**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

**S** **U B O T I C A**

**PROJEKAT**

**iz predmeta Web programiranje**

|  |  |
| --- | --- |
| **KANDIDAT(I)**  **Samanta Hozic**  **Tamara Baclija** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SUBOTICA, 2024.** | |

Sadržaj

[1. Opis zadatka 3](#_Toc156903284)

[2. Realizacija zadatka 4](#_Toc156903285)

[3. Struktura baze podataka (MySQL Workbench) 5](#_Toc156903286)

[4. Opis funkcionalnosti 6](#_Toc156903287)

[5. Korišćena literatura 7](#_Toc156903288)

# Opis zadatka

Opisati ukratko šta je bio projektni zadatak**. Navesti web adresu projekta.**

**Naš zadatak je bio kreiranje potpuno funkcionalne veb stranice za šetanje pasa pod imenom "Paws and Paths". Veb stranica omogućava korisnicima da se prijave kao obični korisnici koji žele da zakažu šetnju pasa ili kao "walkeri" - korisnici koji žele da šetaju pse.**

**Web adresa projekta:** https://42.stud.vts.su.ac.rs/setnja\_pasa/

**Struktura Stranica:**

1. **Main Page (Glavna stranica):**

Upoznaje korisnike sa našim ciljem i delatnostima.

Prikazuje 5 najboljih i 5 najaktivnijih šetača.

Omogućava prijavu na newsletter za obaveštenja o planovima i promenama.

1. **About Page (O nama):**

Sadrži informacije koje mogu biti zanimljive korisnicima, kao što su najčešće postavljana pitanja, naša misija i vizija.

1. **Contact Page (Kontakt):**

Prikazuje našu lokaciju i omogućava korisnicima da kontaktiraju glavnu organizaciju u slučaju određenih problema ili pitanja.

1. **Our Walkers Page (Naši šetači):**

Omogućava korisnicima da pregledaju šetače i da putem dugmeta na njihovim karticama vide pojedinačne profile šetača.

1. **Booking Page (Rezervacija):**

Omogućava korisnicima da zakažu šetnju pasa, gde unose rasu, starost psa, kao i vreme i mesto šetnje.

1. **Diary Page (Dnevnik):**

Omogućava šetačima da vode evidenciju o svojim zakazanim šetnjama i unose dodatne informacije o njima.

# Realizacija zadatka

U ovom delu treba prikazati i delove programskog koda. Programski kod napisati uz pomoć fonta **Courier, veličina slova 10**.

**Sigup I slanje maila**

<?php

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

use PHPMailer\PHPMailer\SMTP;

require 'PHPMailer/src/Exception.php';

require 'PHPMailer/src/PHPMailer.php';

require 'PHPMailer/src/SMTP.php';

require\_once 'db\_config.php';

if ($\_SERVER['REQUEST\_METHOD'] == "POST") {

$firstname = strip\_tags($\_POST['firstName']);

$lastname = strip\_tags($\_POST['lastName']);

$username = strip\_tags($\_POST['email']);

$phone = strip\_tags($\_POST['phone']);

$address = strip\_tags($\_POST['address']);

$city = strip\_tags($\_POST['city']);

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

$hashed\_password = password\_hash($password, PASSWORD\_BCRYPT);

$country = isset($\_POST['country']) ? strip\_tags($\_POST['country']) : "America";

$user\_type = $\_POST['userType'];

$username\_without\_at = '';

if (strpos($username, '@gmail.com') !== false)

{

$username\_without\_at = str\_replace('@gmail.com', '', $username);

}

else

{

die("Please enter a valid e-mail address");

}

$five\_digit\_code = rand(0, 99999);

$activation\_code = str\_pad($five\_digit\_code, 5, '0', STR\_PAD\_LEFT);

$file\_name = '';

$file\_temp = '';

$file\_type = '';

if ($user\_type == 'premium') {

if (is\_uploaded\_file($\_FILES['image']['tmp\_name'])) {

$file\_name = $\_FILES['image']["name"];

$file\_temp = $\_FILES["image"]["tmp\_name"];

$file\_type = $\_FILES["image"]["type"];

if (exif\_imagetype($file\_temp) != IMAGETYPE\_JPEG) {

die("The file type is not a JPEG");

}

$new\_file\_name = $username\_without\_at . ".jpg";

$upload = "imgs/walkers/$new\_file\_name";

if ($\_FILES['image']["error"] > 0) {

die("Something went wrong during file upload!");

} else {

move\_uploaded\_file($file\_temp, $upload);

$sql = "INSERT INTO walker (email, password, fname, lname, phone\_number, address, city, country, photo, rating\_enable, activation\_code) VALUES (:username, :hashed\_password, :firstname, :lastname, :phone, :address, :city, :country, :photo\_path, :rating\_enable, :activation\_code)";

if (!empty($dbh)) {

$query = $dbh->prepare($sql);

$query->bindParam(':username', $username\_without\_at, PDO::PARAM\_STR);

$query->bindParam(':hashed\_password', $hashed\_password, PDO::PARAM\_STR);

$query->bindParam(':firstname', $firstname, PDO::PARAM\_STR);

$query->bindParam(':lastname', $lastname, PDO::PARAM\_STR);

$query->bindParam(':phone', $phone, PDO::PARAM\_INT);

$query->bindParam(':address', $address, PDO::PARAM\_STR);

$query->bindParam(':city', $city, PDO::PARAM\_STR);

$query->bindParam(':country', $country, PDO::PARAM\_STR);

$query->bindParam(':photo\_path', $upload, PDO::PARAM\_STR);

$query->bindParam(':rating\_enable', $activation\_code, PDO::PARAM\_INT);

$query->bindParam(':activation\_code', $activation\_code, PDO::PARAM\_INT);

if ($query->execute())

{

$mail = new PHPMailer(true);

$mailMessage = "Your activation code is: $activation\_code <br> DO NOT REPLY TO THIS EMAIL";

$mailTitle = "Activation Code";

$mailTo = $username;

$mail->SMTPDebug = SMTP::DEBUG\_SERVER;

$mail->isSMTP();

$mail->Host = 'mail.42.stud.vts.su.ac.rs';

$mail->SMTPAuth = true;

$mail->Username = 'four2';

$mail->Password = '6pxxIS2mnrjtjIz';

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

$mail->setFrom('four2@42.stud.vts.su.ac.rs', 'Mailer');

$mail->addAddress($mailTo);

$mail->isHTML(true);

$mail->Subject = $mailTitle;

$mail->Body = $mailMessage;

$mail->AltBody = $mailMessage;

$mail->send();

echo "<script>alert('Sign up successful! Wait for an admin to approve, and insert a code during your first login. The code has been sent to your email');window.location.href = 'index.php';</script>";

exit();

}

else

{

echo "<script>alert('Sign in failed. Try again later');window.location.href = 'index.php';</script>";

}

}

}

}

} elseif ($user\_type == 'regular') {

try {

$sql = 'INSERT INTO user (email, password, fname, lname, phone\_number, address, city, country, activation\_code) VALUES (:username, :hashed\_password, :firstname, :lastname, :phone, :address, :city, :country, :activation\_code)';

$query = $dbh->prepare($sql);

$query->bindParam(':username', $username\_without\_at, PDO::PARAM\_STR);

$query->bindParam(':hashed\_password', $hashed\_password, PDO::PARAM\_STR);

$query->bindParam(':firstname', $firstname, PDO::PARAM\_STR);

$query->bindParam(':lastname', $lastname, PDO::PARAM\_STR);

$query->bindParam(':phone', $phone, PDO::PARAM\_STR);

$query->bindParam(':address', $address, PDO::PARAM\_STR);

$query->bindParam(':city', $city, PDO::PARAM\_STR);

$query->bindParam(':country', $country, PDO::PARAM\_STR);

$query->bindParam(':activation\_code', $activation\_code, PDO::PARAM\_STR);

if ($query->execute()) {

echo 'Sign Up successful! But wait for the admin to approve...<br>';

} else {

echo "There has been an error with registering a user!";

}

$mail = new PHPMailer(true);

$mailMessage = "Your activation code is: $activation\_code <br> DO NOT REPLY TO THIS EMAIL";

$mailTitle = "Activation Code";

$mailTo = $username;

$mail->SMTPDebug = SMTP::DEBUG\_SERVER;

$mail->isSMTP();

$mail->Host = 'mail.42.stud.vts.su.ac.rs';

$mail->SMTPAuth = true;

$mail->Username = 'four2';

$mail->Password = '6pxxIS2mnrjtjIz';

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

$mail->setFrom('four2@42.stud.vts.su.ac.rs', 'Mailer');

$mail->addAddress($mailTo);

$mail->isHTML(true);

$mail->Subject = $mailTitle;

$mail->Body = $mailMessage;

$mail->AltBody = $mailMessage;

$mail->send();

echo "<script>alert('Sign in successful');window.location.href = 'index.php';</script>";

exit();

} catch (Exception $e) {

echo "<script>alert('Unable to send email');window.location.href = 'index.php';</script>";

}

finally{

echo "<script>alert('Sign in successful');window.location.href = 'index.php';</script>";

exit();

}

}

}

**Javascript za sign up**

window.addEventListener("DOMContentLoaded", init);

function init() {

const form = document.getElementById('signingUpForm');

form.addEventListener('submit', function(event) {

let isValid = validateForm();

if (isValid) {

sendData();

} else {

event.preventDefault();

}

});

}

function validateForm() {

const firstName = document.getElementById('firstName');

const lastName = document.getElementById('lastName');

const email = document.getElementById('email');

const phone = document.getElementById('phone');

const password = document.getElementById('password');

const address = document.getElementById('address');

const city = document.getElementById('city');

const country = document.getElementById('country');

const userType = document.getElementById('userType');

let isValid = true;

if (userType.value === 'premium') {

const image = document.getElementById('image');

if (!image.files.length) {

showErrorMessage(image, 'Please upload a picture');

isValid = false;

} else if (image.files[0].type !== 'image/jpeg') {

showErrorMessage(image, 'Please select a JPEG file');

isValid = false;

} else {

hideErrorMessage(image);

}

}

if (firstName.value.trim() === '') {

showErrorMessage(firstName, 'Please enter your name');

isValid = false;

} else if (firstName.value.length > 25) {

showErrorMessage(firstName, 'Name must be shorter than 25 characters!');

isValid = false;

} else {

hideErrorMessage(firstName);

}

if (lastName.value.trim() === '') {

showErrorMessage(lastName, 'Please enter your last name');

isValid = false;

} else if (lastName.value.length > 25) {

showErrorMessage(lastName, 'Last name must be shorter than 25 characters!');

isValid = false;

} else {

hideErrorMessage(lastName);

}

if (email.value.trim() === '') {

showErrorMessage(email, 'Please enter your email');

isValid = false;

} else if (email.value.length > 50) {

showErrorMessage(email, 'Email must be shorter than 50 characters!');

isValid = false;

} else {

hideErrorMessage(email);

}

if (phone.value.trim() === '') {

showErrorMessage(phone, 'Please enter your phone number');

isValid = false;

} else if (!validatePhoneNumber(phone.value)) {

showErrorMessage(phone, 'Please enter a valid phone number');

isValid = false;

} else {

hideErrorMessage(phone);

}

if (password.value.trim() === '') {

showErrorMessage(password, 'Please enter your password');

isValid = false;

} else if (password.value.length < 9) {

showErrorMessage(password, 'Password should be at least 9 characters long');

isValid = false;

} else {

hideErrorMessage(password);

}

if (address.value.trim() === '') {

showErrorMessage(address, 'Please enter your address');

isValid = false;

} else if (address.value.length < 3) {

showErrorMessage(address, 'Address should be more than 3 characters long');

isValid = false;

} else {

hideErrorMessage(address);

}

if (city.value.trim() === '') {

showErrorMessage(city, 'Please enter your city');

isValid = false;

} else if (city.value.length < 3) {

showErrorMessage(city, 'City name should be more than 3 characters long');

isValid = false;

} else {

hideErrorMessage(city);

}

if (country.value.trim() === '') {

showErrorMessage(country, 'Please enter your country');

isValid = false;

} else if (country.value.length < 3) {

showErrorMessage(country, 'Country name should be more than 3 characters long');

isValid = false;

} else {

hideErrorMessage(country);

}

return isValid;

}

function showErrorMessage(input, message) {

const small = input.nextElementSibling;

small.style.display = 'inline';

small.textContent = message;

}

function hideErrorMessage(input) {

const small = input.nextElementSibling;

small.style.display = 'none';

}

function validatePhoneNumber(phoneNumber) {

const phonePattern = /^\d{6,12}$/;

return phonePattern.test(phoneNumber);

}

function sendData() {

const form = document.getElementById('signingUpForm');

const formData = new FormData(form);

fetch('sign\_up\_processing.php', {

method: 'POST',

body: formData

}).then(response => {

if (response.ok) {

return response.text();

}

throw new Error('Network response was not ok.');

}).then(data => {

console.log(data);

}).catch(error => {

console.error('There was a problem with your fetch operation:', error);

});

}

**Booking kod**

<?php

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

use PHPMailer\PHPMailer\SMTP;

require 'PHPMailer/src/Exception.php';

require 'PHPMailer/src/PHPMailer.php';

require 'PHPMailer/src/SMTP.php';

require\_once 'db\_config.php';

session\_start();

if($\_SERVER['REQUEST\_METHOD'] == "POST") {

$username = $\_SESSION['name'];

$dogName = filter\_var($\_POST['dogName'], FILTER\_SANITIZE\_STRING);

$dogBreed = filter\_var($\_POST['dogBreed'], FILTER\_SANITIZE\_STRING);

$age = filter\_var($\_POST['dogAge'], FILTER\_SANITIZE\_NUMBER\_INT);

$booking\_date = $\_POST['bookingDate'];

$startTime = $\_POST['startTime'];

$endTime = $\_POST['endTime'];

$walkerName = $\_POST['dogWalker'];

$comments = filter\_var($\_POST['comments'], FILTER\_SANITIZE\_STRING);

try {

$sql = "INSERT INTO dog\_walking\_appt (walker, owner\_name, dog\_name, breed, age, additional\_info, booking\_date, start\_time, end\_time)

VALUES (:walker, :owner\_name, :dog\_name, :breed, :age, :info, :booking\_date, :start\_time, :end\_time)";

$query = $dbh->prepare($sql);

$query->bindParam(':walker', $walkerName, PDO::PARAM\_STR);

$query->bindParam(':owner\_name', $username, PDO::PARAM\_STR);

$query->bindParam(':dog\_name', $dogName, PDO::PARAM\_STR);

$query->bindParam(':breed', $dogBreed, PDO::PARAM\_STR);

$query->bindParam(':age', $age, PDO::PARAM\_INT);

$query->bindParam(':info', $comments, PDO::PARAM\_STR);

$query->bindParam(':booking\_date', $booking\_date, PDO::PARAM\_STR);

$query->bindParam(':start\_time', $startTime, PDO::PARAM\_STR);

$query->bindParam(':end\_time', $endTime, PDO::PARAM\_STR);

$query->execute();

$mail = new PHPMailer(true);

$mailMessage = "You have booked a dog walking of $dogName, on $booking\_date starting at $startTime and ending at $endTime.<br><br>The information you have saved about your dog will be sent to the walker. If you have any special requests you'd like to discuss with the walker, please contact them directly through our website or on their phone which is provided on their profile.<br><br><b>Hope your little one has a great time! <br><i>Paws&Pets</i></b>";

$mailTitle = "Dog Walking appointment";

$mailTo = $username . "@gmail.com"; // You might want to use a proper email validation and retrieval

//Server settings

$mail->SMTPDebug = 0; //Disable verbose debug output

$mail->isSMTP(); //Send using SMTP

$mail->Host = 'mail.42.stud.vts.su.ac.rs'; //Set the SMTP server to send through

$mail->SMTPAuth = true; //Enable SMTP authentication

$mail->Username = 'four2'; //SMTP username

$mail->Password = '6pxxIS2mnrjtjIz'; //SMTP password

$mail->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS; //Enable TLS encryption

$mail->Port = 587; //TCP port to connect to

$mail->setFrom('four2@42.stud.vts.su.ac.rs', 'Mailer');

$mail->addAddress($mailTo); //Add a recipient

//Content

$mail->isHTML(true); //Set email format to HTML

$mail->Subject = $mailTitle;

$mail->Body = $mailMessage;

$mail->AltBody = strip\_tags($mailMessage);

$mail->send();

echo "<script>

alert('An email has been sent to you about the data, please check if everything is correct');

window.location.href = 'booking.php';

</script>";

} catch (PDOException $e) {

echo "<script>alert('An error occurred while saving your booking: " . $e->getMessage() . "'); window.location.href = 'booking.php';</script>";

} catch (Exception $e) {

echo "<script>alert('An error occurred while sending the email: " . $e->getMessage() . "'); window.location.href = 'booking.php';</script>";

}

}

?>

**Profil kod**

<?php

include "navigation.php";

require\_once "db\_config.php";

session\_start();

if (!isset($\_SESSION['name']) || !isset($\_SESSION['role'])) {

echo "<br><br><br>You are not logged in";

exit();

}

$user\_type = $\_SESSION['role'];

$username = $\_SESSION['name'];

$walker\_id = isset($\_GET['walker\_id']) ? $\_GET['walker\_id'] : null;

$dbh->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = "";

$params = [];

if ($walker\_id) {

$sql = "SELECT \* FROM walker WHERE walker\_id = :walker\_id AND NOT (walker\_id=4224 OR walker\_id=2442)";

$params = [':walker\_id' => $walker\_id];

} elseif ($user\_type == 'walker' || $user\_type == 'admin') {

$sql = "SELECT \* FROM walker WHERE email = :username AND NOT (walker\_id=4224 OR walker\_id=2442)";

$params = [':username' => $username];

} elseif ($user\_type == 'user') {

$sql = "SELECT \* FROM user WHERE email = :username";

$params = [':username' => $username];

} else {

echo "Invalid user type";

exit();

}

$query = $dbh->prepare($sql);

$query->execute($params);

$result = $query->fetch(PDO::FETCH\_ASSOC);

if (!$result) {

echo "No user found";

exit();

}

$canEditProfile = true;

$canChangePassword = $user\_type == 'walker' || $user\_type == 'admin' || ($user\_type == 'user' && !$walker\_id);

$canContact = $user\_type == 'user' && $walker\_id;

$canHire = $user\_type == 'user' && $walker\_id;

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Profile</title>

<link rel="stylesheet" href="path/to/bootstrap.css">

</head>

<body>

<?php if(isset($\_GET['err'])){

echo "<br><br><br><td style='border: 1px solid red;'>Mew password cannot be the same as the old password</td>";

} else if(isset($\_GET['succ']) && $\_GET['succ']==1){

echo "<br><br><br>Password changed successfully";

} else if(isset($\_GET['succ']) && $\_GET['succ']==2){

echo "<br><br><br>An unknown error has occurred, try again later";

}

?>

<div class="container rounded bg-white mt-5 mb-5">

<div class="row profile-row">

<div class="col-md-4 border-right">

<div class="d-flex flex-column align-items-center text-center p-3 py-5">

<img class="profile-picture mt-5" src="<?php echo htmlspecialchars(isset($result['photo']) ? $result['photo'] : 'imgs/default-pfp.jpg'); ?>" ">

<span class="font-weight-bold"><?php echo htmlspecialchars($result['email']); ?></span>

<span class="text-black-50"><?php echo htmlspecialchars($result['email']) . "@gmail.com"; ?></span>

<?php if (!empty($result['about'])): ?>

<p><?php echo htmlspecialchars($result['about']); ?></p>

<?php endif; ?>

<?php if (!isset($\_GET['walker\_id']) && isset($\_SESSION['role'])): ?>

<button class="btn btn-primary mt-3" type="button" id="edit" onclick="editProfile()">Edit</button>

<div class="col-md-12">

<button type="button" id="saveButton" class="btn btn-primary mt-3" style="display: none;" onclick="submitForm()">Save</button>

</div>

<button class="btn btn-primary mt-3" id="changePasswordButton" onclick="showPasswordChange()">Change Password</button>

<?php endif; ?>

<?php

if(isset($\_SESSION['role'])): ?>

<button class="btn btn-primary mt-3" id="delete"><a href="delete\_account.php" style="color: #f2f2f2">Delete account</a></button>

<?php

endif;

if ($canContact): ?>

<button class="btn btn-primary mt-3">Contact</button>

<?php endif; ?>

<?php if ($canHire): ?>

<a href="booking.php?walker=<?php echo $walker\_id; ?>" class="btn btn-primary mt-3">Hire</a>

<?php endif; ?>

</div>

</div>

<div class="col-md-6 border-right">

<form id="profileForm" method="post" action="update\_profile.php">

<div class="p-3 py-5">

<div class="d-flex justify-content-between align-items-center mb-3">

<h4 class="text-right"><?php echo htmlspecialchars($result['fname'] . " " . htmlspecialchars($result['lname'])); ?></h4>

</div>

<div class="row mt-2">

<div class="col-md-6">

<label for="pfname" class="labels"><b>Name</b></label>

<input id="pfname" name="pfname" type="text" class="form-control" placeholder="first name" value="<?php echo htmlspecialchars($result['fname']); ?>" readonly>

</div>

<div class="col-md-6">

<label for="pflname" class="labels"><b>Surname</b></label>

<input id="pflname" name="pflname" type="text" class="form-control" value="<?php echo htmlspecialchars($result['lname']); ?>" placeholder="surname" readonly>

</div>

</div>

<div class="row mt-3">

<div class="col-md-12">

<label for="pfphone" class="labels"><b>Phone Number</b></label>

<input id="pfphone" name="pfphone" type="text" class="form-control" placeholder="enter phone number" value="<?php echo htmlspecialchars($result['phone\_number']); ?>" readonly>

</div>

<div class="col-md-12">

<label for="pfaddress" class="labels"><b>Address</b></label>

<input id="pfaddress" name="pfaddress" type="text" class="form-control" placeholder="enter address" value="<?php echo htmlspecialchars($result['address']); ?>" readonly>

</div>

<div class="col-md-12">

<label for="pfemail" class="labels"><b>Email</b></label>

<input id="pfemail" name="pfemail" type="text" class="form-control" placeholder="enter email id" value="<?php echo htmlspecialchars($result['email']) . "@gmail.com"; ?>" readonly>

</div>

</div>

<div class="row mt-3">

<div class="col-md-6">

<label for="pfcountry" class="labels"><b>Country</b></label>

<input id="pfcountry" name="pfcountry" type="text" class="form-control" placeholder="country" value="<?php echo htmlspecialchars($result['country']); ?>" readonly>

</div>

<div class="col-md-6">

<label for="pfcity" class="labels"><b>City</b></label>

<input id="pfcity" name="pfcity" type="text" class="form-control" value="<?php echo htmlspecialchars($result['city']); ?>" placeholder="city" readonly>

</div>

</div>

<?php if ($user\_type == 'walker' || $user\_type == 'admin'): ?>

<div class="col-md-12">

<label for="about" class="labels"><b>About</b></label>

<input id="about" name="about" type="text" class="form-control" value="<?php echo htmlspecialchars($result['about']); ?>" placeholder="about" readonly>

</div>

<?php endif; ?>

<?php if ($\_SESSION['role'] == 'walker' || $\_SESSION['role'] == 'admin'): ?>

<div class="col-md-12">

<input type="checkbox" id="myCheckbox" name="myCheckbox" <?php echo ($result['accept'] == 1) ? 'checked' : ''; ?> disabled>

<label for="myCheckbox"><?php echo ($result['accept'] == 1) ? 'I am accepting walks' : 'I am not accepting walks'; ?></label>

</div>

<?php endif; ?>

<div class="col-md-12 mt-4">

<button type="submit" id="saveButton" class="btn btn-primary" style="display: none;">Save</button>

</div>

</div>

</form>

</div>

</div>

</div>

<hr >

<?php if (isset($\_GET['walker\_id'])): ?>

<div class="col-md-12">

<label for="rating" class="labels">Rating & Comments</label

</div>

<?php

$sql = $dbh->prepare("SELECT \* FROM walks WHERE walker IN (SELECT email FROM walker WHERE walker\_id=:walker\_id)");

$sql->bindParam(':walker\_id', $walker\_id, PDO::PARAM\_INT);

if ($sql->execute()) {

$walks = $sql->fetchAll(PDO::FETCH\_ASSOC);

echo "<table style='border: 2px solid black;'>";

echo "<tr><th>Walker View</th><th>User View</th><th>Path</th><th>Duration</th><th>Rating</th></tr>";

foreach ($walks as $walk) {

echo "<tr style='border: 1px solid black;'>";

echo "<td style='border: 1px solid black;'>" . htmlspecialchars($walk["walker\_view"]) . "</td>";

echo "<td style='border: 1px solid black;'>" . htmlspecialchars($walk["user\_view"]) . "</td>";

echo "<td style='border: 1px solid black;'>" . htmlspecialchars($walk["path"]) . "</td>";

echo "<td style='border: 1px solid black;'>" . htmlspecialchars($walk["duration"]) . "</td>";

echo "<td style='border: 1px solid black;'>" . htmlspecialchars($walk["walk\_rating"]) . "</td>";

echo "</tr>";

}

echo "</table>";

}

?>

<?php endif; ?>

<script>

function editProfile() {

var inputs = document.querySelectorAll('#profileForm input');

inputs.forEach(function(input) {

input.removeAttribute('readonly');

});

var checkbox = document.getElementById('myCheckbox');

if (checkbox) {

checkbox.removeAttribute('disabled');

}

document.getElementById('saveButton').style.display = 'inline';

}

function changePassword() {

window.location.href = 'change\_password.php';

}

function submitForm() {

document.getElementById('profileForm').submit();

document.getElementById('saveButton').style.display = 'none';

}

function showPasswordChange() {

document.getElementById('passwordModal').style.display = 'block';

}

document.querySelector('.close').onclick = function() {

document.getElementById('passwordModal').style.display = 'none';

}

document.getElementById('passwordForm').addEventListener('submit', function(event) {

event.preventDefault();

var formData = new FormData(this);

formData.append('username', "<?php echo htmlspecialchars($result['email']); ?>");

fetch('update\_password.php', {

method: 'POST',

body: formData

}).then(response => response.text())

.then(data => {

alert(data);

document.getElementById('passwordModal').style.display = 'none';

}).catch(error => console.error('Error:', error));

});

</script>

<div id="passwordModal" class="modal" style="display: none;">

<div class="modal-content">

<span class="close">&times;</span>

<form id="passwordForm" method="post" action="update\_password.php">

<label for="newPassword">New Password:</label>

<input type="password" id="newPassword" name="newPassword" required>

<button type="submit">Change Password</button>

</form>

</div>

</div>

<style>

.modal {

display: none;

position: fixed;

z-index: 1000;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: hidden;

outline: 0;

background-color: rgba(0, 0, 0, 0.5);

justify-content: center;

align-items: center;

}

.modal-content {

position: relative;

display: flex;

flex-direction: column;

width: 100%;

max-width: 800px;

background-color: #fff;

border: 1px solid rgba(0, 0, 0, .2);

border-radius: 0.3rem;

box-shadow: 0 3px 9px rgba(0, 0, 0, .5);

padding: 20px;

}

.modal-header {

display: flex;

align-items: flex-start;

justify-content: space-between;

padding: 1rem;

border-bottom: 1px solid #e9ecef;

border-top-left-radius: 0.3rem;

border-top-right-radius: 0.3rem;

}

. profile-row .col-md-6,

.profile-row .col-md-4 {

margin: 0px 15px;

}

.modal-title {

margin-bottom: 0;

line-height: 1.5;

}

.close {

font-size: 1.5rem;

font-weight: 700;

line-height: 1;

color: #000;

opacity: .5;

cursor: pointer;

}

.close:hover {

color: #000;

text-decoration: none;

opacity: .75;

}

.modal-body {

position: relative;

flex: 1 1 auto;

padding: 1rem;

}

.modal-footer {

display: flex;

flex-wrap: wrap;

align-items: center;

justify-content: flex-end;

padding: 0.75rem;

border-top: 1px solid #e9ecef;

border-bottom-right-radius: 0.3rem;

border-bottom-left-radius: 0.3rem;

}

.btn {

margin-top: 10px;

}

.modal-backdrop {

position: fixed;

top: 0;

left: 0;

z-index: 1040;

width: 100vw;

height: 100vh;

background-color: #000;

opacity: .5;

}

.profile-form {

display: flex;

flex-direction: column;

gap: 15px;

}

.profile-form .form-control {

border-radius: 0.2rem;

}

.profile-form label {

font-weight: bold;

}

.profile-picture {

max-width: 150px;

border-radius: 50%;

margin-bottom: 20px;

}

.row{

justify-content:center;

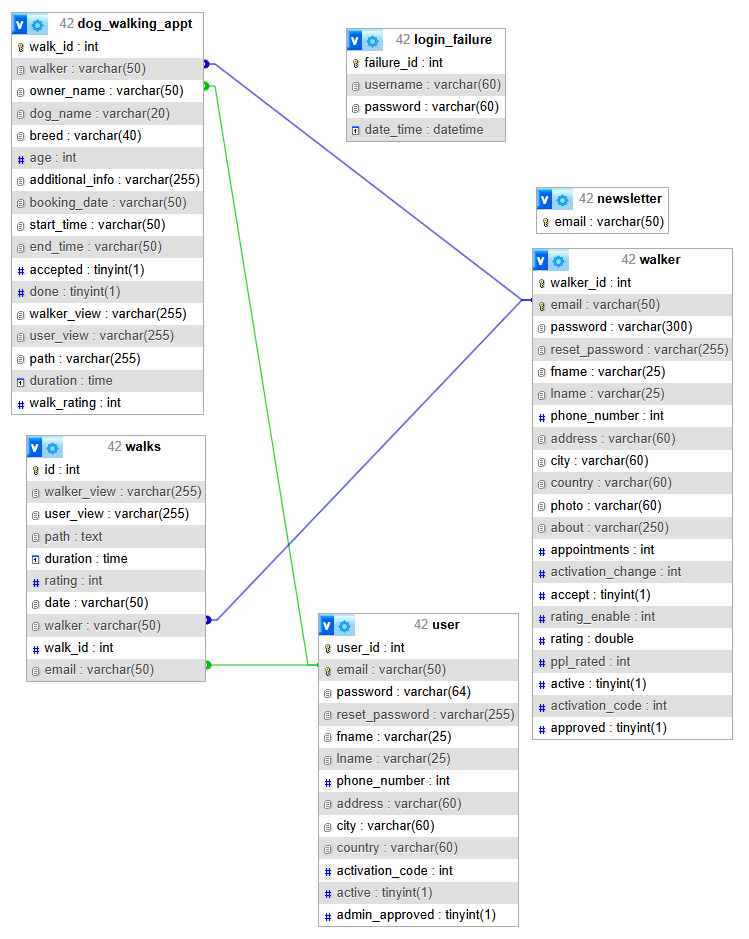
}

</style>

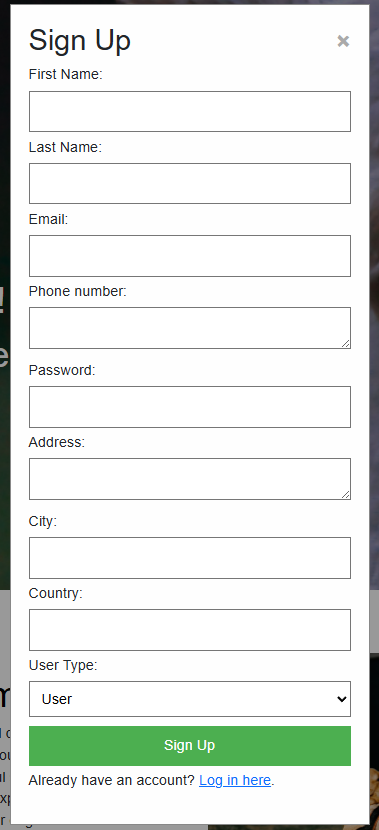
</body>

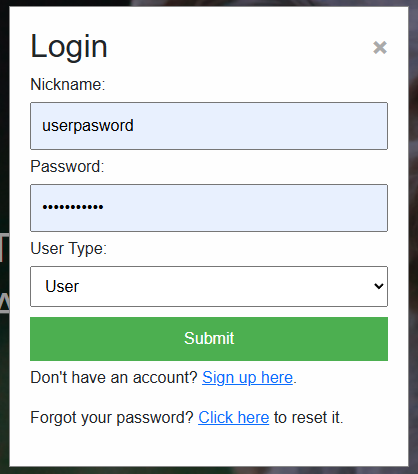
</html

# Struktura baze podataka (MySQL Workbench)



# Opis funkcionalnosti

**1.) Log In & Sign Up Forme**



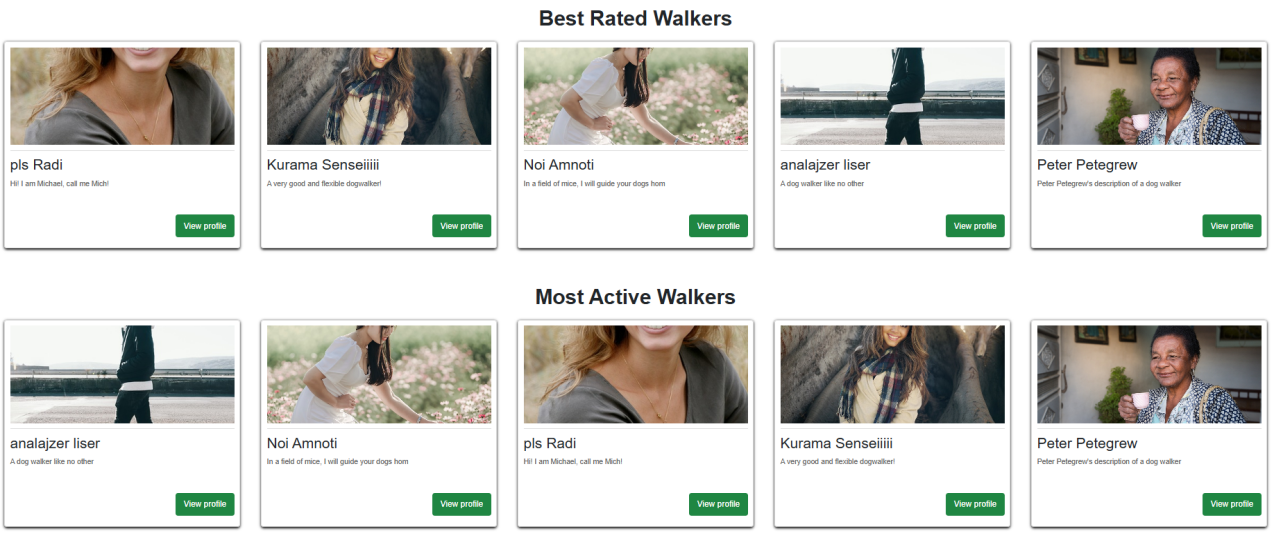
**Registracija (Sign Up):**

* Korisnici unose svoje podatke i biraju da li su obični korisnici (user) ili šetači (walker).
* Nakon što pritisnu dugme "Sign Up", dobijaju verifikacioni kod putem email-a.

**Prijava (Log In):**

* Korisnici unose svoje podatke i biraju da li su obični korisnici (user) ili šetači (walker).
* Prilikom prve prijave unose verifikacioni kod koji su dobili putem email-a.

**2.)Kartice za najbolje I najaktivnije šetače**

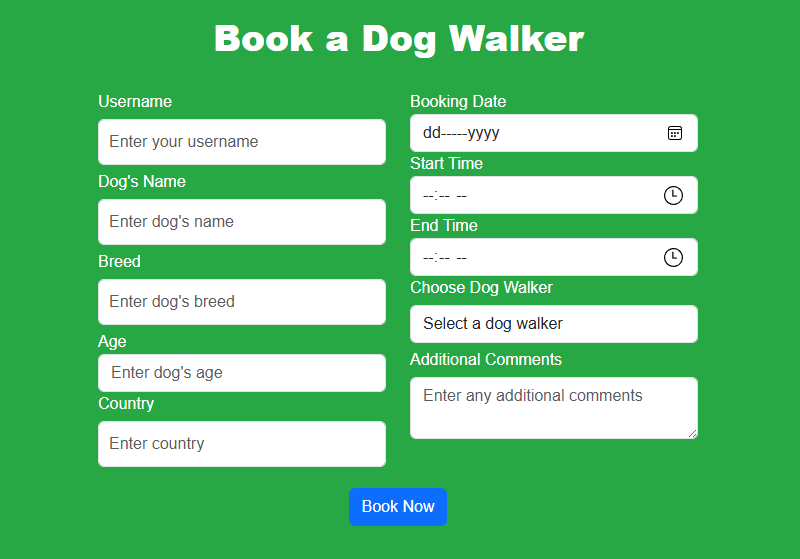


Kartice korisnicima pružaju funkcionalnost, pritiskom na "View Profile", otvara se popup prozor sa detaljnim profilom šetača pasa. Ova interakcija omogućava korisnicima da detaljno pregledaju profil šetača i ostvare kontakt s njim radi dogovora o šetnji psa.

**V**iew Profile**:** Pritiskom na ovu opciju, korisnicima se prikazuje popup prozor sa sveobuhvatnim informacijama o šetaču psa, što uključuje njegovo ime, ocene, opis usluga, kao i dostupnost.

**Zakazivanje šetnje:** Na kartici postoji opcija za zakazivanje šetnje s odabranim šetačem psa. Prilikom pritiska dugmeta "Hire" korisnik ce biti prebačen na stranicu booking.

**3.)Booking**



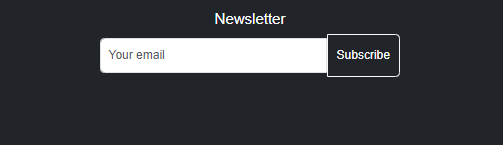
Sistem za rezervaciju šetnje pasa omogućava korisnicima da nakon unosa svih potrebnih informacija zakazuju šetnju. Korisnici mogu odabrati željenog šetača pasa za rezervaciju, a ako su došli do stranice za rezervaciju pritiskom na dugme "Hire" na kartici šetača, polje za šetača automatski se postavlja na tog šetača.

**Ključne funkcionalnosti sistema za rezervaciju:**

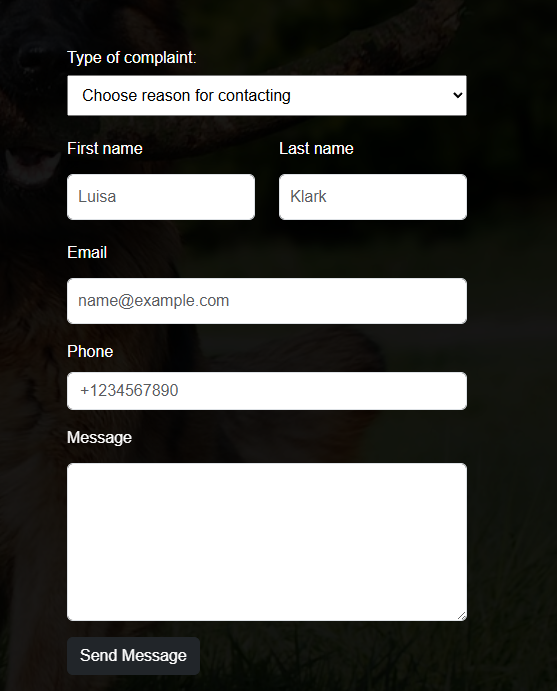
**Unos podataka:** Korisnici unose sve relevantne podatke potrebne za rezervaciju šetnje, uključujući datum, vreme, trajanje šetnje i preferencije (ako postoje).

**Izbor šetača:** Korisnici mogu pregledati profile šetača pasa i odabrati željenog šetača za rezervaciju. Pritiskom na dugme "Hire" na kartici šetača, polje za šetača u obrascu za rezervaciju automatski se popunjava informacijama tog šetača.

**4.)Newsletter**



Sistem za newsletter omogućava korisnicima da se prijave za primanje obaveštenja sa naše stranice o novim događajima i dešavanjima. Kada korisnik unese svoju email adresu u polje za newsletter, počinje da dobija redovna obaveštenja putem email-a.



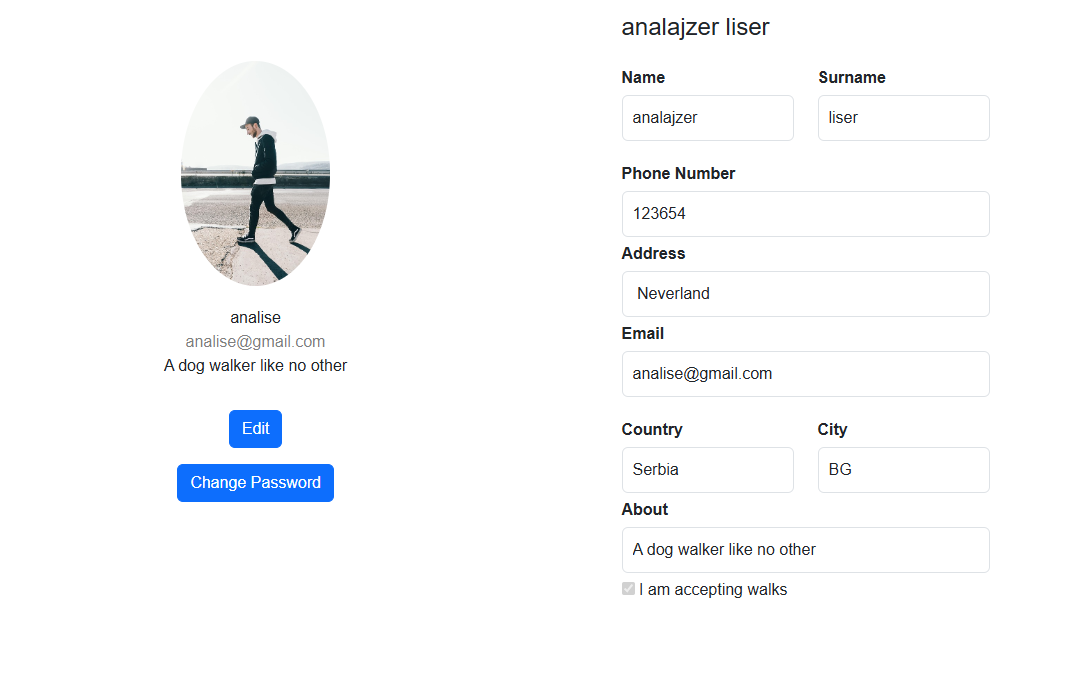
**5.)Kontakt Forma**

· **Popunjavanje podataka:** Korisnici unose svoje ime, email adresu, naslov poruke i tekst poruke u odgovarajuća polja forme.

· **Slanje poruke:** Nakon popunjavanja formulara, korisnik može poslati poruku pritiskom na dugme za slanje.

·

**6.)Profil korisnika/šetača**



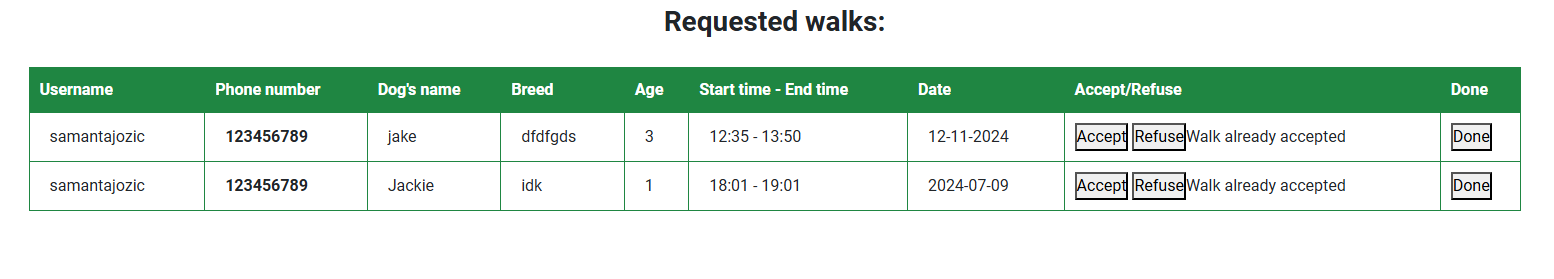
**Uređivanje informacija:** Korisnici i šetači pasa mogu pristupiti svom profilu i izmeniti sve relevantne informacije, kao što su ime, kontakt podaci, biografski podaci, slike i ostale detalje koji su im bitni.

**Promena lozinke:** Na profilu postoji opcija za promenu lozinke. Korisnici mogu zatražiti promenu lozinke unoseći trenutnu lozinku i novu lozinku koju žele postaviti.

**Brisanje profila:** Korisnici imaju mogućnost da obrišu svoj profil sa platforme. Ova funkcija omogućava korisnicima da trajno uklone svoje podatke i prestanu koristiti usluge sistema.

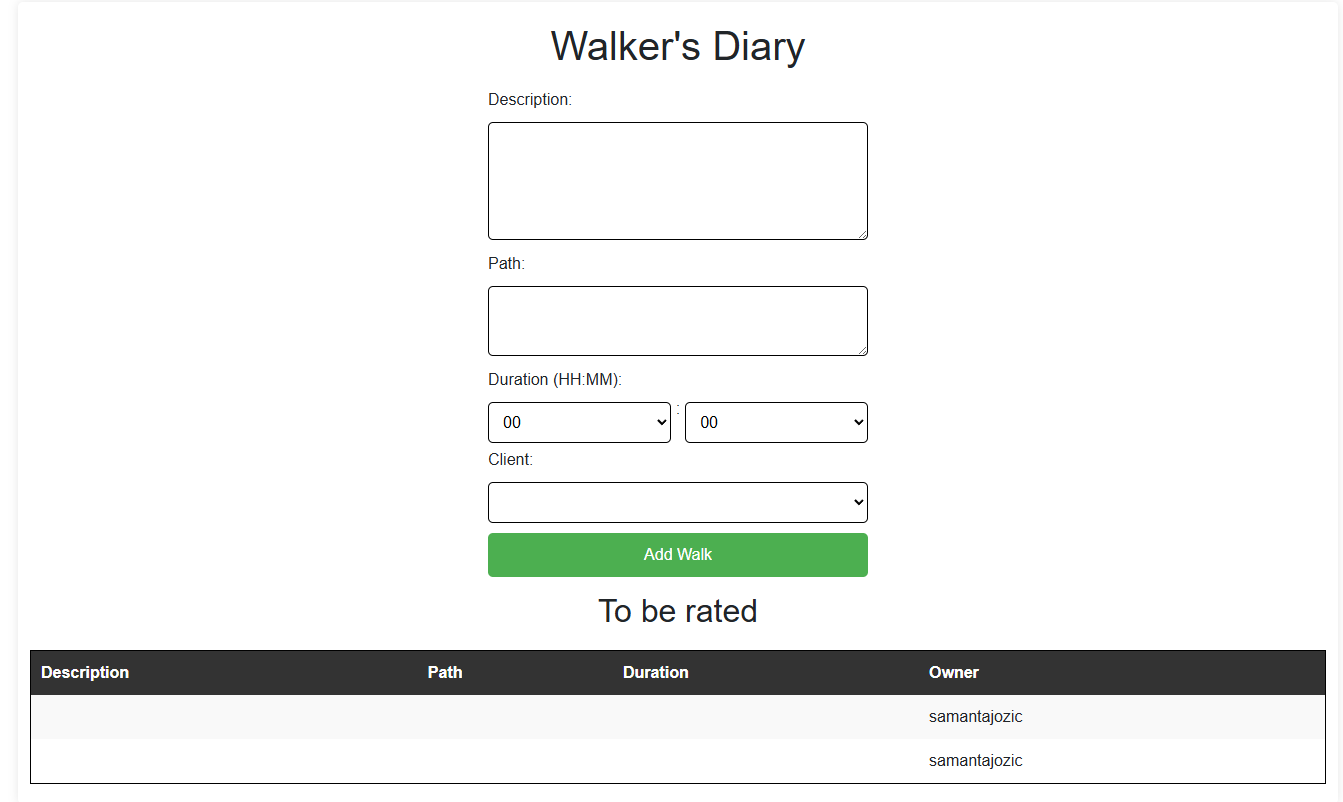
**Čuvanje promena:** Nakon što izvrše željene promene na svom profilu ili promene lozinke, korisnici i šetači pasa mogu potvrditi svoje promene i sačuvati ih.

**7.)Potvrda o setnji psa**



Sistem omogućava šetačima pasa da potvrde ili odbiju određenu šetnju, dok korisnici dobijaju email obaveštenje o statusu šetnje - da li je šetnja prihvaćena ili odbijena.

**8.)Šetačev dnevnik**



****Dnevnik šetača:****

**Vođenje zapisnika:** Šetači mogu voditi kratak zapisnik o svakoj šetnji koju obave. To uključuje beleženje vremena šetnje, lokacije, aktivnosti i opšti utisak o šetnji.

**Ocena kvaliteta šetnje:** Nakon svake šetnje, šetač može oceniti kvalitet šetnje na osnovu različitih faktora kao što su ponašanje psa, uslovi šetnje i interakcija sa psom

**Dnevnik korisnika:**

**Vođenje zapisnika:** Korisnici mogu da ocenjuju svaku šetnju koju njihov pas ima sa određenim šetačem. Ovo uključuje beleženje zadovoljstva sa uslugom šetača, ponašanjem psa tokom i posle šetnje.

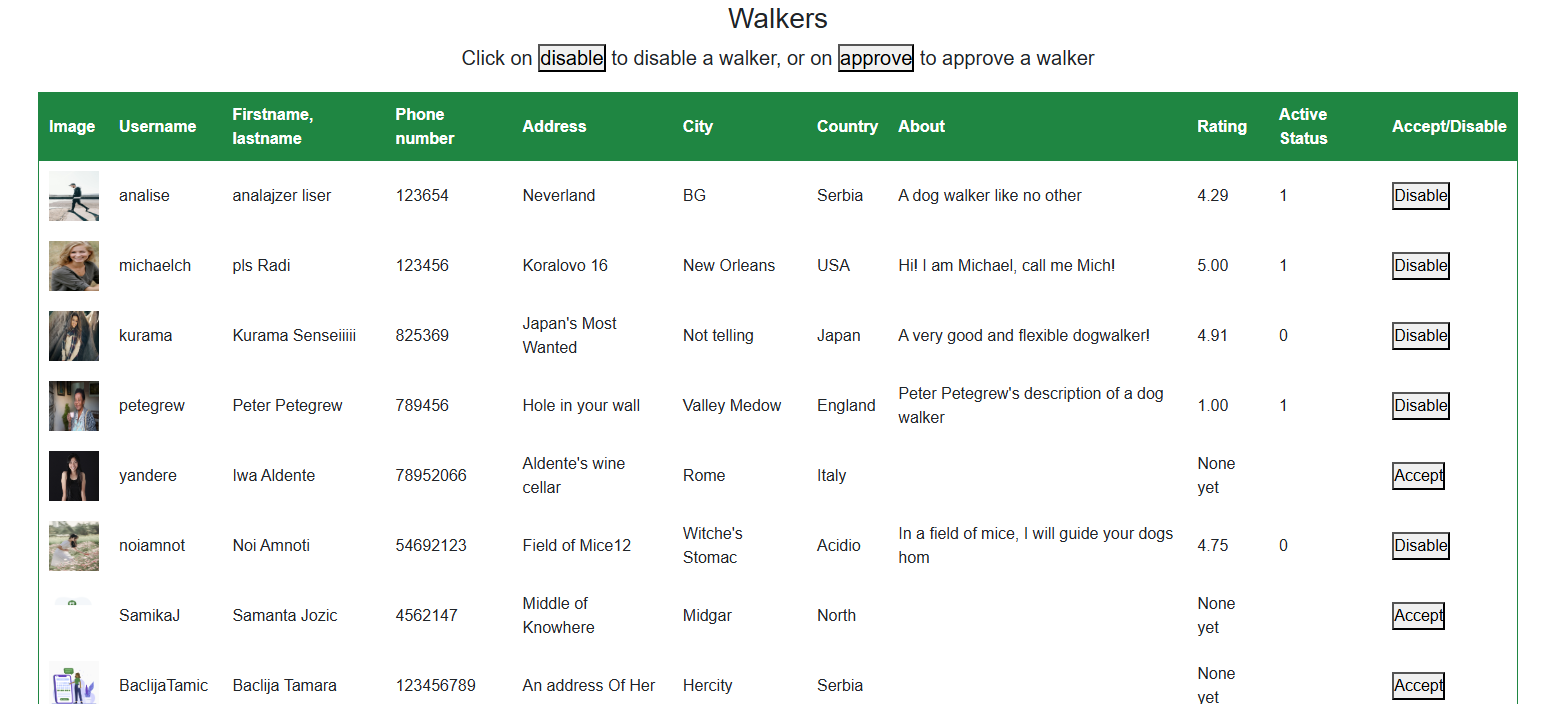
**Ocena šetača:** Korisnici mogu oceniti šetača na osnovu iskustva sa njim, što uključuje pouzdanost, pažnju prema psu i opšti utisak o njihovoj profesionalnosti.

1. **)Admin tabele**

**Tablica za odobravanje korisnika:**

**Pregled zahteva:** Administrator može videti sve zahteve koje korisnici podnose za kreiranje profila.

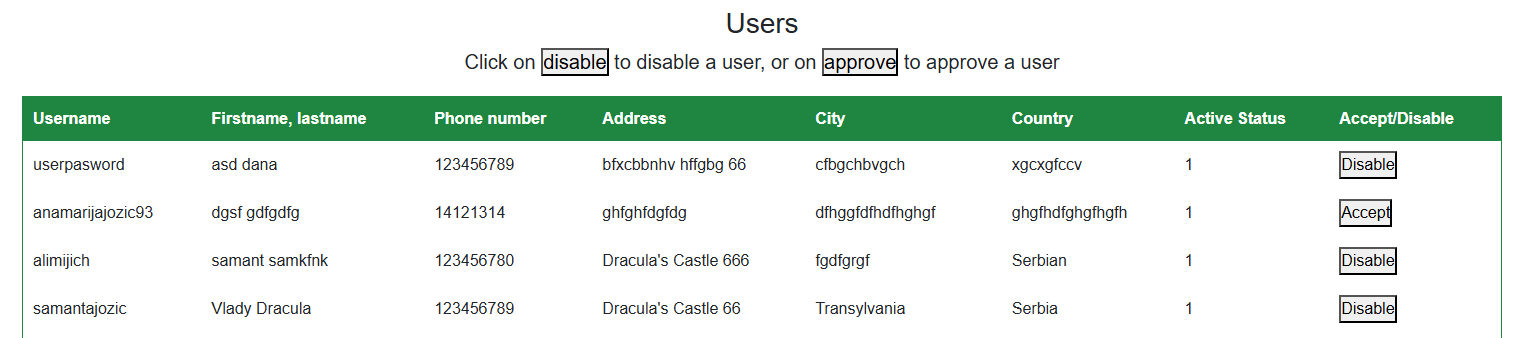
**Odobravanje ili odbijanje zahteva:** Administrator ima mogućnost da prihvati ili odbije zahtev za kreiranje profila korisnika na osnovu zadatih kriterijuma ili pravila platforme.



**Tablica za odobravanje šetača pasa:**

**Pregled zahteva:** Administrator može pregledati sve zahteve koje šetači pasa podnose za registraciju na platformi.

**Odobravanje ili odbijanje zahteva:** Administrator može odlučiti da prihvati ili odbije zahtev za registraciju šetača pasa, uzimajući u obzir aspekte kao što su isk



# Korišćena literatura

W1. <http://www.w3schools.com/>

W2. <http://www.jquery.com>

W3. <https://getbootstrap.com/>

W4. https://www.php.net/