_user_error"></p>`;
            button_text += `<button
id="admin_delete_user_submit">${language_content.admin_template.admin_controls.admin_delete_u
ser_submit}</button>`;
            $("#admin_extra_space").html(button_text);
            $("#admin_delete_user_submit").on("click", admin_delete_user_send);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_user_send() {
    var user = document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_option").selectedIndex].value;
    document.getElementById("admin_delete_user_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_user.php",
        method: "POST",
        dataType: "json",
        data: {
            user: user
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_user");
                fill_admin_controls("admin_delete_user");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_user_error").innerHTML = language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_user_error").innerHTML = language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_post_show() {
    $.ajax({
        url: "models/admin/posts.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (p) {
                text += `<option value="${p.ID}">${lang=="eng"?p.title:(lang=="srb"?p.SRBtitle:"Error")}</option>`;
            });
            $("#admin_extra_option").html(text);
            var button_text = `<p id="admin_delete_post_error"></p>`;
            button_text += `<button id="admin_delete_post_submit">${language_content.admin_template.admin_controls.admin_delete_post_submit}</button>`;
            $("#admin_extra_space").html(button_text);
            $("#admin_delete_post_submit").on("click", admin_delete_post_send);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_post_send() {
    var post = 
document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_op
tion").selectedIndex].value;
    document.getElementById("admin_delete_post_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_post.php",
        method: "POST",
        dataType: "json",
        data: {
            post: post
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_post");
                fill_admin_controls("admin_delete_post");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_post_error").innerHTML = 
language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_post_error").innerHTML = 
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_tag_show() {
    $.ajax({
        url: "models/admin/tags.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (t) {
```

```
            text += `<option
value="${t.ID}">${lang=="eng"?t.name:(lang=="srb"?t.SRBname:"Error")}</option>`;
         });
         $("#admin_extra_option").html(text);
         var button_text = `<p id="admin_delete_tag_error"></p>`;
         button_text += `<button
id="admin_delete_tag_submit">${language_content.admin_template.admin_controls.admin_delete_ta
g_submit}</button>`;
         $("#admin_extra_space").html(button_text);
         $("#admin_delete_tag_submit").on("click", admin_delete_tag_send);
      },
      error: function (err) {
         write_error_to_file(err);
         $("#admin_extra_space").html(err.responseText);
      }
   });
}
function admin_delete_tag_send() {
   var tag =
document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_op
tion").selectedIndex].value;
   document.getElementById("admin_delete_tag_error").innerHTML = "";
   $.ajax({
      url: "models/admin/delete_tag.php",
      method: "POST",
      dataType: "json",
      data: {
         tag: tag
      },
      success: function (data) {
         if (data == "Successful") {
            show_notification("delete_tag");
            fill_admin_controls("admin_delete_tag");
         }
      },
      error: function (err) {
         write_error_to_file(err.responseJSON);
         switch(err.responseJSON) {
            case "database_error":
               document.getElementById("admin_delete_tag_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
               break;
            default:
               document.getElementById("admin_delete_tag_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
         }
      }
   });
```

```
}
function admin_delete_picture_show() {
    var comments;
    $.ajax({
        url: "models/admin/pictures.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            pictures = data;
            var text = "";
            pictures.forEach(function (p) {
                text += `<option value="${p.ID}">${lang=="eng"?p.alt:(lang=="srb"?p.SRBalt:"Error")}</option>`;
            });
            $("#admin_extra_option").html(text);
            admin_delete_picture_show_second(pictures, pictures[0].ID);
            $("#admin_extra_option").on('change', function() {
                admin_delete_picture_show_second(pictures, $(this).val());
            });
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_picture_show_second(pictures, picture_id) {
    pictures.forEach(function (p) {
        if (p.ID != picture_id) { return; }
        var text = `<img src="${p.src}" alt="${lang=="eng"?p.alt:(lang=="srb"?p.SRBalt:"Error")}"/>`;
        text += `<p id="admin_delete_picture_error"></p>`;
        text += `<button id="admin_delete_picture_submit">${language_content.admin_template.admin_controls.admin_delete_picture_submit}</button>`;
        $("#admin_extra_space").html(text);
    });
    $("#admin_delete_picture_submit").on("click", admin_delete_picture_send);
}
function admin_delete_picture_send() {
    var picture = document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_option").selectedIndex].value;
    document.getElementById("admin_delete_picture_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_picture.php",
        method: "POST",
        dataType: "json",
        data: {
            picture: picture
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_picture");
                fill_admin_controls("admin_delete_picture");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_picture_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_picture_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_comment_show() {
    var comments;
    $.ajax({
        url: "models/admin/comments.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            comments = data;
            var text = "";
            comments.forEach(function (c) {
                text += `<option value="${c.ID}">${c.text}</option>`;
            });
            $("#admin_extra_option").html(text);
            admin_delete_comment_show_second(comments, comments[0].ID);
            $("#admin_extra_option").on('change', function() {
                admin_delete_comment_show_second(comments, $(this).val());
            });
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_comment_show_second(comments, comment_id) {
    comments.forEach(function (c) {
        if (c.ID != comment_id) { return; }
        var text = `<span>${c.text}</span>`;
        text += `<p id="admin_delete_comment_error"></p>`;
        text += `<button
id="admin_delete_comment_submit">${language_content.admin_template.admin_controls.admin_del
ete_comment_submit}</button>`;
        $("#admin_extra_space").html(text);
    });
    $("#admin_delete_comment_submit").on("click", admin_delete_comment_send);
}
function admin_delete_comment_send() {
    var comment =
document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_op
tion").selectedIndex].value;
    document.getElementById("admin_delete_comment_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_comment.php",
        method: "POST",
        dataType: "json",
        data: {
            comment: comment
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_comment");
                fill_admin_controls("admin_delete_comment");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_comment_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_comment_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_newsletter_show() {
    $.ajax({
        url: "models/admin/newsletter.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (s) {
                text += `<option value="${s.ID}">${s.email}</option>`;
            });
            $("#admin_extra_option").html(text);
            var button_text = `<p id="admin_delete_newsletter_error"></p>`;
            button_text += `<button id="admin_delete_newsletter_submit">${language_content.admin_template.admin_controls.admin_delete_newsletter_submit}</button>`;
            $("#admin_extra_space").html(button_text);
            $("#admin_delete_newsletter_submit").on("click", admin_delete_newsletter_send);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_newsletter_send() {
    var newsletter = document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_option").selectedIndex].value;
    document.getElementById("admin_delete_newsletter_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_newsletter.php",
        method: "POST",
        dataType: "json",
        data: {
            newsletter: newsletter
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_newsletter");
                fill_admin_controls("admin_delete_newsletter");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_newsletter_error").innerHTML = language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_newsletter_error").innerHTML = language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function admin_delete_post_tag_show() {
    $.ajax({
        url: "models/admin/post_tag.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (pt) {
                text += `<option value="${pt.postID}-
${pt.tagID}">${lang=="eng"?pt.postTitle:(lang=="srb"?pt.postSRBTitle:"Error")} -
${lang=="eng"?pt.tagName:(lang=="srb"?pt.tagSRBName:"Error")}</option>`;
            });
            $("#admin_extra_option").html(text);
            var button_text = `<p id="admin_delete_post_tag_error"></p>`;
            button_text += `<button
id="admin_delete_post_tag_submit">${language_content.admin_template.admin_controls.admin_dele
te_post_tag_submit}</button>`;
            $("#admin_extra_space").html(button_text);
            $("#admin_delete_post_tag_submit").on("click", admin_delete_post_tag_send);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#admin_extra_space").html(err.responseText);
        }
    });
}
function admin_delete_post_tag_send() {
    var post_tag =
document.getElementById("admin_extra_option").options[document.getElementById("admin_extra_op
tion").selectedIndex].value;
    document.getElementById("admin_delete_post_tag_error").innerHTML = "";
    $.ajax({
        url: "models/admin/delete_post_tag.php",
        method: "POST",
        dataType: "json",
        data: {
            post_tag: post_tag
        },
        success: function (data) {
            if (data == "Successful") {
                show_notification("delete_post_tag");
                fill_admin_controls("admin_delete_post_tag");
            }
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("admin_delete_post_tag_error").innerHTML =
language_content.admin_template.admin_errors.database_error;
                    break;
                default:
                    document.getElementById("admin_delete_post_tag_error").innerHTML =
language_content.admin_template.admin_errors.other_error;
            }
        }
    });
}
function get_page_access_info() {
    $.ajax({
        url: "models/page_access_info.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var sum = data.admin_template + data.appetizer + data.contact_template + data.dessert +
data.main_course + data.main_template + data.page_404;
            $(".admin_controls .visiting_main span:last-child").html((data.main_template / sum *
100).toFixed(2) + "%");
            $(".admin_controls .visiting_appetizer span:last-child").html((data.appetizer / sum *
100).toFixed(2) + "%");
            $(".admin_controls .visiting_main_course span:last-child").html((data.main_course / sum *
100).toFixed(2) + "%");
            $(".admin_controls .visiting_dessert span:last-child").html((data.dessert / sum * 100).toFixed(2) +
"%");
            $(".admin_controls .visiting_contact span:last-child").html((data.contact_template / sum *
100).toFixed(2) + "%");
            $(".admin_controls .visiting_admin span:last-child").html((data.admin_template / sum *
100).toFixed(2) + "%");
            $(".admin_controls .visiting_404 span:last-child").html((data.page_404 / sum * 100).toFixed(2) +
"%");
        },
        error: function (err) {
            write_error_to_file(err);
            $(".admin_controls
.admin_file_holder").html(language_content.admin_template.admin_errors.other_error);
        }
    });
}
function get_logged_users() {
    $.ajax({
        url: "models/logged_users.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            $(".admin_controls .number_logged_users span:last-child").html(data[0].logged_users);
        },
        error: function (err) {
            write_error_to_file(err);
            $(".admin_controls
.admin_file_holder").html(language_content.admin_template.admin_errors.other_error);
        }
    });
}
function export_to_excel() {
    $.ajax({
        url: "models/export_excel.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            show_notification("excel_saved");
        },
        error: function (err) {
            if (err.status == 200) {
                show_notification("excel_saved");
            } else {
                alert(err.responseText);
            }
        }
    });
}
```

3.57. assets/js/contact_template.js

```
function start_contact() {
    setTimeout(start_second_contact, 50);
}
function start_second_contact() {
    if (!$("#contact_template:visible").length) { setTimeout(start_second_contact, 50); } else {
load_contact(); }
}
function load_contact() {
    $("#contact_template .contact_area h2").html(language_content.contact_template.title);
    $("#contact_template .contact_area
button").html(language_content.contact_template.send_contact);
    $("#contact_template .contact_area #message_mail").attr("placeholder",
language_content.contact_template.send_mail);

    $("#contact_template .author h2").html(language_content.contact_template.author);
    $("#contact_template .author #message_mail").attr("alt",
language_content.contact_template.author);
```

```
  $("#contact_template .author .author_name p:first-
child").html(language_content.contact_template.author_name);
  $("#contact_template .author .author_birth p:first-
child").html(language_content.contact_template.author_birth);
  $("#contact_template .author .author_mail p:first-
child").html(language_content.contact_template.author_mail);
  $("#contact_template .author .author_number p:first-
child").html(language_content.contact_template.author_number);
  $("#contact_template .author .author_work p:first-
child").html(language_content.contact_template.author_work);
  $("#contact_template .author
button").html(language_content.contact_template.download_author_data);

  document.getElementById("message_mail").addEventListener("keyup", function (e) { if (e.key ==
"Enter") { mail_send_work(); } });
  $("#sent_contact").on("click", mail_send_work);
  $("#download_author_data").on("click", download_author_data);
}
function mail_send_work() {
  var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
    mail = document.getElementById("message_mail").value,
    text = document.getElementById("send_message").value;
  if (mail == "" || text == "") {
    document.getElementById("message_error").innerHTML =
language_content.contact_template.empty_field;
    return;
  }
  if (!email_pat.test(mail)) {
    document.getElementById("message_error").innerHTML =
language_content.contact_template.mail_invalid;
    return;
  }
  document.getElementById("message_error").innerHTML = "";
  $.ajax({
    url: "models/contact.php",
    method: "POST",
    dataType: "json",
    data: {
      mail: mail,
      text: text
    },
    success: function (data) {
      if (data == "Successful") {
        show_notification("contact_mail_sent");
        document.getElementById("message_mail").value = "";
        document.getElementById("send_message").value = "";
      }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
          document.getElementById("message_error").innerHTML =
language_content.contact_template.database_error;
          break;
        case "empty_field":
          document.getElementById("message_error").innerHTML =
language_content.contact_template.empty_field;
          break;
        case "mail_invalid":
          document.getElementById("message_error").innerHTML =
language_content.contact_template.mail_invalid;
          break;
        default:
          document.getElementById("message_error").innerHTML =
language_content.contact_template.other_error;
      }
    }
  });
}
function download_author_data() {
  $.ajax({
    url: "models/save_word.php",
    method: "POST",
    dataType: "json",
    data: {
      author_name: $("#contact_template .author .author_name p:first-child").html(),
      author_name_val: $("#contact_template .author .author_name p:last-child").html(),
      author_birth: $("#contact_template .author .author_birth p:first-child").html(),
      author_birth_val: $("#contact_template .author .author_birth p:last-child").html(),
      author_mail: $("#contact_template .author .author_mail p:first-child").html(),
      author_mail_val: $("#contact_template .author .author_mail p:last-child").html(),
      author_number: $("#contact_template .author .author_number p:first-child").html(),
      author_number_val: $("#contact_template .author .author_number p:last-child").html(),
      author_work: $("#contact_template .author .author_work p:first-child").html(),
      author_work_val: $("#contact_template .author .author_work p:last-child").html()
    },
    success: function (data) {
      show_notification("word_saved");
    },
    error: function (err) {
      if (err.status == 200) {
        show_notification("word_saved");
      } else {
        alert(err.responseText);
      }
```

```
        }
    });
}
```

3.58. assets/js/food_post_template.js

```
function start_food_posts() {
    if (!$("#main_panel_load_food:visible").length) { setTimeout(start_food_posts, 50); } else {
load_food_posts(); }
}
function load_food_posts() {
    var sel_tag = selected_tags,
        type_id = food_types.indexOf(selected_food_type) + 2;
    if (sel_tag.length == 0) { sel_tag = [""]; }
    $.ajax({
        url: "models/posts.php",
        method: "GET",
        dataType: "json",
        data: {
            type_id: type_id,
            selected_tags: sel_tag
        },
        success: function (data) {
            post_to_array(data);
        },
        error: function (err) {
            write_error_to_file(err);
            post_to_array(err.responseText);
        }
    });
}
function post_to_array(data, post_id) {
    posts = [];
    data.forEach(function (longP) {
        var has = false;
        posts.forEach(function (post) {
            if (post.id == longP.id) { has = true; }
        });
        if (has) {
            posts.forEach(function (post) {
                if (post.id != longP.id) { return; }
                var hasimage = false,
                    hastag = false;
                post.images.forEach(function (pi) {
                    if (pi.picid == longP.picid) { hasimage = true; }
                });
                post.tags.forEach(function (pt) {
                    if (pt.tagid == longP.tagid) { hastag = true; }
                });
                if (!hasimage) {
                    var image = {
                        picsrc: longP.picsrc,
                        picSRBalt: longP.picSRBalt,
                        picalt: longP.picalt,
                        picid: longP.picid
                    };
                    post.images.push(image);
                }
                if (!hastag) {
                    var tag = {
                        tagid: longP.tagid,
                        tagSRBname: longP.tagSRBname,
                        tagname: longP.tagname
                    };
                    post.tags.push(tag);
                }
            });
        } else {
            var el = {
                SRBtext: longP.SRBtext,
                SRBtitle: longP.SRBtitle,
                id: longP.id,
                images: [{
                    picsrc: longP.picsrc,
                    picSRBalt: longP.picSRBalt,
                    picalt: longP.picalt,
                    picid: longP.picid
                }],
                last_name: longP.last_name,
                name: longP.name,
                postDate: longP.postDate,
                postText: longP.postText,
                tags: [{
                    tagid: longP.tagid,
                    tagSRBname: longP.tagSRBname,
                    tagname: longP.tagname
                }],
                title: longP.title,
                commentNumber: longP.commentNumber
            };
            posts.push(el);
        }
    });
    fill_mid_posts(posts, post_id);
}
```

```
function fill_mid_posts(posts, post_id) {
    var text = "";
    posts.forEach(function (post) {
        text += `<div class="post_header"><ul class="post_tags">`;
        post.tags.forEach(function (t) {
            text += `<li><span data-
id="${t.tagid}">${lang=="eng"?t.tagname:(lang=="srb"?t.tagSRBname:"Error")}</span></li>`;
        })
        text += "</ul>";
        text += `<h2>${lang=="eng"?post.title:(lang=="srb"?post.SRBtitle:"Error")}</h2>`;
        text += `<div class="post_image"><button data-way="back" data-image="${post.images[0].picid}"
data-post="${post.id}"><span class="fas fa-arrow-left"></span></button><img
src="${post.images[0].picsrc}"
alt="${lang=="eng"?post.images[0].picalt:(lang=="srb"?post.images[0].picSRBalt:"Error")}" data-
id="${post.images[0].picid}"/><button data-way="forward" data-image="${post.images[0].picid}" data-
post="${post.id}"><span class="fas fa-arrow-right"></span></button></div>`;
        text += `<ul class="post_details"><li><span class="fa fa-calendar" aria-hidden="true"></span>
${post.postDate}</li><li><span class="far fa-user"></span> ${language_content.food_posts.by}
${post.name} ${post.last_name}</li><li><span class="fas fa-comment"></span>
${post.commentNumber} ${language_content.food_posts.comments}</li></ul>`;
        text += "</div>";
        if (post_id) {
            text += `<p
class="bigger_letter">${lang=="eng"?post.postText:(lang=="srb"?post.SRBtext:"Error")}</p>`;
        } else {
            text += `<p class="bigger_letter">${lang=="eng"?(post.postText.substring(0,146) +
"..."):(lang=="srb"?(post.SRBtext.substring(0,146) + "..."):"Error")}</p>`;
            text += `<div class="post_button"><button data-
id="${post.id}">${language_content.food_posts.continue_reading}</button></div>`;
        }
    });
    if (posts.length == 0) {
        text += `<h2>${language_content.food_posts.no_posts}</h2>`;
    }
    if (food_types.indexOf(page_content) < 0) { return; }
    document.getElementById("main_panel_load_food").innerHTML = text;
    if (post_id) { get_post_comments(post_id); }
    $(".post_header .post_image button").on("click", function(e) {
        var div_tag = $(this).parent(),
            way = $(this).attr("data-way"),
            post_id = $(this).attr("data-post"),
            image_id = $(this).attr("data-image"),
            image_tag = $(this).parent().children('img');
        e.preventDefault();
        posts.forEach(function(p) {
            if(p.id != post_id) { return; }
            for (var i = 0; i < p.images.length; i++) {
                if (p.images[i].picid == image_id) {
                    if (way == "forward") {
                        var next_num = i + 1;
                        if (next_num == p.images.length) { next_num = 0; }
                        image_tag.attr("alt",
lang=="eng"?p.images[next_num].picalt:(lang=="srb"?pp.images[next_num].picSRBalt:"Error"));
                        image_tag.attr("src", p.images[next_num].picsrc);
                        div_tag.children("button").eq(0).attr("data-image", p.images[next_num].picid);
                        div_tag.children("button").eq(1).attr("data-image", p.images[next_num].picid);
                    }
                    if (way == "back") {
                        var prev_num = i - 1;
                        if (prev_num < 0) { prev_num = p.images.length - 1; }
                        image_tag.attr("alt",
lang=="eng"?p.images[prev_num].picalt:(lang=="srb"?pp.images[prev_num].picSRBalt:"Error"));
                        image_tag.attr("src", p.images[prev_num].picsrc);
                        div_tag.children("button").eq(0).attr("data-image", p.images[prev_num].picid);
                        div_tag.children("button").eq(1).attr("data-image", p.images[prev_num].picid);
                    }
                }
            }
        });
    });
    $(".post_button button").on("click", function() {
        get_single_post($(this).attr("data-id"));
    });
    finished_loading_posts();
}
function finished_loading_posts() {
    search_work();
    how_it_started_work();
    subscribe_newsletter_work();
    random_quote_work();
    rss_click();
    login_work();
    add_post_work();
    $("#logout").on("click", function() {
        $.ajax({
            url: "models/logout.php",
            method: "GET",
            dataType: "json",
            success: function (data) {
                show_notification("logout", true);
            },
            error: function (err) {
                write_error_to_file(err);
                window.location.reload();
            }
        });
```

```
  });
}
function search_work() {
  $("#search").attr("placeholder", language_content.food_posts.search);
  document.getElementById("search").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { search_posts(); }
  });
  $(".search_field button").on("click", search_posts);
}
function search_posts() {
  var sel_tag = selected_tags,
    text = document.getElementById("search").value,
    type_id = food_types.indexOf(selected_food_type) + 2;
  if (sel_tag.length == 0) { sel_tag = [""]; }
  $.ajax({
    url: "models/posts.php",
    method: "GET",
    dataType: "json",
    data: {
      type_id: type_id,
      text: text,
      selected_tags: sel_tag
    },
    success: function (data) {
      post_to_array(data);
    },
    error: function (err) {
      write_error_to_file(err);
      post_to_array(err.responseText);
    }
  });
}
function how_it_started_work() {
  $("#main_panel_load_edit .how_it_started
h2").html(language_content.food_posts.how_it_started_title);
  $("#main_panel_load_edit .how_it_started
p").html(language_content.food_posts.how_it_started_text);
}
function subscribe_newsletter_work() {
  $("#subscribe").attr("placeholder", language_content.food_posts.subscribe);
  $("#main_panel_load_edit .subscribe_newsletter
h2").html(language_content.food_posts.subscribe_newsletter_title);
  $("#main_panel_load_edit .subscribe_newsletter
#subscribe_text").html(language_content.food_posts.subscribe_newsletter_text);
  document.getElementById("subscribe").addEventListener("keyup", function (e) { if (e.key == "Enter") {
subscribe_send(); } });
}
function subscribe_send() {
  var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
    subscription_mail = document.getElementById("subscribe").value;
  if (subscription_mail == "") {
    document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.subscribe_empty_field;
    return;
  }
  if (!email_pat.test(subscription_mail)) {
    document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.not_valid_email;
    return;
  }
  document.getElementById("subscribe_error").innerHTML = "";
  $.ajax({
    url: "models/subscribe.php",
    method: "POST",
    dataType: "json",
    data: {
      subscription_mail: subscription_mail
    },
    success: function (data) {
      if (data == "Successful") {
        show_notification("subscription");
        document.getElementById("subscribe").value = "";
      }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
          document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.database_error;
          break;
        case "subscribe_empty_field":
          document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.subscribe_empty_field;
          break;
        case "subscription_mail_exists":
          document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.subscription_mail_exists;
          break;
        case "not_valid_email":
          document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.not_valid_email;
          break;
        default:
          document.getElementById("subscribe_error").innerHTML =
language_content.food_posts.other_error;
```

```
            }
        }
    });
}
function random_quote_work() {
    $("#main_panel_load_edit .quote_field h2").html(language_content.food_posts.quote_title);
    $.ajax({
        url: "models/random_quote.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "<span>\"</span>";
            text += lang=="eng"?data.text:(lang=="srb"?data.SRBtext:"Error");
            text += "<span>\"</span>";
            $("#main_panel_load_edit .quote_field p").html(text);
        },
        error: function (err) {
            write_error_to_file(err);
            $("#main_panel_load_edit .quote_field p").html(err.responseText);
        }
    });
}
function get_single_post(data) {
    var post_id = data;
    $.ajax({
        url: "models/single_post.php",
        method: "POST",
        dataType: "json",
        data: {
            postId: data
        },
        success: function (data) {
            post_to_array(data, post_id);
        },
        error: function (err) {
            write_error_to_file(err);
            post_to_array(err.responseText);
        }
    });
}
function get_post_comments(post_id) {
    $.ajax({
        url: "models/comments.php",
        method: "GET",
        dataType: "json",
        data: {
            postId: post_id
        },
        success: function (data) {
            document.getElementById("main_panel_load_food").innerHTML += `<h2
class="comment_content">${language_content.food_posts.comments}:</h2>`;
            if (data.length) {
                data.forEach(function(c) {
                    document.getElementById("main_panel_load_food").innerHTML += `<div
class="comment_holder"><p>${c.userName}
${c.userLastName}</p><p>${c.commentText}</p><p>${c.commentDate}</p></div><div
id="add_comment"></div>`;
                });
            } else {
                document.getElementById("main_panel_load_food").innerHTML +=
`<h3>${language_content.food_posts.no_comment_show}</h3><div id="add_comment"></div>`;
            }
            add_comment_view(post_id);
        },
        error: function (err) {
            write_error_to_file(err);
            document.getElementById("main_panel_load_food").innerHTML = err.responseText;
        }
    });
}
function rss_click() {
    $(".rss_feed button").on("click", function() {
        var location = window.location.href,
            index_location = location.indexOf("index.php"),
            new_location = location.substring(0, index_location),
            rss_location = new_location + "data/rss-" + lang + ".xml",
            textArea = document.createElement("textarea");
        textArea.value = rss_location;
        textArea.className = "always_hidden";
        document.body.appendChild(textArea);
        textArea.focus();
        textArea.select();
        document.execCommand('copy');
        document.body.removeChild(textArea);
        show_notification("rss");
    });
}
function login_work() {
    if (!$(".login_register:visible").length) { setTimeout(login_work, 50); } else { start_login_register(); }
}
function add_post_work() {
    if (!$("#add_post form:visible").length) { setTimeout(add_post_work, 50); } else { start_add_post(); }
}
function add_comment_view(post_id) {
    document.getElementById("add_comment").innerHTML =
`<h2>${language_content.food_posts.add_comment_title}</h2><textarea rows="8" cols="60"
```

```
name="add_comment_area" id="add_comment_area"></textarea><p
id="add_comment_error"></p><button
id="add_comment_button">${language_content.food_posts.add_comment_button}</button>`;
  comment_work(post_id);
}
function comment_work(post_id) {
  document.getElementById("add_comment_button").addEventListener("click", function() {
    var comment_text = document.getElementById("add_comment_area").value;
    if (comment_text.trim().length < 5) {
      document.getElementById("add_comment_error").innerHTML =
language_content.food_posts.empty_comment;
      return;
    }
    document.getElementById("add_comment_error").innerHTML = "";
    $.ajax({
      url: "models/comment.php",
      method: "POST",
      dataType: "json",
      data: {
        post_id: post_id,
        comment_text: comment_text.trim()
      },
      success: function (data) {
        document.getElementById("add_comment_area").value = "";
        show_notification("comment", true);
      },
      error: function (err) {
        write_error_to_file(err);
        switch(err.responseJSON) {
          case "empty_comment":
            document.getElementById("add_comment_error").innerHTML =
language_content.food_posts.empty_comment;
            break;
          case "log_to_comment":
            document.getElementById("add_comment_error").innerHTML =
language_content.food_posts.log_to_comment;
            break;
          case "database_error":
            document.getElementById("add_comment_error").innerHTML =
language_content.food_posts.database_error;
            break;
          default:
            document.getElementById("add_comment_error").innerHTML =
language_content.food_posts.other_error;
        }
      }
    });
  });
```

```
}
```

3.59. models/admin/post_tag.php

```
<?php
    header("Content-Type: application/json");
    include("../../config/connection.php");
    $query = "SELECT p.title AS postTitle, p.SRBtitle AS postSRBTitle, p.ID AS postID, t.name AS tagName,
t.SRBname AS tagSRBName, t.ID AS tagID FROM post_tag pt INNER JOIN post p ON pt.ID_post = p.ID
INNER JOIN tags t ON pt.ID_tag = t.ID";
    $posttags = executeQuery($query);
    echo json_encode($posttags);
?>
```

3.60. assets/js/footer_template.js

```
var selected_tags = [];

function fill_navigation_footer(text) {
    document.getElementById("navigation_footer").innerHTML = text +
document.getElementById("navigation_footer").innerHTML;
    $("#navigation_footer li a").on("click", function(e) {
        if ($.inArray($(this).attr("href"), food_types) !== -1) {
            selected_food_type = $(this).attr("href");
            start_food_posts();
        } else {
            selected_food_type = "";
            if ($(this).attr("href") == "index.php?page=main_template") {
                language_main_template();
            } else if ($(this).attr("href") == "index.php?page=admin_template") {
                start_admin();
            } else {
                start_contact();
            }
        }
    });
}

function fill_tags_footer(text) {
    document.getElementById("tags_footer").innerHTML = text;
    $("#tags_footer span").on("click", function(e) {
        if(selected_tags.indexOf($(this).attr("data-id")) >= 0) {
            for (var i = 0; i < selected_tags.length; i++) {
                if ( selected_tags[i] === $(this).attr("data-id")) {
                    selected_tags.splice(i, 1);
                }
            }
            $(this).removeClass("selected_footer_tag");
        } else {
            selected_tags.push($(this).attr("data-id"));
            $(this).addClass("selected_footer_tag");
        }
        filter_tags();
    });
}

function filter_tags() {
    load_food_posts();
}

function fill_top_posts_footer(text) {
    document.getElementById("recent_posts_footer").innerHTML = text;
}

function language_footer_template() {
    $(".footer_navigation h3").html(language_content.footer_template.navigation);
    $(".footer_recents h3").html(language_content.footer_template.recent_posts);
    $(".footer_tags h3").html(language_content.footer_template.tags);
}
```

3.61. assets/js/header_template.js

```
function fill_navigation_header(text) {
    document.getElementById("header_content").innerHTML = text;
}

$(".header .language img").on("click", function() {
    set_language($(this).attr("data-img"));
});
$("#menu_click").click(function() {
    if ($("#header_content").hasClass("show_menu")) {
        $("#header_content").removeClass("show_menu");
    } else {
        $("#header_content").addClass("show_menu");
    }
});

function set_language(lang) {
    localStorage.setItem("language", lang);
    location.reload();
}

function language_header_template() {
    document.getElementsByClassName("language")[0].firstElementChild.alt =
language_content.header_template.serbian_flag;
    document.getElementsByClassName("language")[0].lastElementChild.alt =
language_content.header_template.uk_flag;
}
```

3.62. assets/js/login_template.js

```
function start_login_register() {
   $(".login_param label[for='register_email']").html(language_content.login_template.register_email);
   $(".login_param #register_email").attr("placeholder",
language_content.login_template.register_email_placeholder);

   $(".login_param label[for='register_name']").html(language_content.login_template.register_name);
   $(".login_param #register_name").attr("placeholder",
language_content.login_template.register_name_placeholder);

   $(".login_param
label[for='register_lastname']").html(language_content.login_template.register_lastname);
   $(".login_param #register_lastname").attr("placeholder",
language_content.login_template.register_lastname_placeholder);

   $(".login_param
label[for='register_password']").html(language_content.login_template.register_password);
   $(".login_param
label[for='register_password_repeat']").html(language_content.login_template.register_password_repe
at);

   $("#register_button").html(language_content.login_template.register);
   $("#login_button").html(language_content.login_template.login);
   $("#forgot_password").html(language_content.login_template.forgot_password);

   $(".login_param label[for='login_email']").html(language_content.login_template.register_email);
   $(".login_param #login_email").attr("placeholder",
language_content.login_template.register_email_placeholder);

   $(".login_param
label[for='login_password']").html(language_content.login_template.register_password);

   $(".login_register .register_title").html(language_content.login_template.register);
   $(".login_register .login_title").html(language_content.login_template.login);
   if ($("#login_email:visible").length) { login_register_events(); }
}
function login_register_events() {
   document.getElementById("login_email").addEventListener("keyup", function (e) {
      if (e.key == "Enter") { login_request(); }
   });
   document.getElementById("login_password").addEventListener("keyup", function (e) {
      if (e.key == "Enter") { login_request(); }
   });
   $("#login_button").on("click", login_request);
```

```
  document.getElementById("register_email").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { register_request(); }
  });
  document.getElementById("register_name").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { register_request(); }
  });
  document.getElementById("register_lastname").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { register_request(); }
  });
  document.getElementById("register_password").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { register_request(); }
  });
  document.getElementById("register_password_repeat").addEventListener("keyup", function (e) {
    if (e.key == "Enter") { register_request(); }
  });
  $("#register_button").on("click", register_request);

  $("#forgot_password").on("click", forgot_password);
}
function login_request() {
  var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
    email = document.getElementById("login_email").value,
    password = document.getElementById("login_password").value;
  if (email == "" || password == "") {
    document.getElementById("login_error").innerHTML =
language_content.login_template.empty_field;
    return;
  }
  if (password.length < 6) {
    document.getElementById("login_error").innerHTML =
language_content.login_template.password_short;
    return;
  }
  if (!email_pat.test(email)) {
    document.getElementById("login_error").innerHTML =
language_content.login_template.not_valid_email;
    return;
  }
  document.getElementById("login_error").innerHTML = "";
  $.ajax({
    url: "models/login.php",
    method: "POST",
    dataType: "json",
    data: {
      email: email,
      password: password
    },
    success: function (data) {
      if (data == "Successful") {
        show_notification("login", true);
      }
      if (data == "wrong_params") {
        document.getElementById("login_error").innerHTML =
language_content.login_template.wrong_params;
      }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
          document.getElementById("login_error").innerHTML =
language_content.login_template.database_error;
          break;
        case "empty_field":
          document.getElementById("login_error").innerHTML =
language_content.login_template.empty_field;
          break;
        case "not_valid_email":
          document.getElementById("login_error").innerHTML =
language_content.login_template.not_valid_email;
          break;
        case "password_short":
          document.getElementById("login_error").innerHTML =
language_content.login_template.password_short;
          break;
        default:
          document.getElementById("login_error").innerHTML =
language_content.login_template.other_error;
      }
    }
  });
}
function register_request() {
  var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
    name_pat = /^[A-Z][a-z]+$/,
    lastname_pat = /^([A-Z][a-z]+)+$/,
    email = document.getElementById("register_email").value,
    name = document.getElementById("register_name").value,
    lastname = document.getElementById("register_lastname").value,
    password = document.getElementById("register_password").value,
    password_repeat = document.getElementById("register_password_repeat").value;
  if (email == "" || name == "" || lastname == "" || password == "" || password_repeat == "") {
    document.getElementById("register_error").innerHTML =
language_content.login_template.empty_field;
    return;
  }
  if (password.length < 6) {
    document.getElementById("register_error").innerHTML =
language_content.login_template.password_short;
    return;
  }
  if (password != password_repeat) {
    document.getElementById("register_error").innerHTML =
language_content.login_template.passwords_not_match;
    return;
  }
  if (!email_pat.test(email)) {
    document.getElementById("register_error").innerHTML =
language_content.login_template.not_valid_email;
    return;
  }
  if (!name_pat.test(name) || !lastname_pat.test(lastname)) {
    document.getElementById("register_error").innerHTML =
language_content.login_template.name_invalid;
    return;
  }
  document.getElementById("register_error").innerHTML = "";
  $.ajax({
    url: "models/register.php",
    method: "POST",
    dataType: "json",
    data: {
      email: email,
      name: name,
      lastname: lastname,
      password: password,
      password_repeat: password_repeat
    },
    success: function (data) {
      if (data == "Successful") {
        show_notification("register");
        document.getElementById("register_email").value = "";
        document.getElementById("register_name").value = "";
        document.getElementById("register_lastname").value = "";
        document.getElementById("register_password").value = "";
        document.getElementById("register_password_repeat").value = "";
      }
    },
    error: function (err) {
      write_error_to_file(err.responseJSON);
      switch(err.responseJSON) {
        case "database_error":
          document.getElementById("register_error").innerHTML =
language_content.login_template.database_error;
```

```
          break;
        case "empty_field":
          document.getElementById("register_error").innerHTML =
language_content.login_template.empty_field;
          break;
        case "passwords_not_match":
          document.getElementById("register_error").innerHTML =
language_content.login_template.passwords_not_match;
          break;
        case "name_invalid":
          document.getElementById("register_error").innerHTML =
language_content.login_template.name_invalid;
          break;
        case "email_taken":
          document.getElementById("register_error").innerHTML =
language_content.login_template.email_taken;
          break;
        case "not_valid_email":
          document.getElementById("register_error").innerHTML =
language_content.login_template.not_valid_email;
          break;
        case "password_short":
          document.getElementById("register_error").innerHTML =
language_content.login_template.password_short;
          break;
        default:
          document.getElementById("register_error").innerHTML =
language_content.login_template.other_error;
      }
    }
  });
}
function forgot_password() {
  var email_pat = /^[a-zA-Z0-9.!#$%&’*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/,
    email = document.getElementById("login_email").value;
  if (email == "") {
    document.getElementById("login_error").innerHTML =
language_content.login_template.empty_email;
    return;
  }
  if (!email_pat.test(email)) {
    document.getElementById("login_error").innerHTML =
language_content.login_template.not_valid_email;
    return;
  }
  document.getElementById("login_error").innerHTML = "";
  $.ajax({
    url: "models/forgot_password.php",
        method: "POST",
        dataType: "json",
        data: {
            email: email
        },
        success: function (data) {
            show_notification("forgot_password");
            document.getElementById("login_email").value = "";
        },
        error: function (err) {
            write_error_to_file(err.responseJSON);
            switch(err.responseJSON) {
                case "database_error":
                    document.getElementById("login_error").innerHTML =
language_content.login_template.database_error;
                    break;
                case "empty_email":
                    document.getElementById("login_error").innerHTML =
language_content.login_template.empty_email;
                    break;
                case "not_valid_email":
                    document.getElementById("login_error").innerHTML =
language_content.login_template.not_valid_email;
                    break;
                default:
                    document.getElementById("login_error").innerHTML =
language_content.login_template.other_error;
            }
        }
    });
}
```

3.63. assets/js/main_template.js

```
function start_main_template() {
    if (!$(".landing_screen") || !$(".landing_screen h2:visible").length || !$(".landing_screen
img:visible").length) {
        setTimeout(start_main_template, 50);
    } else {
        var elArray = document.getElementsByClassName("landing_screen");
        setTimeout(function() {
            var firstChild = elArray[0];
            fill_first_child(firstChild);
        }, 10);
    }
}
function language_main_template() {
   setTimeout(start_main_template, 50);
}
function fill_first_child(firstChild) {
   firstChild.firstElementChild.innerHTML = language_content.main_template.title;
   firstChild.lastElementChild.alt = language_content.main_template.baking_image;
}
```

3.64. assets/js/main.js

```
var lang = "eng",
   language_content = "",
   food_types = ['appetizer', 'main_course', 'dessert'],
   selected_food_type = '',
   posts = [],
   page = window.location.href.substr(window.location.href.indexOf("?"), window.location.href.length -
1),
   page_content = page.substr(page.indexOf("=") + 1, page.length - 1);

$(document).ready(function () {
   get_language();
   get_content();
   get_navigation("header");
   get_navigation("footer");
   get_tags("footer");
   get_top_posts("footer");
});

function get_language() {
   if (window.localStorage.getItem("language")) {
      lang = window.localStorage.getItem("language");
   } else {
      lang = "eng";
   }
}

function language_head() {
   if (lang == "eng") {
      document.getElementsByTagName("head")[0].innerHTML += `<title>25Hour Kitchen</title>
         <meta name="keywords" content="25, hours, kitchen, meal, food"/>
           <meta name="description" content="The best blog for your favourite meals and recipes!"/>
           <meta name="abstract" content="25Hour Kitchen with RarOks 1008."/>`;
   }
   if (lang == "srb") {
      document.getElementsByTagName("head")[0].innerHTML += `<meta charset="utf-8"/>
         <title>25Sati Kuhinja</title>
         <meta name="keywords" content="25, sat, kuhinja, obrok, hrana"/>
           <meta name="description" content="Najbolji blog za Vašu omiljenu hranu i recepte!"/>
           <meta name="abstract" content="25Sati Kuhinja sa RarOks-om 1008."/>`;
```

```
    }
    $("h1").html(language_content.page_title);
}

function get_content() {
    $.ajax({
        url: "assets/lang/" + lang + ".json",
        method: "GET",
        dataType: "json",
        success: function (data) {
            language_content = data;
            language_head();
            language_header_template();
            language_footer_template();
            selected_food_type = "";
            switch(page_content) {
                case "main_template":
                    language_main_template();
                    break;
                case "appetizer":
                case "main_course":
                case "dessert":
                    selected_food_type = page_content;
                    start_food_posts();
                    break;
                case "contact_template":
                    start_contact();
                    break;
                case "admin_template":
                    start_admin();
                    break;
                default:
                    page_content = "main_template";
                    language_main_template();
                    break;
            }
        },
        error: function (err) {
            write_error_to_file(err);
            window.location.reload();
        }
    });
}

function get_top_posts(which) {
    $.ajax({
        url: "models/top_posts.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (p) {
                text += `<div class="top_post"><h4>${lang=="eng"?p.title:(lang=="srb"?p.SRBtitle:"Error")}</h4><div><span>${p.postDate}</span><span>${p.name} ${p.lastname}</span></div></div>`;
            });
            if (which == "footer") {
                fill_top_posts_footer(text);
            }
        },
        error: function (err) {
            write_error_to_file(err);
            fill_top_posts_footer(err.responseText);
        }
    });
}

function get_navigation(which) {
    $.ajax({
        url: "models/navigation.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (n) {
                text += `<li><a href="index.php?page=${n.Link}">${lang=="eng"?n.Title:(lang=="srb"?n.SRBTitle:"Error")}</a></li>`;
            });
            if (which == "header") {
                fill_navigation_header(text);
            }
            if (which == "footer") {
                fill_navigation_footer(text);
            }
        },
        error: function (err) {
            write_error_to_file(err.responseText);
            if (which == "header") {
                fill_navigation_header(err.responseText);
            }
            if (which == "footer") {
                fill_navigation_footer(err.responseText);
            }
        }
    });
}
```

```
function get_tags(which) {
    $.ajax({
        url: "models/tags.php",
        method: "GET",
        dataType: "json",
        success: function (data) {
            var text = "";
            data.forEach(function (n) {
                text += `<span data-
id="${n.ID}">${lang=="eng"?n.name:(lang=="srb"?n.SRBname:"Error")}</span>`;
            });
            if (which == "footer") {
                fill_tags_footer(text);
            }
        },
        error: function (err) {
            write_error_to_file(err);
            fill_tags_footer(err.responseText);
        }
    });
}

function show_notification(which, reload) {
    $(".notification").css({"display": "flex", "opacity": 1});
    switch (which) {
        case "rss":
            document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.rss_title;
            document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.rss_text;
            break;
        case "login":
            document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.login_title;
            document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.login_text;
            break;
        case "register":
            document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.register_title;
            document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.register_text;
            break;
        case "logout":
            document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.logout_title;
```

```
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.logout_text;
      break;
    case "comment":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.comment_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.comment_text;
      break;
    case "insert_user":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.insert_user_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.insert_user_text;
      break;
    case "insert_tag":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.insert_tag_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.insert_tag_text;
      break;
    case "insert_post_tag":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.insert_post_tag_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.insert_post_tag_text;
      break;
    case "update_user":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.update_user_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.update_user_text;
      break;
    case "update_post":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.update_post_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.update_post_text;
      break;
    case "update_tag":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.update_tag_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.update_tag_text;
      break;
    case "delete_user":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_user_title;
```

```
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_user_text;
      break;
    case "delete_post":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_post_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_post_text;
      break;
    case "delete_tag":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_tag_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_tag_text;
      break;
    case "delete_picture":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_picture_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_picture_text;
      break;
    case "delete_comment":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_comment_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_comment_text;
      break;
    case "delete_newsletter":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_newsletter_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_newsletter_text;
      break;
    case "delete_post_tag":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.delete_post_tag_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.delete_post_tag_text;
      break;
    case "subscription":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.subscription_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.subscription_text;
      break;
    case "contact_mail_sent":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.contact_mail_sent_title;
```

```
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.contact_mail_sent_text;
      break;
    case "word_saved":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.word_saved_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.word_saved_text;
      break;
    case "excel_saved":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.excel_saved_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.excel_saved_text;
      break;
    case "forgot_password":
      document.getElementsByClassName("notification")[0].firstElementChild.innerHTML =
language_content.notification.forgot_password_title;
      document.getElementsByClassName("notification")[0].lastElementChild.innerHTML =
language_content.notification.forgot_password_text;
      break;
  }
  $(".notification").animate({
    opacity: 0
  }, 5000, function() {
    $(".notification").css("display", "none");
    if (reload) { window.location.reload(); }
  });
}
function write_error_to_file(data) {
  $.ajax({
    url: "models/write_error.php",
    method: "POST",
    dataType: "json",
    data: {
      text: data
    },
    success: function (data) {
      console.log("Wrote error successfully");
    },
    error: function (err) {
      console.log("Could not write error");
    }
  });
}
```

3.65. assets/css/style.css

```
@font-face {
    font-family: IndieFlower;
    src: url(../fonts/IndieFlower.ttf);
}
@font-face {
  font-family: Slabo;
  src: url(../fonts/Slabo27px-Regular.ttf);
}
* {
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
h1 {
    position: absolute;
    top: 0;
    z-index: -10;
    visibility: hidden;
}
#main_panel {
    margin: 0px auto;
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
    font-family: Slabo;
}
textarea.always_hidden {
    width: 0px;
    height: 0px;
}
.notification {
    position: sticky;
    display: none;
    flex-direction: column;
    align-items: center;
    bottom: 200px;
    width: 40%;
    margin-left: 30%;
    background-color: rgba(255, 255, 255, 0.8);
    font-family: Slabo;
}
.notification span {
    padding: 10px;
    font-weight: bold;
}
.notification span:first-child {
    font-size: 24px;
}
.notification span:last-child {
    font-size: 18px;
}
@media screen and (max-width: 600px) {
    #main_panel {
        flex-direction: column;
        align-items: center;
    }
}
```

3.66. assets/css/food_post_template.css

```
#main_panel_load_food {
    margin-top: 10px;
    width: 60%;
}
#main_panel_load_food .post_header {
    padding-top: 20px;
    padding-bottom: 20px;
    background-color: #fafafa;
}
#main_panel_load_food .post_header .post_tags, #main_panel_load_food .post_details {
    list-style-type: none;
    display: flex;
    justify-content: center;
}
#main_panel_load_food .post_header .post_tags li, #main_panel_load_food .post_details li {
    padding-left: 5px;
    padding-right: 5px;
    border-left: 1px solid #f6490d;
    border-right: 1px solid #f6490d;
    color: #f6490d;
    font-size: 16px;
}
#main_panel_load_food h2 {
    font-style: italic;
    text-align: center;
    margin-top: 20px;
    margin-bottom: 20px;
    font-size: 24px;
}
#main_panel_load_food .post_image {
    width: 100%;
    position: relative;
    display: flex;
    flex-direction: column;
```

```
   align-items: center;
}
#main_panel_load_food .post_image button {
   position: absolute;
   font-size: 20px;
   padding: 13px 15px;
   bottom: 20px;
   background-color: #fafafa;
   opacity: 0.5;
   cursor: pointer;
   border: none;
   border-radius: 2px;
}
#main_panel_load_food .post_image button:first-child {
   left: 0;
}
#main_panel_load_food .post_image button:last-child {
   right: 0;
}
#main_panel_load_food .post_image img {
   max-width: 100%;
   object-fit: scale-down;
   margin-top: 20px;
   margin-bottom: 20px;
}
#main_panel_load_food p {
   font-size: 15px;
   text-align: justify;
   line-height: 25px;
}
#main_panel_load_food .bigger_letter::first-letter {
   font-size: 45px;
   float: left;
   line-height: 20px;
}
#main_panel_load_food .post_button {
   display: flex;
   justify-content: center;
   margin-top: 15px;
   margin-bottom: 50px;
}
#main_panel_load_food .post_button button {
   padding: 7px 14px;
   cursor: pointer;
   font-size: 15px;
   border: none;
   background-color: rgba(12, 221, 5, 0.8);
   border-radius: 3px;
```

```
}
#main_panel_load_edit {
   margin-top: 10px;
   width: 30%;
   padding: 5px;
}
#main_panel_load_edit div {
   margin-top: 100px;
}
#main_panel_load_edit .search_field {
   display: flex;
}
#main_panel_load_edit .search_field input, #main_panel_load_edit .search_field button,
#main_panel_load_edit .subscribe_newsletter input, #add_comment button {
   font-family: Slabo;
   font-size: 16px;
   padding: 5px;
   border: 1px solid #000;
   background-color: #fafafa;
}
#main_panel_load_edit .search_field input {
   width: 80%;
   border-right: 1px solid rgba(0, 0, 0, 0.2);
}
#main_panel_load_edit .how_it_started h2, #main_panel_load_edit .subscribe_newsletter h2,
#main_panel_load_edit .quote_field h2 {
   margin-bottom: 10px;
   text-align: center;
}
#main_panel_load_edit .how_it_started #subscribe_text, #main_panel_load_edit .subscribe_newsletter
p, #main_panel_load_edit .quote_field p {
   text-align: justify;
}
#main_panel_load_edit .subscribe_newsletter #subscribe_error {
   min-height: 20px;
   color: #ff0000;
   text-align: center;
}
#main_panel_load_edit .rss_feed {
   display: flex;
   justify-content: center;
}
#main_panel_load_edit .rss_feed button {
   border: none;
   cursor: pointer;
   font-size: 40px;
   background-color: transparent;
}
```

```
#main_panel_load_edit .subscribe_newsletter input {
   margin-left: 5%;
   width: 90%;
}
#main_panel_load_edit .subscribe_newsletter p {
   margin-top: 10px;
}
#main_panel_load_edit .quote_field p {
   font-size: 25px;
   font-style: italic;
   text-decoration: underline;
}
#main_panel_load_edit .quote_field p span {
   color: #ff0000;
}
.comment_holder {
   padding: 5px;
   background-color: #fafafa;
   border-radius: 5px;
   margin-bottom: 50px;
}
.comment_holder p {
   font-size: 18px !important;
}
.comment_holder p:first-child {
   margin-left: 20px;
   font-style: italic;
   text-decoration: underline;
}
.comment_holder p:nth-child(2) {
   text-align: center !important;
}
.comment_holder p:last-child {
   font-weight: bold;
   text-align: right !important;
}
#new_post_area {
   width: 80%;
   padding: 10px;
   margin: 25px 0px;
   background-color: #fafafa;
   border-radius: 2px;
   display: flex;
   justify-content: space-between;
   flex-wrap: wrap;
}
#new_post_area .add_post_area {
   width: 100%;
```

```
  display: flex;
  flex-direction: column;
}
#new_post_area .add_post_area .welcome_user {
  display: flex;
  justify-content: space-between;
  font-size: 24px;
  margin-bottom: 10px;
}
#new_post_area .add_post_area .welcome_user span:last-child {
  text-decoration: underline;
  font-style: italic;
}
#new_post_area .add_post_area .welcome_user button {
  font-size: 18px;
  padding: 5px 10px;
  background-color: #fff;
  border: 1px solid #000;
  cursor: pointer;
}
#add_comment {
  display: flex;
  flex-direction: column;
  margin-bottom: 20px;
  justify-content: center;
  align-items:center;
}
#add_comment button {
  width: 200px;
  cursor: pointer;
  margin-top: 10px;
}
#add_comment #add_comment_error {
  min-height: 22px;
  margin-top: 5px;
  color: red;
  font-size: 16px;
}
#add_comment textarea {
  font-size: 16px;
  border: 1px solid #000;
  background-color: #fff;
  padding: 5px;
  border-radius: 4px;
}
#main_panel_load_food #page_numbers {
  display: flex;
  justify-content: center;
```

```
    margin-bottom: 30px;
}
#main_panel_load_food #page_numbers button {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    cursor: pointer;
    margin-left: 10px;
}

@media screen and (max-width: 760px) {
    #new_post_area {
        width: 98%;
        flex-direction: column;
    }
}
@media screen and (max-width: 600px) {
    #main_panel_load_food {
        width: 95%;
    }
    #main_panel_load_edit {
        width: 95%;
    }
}
```

3.67. assets/css/footer_template.css

```
.footer {
    width: 100%;
    color: #ffffff;
    font-family: Slabo;
}
.footer p {
    background-color: #0cdd05;
    font-size: 18px;
    text-align: center;
    padding: 5px;
}
.footer .footer_holder .footer_part {
    width: 33%;
    background-color: black;
    padding: 10px;
}
.footer .footer_holder {
    display: flex;
    justify-content: space-between;
```

```
   background-color: #0cdd05;
}
.footer .footer_holder h3 {
   color: #00A78C;
   text-align: center;
}
#navigation_footer {
   list-style-type: none;
   margin-left: 3%;
}
#navigation_footer li {
   margin-top: 5px;
}
#navigation_footer li a {
   text-decoration: none;
   color: #00A78C;
}
#navigation_footer li a:hover {
   color: #0cdd05;
}
#tags_footer {
   padding: 10px;
   display: flex;
   flex-wrap: wrap;
}
#tags_footer span {
   padding: 7px;
   color: #00A78C;
   border: 1px solid #00A78C;
   margin-top: 7px;
   margin-left: 7px;
   border-radius: 5px;
   cursor: pointer;
}
#tags_footer span:hover {
   color: #0cdd05;
   border: 1px solid #0cdd05;
}
.selected_footer_tag {
   color: #0073F3 !important;
   border: 1px solid #0073F3 !important;
}
.footer_part.footer_recents {
   color: #e8e8e8;
}
.footer_part.footer_recents h3 {
   margin-bottom: 10px;
}
```

```
.footer_part.footer_recents .top_post {
   padding-left: 10px;
   padding-right: 10px;
   margin-bottom: 6px;
}
.footer_part.footer_recents .top_post h4 {
   text-align: center;
}
.footer_part.footer_recents .top_post div {
   display: flex;
   justify-content: space-between;
   margin-top: 3px;
   font-style: italic;
}
@media screen and (max-width: 660px) {
   .footer .footer_holder {
      flex-wrap: wrap;
      justify-content: center;
   }
   .footer .footer_holder .footer_part {
      margin-top: 10px;
      width: 98%;
   }
}
```

3.68. assets/css/header_template.css

```
.header {
   width: 100%;
   height: 50px;
   padding: 5px;
   background-color: #0cdd05;
   display: flex;
   justify-content: space-around;
   align-items: center;
   color: #ffffff;
   font-family: IndieFlower;
}
.header .language {
   width: 70px;
   padding: 2px;
   display: flex;
   justify-content: space-between;
}
.header .language img {
   cursor: pointer;
}
.header ul {
   list-style-type: none;
   display: flex;
   width: 80%;
   justify-content: space-around;
   z-index: 2;
}
.header ul li a {
   padding: 2px;
   border-radius: 2px;
   text-decoration: none;
   color: #ffffff;
   font-weight: bold;
   font-size: 30px;
}
.header ul li a:hover {
   background-color: #c2dd05;
}
.header .fas {
   font-size: 30px;
   margin-left: 15px;
   display: none;
}
.header .logo_and_nav {
   display: flex;
   align-items: center;
}
.show_menu {
   height: auto !important;
   display: block !important;
   align-self: flex-start;
   margin-left: 10px;
   background-color: #0cdd05;
}

@media screen and (max-width: 920px) {
   .header {
     justify-content: space-between;
   }
   .header .fas {
     display: block;
   }
   .header ul {
     height: 0;
     display: none;
   }
}
@media screen and (max-width: 440px) {
   .logo {
        display: none;
    }
}
```

3.69. assets/css/login_template.css

```
#new_post_area .login_register {
    width: 49%;
    display: flex;
    flex-direction: column;
}
#new_post_area .login_register .login_param {
    display: flex;
    align-items: center;
    justify-content: space-between;
}
#new_post_area .login_register:last-child {
    padding-left: 5px;
    border-left: 1px dashed red;
}
#new_post_area .login_register:first-child {
    padding-right: 5px;
    border-right: 1px dashed red;
}
#new_post_area .login_register .login_param input, #new_post_area .login_register button {
    font-size: 16px;
    margin-top: 10px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
}
#new_post_area .login_register p {
    min-height: 20px;
    color: #ff0000;
    font-size: 16px;
}
#new_post_area .login_register button {
    width: 120px;
    cursor: pointer;
}
#new_post_area .login_register button#forgot_password {
    width: 160px;
}
@media screen and (max-width: 760px) {
    #new_post_area .login_register {
        width: 98%;
    }
```

```
  #new_post_area .login_register:first-child {
    border: none;
  }
  #new_post_area .login_register:last-child {
    margin-top: 20px;
    border: none;
  }
}
@media screen and (max-width: 400px) {
  #new_post_area .login_register .login_param label {
    width: 70%;
    font-size: 14px;
  }
  #new_post_area .login_register .login_param input {
    width: 29%;
    font-size: 14px;
  }
}
```

3.70. assets/css/main_template.css

```css
.landing_screen {
   position: relative;
   width: 100%;
   display: flex;
   justify-content: center;
}
.landing_screen h2 {
   font-family: IndieFlower;
   font-size: 40px;
   position: absolute;
   top: 20%;
   padding: 10px 30px;
   background-color: rgba(12, 221, 5, 0.4);
   color: #fff;
}
.landing_screen img {
   width: 100%;
}
```

3.71. assets/css/add_post_template.css

```css
.add_post_form h2 {
   font-size: 28px;
}
.add_post_form .add_post_param {
   display: flex;
   align-items: center;
   justify-content: center;
}
.add_post_form .add_post_param {
   margin-top: 15px;
}
.add_post_form .add_post_param label {
   margin-right: 10px;
   font-style: italic;
}
.add_post_form .add_post_param input, .add_post_form .add_post_param textarea, .add_post_form
input[type="submit"] {
   font-size: 16px;
   border: 1px solid #000;
   background-color: #fff;
   padding: 5px;
   border-radius: 4px;
}
.add_post_form p {
   min-height: 20px;
   color: #ff0000;
```

```
    font-size: 16px;
}
.add_post_form input[type="submit"] {
    width: 120px;
    cursor: pointer;
}
@media screen and (max-width: 400px) {
    .add_post_form .add_post_param label, .add_post_form .add_post_param div label  {
        width: 29%;
        font-size: 14px;
    }
    .add_post_form .add_post_param div input {
        width: initial;
    }
    .add_post_form .add_post_param input, .add_post_form .add_post_param div {
        width: 70%;
        font-size: 14px;
    }
}
```

3.72. assets/css/admin_template.css

```
#admin_panel {
    width: 96%;
    margin-left: 2%;
    display: flex;
    flex-direction: column;
    margin-top: 15px;
    margin-bottom: 20px;
    align-items: center;
}
#admin_panel h2 {
    text-align: center;
    margin-bottom: 30px;
}
#admin_panel #admin_option {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    margin-top: 15px;
    width: 300px;
}
#admin_panel .admin_controls {
    width: 90%;
    margin-top: 25px;
    background-color: #fafafa;
    border-radius: 5px;
    padding-top: 10px;
    display: flex;
    flex-direction: column;
    align-items: center;
}
#admin_panel .admin_controls #admin_extra_option {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    margin-top: 15px;
    width: 300px;
    margin-bottom: 50px;
    display: none;
}
#admin_panel .admin_controls #admin_extra_space {
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
}
#admin_panel .admin_controls #admin_extra_space form {
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
}
#admin_panel .admin_controls #admin_extra_space span {
    user-select: none;
    font-size: 20px;
    font-style: italic;
}
#admin_panel .admin_controls #admin_extra_space img {
    width: 60%;
}
#admin_panel .admin_controls #admin_extra_space .admin_option_holder {
    width: 98%;
    margin-left: 1%;
    display: flex;
    justify-content: center;
    align-items: center;
    margin-top: 10px;
    margin-bottom: 10px;
}
#admin_panel .admin_controls #admin_extra_space .admin_option_holder label {
    font-size: 16px;
    margin-right: 20px;
    width: 20%;
    text-align: right;
}
#admin_panel .admin_controls #admin_extra_space .admin_option_holder input, #admin_panel
.admin_controls #admin_extra_space .admin_option_holder select, #admin_panel .admin_controls
#admin_extra_space .admin_option_holder textarea {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    width: 60%;
}
#admin_panel .admin_controls #admin_extra_space button, #admin_panel
.admin_controls.other_controls button {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    width: 200px;
    margin-top: 20px;
    margin-bottom: 30px;
    cursor: pointer;
}
#admin_panel .admin_controls.other_controls button {
    width: 220px;
}
#admin_panel .admin_controls.other_controls a {
    font-size: 24px;
    text-decoration: none;
}
#admin_panel .admin_controls .admin_file_holder {
    width: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
    margin-top: 10px;
    font-size: 20px;
}

#admin_panel .admin_controls #admin_extra_space p {
    min-height: 20px;
    color: #ff0000;
    font-size: 16px;
}
@media screen and (max-width: 400px) {
    #admin_panel #admin_option, #admin_panel .admin_controls #admin_extra_option {
        width: 99%;
    }
}
```

3.73. assets/css/contact_template.css

```
#contact_template {
    width: 96%;
    margin-left: 2%;
    display: flex;
    justify-content: space-between;
    flex-wrap: wrap;
    padding: 10px;
}
#contact_template .contact_area {
    display: flex;
    flex-direction: column;
    width: 65%;
}
#contact_template .contact_area h2 {
    margin-top: 10px;
    margin-left: 20px;
}
#contact_template .contact_area #message_error {
    min-height: 20px;
    color: #ff0000;
}
#contact_template .contact_area input, #contact_template .contact_area textarea, #contact_template
.contact_area button, #contact_template .author button {
    font-size: 16px;
    border: 1px solid #000;
    background-color: #fff;
    padding: 5px;
    border-radius: 4px;
    margin-top: 15px;
}
#contact_template .contact_area button, #contact_template .author button {
    width: 200px;
    cursor: pointer;
}
#contact_template .author {
    width: 33%;
    display: flex;
    flex-direction: column;
    align-items: center;
}
#contact_template .author h2 {
    margin-top: 20px;
}
#contact_template .author img {
    margin-top: 20px;
    width: 80%;
}
#contact_template .author .author_data {
    margin-top: 10px;
    display: flex;
}
#contact_template .author .author_data p {
    text-align: center;
}
#contact_template .author .author_data p:first-child {
    margin-right: 5px;
}
@media screen and (max-width: 820px) {
    #contact_template {
        flex-direction: column;
    }
    #contact_template .contact_area, #contact_template .author {
        width: 100%;
    }
}
```

3.74. assets/css/page_404.css

```
.page_404 {
    width: 100%;
}
.page_404 img {
    width: 100%;
}
```

3.75. data/errors.txt (PRIMER)

```
ERROR_TYPE	11-06-2019 19:01:05
```

3.76. data/log.txt (PRIMER)

```
main_template	10-06-2019 20:41:36	::1
appetizer	10-06-2019 20:41:37	::1
main_course	10-06-2019 20:41:38	::1
dessert	10-06-2019 20:41:39	::1
contact_template	10-06-2019 20:41:40	::1
admin_template	10-06-2019 20:41:42	::1
page_404	10-06-2019 20:41:46	::1
admin_template	10-06-2019 21:34:14	::1
admin_template	10-06-2019 21:36:23	::1
admin_template	10-06-2019 21:39:59	::1
```

```
admin_template        10-06-2019 21:40:48    ::1
admin_template        10-06-2019 21:41:07    ::1
appetizer      10-06-2019 21:44:44    ::1
appetizer      10-06-2019 21:44:56    ::1
appetizer      10-06-2019 21:44:59    ::1
appetizer      10-06-2019 21:45:07    ::1
appetizer      10-06-2019 21:46:10    ::1
appetizer      10-06-2019 21:46:15    ::1
appetizer      10-06-2019 21:47:23    ::1
appetizer      10-06-2019 21:47:44    ::1
main_course    10-06-2019 21:47:49    ::1
main_template 10-06-2019 21:47:50    ::1
contact_template      10-06-2019 21:47:52    ::1
main_course    10-06-2019 21:50:13    ::1
main_course    10-06-2019 21:51:23    ::1
main_course    10-06-2019 21:51:35    ::1
admin_template        10-06-2019 21:51:39    ::1
admin_template        10-06-2019 21:51:53    ::1
admin_template        10-06-2019 21:52:06    ::1
admin_template        10-06-2019 21:52:20    ::1
admin_template        10-06-2019 21:54:49    ::1
appetizer      10-06-2019 21:55:14    ::1
appetizer      10-06-2019 21:55:20    ::1
appetizer      10-06-2019 21:55:31    ::1
main_course    10-06-2019 21:55:32    ::1
main_course    10-06-2019 21:55:38    ::1
main_course    10-06-2019 21:55:50    ::1
admin_template        10-06-2019 21:55:57    ::1
admin_template        10-06-2019 21:57:15    ::1
admin_template        10-06-2019 21:57:36    ::1
admin_template        10-06-2019 21:59:27    ::1
admin_template        10-06-2019 22:02:11    ::1
admin_template        10-06-2019 22:02:31    ::1
admin_template        10-06-2019 22:04:36    ::1
admin_template        11-06-2019 17:29:40    ::1
admin_template        11-06-2019 17:29:49    ::1
main_course    11-06-2019 17:36:21    ::1
main_course    11-06-2019 17:36:28    ::1
main_course    11-06-2019 17:37:03    ::1
main_course    11-06-2019 17:42:14    ::1
dessert 11-06-2019 17:42:17    ::1
contact_template      11-06-2019 17:42:18    ::1
main_course    11-06-2019 17:42:19    ::1
appetizer      11-06-2019 17:42:19    ::1
main_template 11-06-2019 17:42:20    ::1
main_course    11-06-2019 17:42:23    ::1
dessert 11-06-2019 17:42:24    ::1
contact_template      11-06-2019 17:42:25    ::1
contact_template        11-06-2019 17:42:26     ::1
contact_template        11-06-2019 17:42:28     ::1
appetizer       11-06-2019 17:42:29     ::1
appetizer       11-06-2019 17:43:31     ::1
contact_template        11-06-2019 17:43:33     ::1
dessert 11-06-2019 17:43:34     ::1
dessert 11-06-2019 17:43:35     ::1
dessert 11-06-2019 17:58:03     ::1
admin_template          11-06-2019 17:58:06     ::1
admin_template          11-06-2019 17:59:16     ::1
admin_template          11-06-2019 18:00:29     ::1
admin_template          11-06-2019 18:01:05     ::1
admin_template          11-06-2019 18:01:09     ::1
admin_template          11-06-2019 18:01:29     ::1
admin_template          11-06-2019 18:01:34     ::1
admin_template          11-06-2019 18:03:55     ::1
admin_template          11-06-2019 18:04:17     ::1
admin_template          11-06-2019 18:04:34     ::1
admin_template          11-06-2019 18:04:40     ::1
admin_template          11-06-2019 18:04:55     ::1
admin_template          11-06-2019 18:30:17     ::1
admin_template          11-06-2019 18:31:02     ::1
admin_template          11-06-2019 18:31:41     ::1
admin_template          11-06-2019 18:37:06     ::1
admin_template          11-06-2019 18:38:43     ::1
admin_template          11-06-2019 18:43:04     ::1
main_course     11-06-2019 18:45:15     ::1
main_course     11-06-2019 18:45:28     ::1
main_course     11-06-2019 18:45:36     ::1
main_course     11-06-2019 18:48:02     ::1
main_course     11-06-2019 18:48:05     ::1
main_course     11-06-2019 18:48:07     ::1
main_course     11-06-2019 18:49:44     ::1
main_course     11-06-2019 19:00:40     ::1
main_course     11-06-2019 19:00:58     ::1
main_course     11-06-2019 19:01:47     ::1
main_course     11-06-2019 19:03:36     ::1
main_course     11-06-2019 19:04:33     ::1
main_course     11-06-2019 19:52:09     ::1
main_course     11-06-2019 19:52:11     ::1
dessert 11-06-2019 19:52:18     ::1
appetizer       11-06-2019 19:52:23     ::1
appetizer       11-06-2019 19:52:35     ::1
main_course     11-06-2019 19:52:45     ::1
main_course     11-06-2019 19:59:43     ::1
main_course     11-06-2019 20:01:48     ::1
dessert 11-06-2019 20:01:50     ::1
main_course     11-06-2019 20:01:51     ::1
```

```
appetizer      11-06-2019 20:01:52    ::1
appetizer      11-06-2019 20:03:49    ::1
appetizer      11-06-2019 20:05:32    ::1
appetizer      11-06-2019 20:06:26    ::1
appetizer      11-06-2019 20:07:08    ::1
appetizer      11-06-2019 20:07:19    ::1
appetizer      11-06-2019 20:07:53    ::1
appetizer      11-06-2019 20:08:03    ::1
appetizer      11-06-2019 20:08:12    ::1
appetizer      11-06-2019 20:08:20    ::1
appetizer      11-06-2019 20:08:29    ::1
appetizer      11-06-2019 20:10:01    ::1
appetizer      11-06-2019 20:10:13    ::1
appetizer      11-06-2019 20:12:23    ::1
appetizer      11-06-2019 20:12:30    ::1
appetizer      11-06-2019 20:16:23    ::1
appetizer      11-06-2019 20:16:47    ::1
appetizer      11-06-2019 20:17:47    ::1
appetizer      11-06-2019 20:17:55    ::1
appetizer      11-06-2019 20:19:00    ::1
appetizer      11-06-2019 20:19:47    ::1
appetizer      11-06-2019 20:20:20    ::1
appetizer      11-06-2019 20:20:47    ::1
appetizer      11-06-2019 20:21:13    ::1
appetizer      11-06-2019 20:23:05    ::1
main_course    11-06-2019 20:23:10    ::1
page_404       11-06-2019 20:27:03    ::1
page_404       11-06-2019 20:27:07    ::1
```

3.77. data/rss-eng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
<channel>
	<title>25Hour Kitchen Posts</title>
	<link>http://25h-kitchen.epizy.com/index.php</link>
	<description>All newest recipes on one place.</description>

	<item>
		<title>Buns</title>
		<link>http://25h-kitchen.epizy.com/index.php</link>
		<description>Fantasticna Kuvarka</description>
	</item>
	<item>
		<title>Milan risotto</title>
		<link>http://25h-kitchen.epizy.com/index.php</link>
		<description>Marko Markovic</description>
	</item>
```

```
    <item>
        <title>Baked tomato</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Broccoli Orecchiette</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Pizza roll</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Marko Markovic</description>
    </item>
    <item>
        <title>Stewed beans</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Stuffed peppers</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Fried chicken</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Beef stew</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Chocolate cake</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Blueberry pie</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Muhallebi</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Blinis with honey</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
</channel>
</rss>
```

3.78. data/rss-srb.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
<channel>
    <title>25Sati Kuhinja Objave</title>
    <link>http://25h-kitchen.epizy.com/index.php</link>
    <description>Svi najnoviji recepti na jednom mestu.</description>

    <item>
        <title>Kiflice</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Milanski rižoto</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Marko Markovic</description>
    </item>
    <item>
        <title>Pečen paradajz</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Brokoli Orecchiette</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Pizza rolat</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Marko Markovic</description>
    </item>
    <item>
        <title>Pohovana boranija</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Punjene paprike</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Pohovana piletina</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
    <item>
        <title>Juneći gulaš</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Čokoladni kolač</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Nesto Tamo</description>
    </item>
    <item>
        <title>Pita od borovnica</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Muhallebi</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Fantasticna Kuvarka</description>
    </item>
    <item>
        <title>Blinis sa medom</title>
        <link>http://25h-kitchen.epizy.com/index.php</link>
        <description>Drugi User</description>
    </item>
</channel>
</rss>
```

3.79. config/.env

```
DBNAME=***********
SERVER=***********
USERNAME=***********
PASSWORD=***********
```

3.80. config/config.php

```
<?php

define("ABSOLUTE_PATH", $_SERVER["DOCUMENT_ROOT"]);

define("ENV_FAJL", ABSOLUTE_PATH."/config/.env");
define("LOG_FAJL", ABSOLUTE_PATH."/data/log.txt");
define("ERROR_FAJL", ABSOLUTE_PATH."/data/errors.txt");

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

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

6.47. models/admin/newsletter.php

```
<?php
   header("Content-Type: application/json");
   include("../../config/connection.php");
   $query = "SELECT email, ID FROM newsletter_subscribers";
   $subscribers = executeQuery($query);
   echo json_encode($subscribers);
?>
```

3.81. config/connection.php

```
<?php

require_once "config.php";

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

function zabeleziPristupStranici ($content) {
   $open = fopen(LOG_FAJL, "a");
   if ($open) {
      $date = date('d-m-Y H:i:s');
      fwrite($open, "{$content}\t{$date}\t{$_SERVER['REMOTE_ADDR']}\t\n");
      fclose($open);
   }
   if (isset($_SESSION["user"])) {
      global $conn;
      $conn -> query("UPDATE users SET LastTimeSeen = NOW() WHERE ID = ".$_SESSION["user"]->ID);
   }
}
```

3.82. robots.txt

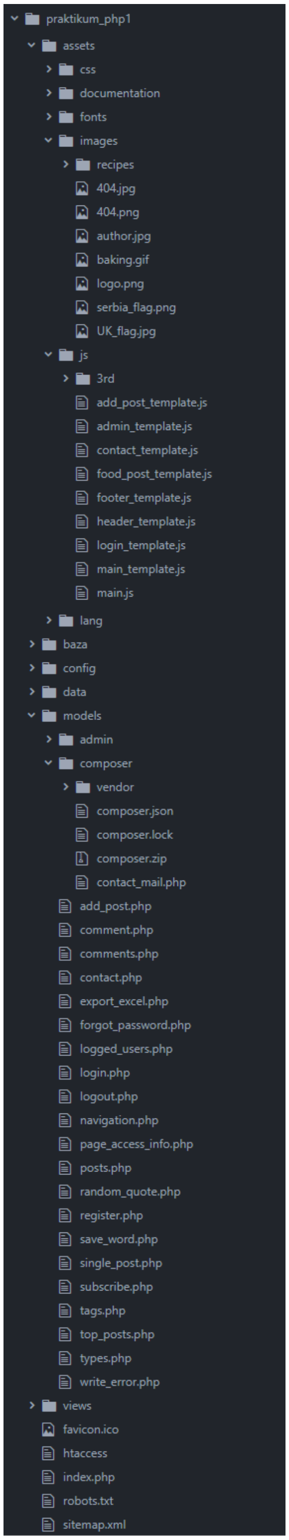
```
User-agent: *
Disallow: /assets/
```

3.83. .htaccess

```
ErrorDocument 404 http://25h-kitchen.epizy.com/index.php?page=404

ErrorDocument 403 http://25h-kitchen.epizy.com/index.php?page=404
Options All -Indexes
```

4. HIJERARHIJSKI PREGLED



```
praktikum_php1
  assets
    css
    documentation
    fonts
    images
      recipes
      404.jpg
      404.png
      author.jpg
      baking.gif
      logo.png
      serbia_flag.png
      UK_flag.jpg
    js
      3rd
      add_post_template.js
      admin_template.js
      contact_template.js
      food_post_template.js
      footer_template.js
      header_template.js
      login_template.js
      main_template.js
      main.js
    lang
  baza
  config
  data
  models
    admin
    composer
      vendor
      composer.json
      composer.lock
      composer.zip
      contact_mail.php
    add_post.php
    comment.php
    comments.php
    contact.php
    export_excel.php
    forgot_password.php
    logged_users.php
    login.php
    logout.php
    navigation.php
    page_access_info.php
    posts.php
    random_quote.php
    register.php
    save_word.php
    single_post.php
    subscribe.php
    tags.php
    top_posts.php
    types.php
    write_error.php
  views
  favicon.ico
  htaccess
  index.php
  robots.txt
  sitemap.xml
```