

Visoka škola strukovnih studija za informacione i komunikacione tehnologije



## Dokumentacija

<http://shopedia-belgrade.000webhostapp.com>

Nikola Nedeljković 2/17

# Sadržaj

Dokumentacija .....	1
Sadržaj .....	2
1. UVOD .....	4
1.1 Korišćeni programski / markap jezici .....	4
1.2 Opis funkcionalnosti.....	4
1.3 Izgled stranica .....	7
2. ORGANIZACIJA.....	8
2.1 Organizacija sajta .....	8
1. ....	9
2.2 Organizacija baze .....	9
2.3 Sitemap.xml .....	10
2.4 Slike stranica sa funkcionalnostima .....	11
3. KODOVI .....	17
3.1 index.php .....	17
3.2. api.php → komunikacija između index i kontrolera .....	18
3.3. data.php → dohvaćanje podataka za prikaz.....	19
3.4 autoload.php .....	20
3.5. DATABASE.php → SQL upiti i kreiranje baze .....	20
3.6 ItemController.php .....	21
3.7 LoginController.php .....	22
3.8. MeasureController.php .....	23
3.9 RegisterController.php .....	24
3.10 RoleController.php.....	24
3.11. ShopController.php .....	25
3.12 UserController.php .....	26
3.13. Item.php .....	26
3.14. Measure.php.....	27

3.15.	Role.php.....	28
3.16.	Shop.php.....	28
3.17.	User.php .....	28
3.18.	footer.php.....	29
3.19.	header.php.....	30
3.20.	404.php.....	31
3.21.	admin.php.....	31
3.22.	login.php.....	32
3.23.	main.php.....	33
3.24.	register.php.....	35
3.25.	teacher.php.....	35
3.26.	style.css.....	36
3.27.	main.js .....	44
4.	HIJERARHIJSKI PREGLED .....	48

# 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
- Sajt je napisan u Atom-u

## 1.2 Opis funkcionalnosti

- **index.php**
  - ✓ Dinamičko učitavanje stranica po stanju korisnika i odabranom linku
- **api.php**
  - ✓ Komunikacija između indexa i kontrolora
- **data.php**
  - ✓ Stranica za dohvatanje podataka koji treba da se prikažu na stranici
- **app/Config/autoload.php**
  - ✓ Autoload skripta
- **app/Config/config.php**
  - ✓ Promenljive za povezivanje na bazu
- **app/Config/DATABASE.php**
  - ✓ Klasa baze podataka
  - ✓ Inicijalizacija i izvršavanje upita
- **app/Controllers/ItemController.php**
  - ✓ Klasa ItemController
  - ✓ Inicijalizacija i rad sa podacima za tabelu item
- **app/Controllers/LoginController.php**
  - ✓ Klasa LoginController

- ✓ Inicijalizacija i rad sa podacima za logovanje korisnika
- **app/Controllers/MeasureController.php**
  - ✓ Klasa MeasureController
  - ✓ Inicijalizacija i rad sa podacima za tabelu measure
- **app/Controllers/RegisterController.php**
  - ✓ Klasa RegisterController
  - ✓ Inicijalizacija i rad sa podacima za registraciju korisnika
- **app/Controllers/RoleController.php**
  - ✓ Klasa RoleController
  - ✓ Inicijalizacija i rad sa podacima za tabelu role
- **app/Controllers/ShopController.php**
  - ✓ Klasa ShopController
  - ✓ Inicijalizacija i rad sa podacima za tabelu shop
- **app/Controllers/UserController.php**
  - ✓ Klasa UserController
  - ✓ Inicijalizacija i rad sa podacima za tabelu user
- **app/Models/Item.php**
  - ✓ Klasa Item
  - ✓ Inicijalizacija i rad sa tabelom item
- **app/Models/Measure.php**
  - ✓ Klasa Measure
  - ✓ Inicijalizacija i rad sa tabelom measure
- **app/Models/Role.php**
  - ✓ Klasa Role
  - ✓ Inicijalizacija i rad sa tabelom role
- **app/Models/Shop.php**
  - ✓ Klasa Shop
  - ✓ Inicijalizacija i rad sa tabelom shop

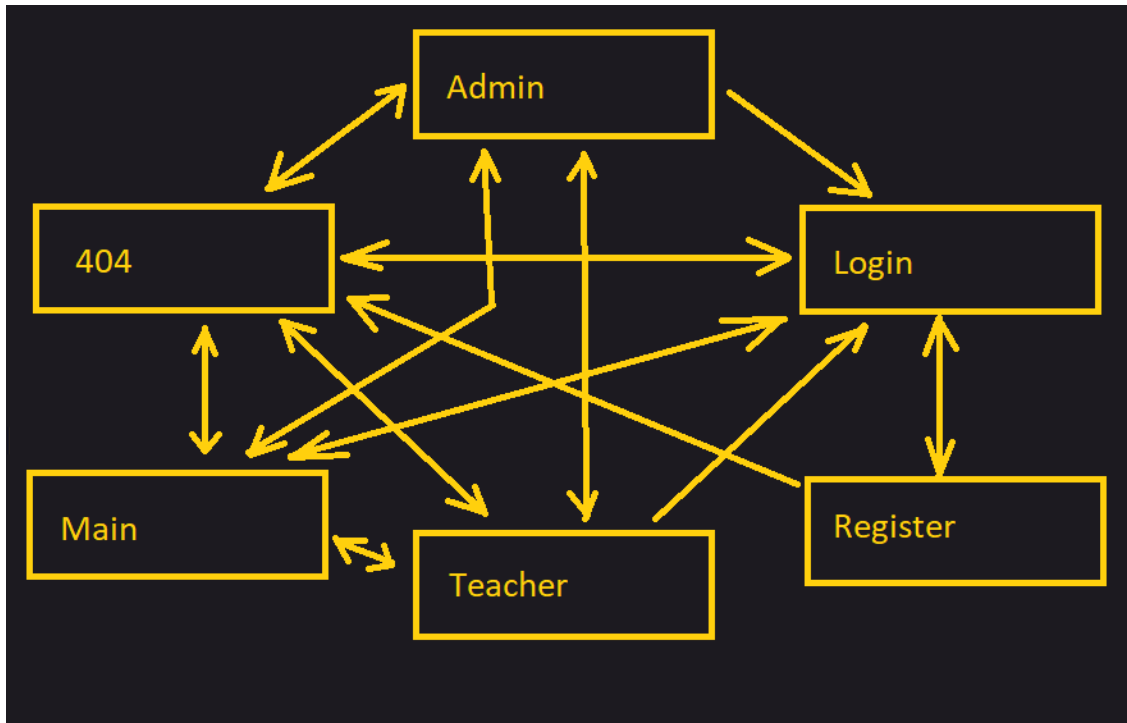
- **app/Models/User.php**
  - ✓ Klasa User
  - ✓ Inicijalizacija i rad sa tabelom user
- **app/Views/fixed/footer.php**
  - ✓ Dinamički prikaz linkova i copyright
- **app/Views/fixed/header.php**
  - ✓ Head tag
  - ✓ Logo
  - ✓ Dinamički prikaz log in/log out dugmića
- **app/Views/pages/404.php**
  - ✓ Error stranica
- **app/Views/pages/admin.php**
  - ✓ Prikaz admin stranice
  - ✓ Funkcionalnosti vezane samo za admin naloge
- **app/Views/pages/login.php**
  - ✓ Prikaz login stranice
  - ✓ Logovanje korisnika
- **app/Views/pages/main.php**
  - ✓ Glavna stranica korisnika
  - ✓ Prikaz, dodavanje i izmena podataka iz šoping liste
  - ✓ Prikaz, dodavanje i izmena radnji u kojima korisnik kupuje
- **app/Views/pages/register.php**
  - ✓ Prikaz register stranice
  - ✓ Registracija korisnika
- **app/Views/pages/teacher.php**
  - ✓ Prikaz stranice o autoru

## 1.3 Izgled stranica



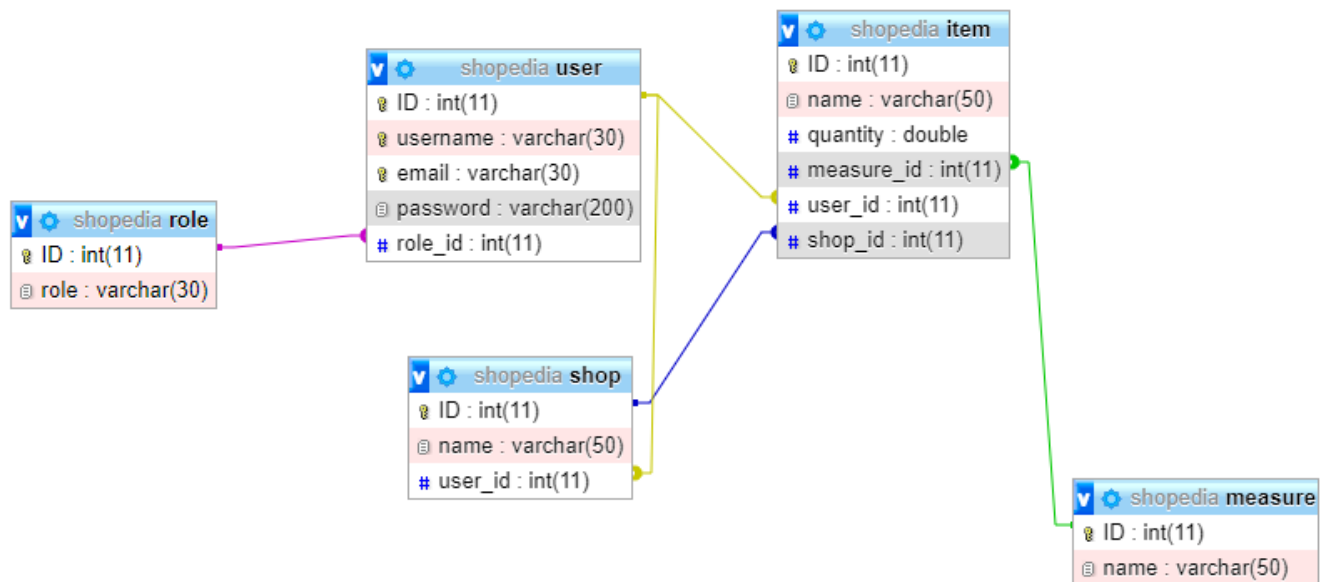
## 2. ORGANIZACIJA

### 2.1 Organizacija sajta





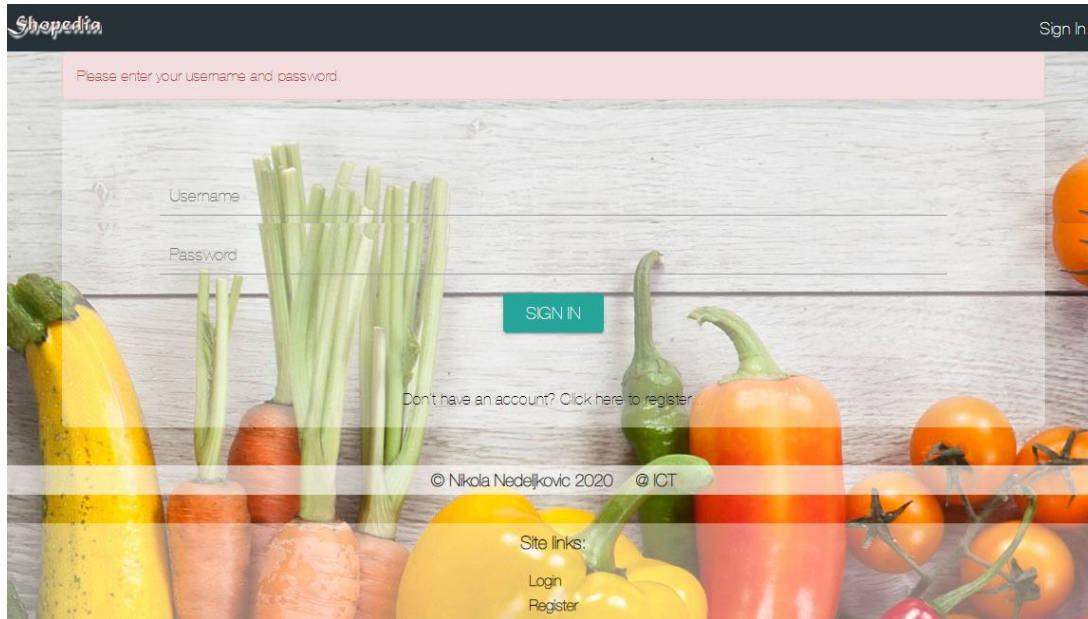
## 2.2 Organizacija baze



## 2.3 Sitemap.xml

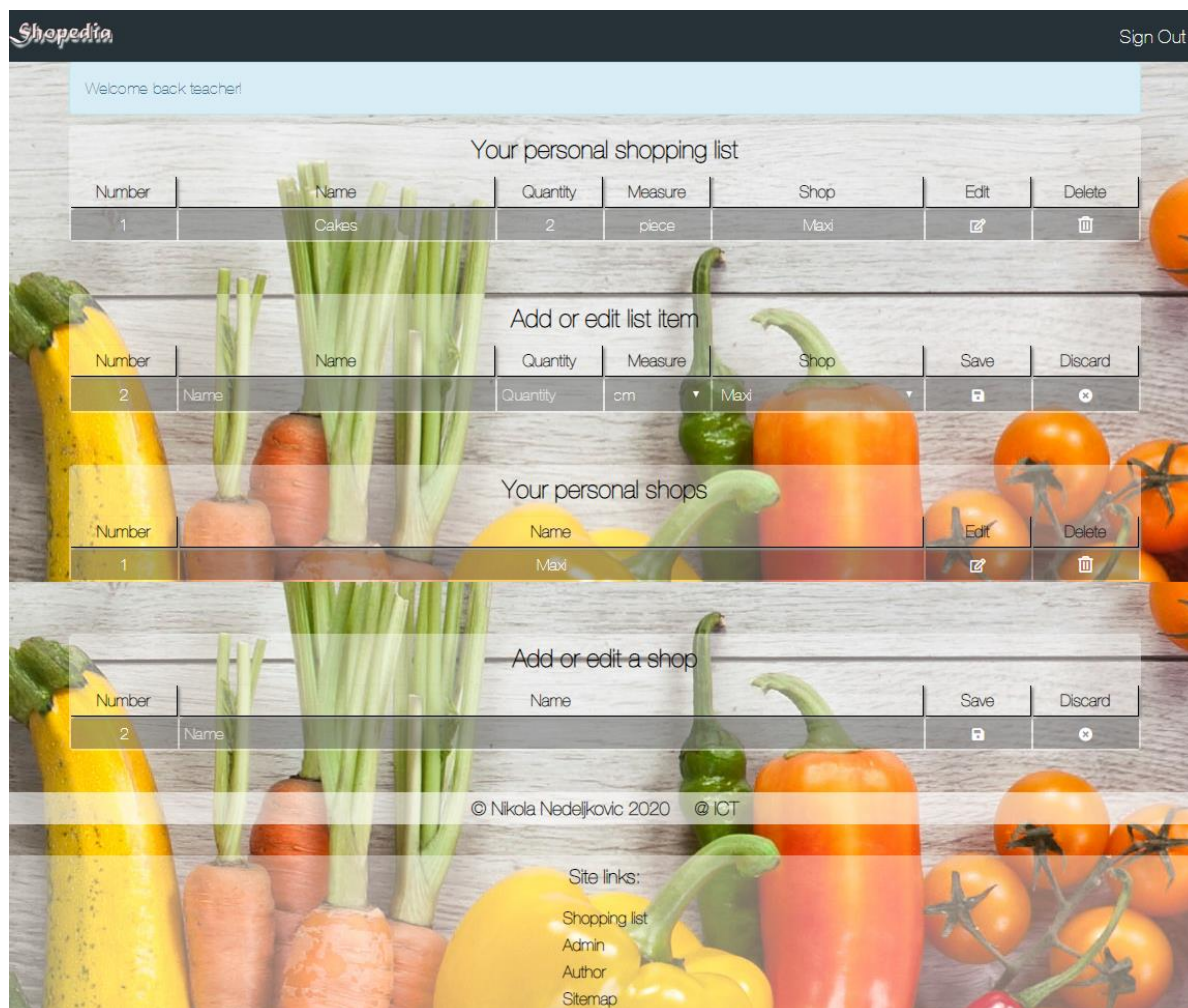
```
<?xml version="1.0"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
  <loc>https://shopedia-belgrade.000webhostapp.com/index.php</loc>
  <lastmod>2020-02-11</lastmod>
  <changefreq>daily</changefreq>
  <priority>1</priority>
</url>
</urlset>
```

## 2.4 Slike stranica sa funkcionalnostima



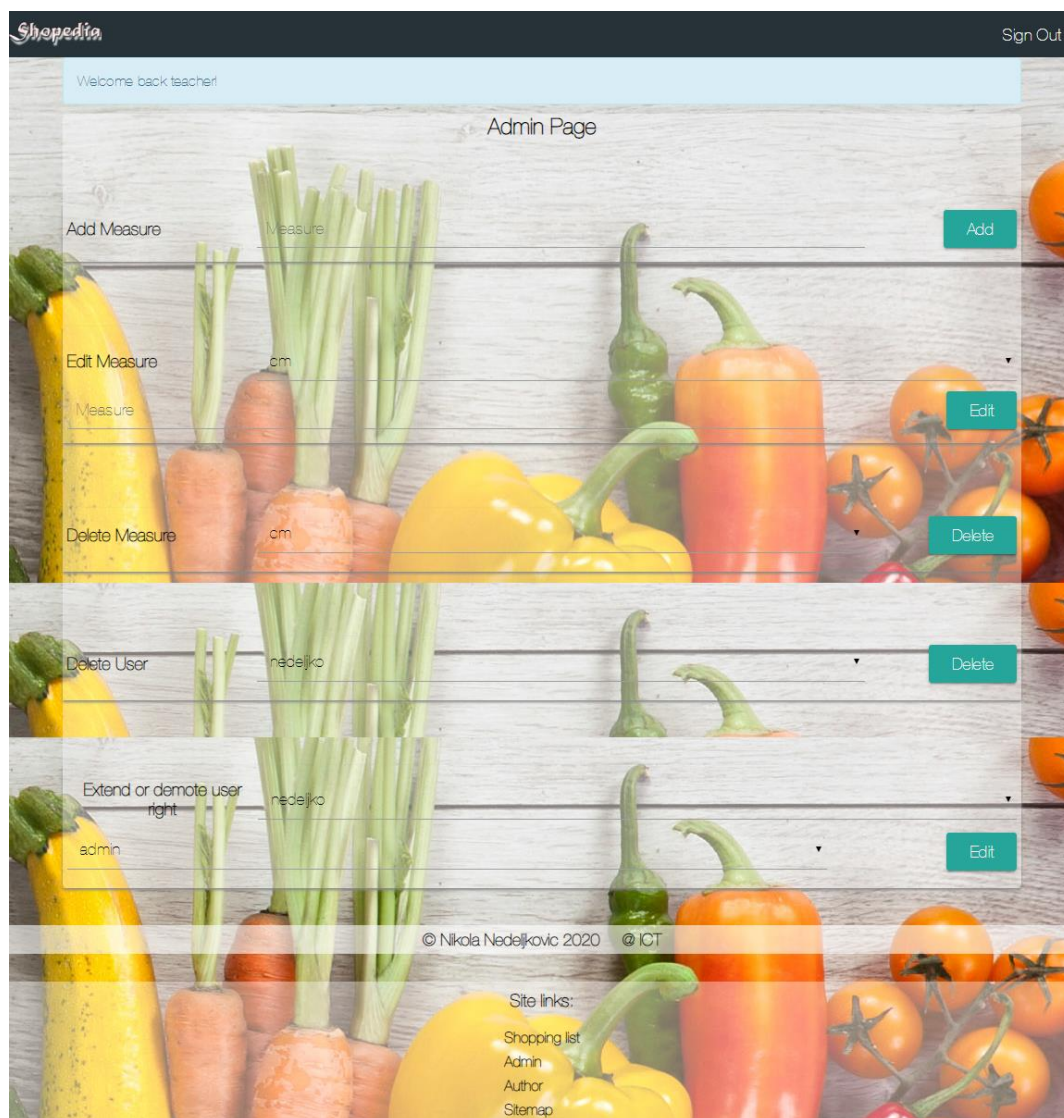
login.php

- Ispisivanje login forme, korisnička provera unetih podataka, prosleđivanje na kontroler i dalja obrada toga. Redirekcija na osnovu dobijenih parametara. Sajtu se ne može pristupiti bez prvog logovanja.



main.php

- Dinamičko ispisivanje podataka i korisnička obrada novih i starih fajlova pred slanje na kontroler



admin.php

- Dinamičko ispisivanje podataka i korisnička obrada novih i starih fajlova pred slanje na kontroler. Kontrole kojima samo admin može pristupiti.



Shopedia

Sign In

Please provide the information to create an account.

Username

Password

Email

REGISTER

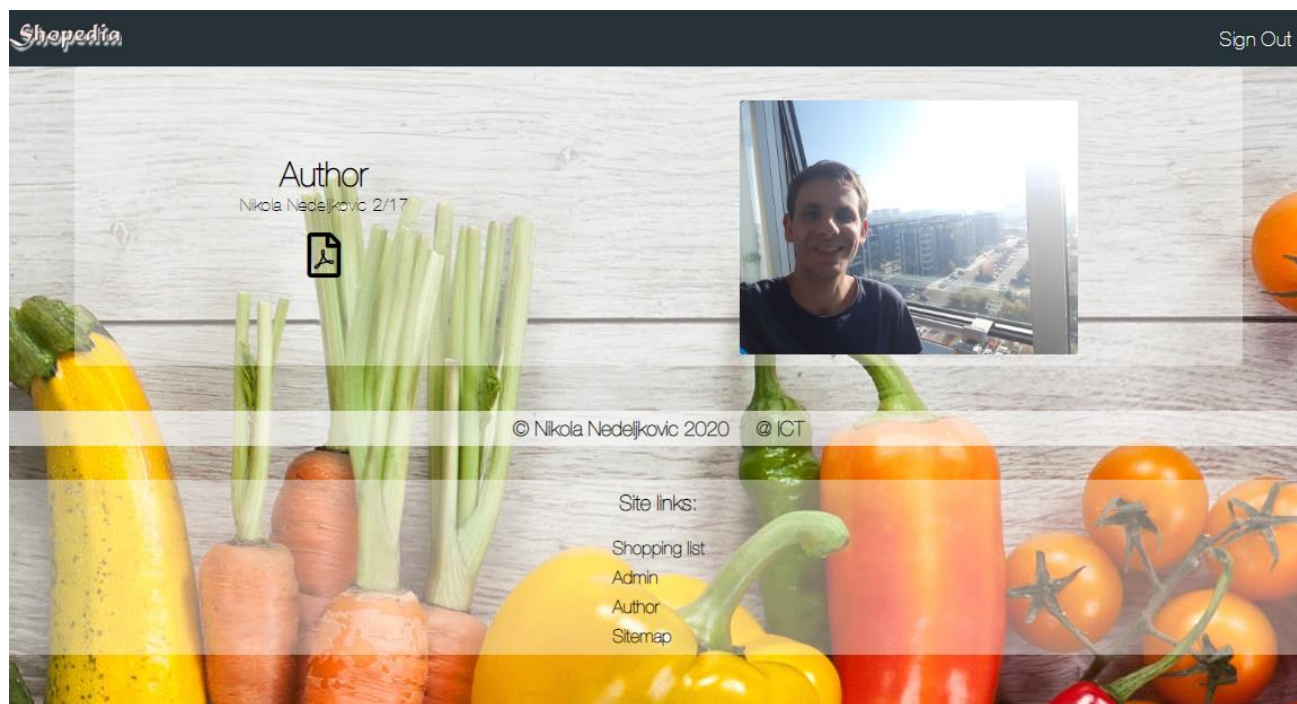
Already have an account? Click here to sign in.

© Nikola Nedeljkovic 2020 @ ICT

Site links:  
Login  
Register

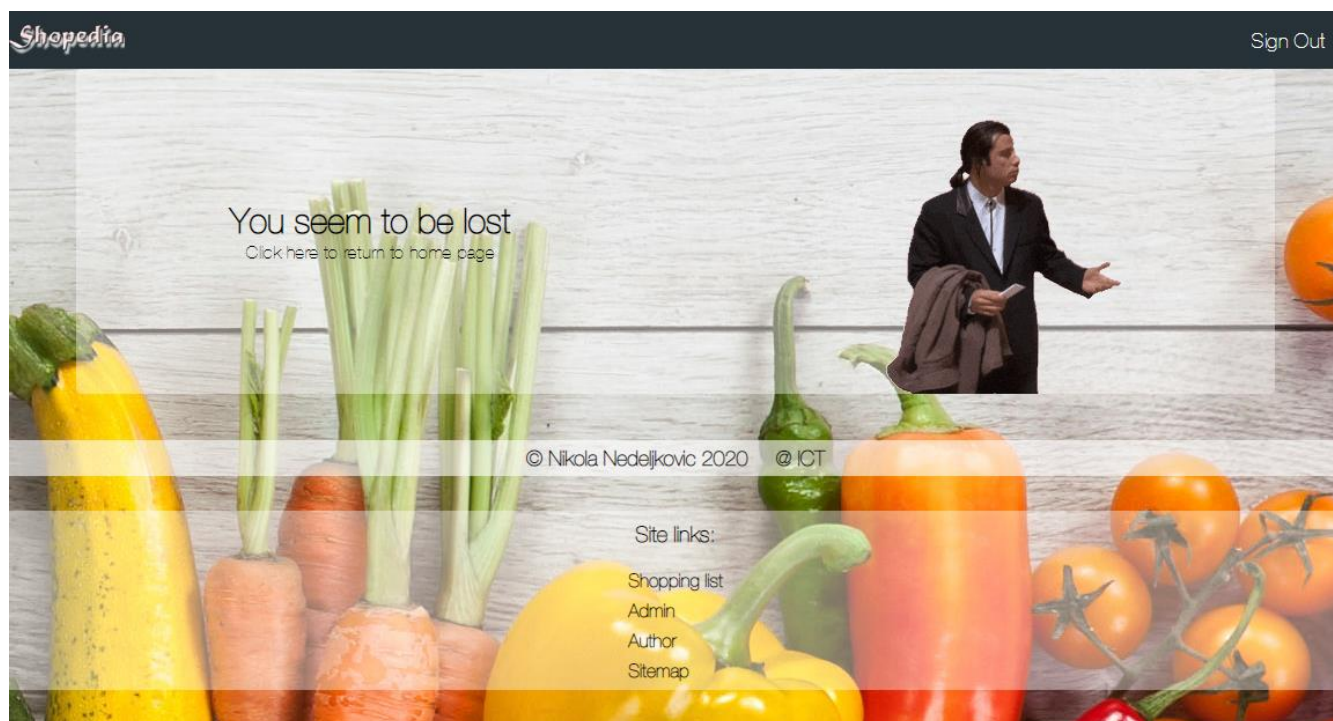
register.php

- Ispisivanje forme za registraciju, korisnička provera unetih podataka, prosleđivanje na kontroler i dalja obrada toga. Redirekcija na osnovu dobijenih parametara.



teacher.php

- Stranica o autoru.



404.php

- 404 stranica za greške.



## 3. KODOVI

### 3.1 index.php

```
<?php
    session_start();
include("app/Views/fixed/header.php");
    if (isset($_GET["page"])) {
        switch($_GET["page"]) {
            case "login":
                if(isset($_SESSION['user'])) {
                    include "app/Views/pages/main.php";
                } else {
                    include "app/Views/pages/login.php";
                }
                break;
            case "register":
                if(isset($_SESSION['user'])) {
                    include "app/Views/pages/main.php";
                } else {
                    include "app/Views/pages/register.php";
                }
                break;
            case "main":
                include "app/Views/pages/main.php";
                break;
            case "admin":
                if(isset($_SESSION['user'])) {
                    if ((($_SESSION['user']->role_id != 1) &&
($SESSION['user']->role_id != 3)) {
                        include "app/Views/pages/main.php";
                    } else {
                        include "app/Views/pages/admin.php";
                    }
                } else {
                    include "app/Views/pages/login.php";
                }
                break;
            case "teacher":
                if(isset($_SESSION['user'])) {
                    if ($_SESSION['user']->role_id != 3) {
                        include "app/Views/pages/main.php";
                    } else {
                        include "app/Views/pages/teacher.php";
                    }
                } else {
                    include "app/Views/pages/login.php";
                }
                break;
            default:
                include "app/Views/pages/404.php";
                break;
        }
    } else {
        if(!isset($_SESSION['user'])) {
```

```

        include "app/Views/pages/login.php";
    } else {
        include "app/Views/pages/main.php";
    }
}
include("app/Views/fixed/footer.php");
?>

```

### 3.2. api.php → komunikacija između index i kontrolera

```

<?php
require_once "app/Config/autoload.php";
require "app/Config/config.php";
use App\Controllers\LoginController;
use App\Controllers\RegisterController;
use App\Controllers\ShopController;
use App\Controllers\UserController;
use App\Controllers\MeasureController;
use App\Controllers\ItemController;
$login = new LoginController();
$register = new RegisterController();
$shopcont = new ShopController();
$usercont = new UserController();
$measurecont = new MeasureController();
$itemcont = new ItemController();
session_start();

function logFunction ($content) {
    $log path = $ SERVER["DOCUMENT_ROOT"] . "/data/log.txt";
    $open = fopen($log path, "a");
    if ($open && isset($ SESSION["user"])) {
        $date = date('d-m-Y H:i:s');
        fwrite($open, "{$content}\t{$date}\t{$_SESSION['user']}-
>username}\t\n");
        fclose($open);
    }
}

logFunction($_GET['page']);

switch ($_GET['page']) {
    case 'login':
        $login->login($_POST);
        break;
    case 'logout':
        if(isset($_SESSION['user'])) {
            unset($_SESSION['user']);
            session_destroy();
        }
        header("Location: index.php?page=login");
        die();
        break;
    case 'register':
        $register->register($_POST);
        break;
    case 'shop_edit':
        if ($_POST["shop_add_edit_id"] != "") {

```

```

        $shopcont->updateShop($_POST);
    } else {
        $shopcont->addShop($_POST, $_SESSION['user']->ID);
    }
    break;
case 'shop_delete':
    $shopcont->deleteShop($_GET['id']);
    break;
case 'item_edit':
    if ($_POST["item_add_edit_id"] != "") {
        $itemcont->updateItem($_POST, $_SESSION['user']->ID);
    } else {
        $itemcont->addItem($_POST, $_SESSION['user']->ID);
    }
    break;
case 'item_delete':
    $itemcont->deleteItem($_GET['id']);
    break;
case 'add_measure':
    $measurecont->addMeasure($_POST);
    break;
case 'edit_measure':
    $measurecont->updateMeasure($_POST);
    break;
case 'delete_measure':
    $measurecont->deleteMeasure($_POST);
    break;
case 'edit_user':
    $usercont->editUserRight($_POST);
    break;
case 'delete_user':
    $usercont->deleteUser($_POST);
    break;
default:
    header("Location: index.php?page=main");
    die();
    break;
}
?>

```

### 3.3. data.php → dohvaćanje podataka za prikaz

```

<?php
require_once "app/Config/autoload.php";
require "app/Config/config.php";
use App\Controllers\ShopController;
use App\Controllers\RoleController;
use App\Controllers\UserController;
use App\Controllers\MeasureController;
use App\Controllers\ItemController;
$shopcont = new ShopController();
$rolecont = new RoleController();
$usercont = new UserController();
$measurecont = new MeasureController();
$itemcont = new ItemController();

$item_data = $itemcont->getItems($_SESSION['user']->ID);

```

```

    $measure_data = $measurecont->getMeasures();
    $shop_data = $shopcont->getShops($_SESSION['user']->ID);
    $user_data = $usercont->getUsers($_SESSION['user']->ID);
    $role_data = $rolecont->getRoles();
?>

```

### 3.4 autoload.php

```

<?php
// AUTOLOAD
spl_autoload_register(function ($path) {
    $path = str_replace("\\", DIRECTORY_SEPARATOR, $path);
    $path[0] = strtolower($path[0]);
    $path .= ".php";
    require_once $path;
});

```

### 3.5. DATABASE.php → SQL upiti i kreiranje baze

```

<?php
namespace App\Config;

class DATABASE {

    private $connection;
    private static $database;

    private function __construct() {
        $this->connect();
    }

    public static function instance() {
        if(self::$database === null){
            self::$database = new DATABASE();
        }
        return self::$database;
    }

    private function connect() {
        try {
            $this->connection = new
            \PDO("mysql:host=".SERVER.";dbname=".DBNAME.";charset=utf8", USERNAME,
            PASSWORD);
            $this->connection->setAttribute(\PDO::ATTR_DEFAULT_FETCH_MODE,
            \PDO::FETCH_OBJ);
            $this->connection->setAttribute(\PDO::ATTR_ERRMODE,
            \PDO::ERRMODE_EXCEPTION);
        }
        catch(PDOException $ex) {
            echo $ex->getMessage();
        }
    }

    public function executeQuery(string $query) {
        return $this->connection->query($query)->fetchAll();
    }
}

```

```

    public function executeQueryAll(string $query, Array $params) {
        $prepare = $this->connection->prepare($query);
        $prepare->execute($params);
        return $prepare->fetchAll();
    }

    public function executeQueryOneRow(string $query, Array $params) {
        $prepare = $this->connection->prepare($query);
        $prepare->execute($params);
        return $prepare->fetch();
    }

    public function executeInsert(string $query, Array $params) {
        $prepare = $this->connection->prepare($query);
        $prepare->execute($params);
    }

    public function executeUpdate(string $query, Array $params) {
        $prepare = $this->connection->prepare($query);
        $prepare->execute($params);
    }

    public function executeDelete(string $query, Array $params) {
        $prepare = $this->connection->prepare($query);
        $prepare->execute($params);
    }
}

```

### 3.6 ItemController.php

```

<?php
namespace App\Controllers;
use App\Models\Item;
use App\Config\DATABASE;

class ItemController {
    private $model;

    public function __construct() {
        $this->model = new Item(DATABASE::instance());
    }

    public function getItems($user_id) {
        return $this->model->getItems($user_id);
    }

    public function addItem($request, $user_id) {
        if(isset($request['item_add_edit_submit'])) {
            $name = $request['item_add_edit_name'];
            $quantity = $request['item_add_edit_quantity'];
            $measure = $request['item_add_edit_measure'];
            $shop = $request['item_add_edit_shop'];

            if ($name != "" && $quantity != "" && $measure != "" && $shop
            != "") {
                $this->model->addItem($name, $quantity, $measure,
                $user_id, $shop);
            }
        }
    }
}

```

```

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

public function updateItem($request) {
    if(isset($request['item_add_edit_submit'])) {
        $name = $request['item_add_edit_name'];
        $id = $request['item_add_edit_id'];
        $quantity = $request['item_add_edit_quantity'];
        $measure = $request['item_add_edit_measure'];
        $shop = $request['item_add_edit_shop'];

        if ($name != "" && $quantity != "" && $measure != "" && $shop
!= "") {
            $this->model->updateItem($name, $quantity, $measure,
$shop, $id);
        }
    }
    header("Location: index.php?page=main");
}

public function deleteItem($id) {
    $this->model->deleteItem($id);
    header("Location: index.php?page=main");
}
}

```

### 3.7 LoginController.php

```

<?php
namespace App\Controllers;
use App\Models\User;
use App\Config\DATABASE;

class LoginController {
    private $model;

    public function __construct() {
        $this->model = new User(DATABASE::instance());
    }

    public function login($request) {
        if(isset($request['signin'])) {
            $username = $request['username'];
            $password = $request['password'];

            $user = $this->model->getUser($username, $password);

            if($user) {
                $_SESSION['user'] = $user;
            } else {
                $_SESSION['l_error'] ="User not registered or wrong
credentials!";
            }
            header("Location: index.php?page=login");
        } else {

```

```

        \http_response_code(404);
    }
}
}

```

### 3.8. MeasureController.php

```

<?php
namespace App\Controllers;
use App\Models\Measure;
use App\Config\DATABASE;

class MeasureController {
    private $model;

    public function __construct() {
        $this->model = new Measure(DATABASE::instance());
    }

    public function getMeasures() {
        return $this->model->getMeasures();
    }

    public function addMeasure($request) {
        if(isset($request['submit_add_measure'])) {
            $name = $request['add_measure'];

            if ($name != "") {
                $this->model->addMeasure($name);
            }
        }
        header("Location: index.php?page=admin");
    }

    public function updateMeasure($request) {
        if(isset($request['submit_edit_measure'])) {
            $name = $request['edit_measure'];
            $id = $request['edit_measure_id'];

            if ($name != "") {
                $this->model->updateMeasure($name, $id);
            }
        }
        header("Location: index.php?page=admin");
    }

    public function deleteMeasure($request) {
        if(isset($request['submit_delete_measure'])) {
            $id = $request['measure_delete_select'];

            if ($id != "") {
                $this->model->deleteMeasure($id);
            }
        }
        header("Location: index.php?page=admin");
    }
}

```

### 3.9 RegisterController.php

```
<?php
namespace App\Controllers;

use App\Models\User;
use App\Config\DATABASE;

class RegisterController {
    private $model;

    public function __construct() {
        $this->model = new User(DATABASE::instance());
    }

    public function register($request) {
        if(isset($request['register'])) {
            $username = $request['username'];
            $password = $request['password'];
            $email = $request['email'];

            if(!filter_var($email, FILTER_VALIDATE_EMAIL)) {
                $_SESSION['r_error'] = "Email not valid!";
            }
            else {
                $user = $this->model->existsUser($username, $email);

                if($user) {
                    $_SESSION['r_error'] = "Email or username already in use!";
                } else {
                    $this->model->createUser($username, $email, $password);
                    $_SESSION['r_error'] = "User created successfully, you can now login with your credentials!";
                }
            }
            header("Location: index.php?page=register");
        } else {
            \http_response_code(404);
        }
    }
}
```

### 3.10 RoleController.php

```
<?php
namespace App\Controllers;

use App\Models\Role;
use App\Config\DATABASE;

class RoleController {
    private $model;

    public function __construct() {
        $this->model = new Role(DATABASE::instance());
    }
}
```



```
        public function getRoles() {  
            return $this->model->getRoles();  
        }  
    }  
}
```

### 3.11. ShopController.php

```
<?php
```

```
namespace App\Controllers;  
use App\Models\Shop;  
use App\Config\DATABASE;  
  
class ShopController {  
    private $model;  
  
    public function __construct() {  
        $this->model = new Shop(DATABASE::instance());  
    }  
  
    public function getShops($user_id) {  
        return $this->model->getShops($user_id);  
    }  
  
    public function addShop($request, $user_id) {  
        if(isset($request['shop_add_edit_submit'])) {  
            $name = $request['shop_add_edit_name'];  
  
            if ($name != "") {  
                $this->model->addShop($name, $user_id);  
            }  
            header("Location: index.php?page=main");  
        }  
  
    public function updateShop($request) {  
        if(isset($request['shop_add_edit_submit'])) {  
            $name = $request['shop_add_edit_name'];  
            $id = $request['shop_add_edit_id'];  
  
            if ($name != "") {  
                $this->model->updateShop($name, $id);  
            }  
            header("Location: index.php?page=main");  
        }  
  
    public function deleteShop($id) {  
        $this->model->deleteShop($id);  
        header("Location: index.php?page=main");  
    }  
}
```

## 3.12 UserController.php

```
<?php
namespace App\Controllers;
use App\Models\User;
use App\Config\DATABASE;

class UserController {
    private $model;

    public function __construct() {
        $this->model = new User(DATABASE::instance());
    }

    public function getUsers($id) {
        return $this->model->getUsers($id);
    }

    public function editUserRight($request) {
        if(isset($request['submit_edit_user_id'])) {
            $user = $request['edit_user_right_id'];
            $role = $request['user_right_select'];

            if ($user != "" && $role != "") {
                $this->model->editUserRight($role, $user);
            }
        }
        header("Location: index.php?page=admin");
    }

    public function deleteUser($request) {
        if(isset($request['submit_delete_user'])) {
            $id = $request['user_delete_select'];

            if ($id != "") {
                $this->model->deleteUser($id);
            }
        }
        header("Location: index.php?page=admin");
    }
}
```

## 3.13. Item.php

```
<?php
namespace App\Models;
use App\Config\DATABASE;

class Item {
    private $database;

    public function __construct(DATABASE $database) {
        $this->database = $database;
    }

    public function getItems($user_id) {
```

```

        return $this->database->executeQueryAll("SELECT i.ID as item_id,
i.name AS name, i.quantity AS quantity, m.name AS measure, s.name AS shop,
m.ID AS measure_id, s.ID AS shop_id FROM item i INNER JOIN measure m ON
i.measure_id = m.ID INNER JOIN shop s ON i.shop_id = s.ID WHERE i.user_id =
?", [$user_id]);
    }

    public function addItem($name, $quantity, $measure, $user_id, $shop)
    {
        $this->database->executeInsert("INSERT INTO item (name, quantity,
measure_id, user_id, shop_id) VALUES (?, ?, ?, ?, ?)", [$name, $quantity,
$measure, $user_id, $shop]);
    }

    public function updateItem($name, $quantity, $measure, $shop, $id) {
        $this->database->executeUpdate("UPDATE item SET name = ?,
quantity = ?, measure_id = ?, shop_id = ? WHERE ID = ?", [$name, $quantity,
$measure, $shop, $id]);
    }

    public function deleteItem($id) {
        $this->database->executeDelete("DELETE FROM item WHERE ID = ?",
[$id]);
    }
}

```

### 3.14. Measure.php

```

<?php
namespace App\Models;
use App\Config\DATABASE;

class Measure {
    private $database;

    public function __construct(DATABASE $database) {
        $this->database = $database;
    }

    public function getMeasures() {
        return $this->database->executeQuery("SELECT * FROM measure");
    }

    public function addMeasure($name) {
        $this->database->executeInsert("INSERT INTO measure (name) VALUES
(?)", [$name]);
    }

    public function updateMeasure($name, $id) {
        $this->database->executeUpdate("UPDATE measure SET name = ? WHERE
ID = ?", [$name, $id]);
    }

    public function deleteMeasure($id) {
        $this->database->executeDelete("DELETE FROM measure WHERE ID =
?", [$id]);
    }
}

```

```
}
```

### 3.15. Role.php

```
<?php
namespace App\Models;
use App\Config\DATABASE;

class Role {
    private $database;

    public function __construct(DATABASE $database) {
        $this->database = $database;
    }

    public function getRoles() {
        return $this->database->executeQuery("SELECT * FROM role");
    }
}
```

### 3.16. Shop.php

```
<?php
namespace App\Models;
use App\Config\DATABASE;

class Shop {
    private $database;

    public function __construct(DATABASE $database) {
        $this->database = $database;
    }

    public function getShops($user_id) {
        return $this->database->executeQueryAll("SELECT ID, name FROM
shop WHERE user_id = ?", [$user_id]);
    }

    public function addShop($name, $user_id) {
        $this->database->executeInsert("INSERT INTO shop (name, user_id)
VALUES (?, ?)", [$name, $user_id]);
    }

    public function updateShop($name, $id) {
        $this->database->executeUpdate("UPDATE shop SET name = ? WHERE ID
= ?", [$name, $id]);
    }

    public function deleteShop($id) {
        $this->database->executeDelete("DELETE FROM shop WHERE ID = ?",
[$id]);
    }
}
```

### 3.17. User.php

```
<?php
```

```
namespace App\Models;
use App\Config\DATABASE;

class User {
    private $database;

    public function __construct(DATABASE $database) {
        $this->database = $database;
    }

    public function getUsers($id) {
        return $this->database->executeQueryAll("SELECT ID, username,
role_id FROM user WHERE ID != ?", [$id]);
    }

    public function getUser($username, $password) {
        return $this->database->executeQueryOneRow("SELECT ID, username,
email, role_id FROM user WHERE username = ? AND password=MD5(?)", [$username,
$password]);
    }

    public function existsUser($username, $email) {
        return $this->database->executeQueryOneRow("SELECT username,
email, role_id FROM user WHERE username = ? OR email = ?", [$username,
$email]);
    }

    public function createUser($username, $email, $password) {
        $this->database->executeInsert("INSERT INTO user(username, email,
password, role_id) VALUES (?, ?, MD5(?), 2)", [$username, $email,
$password]);
    }

    public function editUserRight($role, $user) {
        $this->database->executeUpdate("UPDATE user SET role_id = ? WHERE
ID = ?", [$role, $user]);
    }

    public function deleteUser($id) {
        $this->database->executeDelete("DELETE FROM user WHERE ID = ?",
[$id]);
    }
}
```

### 3.18. footer.php

```
<div class="footer_panel">
  <p><a href="https://www.linkedin.com/in/nikola-nedeljkovic-8110b5150">&copy; Nikola Nedeljkovic 2020 &nbsp;&nbsp;&nbsp; @ ICT</a></p>
  <div class="site_links">
    <h3>Site links:</h3>
    <ul>
      <?php
        if(isset($_SESSION['user'])) {
          echo "<li><a
href=\"index.php?page=main\">Shopping list</a></li>";
```

```

        if (($SESSION['user']->role_id == 1) ||
($ SESSION['user']->role id == 3)) {
            echo "<li><a
href=\"index.php?page=admin\">Admin</a></li>";
        }
        if ($SESSION['user']->role_id == 3) {
            echo "<li><a
href=\"index.php?page=teacher\">Author</a></li>";
            echo "<li><a
href=\"sitemap.xml\">Sitemap</a></li>";
        }
    } else {
        echo "<li><a
href=\"index.php?page=login\">Login</a></li>";
        echo "<li><a
href=\"index.php?page=register\">Register</a></li>";
    }
    ?>
</ul>
</div>
</div>
</body>
<script type="text/javascript" src="assets/js/main.js"></script>
</html>

```

### 3.19. header.php

```

<!DOCTYPE html>
<html>
    <head>
        <title>Shopedia</title>
        <meta name="viewport" content="width=device-width, initial-
scale=1.0"/>
        <meta name="author" content="Nikola Nedeljkovic"/>
        <meta name="copyright" content="RarOks 1008 Entertainment @ 2020"/>
        <meta name="robots" content="index,follow"/>
        <meta name="keywords" content="shopping, list, grocery"/>
        <meta name="description" content="Here you can add your own shopping
list and use it from all devices."/>
        <meta name="abstract" content="Shop with Shopedia."/>
        <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"/>
    </head>
    <body>
        <div class="header_holder">
            <a href="index.php?page=main">
                <h1>Shopedia</h1>
            </a>
            <?php if(isset($SESSION["user"])) : ?>
                <a href="api.php?page=logout"
class="login_logout_button">Sign Out</a>
            <?php else : ?>

```

```

        <a href="index.php?page=login"
class="login_logout_button">Sign In</a>
        <?php endif; ?>
    </div>

```

### 3.20. 404.php

```

<div class="fourofour">
    <div class="lost_text">
        <h2>You seem to be lost</h2>
        <a href="index.php?page=main">Click here to return to home page</a>
    </div>
    
</div>

```

### 3.21. admin.php

```

<?php include_once "data.php"; ?>
<p id="welcome_sign">Welcome back <?php echo $_SESSION['user']->username;
?>!</p>
<div class="admin_panel">
    <h2>Admin Page</h2>
    <div class="admin_item">
        <h3>Add Measure</h3>
        <form onsubmit="return add_measure_check()" method="POST"
action="api.php?page=add_measure">
            <input type="text" name="add_measure" id="add_measure"
Placeholder="Measure"/>
            <button type="submit" name="submit_add_measure">Add</button>
        </form>
    </div>
    <div class="admin_item with_more">
        <div class="more_admin_options">
            <h3>Edit Measure</h3>
            <select id="measure_edit_select"
onchange="admin_measure_change()">
                <?php foreach($measure_data as $measure): ?>
                    <option value="<?=$measure->ID;?>"><?=$measure->
>name;?></option>
                <?php endforeach; ?>
            </select>
        </div>
        <form class="more_admin_options" onsubmit="return
edit_measure_check()" method="POST" action="api.php?page=edit_measure">
            <input type="hidden" name="edit_measure_id"
id="edit_measure_id"/>
            <input type="text" name="edit_measure" id="edit_measure"
Placeholder="Measure"/>
            <button type="submit" name="submit_edit_measure">Edit</button>
        </form>
    </div>
    <div class="admin_item">
        <h3>Delete Measure</h3>
        <form onsubmit="return delete_measure_check()" method="POST"
action="api.php?page=delete_measure">
            <select id="measure_delete_select" name="measure_delete_select">
                <?php foreach($measure_data as $measure): ?>

```

```

        <option value="<?=$measure->ID;?>"><?=$measure-
>name;?></option>
        <?php endforeach; ?>
    </select>
    <button type="submit"
name="submit_delete_measure">Delete</button>
    </form>
</div>
<div class="admin_item">
    <h3>Delete User</h3>
    <form method="POST" action="api.php?page=delete_user">
        <select id="user_delete_select" name="user_delete_select">
            <?php foreach($user_data as $user): ?>
                <option value="<?=$user->ID;?>"><?=$user-
>username;?></option>
            <?php endforeach; ?>
        </select>
        <button type="submit" name="submit_delete_user">Delete</button>
    </form>
</div>
<div class="admin_item with_more">
    <div class="more_admin_options">
        <h3>Extend or demote user right</h3>
        <select id="user_change_select" onchange="admin_user_change()">
            <?php foreach($user_data as $user): ?>
                <option value="<?=$user->ID;?>" data-id="<?=$user-
>role_id;?>"><?=$user->username;?></option>
            <?php endforeach; ?>
        </select>
    </div>
    <form class="more_admin_options" onsubmit="return edit_user_check()"
method="POST" action="api.php?page=edit_user">
        <input type="hidden" name="edit_user_right_id"
id="edit_user_right_id"/>
        <select id="user_right_select" name="user_right_select">
            <?php foreach($role_data as $role): ?>
                <option value="<?=$role->ID;?>"><?=$role-
>role;?></option>
            <?php endforeach; ?>
        </select>
        <button type="submit" name="submit_edit_user_id">Edit</button>
    </div>
</div>
</div>

```

### 3.22. login.php

```

<p id="login_error"><?php
    if(isset($_SESSION['l_error'])) {
        echo $_SESSION['l_error'];
    } else {
        echo "Please enter your username and password.";
    } ?></p>
<div class="login_panel">
    <form onsubmit="return login_check()" method="POST"
action="api.php?page=login">

```



```

        <input type="text" name="username" id="username"
placeholder="Username"/>
        <input type="password" name="password" id="password"
placeholder="Password"/>
        <button type="submit" name="signin" id="signin">SIGN IN</button>
    </form>
    <a href="index.php?page=register">Don't have an account? Click here to
register...</a>
</div>

```

### 3.23. main.php

```

<?php include_once "data.php"; ?>
<p id="welcome_sign">Welcome back <?php echo $_SESSION['user']->username;
?>!</p>
<div class="main_panel">
    <h2>Your personal shopping list</h2>
    <table>
        <tr>
            <th>Number</th>
            <th>Name</th>
            <th>Quantity</th>
            <th>Measure</th>
            <th>Shop</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>
        <?php $item_num = 1; foreach($item_data as $item): ?>
            <tr>
                <td id="itemnum<?=$item_num;?>"><?=$item_num;?></td>
                <td id="itemname<?=$item_num;?>"><?=$item->name;?></td>
                <td id="itemquantity<?=$item_num;?>"><?=$item->quantity;?></td>
                <td id="itemmeasure<?=$item_num;?>" data-id="<?=$item-
>measure_id;?>"><?=$item->measure;?></td>
                <td id="itemshop<?=$item_num;?>" data-id="<?=$item-
>shop_id;?>"><?=$item->shop;?></td>
                <td><button onclick="item_edit_click(<?=$item_num;?>, <?=$item-
>item_id;?>)"><span class="far fa-edit"></span></button></td>
                <td><a href="api.php?page=item_delete&id=<?=$item-
>item_id;?>"><span class="far fa-trash-alt"></span></a></td>
            </tr>
            <?php $item_num++; endforeach; ?>
        </table>
    </div>
    <div class="main_panel not_first">
        <h2>Add or edit list item</h2>
        <table>
            <tr>
                <th>Number</th>
                <th>Name</th>
                <th>Quantity</th>
                <th>Measure</th>
                <th>Shop</th>
                <th>Save</th>
                <th>Discard</th>
            </tr>
            <tr>

```

```

        <form onsubmit="return add_edit_item_check()" method="POST"
action="api.php?page=item_edit">
        <input type="hidden" name="item_add_edit_id"
id="item_add_edit_id" value="" />
        <td id="item_add_edit_number" data-
id="<?=$item_num;?>"><?=$item_num;?></td>
        <td><input type="text" name="item_add_edit_name"
id="item_add_edit_name" placeholder="Name" /></td>
        <td><input type="number" name="item_add_edit_quantity"
id="item_add_edit_quantity" placeholder="Quantity" min="1" /></td>
        <td><select name="item_add_edit_measure"
id="item_add_edit_measure">
                <?php foreach($measure_data as $measure): ?>
                <option value="<?=$measure->ID;?>"><?=$measure-
>name;?></option>
                <?php endforeach; ?>
        </select></td>
        <td><select name="item_add_edit_shop"
id="item_add_edit_shop">
                <?php foreach($shop_data as $shops): ?>
                <option value="<?=$shops->ID;?>"><?=$shops-
>name;?></option>
                <?php endforeach; ?>
        </select></td>
        <td><button type="submit" name="item_add_edit_submit"><span
class="fas fa-save"></span></button></td>
    </form>
    <td><button onclick="item_edit_discard()"><span class="fas fa-
times-circle"></span></button></td>
</tr>
</table>
</div>
<div class="main_panel not_first your_shops">
    <h2>Your personal shops</h2>
    <table>
        <tr>
            <th>Number</th>
            <th>Name</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>
        <?php $shop_num = 1; foreach($shop_data as $shop): ?>
        <tr>
            <td id="shopnum<?=$shop_num;?>"><?=$shop_num;?></td>
            <td id="shopname<?=$shop_num;?>"><?=$shop->name;?></td>
            <td><button onclick="shop_edit_click(<?=$shop_num;?>, <?=$shop-
>ID;?>)"><span class="far fa-edit"></span></button></td>
            <td><a href="api.php?page=shop_delete&id=<?=$shop->ID;?>"><span
class="far fa-trash-alt"></span></a></td>
        </tr>
        <?php $shop_num++; endforeach; ?>
    </table>
</div>
<div class="main_panel not_first your_shops">
    <h2>Add or edit a shop</h2>
    <table>
        <tr>

```

```

        <th>Number</th>
        <th>Name</th>
        <th>Save</th>
        <th>Discard</th>
    </tr>
    <tr>
        <form onsubmit="return add_edit_shop_check()" method="POST"
action="api.php?page=shop_edit">
            <input type="hidden" name="shop_add_edit_id"
id="shop_add_edit_id" value="" />
            <td id="shop_add_edit_number" data-
id="<?=$shop_num;?>"><?=$shop_num;?></td>
            <td><input type="text" name="shop_add_edit_name"
id="shop_add_edit_name" placeholder="Name" /></td>
            <td><button type="submit" name="shop_add_edit_submit"><span
class="fas fa-save"></span></button></td>
        </form>
        <td><button onclick="shop_edit_discard()"><span class="fas fa-
times-circle"></span></button></td>
    </tr>
</table>
</div>

```

### 3.24. register.php

```

<p id="register_error"><?php
    if(isset($_SESSION['r_error'])) {
        echo $_SESSION['r_error'];
    } else {
        echo "Please provide the information to create an account.";
    } ?></p>
<div class="register_panel">
    <form onsubmit="return register_check()" method="POST"
action="api.php?page=register">
        <input type="text" name="username" id="username"
placeholder="Username" />
        <input type="password" name="password" id="password"
placeholder="Password" />
        <input type="email" name="email" id="email" placeholder="Email" />
        <button type="submit" name="register" id="register">REGISTER</button>
    </form>
    <a href="index.php?page=login">Already have an account? Click here to
sign in...</a>
</div>

```

### 3.25. teacher.php

```

<div class="teacher_panel">
    <div class="teacher_text">
        <h2>Author</h2>
        <a href="https://www.linkedin.com/in/nikola-nedeljkovic-
8110b5150">Nikola Nedeljkovic 2/17</a>
        <div><a href="assets/documentation/documentation.pdf"><span
class="far fa-file-pdf"></span></a></div>
    </div>
    
</div>

```

## 3.26. style.css

```
@font-face {
  font-family: HelveticaNeue;
  src: url(../fonts/HelveticaNeue.ttf);
}
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
  font-family: HelveticaNeue;
}
html {
  background: url('../img/food_background.jpg') no-repeat center center
  fixed;
}
h1 {
  width: 100px;
  height: 50px;
  background: url('../img/logo.png') top left no-repeat;
  text-indent: -9999px;
  background-size: cover;
  color: #fff;
  background-color: #263238;
}
.red_border {
  border: 1px solid #f31413 !important;
  border-radius: 3px;
}

/*Header Stranica*/
.header_holder {
  width: 100%;
  background-color: #263238;
  display: flex;
  justify-content: space-between;
  align-items: center
}
.header_holder a {
  color: #fff;
  text-decoration: none;
}
.header_holder .login_logout_button {
  font-size: 18px;
  padding: 5px 15px;
}
.header_holder .login_logout_button:hover {
  background-color: rgba(0,0,0,0.1);
  color: #23527c;
}
@media screen and (max-width: 200px) {
  .header_holder {
    flex-direction: column;
  }
  .header_holder .login_logout_button {
    margin-top: 5px;
    margin-bottom: 5px;
  }
}
```

```
    }  
  }  
  
  /*404 i Teacher Stranice*/  
  .fourofour, .teacher_panel {  
    border-radius: 5px;  
    width: 90%;  
    margin-left: 5%;  
    background-color: rgba(255, 255, 255, 0.4);  
    display: flex;  
    justify-content: space-around;  
    align-items: center;  
  }  
  .fourofour .lost_text, .teacher_panel .teacher_text {  
    text-align: center;  
  }  
  .fourofour .lost_text h2, .teacher_panel .teacher_text h2 {  
    font-size: 30px;  
  }  
  .fourofour .lost_text a, .teacher_panel .teacher_text a {  
    color: #000;  
    text-decoration: none;  
  }  
  .fourofour .lost_text a: hover, .teacher_panel .teacher_text a: hover {  
    text-decoration: underline;  
  }  
  .fourofour img, .teacher_panel img {  
    margin-top: 30px;  
    width: 100%;  
    max-width: 260px;  
  }  
  .teacher_panel img {  
    max-width: 300px;  
    margin-bottom: 10px;  
    border-radius: 3px;  
  }  
  .teacher_panel .teacher_text div {  
    margin-top: 15px;  
  }  
  .teacher_panel .teacher_text div a span {  
    font-size: 40px;  
  }  
  .teacher_panel .teacher_text div a span: hover {  
    color: #999;  
  }  
  @media screen and (max-width: 700px) {  
    .fourofour, .teacher_panel {  
      flex-direction: column;  
    }  
    .fourofour .lost_text, .teacher_panel .teacher_text {  
      margin-top: 30px;  
    }  
    .teacher_panel img {  
      margin-bottom: 0;  
    }  
  }  
}
```

```
/* Login i Register Stranice */
#login_error, #register_error {
  width: 90%;
  margin-left: 5%;
  color: #a94442;
  background-color: #f2dede;
  padding: 15px;
  margin-bottom: 20px;
  border: 1px solid transparent;
  border-color: #ebccd1;
  border-radius: 4px;
  margin-bottom: 10px;
}
.login_panel, .register_panel {
  border-radius: 5px;
  width: 90%;
  margin-left: 5%;
  background-color: rgba(255, 255, 255, 0.4);
  display: flex;
  justify-content: space-around;
  align-items: center;
  flex-direction: column;
}
.login_panel form, .register_panel form {
  width: 100%;
  display: flex;
  justify-content: space-around;
  align-items: center;
  flex-direction: column;
  margin-top: 50px;
  margin-bottom: 50px;
}
.login_panel input, .register_panel input {
  margin-top: 20px;
  width: 80%;
  padding: 10px;
  border: none;
  border-bottom: 1px solid #9e9e9e;
  background-color: transparent;
  font-size: 18px;
}
.login_panel input:focus, .register_panel input:focus {
  outline: none;
  border-bottom: 2px solid #9e9e9e;
}
.login_panel button, .register_panel button {
  margin-top: 20px;
  padding: 10px 25px;
  cursor: pointer;
  color: #fff;
  background-color: #26a69a;
  font-size: 18px;
  margin-bottom: 10px;
  box-shadow: 0 2px 2px 0 rgba(0,0,0,0.14), 0 1px 5px 0 rgba(0,0,0,0.12), 0
3px 1px -2px rgba(0,0,0,0.2);
  border: none;
  border-radius: 3px;
}
```

```
}
.login_panel button:focus, .register_panel button:focus {
    background-color: #1d7d74;
    border-color: #122b40;
}
.login_panel button:hover, .register_panel button:hover {
    box-shadow: 0 3px 3px 0 rgba(0,0,0,0.14), 0 1px 7px 0 rgba(0,0,0,0.12), 0
3px 1px -1px rgba(0,0,0,0.2);
    background-color: #2bbbad;
}
.login_panel a, .register_panel a {
    font-size: 17px;
    text-decoration: none;
    color: #000;
    margin-bottom: 20px;
    text-align: center;
}
.login_panel a:hover, .register_panel a:hover {
    text-decoration: underline;
}

/* Main Stranica */
#welcome_sign {
    width: 90%;
    margin-left: 5%;
    color: #31708f;
    background-color: #d9edf7;
    padding: 15px;
    margin-bottom: 20px;
    border: 1px solid transparent;
    border-color: #bce8f1;
    border-radius: 4px;
    margin-bottom: 10px;
}
.main_panel {
    border-radius: 5px;
    width: 90%;
    margin-left: 5%;
    background-color: rgba(255, 255, 255, 0.4);
    display: flex;
    justify-content: space-around;
    align-items: center;
    flex-direction: column;
}
.main_panel h2 {
    margin-top: 10px;
    margin-bottom: 10px;
}
.main_panel table {
    width: 100%;
    table-layout: fixed;
}
.main_panel table button {
    border: none;
    background-color: transparent;
    color: #fff;
    cursor: pointer;
```

```
}  
.main_panel table input, .main_panel table select {  
    width: 100%;  
    border: none;  
    background-color: transparent;  
    font-size: 16px;  
    color: #fff;  
}  
.main_panel table a {  
    color: #fff;  
}  
.main_panel table select option {  
    color: #000;  
}  
.main_panel table input:focus, .main_panel table select:focus {  
    outline: none;  
}  
.main_panel table input::placeholder {  
    color: #fff;  
    opacity: 0.7;  
}  
.main_panel .not_first {  
    margin-top: 50px;  
}  
.main_panel table tr:nth-child(even) {  
    background-color: rgba(0, 0, 0, 0.3);  
    color: #fff;  
}  
.main_panel table tr:nth-child(odd) {  
    background-color: rgba(0, 0, 0, 0.2);  
    color: #fff;  
}  
.main_panel table tr:first-child {  
    background-color: transparent;  
    color: #000;  
}  
.main_panel table tr:hover {  
    background-color: rgba(0, 0, 0, 0.5);  
}  
.main_panel table tr:first-child:hover {  
    background-color: transparent;  
}  
.main_panel table tr th {  
    border-bottom: 1px solid #000;  
    border-right: 1px solid #000;  
    box-shadow: 2px 2px 3px #888;  
}  
.main_panel table tr td {  
    text-align: center;  
}  
.main_panel table tr th, .main_panel table tr td {  
    padding: 5px;  
    text-overflow: ellipsis;  
    overflow: hidden;  
}  
.main_panel table tr th:nth-child(1), .main_panel table tr th:nth-child(3),  
.main_panel table tr th:nth-child(4),
```



```

.main_panel table tr th:nth-child(6), .main_panel table tr th:nth-child(7),
.main_panel table tr td:nth-child(1),
.main_panel table tr td:nth-child(3), .main_panel table tr td:nth-child(4),
.main_panel table tr td:nth-child(6),
.main_panel table tr td:nth-child(7) {
    width: 10%;
}
.main_panel table tr th:nth-child(2), .main_panel table tr td:nth-child(2) {
    width: 30%;
}
.main_panel table tr th:nth-child(5), .main_panel table tr td:nth-child(5) {
    width: 20%;
}
.your_shops table tr th:nth-child(1), .your_shops table tr th:nth-child(3),
.your_shops table tr th:nth-child(4),
.your_shops table tr td:nth-child(1), .your_shops table tr td:nth-child(3),
.your_shops table tr td:nth-child(4) {
    width: 10%;
}
.your_shops table tr th:nth-child(2), .your_shops table tr td:nth-child(2) {
    width: 70%;
}
@media screen and (max-width: 500px) {
    .main_panel table tr {
        display: table;
        table-layout: fixed;
        width: 100%;
        overflow: hidden;
    }
    .main_panel table tr td {
        display: table-row;
    }
    .main_panel table tr th {
        display: none;
    }
}

/*Admin Stranica*/
.admin_panel {
    border-radius: 5px;
    width: 90%;
    margin-left: 5%;
    background-color: rgba(255, 255, 255, 0.4);
    display: flex;
    justify-content: space-around;
    align-items: center;
    flex-direction: column;
}
.admin_panel .admin_item {
    margin-top: 70px;
    padding: 5px;
    width: 100%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    text-align: center;
    box-shadow: 0px 2px 3px #888;
}

```

```

        padding-bottom: 15px;
    }
    .admin_panel .admin_item form {
        display: flex;
        justify-content: space-between;
        align-items: center;
        width: 80%;
    }
    .admin_panel .admin_item input, .admin_panel .admin_item select {
        width: 80%;
        padding: 10px;
        border: none;
        border-bottom: 1px solid #9e9e9e;
        background-color: transparent;
        font-size: 18px;
    }
    .admin_panel .admin_item form select {
        margin-bottom: 5px;
    }
    .admin_panel .admin_item input:focus, .admin_panel .admin_item select:focus {
        outline: none;
        border-bottom: 2px solid #9e9e9e;
    }
    .admin_panel .admin_item button {
        padding: 10px 25px;
        cursor: pointer;
        color: #fff;
        background-color: #26a69a;
        font-size: 18px;
        box-shadow: 0 2px 2px 0 rgba(0,0,0,0.14), 0 1px 5px 0 rgba(0,0,0,0.12), 0
3px 1px -2px rgba(0,0,0,0.2);
        border: none;
        border-radius: 3px;
    }
    .admin_panel .admin_item button:focus {
        background-color: #1d7d74;
        border-color: #122b40;
    }
    .admin_panel .admin_item button:hover {
        box-shadow: 0 3px 3px 0 rgba(0,0,0,0.14), 0 1px 7px 0 rgba(0,0,0,0.12), 0
3px 1px -1px rgba(0,0,0,0.2);
        background-color: #2bbbad;
    }
    .admin_panel .admin_item.with_more {
        flex-direction: column;
    }
    .admin_panel .admin_item .more_admin_options {
        margin-top: 10px;
        width: 100%;
        display: flex;
        justify-content: space-between;
        align-items: center;
        text-align: center;
    }
    @media screen and (max-width: 300px) {
        .admin_panel, .admin_item, .admin_panel .admin_item .more_admin_options,
        .admin_panel .admin_item form {

```

```
        flex-direction: column;
    }
    .admin_panel, .admin_item input {
        width: 90%;
        margin-top: 5px;
        margin-bottom: 5px;
    }
}

/*Footer Stranica*/
.footer_panel {
    width: 100%;
}
.footer_panel p {
    width: 100%;
    text-align: center;
    background-color: rgba(255, 255, 255, 0.6);
    font-size: 18px;
    text-align: center;
    padding: 5px;
    margin-top: 40px;
}
.footer_panel p a {
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
.footer_panel p a:hover {
    text-decoration: underline;
}
.footer_panel .site_links {
    margin-top: 30px;
    width: 100%;
    background-color: rgba(255, 255, 255, 0.4);
    display: flex;
    flex-direction: column;
    align-items: center;
}
.footer_panel .site_links h3 {
    margin-top: 10px;
    margin-bottom: 20px;
}
.footer_panel .site_links ul {
    list-style-type: none;
}
.footer_panel .site_links ul li {
    margin-bottom: 7px;
}
.footer_panel .site_links ul li a {
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
.footer_panel .site_links ul li a:hover {
    text-decoration: underline;
}
```

### 3.27. main.js

```
function login_check() {
    var username = document.getElementById("username").value,
        password = document.getElementById("password").value;
    if (username == "" || password == "") {
        document.getElementById("login_error").innerHTML = "Username and
password can not be empty";
        return false;
    }
    document.getElementById("login_error").innerHTML = "";
    return true;
}

function register_check() {
    var username = document.getElementById("username").value,
        password = document.getElementById("password").value,
        email = document.getElementById("email").value,
        email_pattern = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-
]+(?:\.[a-zA-Z0-9-]+)*$/;
    if (username == "" || password == "" || email == "") {
        document.getElementById("register_error").innerHTML = "You must not
leave empty fields";
        return false;
    }
    if (!email_pattern.test(email)) {
        document.getElementById("register_error").innerHTML = "Email not
valid";
        return;
    }
    document.getElementById("register_error").innerHTML = "";
    return true;
}

function item_edit_click(number, item_id) {
    var num_field = document.getElementById("itemnum" + number),
        name_field = document.getElementById("itemname" + number),
        quantity_field = document.getElementById("itemquantity" + number),
        measure_field = document.getElementById("itemmeasure" + number),
        shop_field = document.getElementById("itemshop" + number),
        item_add_edit_number_field =
document.getElementById("item_add_edit_number"),
        item_add_edit_name_field =
document.getElementById("item_add_edit_name"),
        item_add_edit_quantity_field =
document.getElementById("item_add_edit_quantity"),
        item_add_edit_measure_field =
document.getElementById("item_add_edit_measure"),
        item_add_edit_shop_field =
document.getElementById("item_add_edit_shop"),
        item_add_edit_id_field = document.getElementById("item_add_edit_id");

    item_add_edit_number_field.innerHTML = num_field.innerHTML;
    item_add_edit_name_field.value = name_field.innerHTML;
    item_add_edit_quantity_field.value = quantity_field.innerHTML;
    item_add_edit_measure_field.value = measure_field.getAttribute('data-
id');
```

```
        item_add_edit_shop_field.value = shop_field.getAttribute('data-id');
        item_add_edit_id_field.value = item_id;
    }

    function item_edit_discard() {
        var item_add_edit_id_field = document.getElementById("item_add_edit_id"),
            item_add_edit_number_field =
document.getElementById("item_add_edit_number"),
            item_add_edit_name_field =
document.getElementById("item_add_edit_name"),
            item_add_edit_quantity_field =
document.getElementById("item_add_edit_quantity"),
            item_add_edit_measure_field =
document.getElementById("item_add_edit_measure"),
            item_add_edit_shop_field =
document.getElementById("item_add_edit_shop");

        item_add_edit_id_field.value = "";
        item_add_edit_number_field.innerHTML =
item_add_edit_number_field.getAttribute('data-id');
        item_add_edit_name_field.value = "";
        item_add_edit_quantity_field.value = "";
        item_add_edit_measure_field.selectedIndex = 0;
        item_add_edit_shop_field.selectedIndex = 0;
    }

    function shop_edit_click(number, shop_id) {
        var num_field = document.getElementById("shopnum" + number),
            name_field = document.getElementById("shopname" + number),
            shop_add_edit_number_field =
document.getElementById("shop_add_edit_number"),
            shop_add_edit_name_field =
document.getElementById("shop_add_edit_name"),
            shop_add_edit_id_field = document.getElementById("shop_add_edit_id");

        shop_add_edit_number_field.innerHTML = num_field.innerHTML;
        shop_add_edit_name_field.value = name_field.innerHTML;
        shop_add_edit_id_field.value = shop_id;
    }

    function shop_edit_discard() {
        var shop_add_edit_number_field =
document.getElementById("shop_add_edit_number"),
            shop_add_edit_name_field = document.getElementById("shop_add_edit_name"),
            shop_add_edit_id_field = document.getElementById("shop_add_edit_id");

        shop_add_edit_id_field.value = "";
        shop_add_edit_number_field.innerHTML =
shop_add_edit_number_field.getAttribute("data-id");
        shop_add_edit_name_field.value = "";
    }

    function add_edit_shop_check() {
        var name = document.getElementById("shop_add_edit_name");
        if (name.value == "") {
            name.classList.add("red_border");
            return false;
        }
    }
}
```

```
    }
    name.classList.remove("red_border");
    return true;
}

function add_edit_item_check() {
    var name = document.getElementById("item_add_edit_name"),
        quantity = document.getElementById("item_add_edit_quantity");
    if (name.value == "") {
        name.classList.add("red_border");
        return false;
    }
    name.classList.remove("red_border");
    if (quantity.value == "") {
        quantity.classList.add("red_border");
        return false;
    }
    quantity.classList.remove("red_border");
    return true;
}

function add_measure_check() {
    var name = document.getElementById("add_measure");
    if (name.value == "") {
        name.classList.add("red_border");
        return false;
    }
    name.classList.remove("red_border");
    return true;
}

function admin_measure_change() {
    var measures = document.getElementById("measure_edit_select"),
        id_field = document.getElementById("edit_measure_id"),
        name_field = document.getElementById("edit_measure");
    id_field.value = measures.value;
    name_field.value = measures.options[measures.selectedIndex].text;
}

function edit_measure_check() {
    var measures = document.getElementById("measure_edit_select"),
        id_field = document.getElementById("edit_measure_id"),
        name_field = document.getElementById("edit_measure");
    if (id_field.value == "") {
        measures.classList.add("red_border");
        return false;
    }
    measures.classList.remove("red_border");
    if (name_field.value == "") {
        name_field.classList.add("red_border");
        return false;
    }
    name_field.classList.remove("red_border");
    return true;
}

function admin_user_change() {
```

```
var user = document.getElementById("user_change_select"),
    id_field = document.getElementById("edit_user_right_id"),
    right = document.getElementById("user_right_select");
id_field.value = user.value;
right.value = user.options[user.selectedIndex].getAttribute('data-id');
}

function edit_user_check() {
var right_id = document.getElementById("edit_user_right_id"),
    right = document.getElementById("user_right_select"),
    user = document.getElementById("user_change_select");
if (right_id.value == "") {
    user.classList.add("red_border");
    return false;
}
user.classList.remove("red_border");
if (right.value == "") {
    right.classList.add("red_border");
    return false;
}
right.classList.remove("red_border");
return true;
}
```

## 4. HIJERARHIJSKI PREGLED

