**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

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

**PROJEKAT**

**iz predmeta Web programiranje**

|  |  |
| --- | --- |
| **KANDIDATI (II)**  **Hana Jakovetić 26123039**  **Bojan Kovačić 26123038** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SUBOTICA, 2025.** | |

Sadržaj:

[1. Opis zadatka 3](#_Toc4957)

[2. Realizacija zadatka 4](#_Toc25580)

[3. Struktura baze podataka (MySQL Workbench) 24](#_Toc11757)

[4. Opis funkcionalnosti 26](#_Toc29774)

[5. Korišćena literatura 37](#_Toc28082)

# Opis zadatka

Projektni zadatak je bio da se razvije responzivna web aplikacija za veterinarsku ordinaciju, koja omogućava korisnicima jednostavno zakazivanje termina za tretmane ljubimaca, upravljanje korisničkim i veterinarskim nalozima, kao i evidenciju tretmana i usluga. Aplikacija obezbeđuje različite nivoe pristupa (gost, korisnik, veterinar, administrator), sigurnu registraciju, automatsku email verifikaciju, rad sa elektronskim kartonom ljubimaca, kao i upravljanje raspoloživim terminima i uslugama.

Aplikacija je izrađena korišćenjem HTML-a, CSS-a, JavaScript-a (sa Fetch API/AJAX), PHP-a (OOP, PDO), MySQL baze i PHPMailer biblioteke, u skladu sa zadatim smernicama predmeta Web programiranje.

**Web adresa projekta:**  
<https://cerouno.stud.vts.su.ac.rs/>

# Realizacija zadatka

**Courier, veličina slova 10**.

· povezivanje na bazu

<?php

class DBConfig {

private string $host = 'localhost';

private string $dbname = 'cerouno';

private string $username = 'cerouno';

private string $password = 'iLWnRwC0ZfrDEyG';

private ?PDO $pdo = null;

public function \_\_construct() {

try {

$dsn = "mysql:host={$this->host};dbname={$this->dbname};charset=utf8mb4";

$this->pdo = new PDO($dsn, $this->username, $this->password, [

PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC

]);

} catch (PDOException $e) {

exit("❌ Greška prilikom povezivanja sa bazom: " . $e->getMessage());

}

}

public function getConnection(): PDO {

return $this->pdo;

}

}

· validacija korisničkog unosa

<?php

session\_start();

require\_once 'db\_config.php';

require\_once 'functions.php';

$ordinacija = new VeterinarskaOrdinacija();

$message = '';

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

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

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

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$message = "Molimo unesite validnu email adresu kao korisničko ime.";

} else {

$role = $ordinacija->loginUser($email, $password);

if ($role === 'not\_active') {

$message = "⚠️ Vaš nalog nije aktiviran. Proverite svoj e-mail i kliknite na aktivacioni link.";

} elseif ($role === 1) {

header("Location: manage\_users.php");

exit;

} elseif ($role === 2 && isset($\_SESSION['vet\_id'])) {

header("Location: vet\_profile.php");

exit;

} elseif ($role === 3) {

header("Location: pet\_information.php");

exit;

} else {

$message = "Pogrešan email ili lozinka ili nedozvoljena uloga.";

}

}

}

?>

· registracija korisnika sa email verifikacijom

<?php

session\_start();

require\_once 'db\_config.php';

require\_once 'functions.php';

$message = '';

$ordinacija = new VeterinarskaOrdinacija();

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

$first = trim($\_POST['first\_name']);

$last = trim($\_POST['last\_name']);

$email = trim($\_POST['email']);

$phone = trim($\_POST['phone']);

$address = trim($\_POST['address']);

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

$confirm = $\_POST['confirm\_password'];

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$message = "Email adresa nije validna.";

} elseif (!$address) {

$message = "Adresa je obavezna.";

} elseif ($phone !== '' && !preg\_match('/^[0-9]+$/', $phone)) {

$message = "Broj telefona može sadržati samo cifre.";

} elseif ($password !== $confirm) {

$message = "Lozinke se ne poklapaju.";

} else {

$result = $ordinacija->registerUser($first, $last, $email, $phone, $address, $password);

if ($result === true) {

header("Location: login.php?registered=true");

exit;

} else {

$message = $result;

}

}

}

· rezervacija termina i slanje koda na email, kao i rad sa Fetch API ili AJAX-om

<?php

session\_start();

require\_once 'db\_config.php';

require\_once 'functions.php';

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

ini\_set('display\_errors', 1);

ini\_set('display\_startup\_errors', 1);

error\_reporting(E\_ALL);

$db = new DBConfig();

$pdo = $db->getConnection();

$ord = new VeterinarskaOrdinacija($pdo);

$user\_id = $\_SESSION['user\_id'];

$veterinari = $ord->getAllVets();

$pets = $ord->getPetsByUser($user\_id);

$services = $ord->getAllServices();

// AJAX: Get free terms

if ($\_SERVER['REQUEST\_METHOD'] === 'GET' && ($\_GET['action'] ?? '') === 'slots') {

$vet\_id = (int)($\_GET['vet\_id'] ?? 0);

$date = $\_GET['date'] ?? '';

$slots = $ord->getAvailableSlots($vet\_id, $date);

echo json\_encode($slots);

exit;

}

// AJAX: Create new term

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && ($\_POST['action'] ?? '') === 'create') {

$slot\_id = (int)($\_POST['slot\_id'] ?? 0);

$pet\_id = (int)($\_POST['pet\_id'] ?? 0);

$date = $\_POST['date'] ?? '';

$service\_id = (int)($\_POST['service\_id'] ?? 0);

$result = $ord->createAppointment($user\_id, $pet\_id, $slot\_id, $date, $service\_id);

echo json\_encode($result);

exit;

}

?>

<!DOCTYPE html>

<html lang="sr">

<head>

<meta charset="UTF-8" />

<title>Rezervacije</title>

<link rel="stylesheet" href="css/css.css" />

</head>

<body>

<header>

<div class="logo">🐾 PetCare</div>

<nav>

<ul>

<li><a href="pet\_information.php">Moj ljubimac</a></li>

<li><a href="pet\_treatments.php">Tretman ljubimca</a></li>

<li><a href="users\_reservation.php" class="active">Rezervacije</a></li>

<li><a href="change\_reservation.php">Promeni rezervaciju</a></li>

<li><a href="user\_information.php">Moj nalog</a></li>

<li><a href="logout.php">Odjavi se</a></li>

</ul>

</nav>

</header>

<main style="padding: 20px;">

<h2>Rezerviši termin</h2>

<form id="reservationForm" class="form-section vet-schedule-form">

<label for="vet\_id" class="form-label vet-form-label">Veterinar:</label>

<select id="vet\_id" name="vet\_id" required class="form-input vet-form-input">

<option value="" disabled selected>-- Izaberi --</option>

<?php foreach ($veterinari as $v): ?>

<option value="<?= $v['vet\_id'] ?>"><?= htmlspecialchars($v['first\_name'] . ' ' . $v['last\_name']) ?></option>

<?php endforeach; ?>

</select>

<label for="date" class="form-label vet-form-labell">Datum:</label>

<input type="date" id="date" name="date" required min="<?= date('Y-m-d') ?>" max="<?= date('Y-m-d', strtotime('+30 days')) ?>">

<button type="button" onclick="fetchSlots()" class="cta-button vet-cta-button">Prikaži termine</button>

<div id="slotsContainer" style="margin-top:20px;"></div>

<div id="slotsContainer" style="margin-top:20px;"></div>

<label for="pet\_id">Ljubimac:</label>

<select id="pet\_id" name="pet\_id" required class="form-input vet-form-input">

<option value="">-- Izaberi --</option>

<?php foreach ($pets as $p): ?>

<option value="<?= $p['id'] ?>"><?= htmlspecialchars($p['name']) ?></option>

<?php endforeach; ?>

</select>

<label for="service\_id">Usluga:</label>

<select id="service\_id" name="service\_id" required class="form-input vet-form-input" >

<option value="">-- Izaberi --</option>

<?php foreach ($services as $s): ?>

<option value="<?= $s['id'] ?>"><?= htmlspecialchars($s['name']) ?></option>

<?php endforeach; ?>

</select>

<button type="submit" class="cta-button vet-cta-button">Rezerviši</button>

</form>

<div id="msgBox" style="margin-top:20px; font-weight: bold;"></div>

</main>

<footer class="custom-footer" >

<div class="footer-content">

&copy; 2025 PetCare Ordinacija. Sva prava zadržana.

</div>

</footer>

<script src="js/users\_reservation.js"></script>

</body>

</html>

public function getAllVets(): array

{

$stmt = $this->pdo->query("

SELECT v.id AS vet\_id, u.first\_name, u.last\_name, u.email

FROM veterinarians v

JOIN users u ON v.user\_id = u.id

ORDER BY u.first\_name, u.last\_name

");

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

public function getPetsByUser1(int $user\_id): array

{

$stmt = $this->pdo->prepare("

SELECT p.id, p.name, pt.name AS type\_name, pb.name AS breed\_name, p.age, p.gender, p.birth\_date, p.photo

FROM pets p

JOIN pet\_owners po ON p.owner\_id = po.id

JOIN pet\_types pt ON p.type\_id = pt.id

JOIN pet\_breeds pb ON p.breed\_id = pb.id

WHERE po.user\_id = ?

ORDER BY p.name

");

$stmt->execute([$user\_id]);

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

public function getAllServices(): array

{

$stmt = $this->pdo->query("SELECT \* FROM services");

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

public function getAvailableSlots(int $vet\_id, string $date): array

{

$dayOfWeekMap = [

'Monday' => 'Ponedeljak',

'Tuesday' => 'Utorak',

'Wednesday' => 'Sreda',

'Thursday' => 'Cetvrtak',

'Friday' => 'Petak',

'Saturday' => 'Subota',

'Sunday' => 'Nedelja'

];

$phpDay = date('l', strtotime($date));

$dbDay = $dayOfWeekMap[$phpDay] ?? null;

if (!$dbDay) return [];

$stmt = $this->pdo->prepare("

SELECT vs.id AS schedule\_id, vs.start\_time, vs.end\_time

FROM veterinarian\_schedule vs

WHERE vs.veterinarian\_id = :vet\_id

AND vs.day\_of\_week = :day\_of\_week

AND NOT EXISTS (

SELECT 1 FROM appointments a

WHERE a.schedule\_id = vs.id

AND a.appointment\_date = :date

AND a.status = 'zakazano'

)

ORDER BY vs.start\_time

");

$stmt->execute([

':vet\_id' => $vet\_id,

':day\_of\_week' => $dbDay,

':date' => $date

]);

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

public function createAppointment(int $user\_id, int $pet\_id, int $schedule\_id, string $date, int $service\_id): array

{

try {

$stmt = $this->pdo->prepare("

SELECT veterinarian\_id, start\_time, end\_time

FROM veterinarian\_schedule

WHERE id = ?

");

$stmt->execute([$schedule\_id]);

$slot = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$slot) {

return ['success' => false, 'message' => '❌ Termin ne postoji.'];

}

// Provera da li je termin već zauzet

$check = $this->pdo->prepare("

SELECT COUNT(\*) FROM appointments

WHERE schedule\_id = ? AND appointment\_date = ? AND status = 'scheduled'

");

$check->execute([$schedule\_id, $date]);

if ($check->fetchColumn() > 0) {

return ['success' => false, 'message' => '❌ Termin je već zauzet.'];

}

// GENERATE CODE

$code = random\_int(100000, 999999);

// CREATE TERM

$stmtInsert = $this->pdo->prepare("

INSERT INTO appointments (

user\_id, pet\_id, veterinarian\_id, schedule\_id, service\_id,

appointment\_date, start\_time, end\_time, status, reservation\_code

) VALUES (?, ?, ?, ?, ?, ?, ?, ?, 'scheduled', ?)

");

$stmtInsert->execute([

$user\_id,

$pet\_id,

$slot['veterinarian\_id'],

$schedule\_id,

$service\_id,

$date,

$slot['start\_time'],

$slot['end\_time'],

$code

]);

// SEND EMIAL

$user = $this->getUserById($user\_id);

if ($user) {

$this->sendReservationConfirmationMail($user['email'], $user['first\_name'], $code);

}

return ['success' => true, 'message' => "✅ Termin zakazan. Kod potvrde: $code"];

} catch (PDOException $e) {

return ['success' => false, 'message' => 'Greška: ' . $e->getMessage()];

}

}

"use strict";

function fetchSlots() {

const vetId = document.getElementById('vet\_id').value;

const date = document.getElementById('date').value;

if (!vetId || !date) return;

fetch(`users\_reservation.php?action=slots&vet\_id=${vetId}&date=${date}`)

.then(res => res.json())

.then(data => {

const container = document.getElementById('slotsContainer');

container.innerHTML = '';

if (!data || data.length === 0) {

container.innerHTML = "<p>Nema slobodnih termina.</p>";

return;

}

const table = document.createElement('table');

table.className = 'styled-table';

const thead = document.createElement('thead');

const headerRow = document.createElement('tr');

['Izaberite', 'Početak', 'Kraj'].forEach(text => {

const th = document.createElement('th');

th.innerText = text;

th.style.border = '1px solid #ccc';

th.style.padding = '8px';

headerRow.appendChild(th);

});

thead.appendChild(headerRow);

table.appendChild(thead);

const tbody = document.createElement('tbody');

data.forEach(slot => {

const tr = document.createElement('tr');

const tdRadio = document.createElement('td');

tdRadio.style.border = '1px solid #ccc';

tdRadio.style.padding = '8px';

const radio = document.createElement('input');

radio.type = 'radio';

radio.name = 'slot\_id';

radio.value = slot.schedule\_id;

radio.required = true;

tdRadio.appendChild(radio);

tr.appendChild(tdRadio);

const tdStart = document.createElement('td');

tdStart.innerText = slot.start\_time;

tdStart.style.border = '1px solid #ccc';

tdStart.style.padding = '8px';

tr.appendChild(tdStart);

const tdEnd = document.createElement('td');

tdEnd.innerText = slot.end\_time;

tdEnd.style.border = '1px solid #ccc';

tdEnd.style.padding = '8px';

tr.appendChild(tdEnd);

tbody.appendChild(tr);

});

table.appendChild(tbody);

container.appendChild(table);

})

.catch(error => {

console.error("Greška pri dohvaćanju termina:", error);

document.getElementById('slotsContainer').innerHTML = "<p style='color:red;'>Greška pri dohvaćanju termina.</p>";

});

}

document.getElementById('reservationForm').addEventListener('submit', function(e) {

e.preventDefault();

const vetId = document.getElementById('vet\_id').value;

const date = document.getElementById('date').value;

const petId = document.getElementById('pet\_id').value;

const serviceId = document.getElementById('service\_id').value;

const slotRadio = document.querySelector('input[name="slot\_id"]:checked');

const box = document.getElementById('msgBox');

if (!vetId || !date || !petId || !slotRadio || !serviceId) {

box.style.color = 'red';

box.innerText = '⚠️ Morate popuniti sva polja i izabrati termin.';

return;

}

const formData = new FormData();

formData.append('action', 'create');

formData.append('vet\_id', vetId);

formData.append('date', date);

formData.append('pet\_id', petId);

formData.append('slot\_id', slotRadio.value);

formData.append('service\_id', serviceId);

fetch('users\_reservation.php', {

method: 'POST',

body: formData

})

.then(res => res.json())

.then(data => {

box.style.color = data.success ? 'green' : 'red';

box.innerText = data.message;

if (data.success) {

document.getElementById('slotsContainer').innerHTML = '';

document.getElementById('reservationForm').reset();

}

})

.catch(error => {

console.error("Greška pri zakazivanju:", error);

box.style.color = 'red';

box.innerText = 'Greška pri zakazivanju termina.';

});

});

· provera termina

<?php

session\_start();

require\_once 'auth.php';

requireVeterinarian();

require\_once 'db\_config.php';

require\_once 'functions.php';

if (!isset($\_SESSION['vet\_id'])) {

header("Location: login.php");

exit;

}

$vet\_id = $\_SESSION['vet\_id'];

$appointment\_id = (int)($\_GET['appointment\_id'] ?? 0);

$error = '';

$details = null;

$code\_input = '';

$db = new DBConfig();

$pdo = $db->getConnection();

$ordinacija = new VeterinarskaOrdinacija($pdo);

if ($appointment\_id <= 0) {

die("❌ Nevažeći ID termina.");

}

if (isset($\_SESSION['allowed\_appointments'][$appointment\_id])) {

$access\_granted = true;

} else {

$access\_granted = false;

}

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

$code\_input = trim($\_GET['code\_input'] ?? '');

} elseif ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$code\_input = trim($\_POST['code\_input'] ?? '');

}

if (!$access\_granted) {

if ($code\_input === '') {

$show\_form = true;

} else {

$stmt = $pdo->prepare("SELECT reservation\_code FROM appointments WHERE id = ? AND veterinarian\_id = ?");

$stmt->execute([$appointment\_id, $vet\_id]);

$row = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$row) {

die("❌ Termin nije pronađen ili nemate pristup.");

}

if ($row['reservation\_code'] !== $code\_input) {

$error = "❌ Nije tačan kod za ovaj termin.";

$show\_form = true;

} else {

$\_SESSION['allowed\_appointments'][$appointment\_id] = true;

$access\_granted = true;

$show\_form = false;

}

}

} else {

$show\_form = false;

}

if ($access\_granted) {

$details = $ordinacija->getTreatmentDetails($appointment\_id, $vet\_id);

if (!$details) {

die("❌ Nema dostupnih podataka za ovaj termin.");

}

$stmt = $pdo->prepare("SELECT pet\_id FROM appointments WHERE id = ?");

$stmt->execute([$appointment\_id]);

$appointmentData = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$appointmentData) {

die("❌ Termin nije pronađen.");

}

$pet\_id = $appointmentData['pet\_id'];

$treatments = $ordinacija->getAllTreatments();

$records = $ordinacija->getTreatmentRecords($appointment\_id);

if (isset($\_GET['delete\_id'])) {

$delete\_id = (int)$\_GET['delete\_id'];

$ordinacija->deleteTreatmentRecord($delete\_id);

header("Location: vet\_treatments\_details.php?appointment\_id=$appointment\_id&code\_input=" . urlencode($code\_input));

exit;

}

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

if (isset($\_POST['no\_show']) && $\_POST['no\_show'] == '1') {

$ordinacija->addPenaltyToOwner((int)$details['user\_id']);

header("Location: vet\_treatments\_details.php?appointment\_id=$appointment\_id&code\_input=" . urlencode($code\_input));

exit;

}

$diagnosis = trim($\_POST['diagnosis'] ?? '');

$treatment\_id = (int)($\_POST['treatment\_id'] ?? 0);

$selected\_treatment = array\_filter($treatments, function($t) use ($treatment\_id) {

return $t['id'] == $treatment\_id;

});

$selected\_treatment = reset($selected\_treatment);

if ($diagnosis && $selected\_treatment) {

$ordinacija->saveTreatmentDetails(

$appointment\_id,

$vet\_id,

$pet\_id,

$diagnosis,

$selected\_treatment['name'],

(float)$selected\_treatment['price']

);

header("Location: vet\_treatments\_details.php?appointment\_id=$appointment\_id&code\_input=" . urlencode($code\_input));

exit;

} else {

$error = "Molimo unesite sve potrebne podatke.";

}

}

}

?>

<!DOCTYPE html>

<html lang="sr">

<head>

<meta charset="UTF-8" />

<title>Detalji tretmana</title>

<link rel="stylesheet" href="css/css.css" />

</head>

<body>

<header>

<div class="logo">🐾 PetCare</div>

<nav>

<ul>

<li><a href="vet\_treatments\_info.php">Tretmani</a></li>

<li><a href="vet\_electronic\_card.php" class="active">Karton</a></li>

<li><a href="vet\_profile.php">Vet profil</a></li>

<li><a href="vet\_schedule.php">Radno vreme</a></li>

<li><a href="logout.php">Odjavi se</a></li>

</ul>

</nav>

</header>

<?php if ($show\_form): ?>

<h2>Unesite kod za pristup kartonu</h2>

<?php if ($error): ?>

<p style="color:red;"><?= htmlspecialchars($error) ?></p>

<?php endif; ?>

<form method="get" action="">

<input type="hidden" name="appointment\_id" value="<?= htmlspecialchars($appointment\_id) ?>" />

<input type="text" name="code\_input" required autocomplete="off" placeholder="Kod potvrde" style="width: 300px;" />

<button type="submit">Potvrdi</button>

</form>

<?php else: ?>

<h2>Detalji tretmana</h2>

<div class="container" style="display: flex; gap: 20px; padding: 20px;">

<div style="width: 25%;">

<img src="images/pets/<?= htmlspecialchars($details['photo']) ?>" alt="Slika ljubimca" style="max-width: 100%; border-radius: 10px;" />

</div>

<div style="width: 75%;">

<p><strong>Ljubimac:</strong> <?= htmlspecialchars($details['pet\_name']) ?></p>

<p><strong>Vlasnik:</strong> <?= htmlspecialchars($details['owner\_name']) ?></p>

<p><strong>Datum:</strong> <?= date('Y-m-d', strtotime($details['appointment\_date'])) ?></p>

<p><strong>Početak:</strong> <?= date('H:i', strtotime($details['start\_time'])) ?></p>

<p><strong>Kraj:</strong> <?= date('H:i', strtotime($details['end\_time'])) ?></p>

<?php if ($error): ?>

<p style="color:red;"><?= htmlspecialchars($error) ?></p>

<?php endif; ?>

<!-- Form to submit new note -->

<form method="post" id="noteForm">

<input type="hidden" name="code\_input" value="<?= htmlspecialchars($code\_input) ?>" />

<label for="diagnosis">Beleška (dijagnoza):</label><br />

<textarea name="diagnosis" id="diagnosis" rows="4" cols="50" required><?= htmlspecialchars($\_POST['diagnosis'] ?? '') ?></textarea><br /><br />

<label for="treatment\_id">Tretman:</label>

<select name="treatment\_id" id="treatment\_id" class="form-input" required>

<option value="">-- Izaberite tretman --</option>

<?php foreach ($treatments as $t): ?>

<option value="<?= $t['id'] ?>" <?= (isset($\_POST['treatment\_id']) && $\_POST['treatment\_id'] == $t['id']) ? 'selected' : '' ?>>

<?= htmlspecialchars($t['name']) ?> (<?= number\_format($t['price'], 2) ?> RSD)

</option>

<?php endforeach; ?>

</select><br /><br />

<button type="submit" id="submitBtn" class="cta-button" style="height: 40px; width: 200px;">Unesi napomenu</button>

</form>

<!-- Display previous notes -->

<h3>Prethodne beleške</h3>

<?php if ($records): ?>

<table border="1" cellpadding="8" cellspacing="0">

<thead>

<tr>

<th>Datum</th>

<th>Veterinar</th>

<th>Dijagnoza</th>

<th>Tretman</th>

<th>Cena</th>

<th>Akcije</th>

</tr>

</thead>

<tbody>

<?php foreach ($records as $r): ?>

<tr>

<td><?= htmlspecialchars($r['created\_at']) ?></td>

<td><?= htmlspecialchars($r['first\_name'] . ' ' . $r['last\_name']) ?></td>

<td><?= htmlspecialchars($r['diagnosis']) ?></td>

<td><?= htmlspecialchars($r['treatment']) ?></td>

<td><?= number\_format($r['price'], 2) ?> RSD</td>

<td>

<form method="get" style="display:inline-block;">

<input type="hidden" name="appointment\_id" value="<?= htmlspecialchars($appointment\_id) ?>" />

<input type="hidden" name="delete\_id" value="<?= $r['id'] ?>" />

<button type="submit" class="cta-button" style="height: 40px; width: 100px;" onclick="return confirm('Da li ste sigurni?')">Obriši</button>

</form>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<?php else: ?>

<p>Još nema unetih beleški.</p>

<?php endif; ?>

</div>

</div>

<?php endif; ?>

<script src="js/vet\_treatments\_details.js"></script>

</body>

</html>

· primer klijentske validacije...

"use strict";

document.querySelector('.login-form').addEventListener('submit', function(e) {

const emailInput = this.username.value.trim();

const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailPattern.test(emailInput)) {

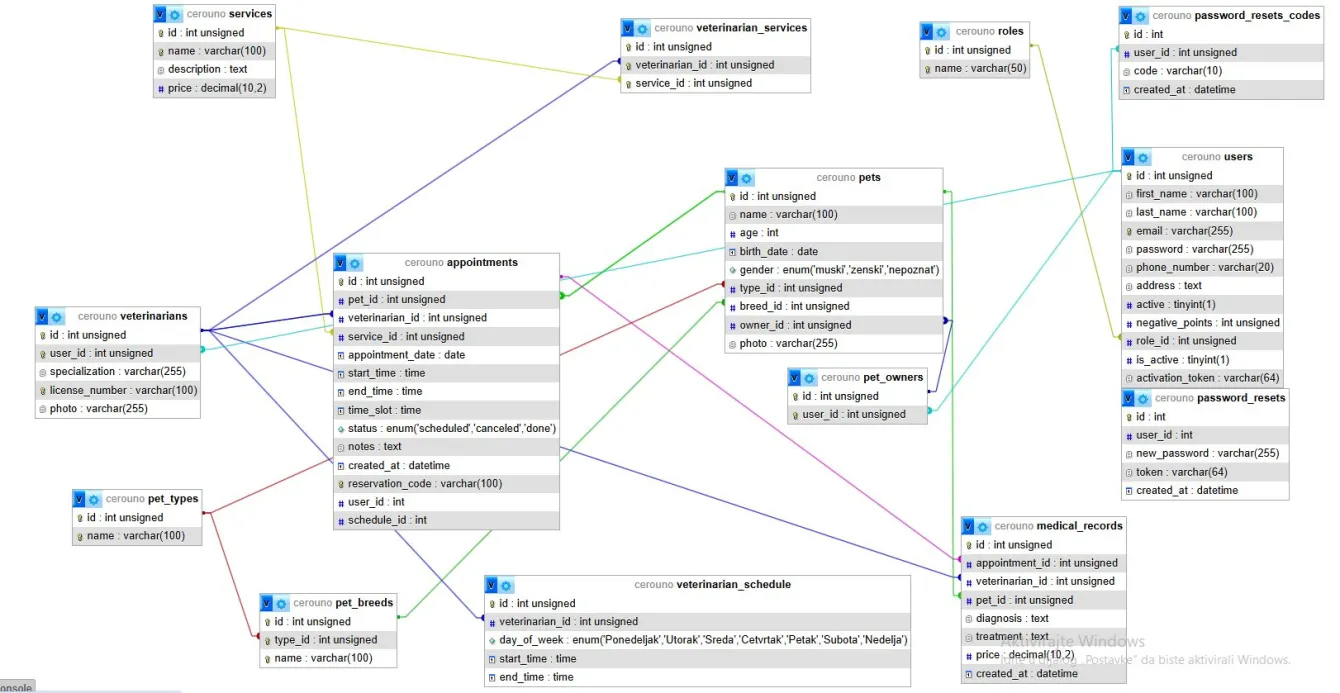
alert('Molimo unesite validnu email adresu kao korisničko ime.');

e.preventDefault();

}

});

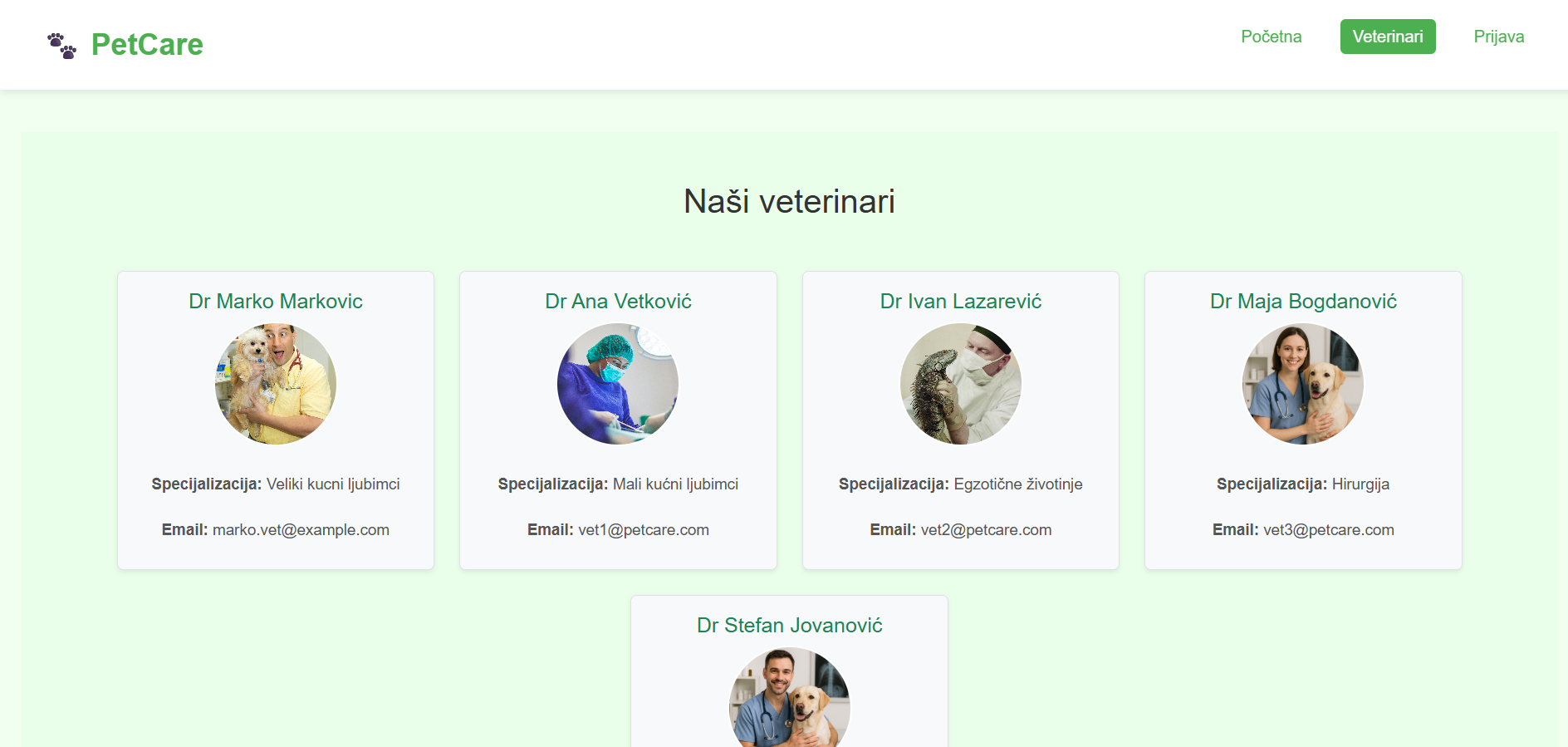
# Struktura baze podataka (MySQL Workbench)

****

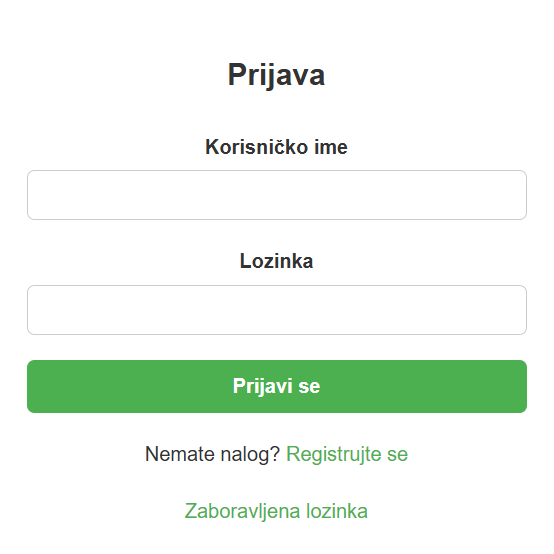
# Opis funkcionalnosti



Početna stranica je stranica koju mogu da vide svi, odnosno to je stranica koju vidi svako ko nije prijavljen na sistem, svi gosti. Tu se gostu sajta pokazuje poruka dobrodošlice, lista usluga koje naša veterinarska ordinacija “PetCare” pruža, kao i galerija slika.

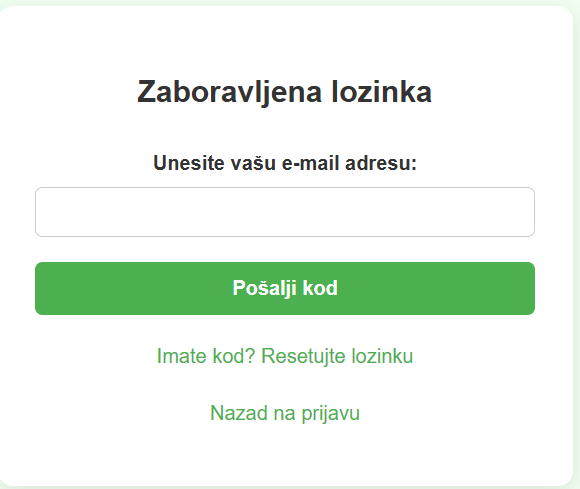


Stranica veterinari prikazuje gostu sajta sve zaposlene veterinare u ordinaciji, njhovu specijalizaciju i mejl.

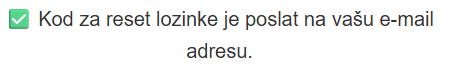


Prijava na sistem se vrši tako što korisnik unosi svoj email kao korisničko ime, a zatim unosi lozinku.

Klikom na link “Zaboravljena lozinka” korisnik može da postavi novu lozinku ukoliko je staru zaboravio.



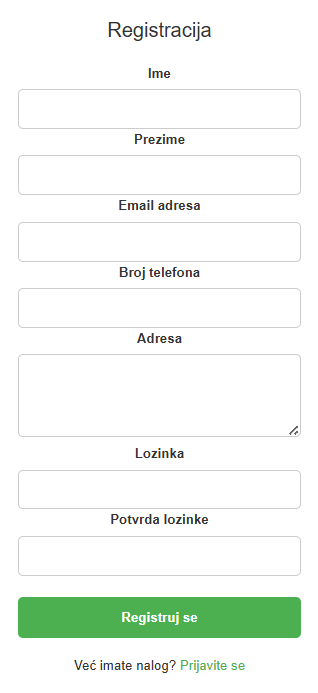
Kada klikne na link, otvara mu se nova forma gde mora uneti svoje korisničko ime (mejl) na koji će mu nakon zahteva stići mejl sa kodom za promenu lozinke.



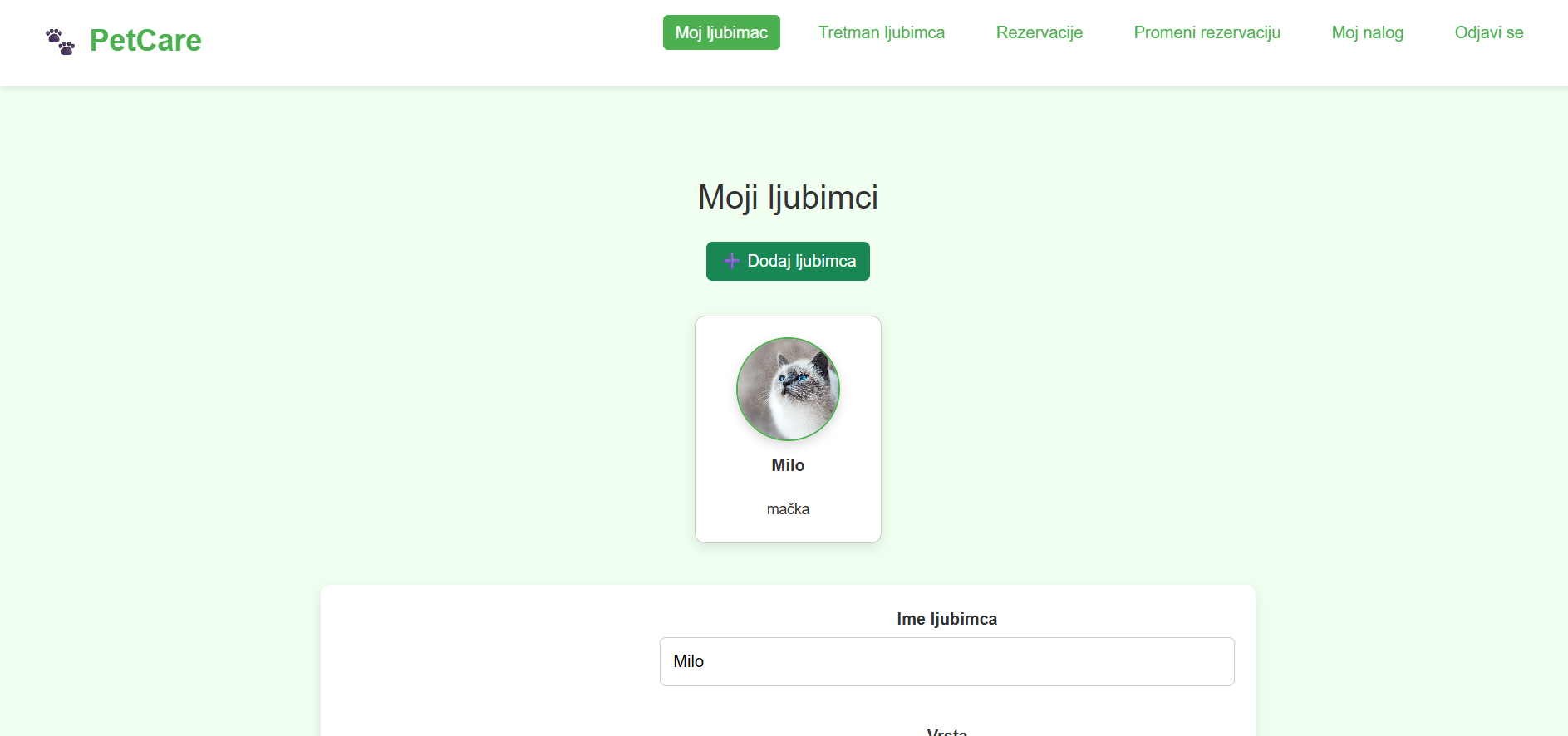
Nakon toga, potrebno je da klikne na link “Imate kod? Resetujte lozinku” i tada mu se otvara forma gde je potrbno uneti kod I novu lozinku.



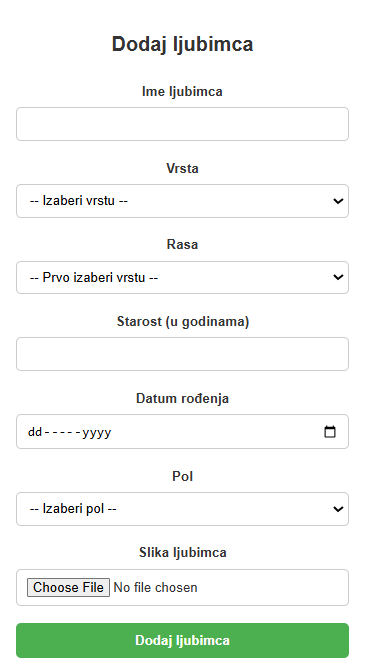
Ukoliko korisnik nema nalog, može da ga napravi tako što će se registrovati na sajt.

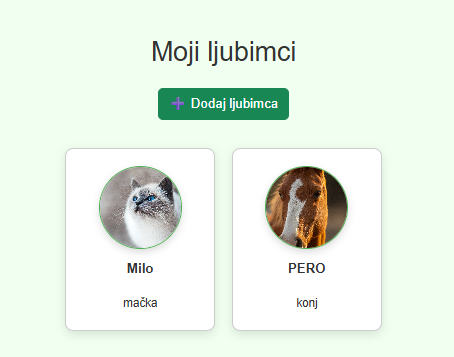


Potrebni podaci su ime, prezime, broj telefona, imejl (koji mora biti jedinstveni, odnosno korisnik ne može imati više naloga sa jednim imejlom), adresa i lozinka.

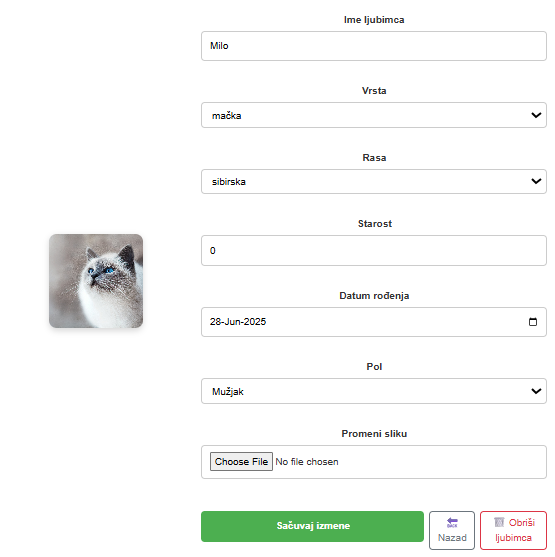


Nakon prijave, korinsiku se otvara stranica ljubimac, gde klikom na zeleno dugme može da unese podatke o svom kućnom ljubimcu.

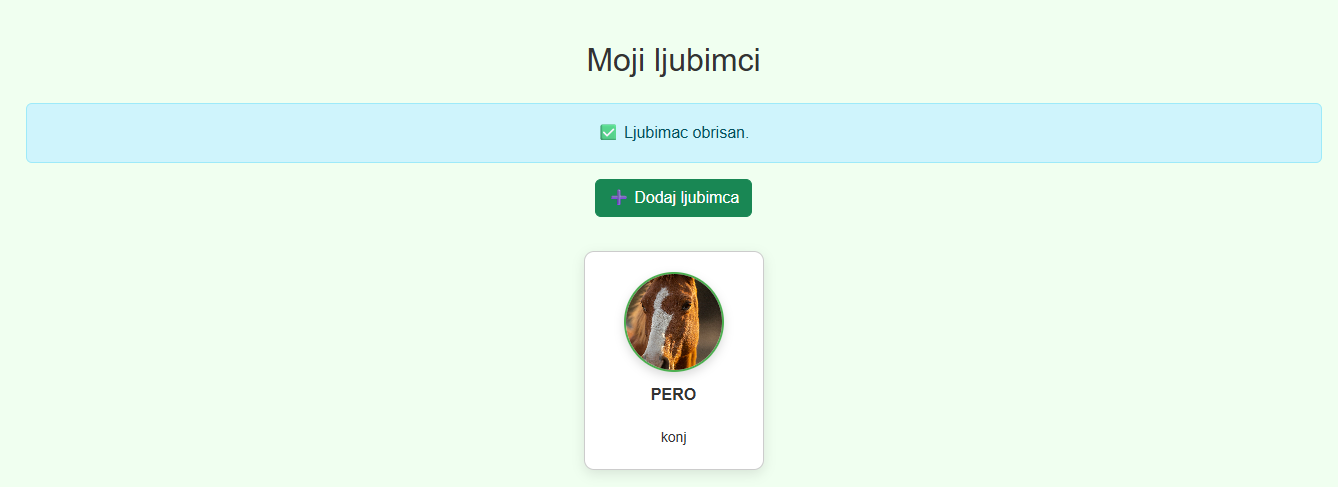




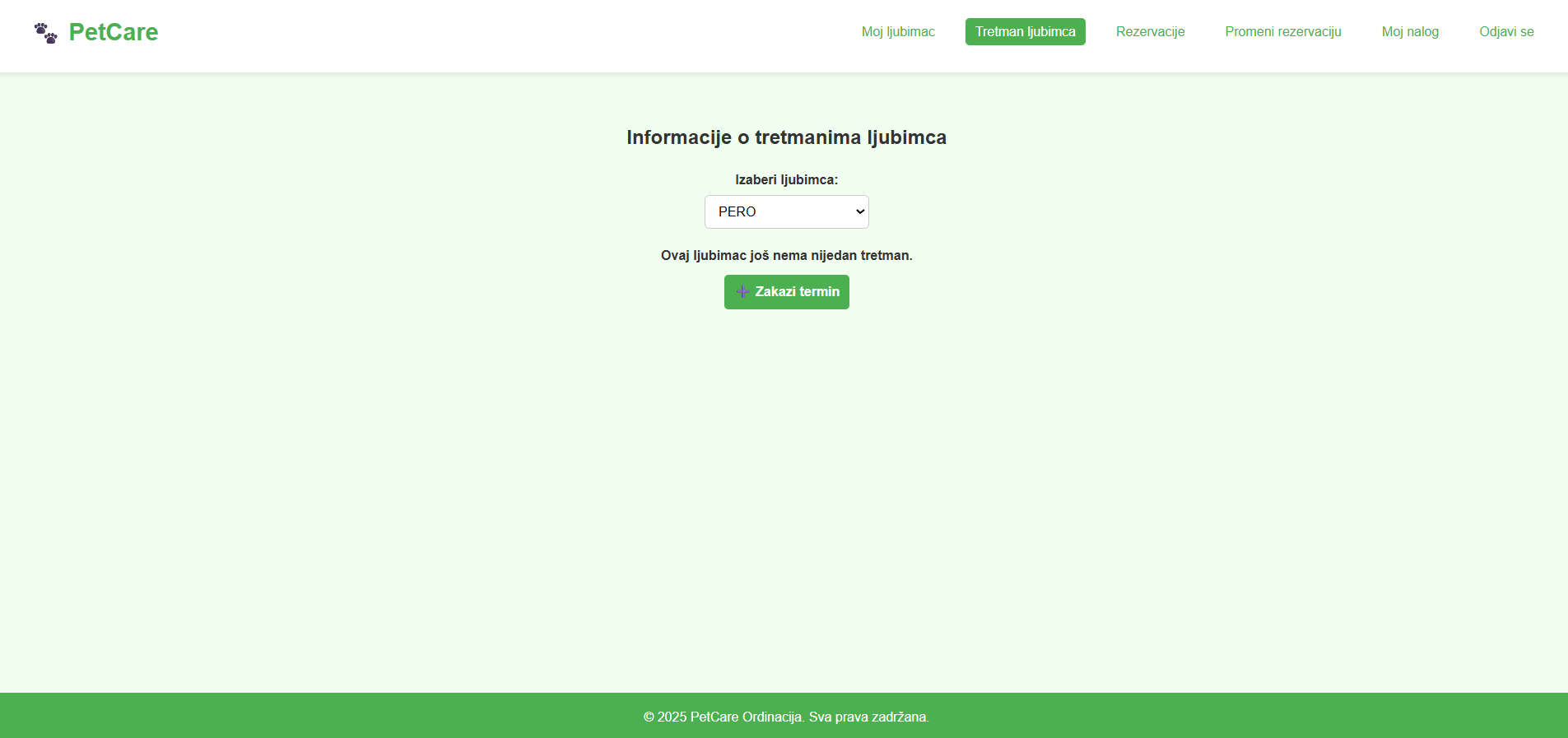
Kada korisnik unese podatke o ljubimcu, pojavićeu se njegova kartica na počtku stranice “Moj ljubimac”.



Takođe, korisnik može i da izmeni podatke o već postojećem ljubimcu tako što će izmeniti podatke u popunjenoj formi koja se nalazi ispod kartica.

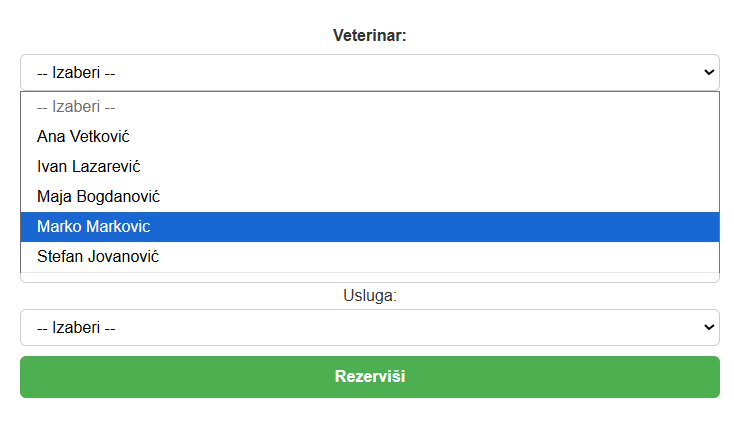


Omogućeno je i brisanje ljubimca, klikom na dugme obriši ljubimca.



Ukoliko korisnik želi da vidi sve zakazane termine za njegove ljubimce može otići na stranicu “Tretman ljubimca”, zatim selektovati za kog ljubimca želi da vidi zakazane termine. Ukoliko nema zakazano, prikazuje mu se dugme “Zakaži termin”.

Kada se klikne na to dugme, otvara se stranica “Rezervacije” gde korisnik može rezervisati termin za svog ljubimca.



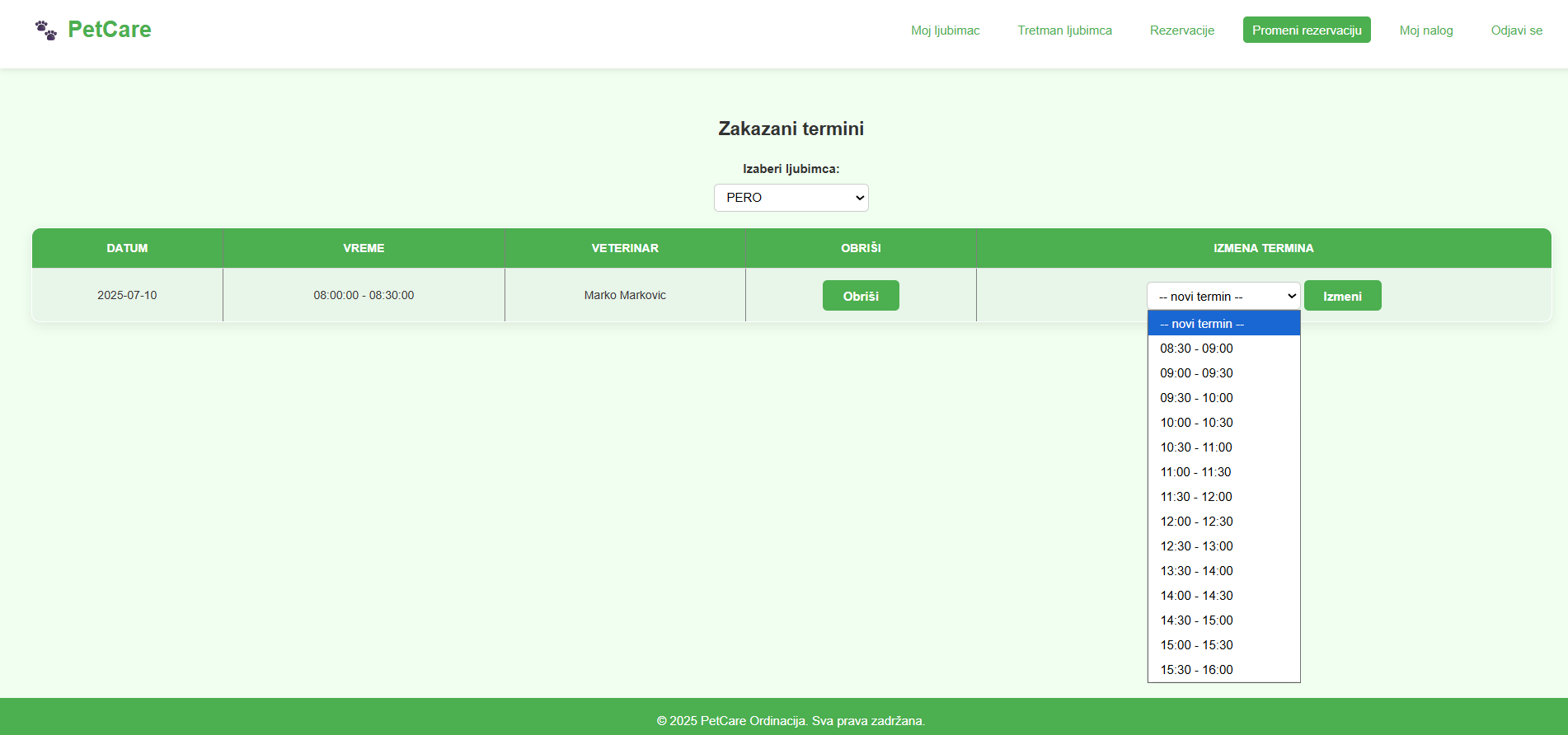


Nakon odabira veterinara i datuma, korisnik treba da pritisne na dugme “Prikaži termine” kako bi video sve termine od izabranog veterinara. Zatim j epotrebno izabrati za kog ljubimca se rezerviše termin i koju uslugu korisnik smatra da je potrebna njegovom ljubimcu.

Ukoliko je termin slobodan korisnik će uspešno rezervisati termin, a ukoliko nije slobodan korisniku neće biti moguća rezervacija.

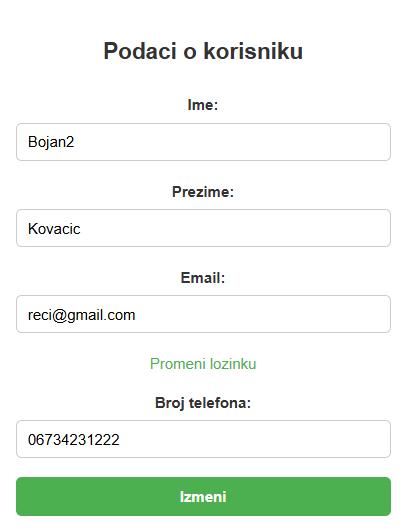


Kada se termin rezerviše, korisniku se ispiše poruka da je uspešno zakazan termin i šalje se mejl sa kodom rezervacije, koji kasnije veterinar unosi pri pristupanju kartona za odgovarajućeg ljubimca.



Promena rezervacije se vrši na stranici “Promeni rezervaciju”. Korisniku je omogućeno da promeni samo vreme termina, ne i datum. Ukoliko želi da promeni datum, potrebno je otkaže zakazani termin i rezerviše novi tog datuma kojeg želi.

Dugme “Obriši” briše rezervaciju.

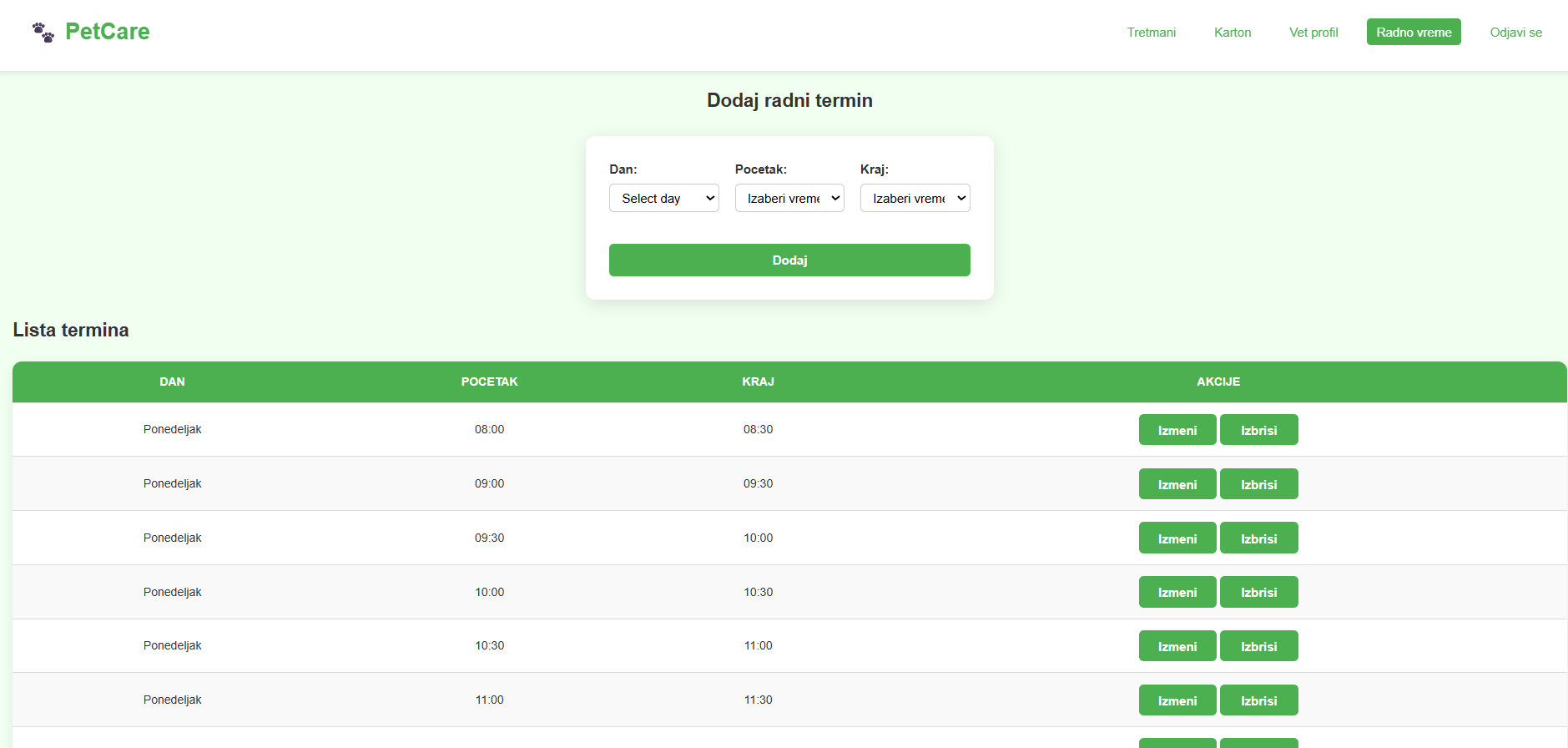


Promena ličnih podataka korsnika je moguća na stranici “Moj nalog”, gde korisnik može da menja ime, prezime i broj telefona. Klikom na potrdi, potvrđuje da se slaže sa time da se njegovi podaci promene.



Kada se korisnik uloguje kao veterinar, ima drugačiji uvid na stranicu od gosta i korisnika.

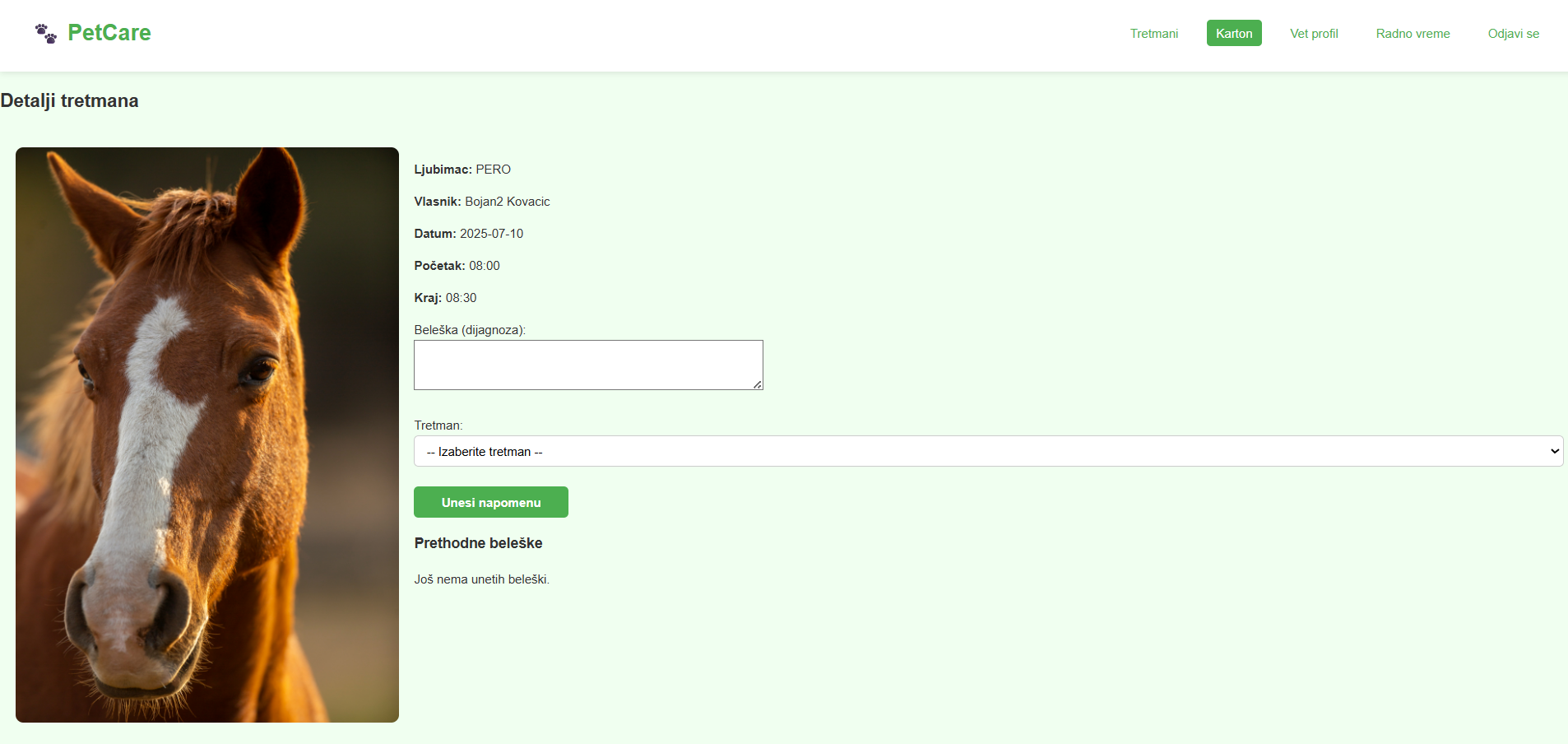
Nakon prijavljivanja, veterinaru se automatski otvara njegov profil, koji može samo da pregleda. Ukoliko želi da izmeni podatke mora se obratiti adminu.



Vetereinar može da na stranici “Radno vreme” pregleda, izmeni i obriše termin kada radi u ordinaciji.



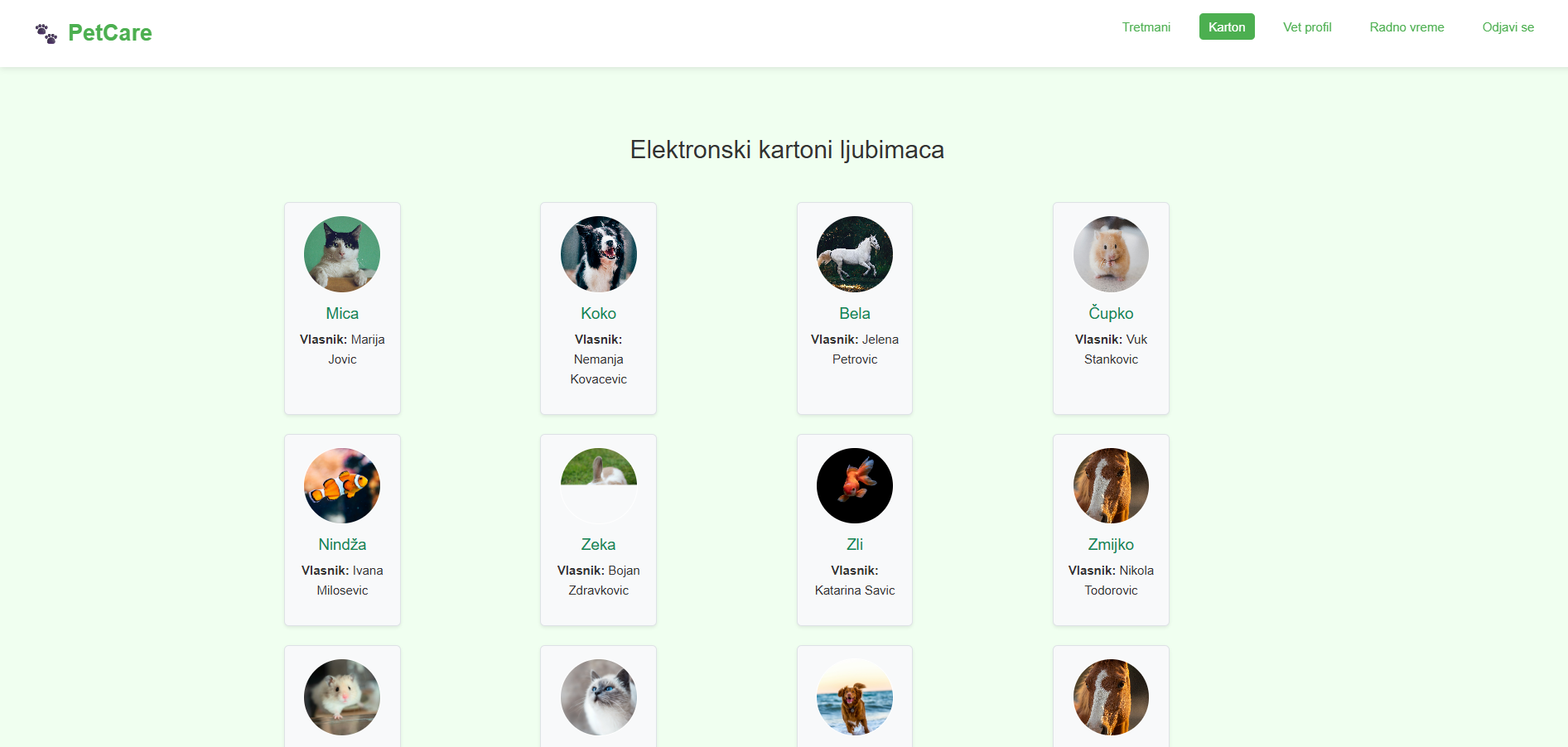
Stranica “Tretmani” sadrži listu zakazanih termina kod ulogovanog veterinara.



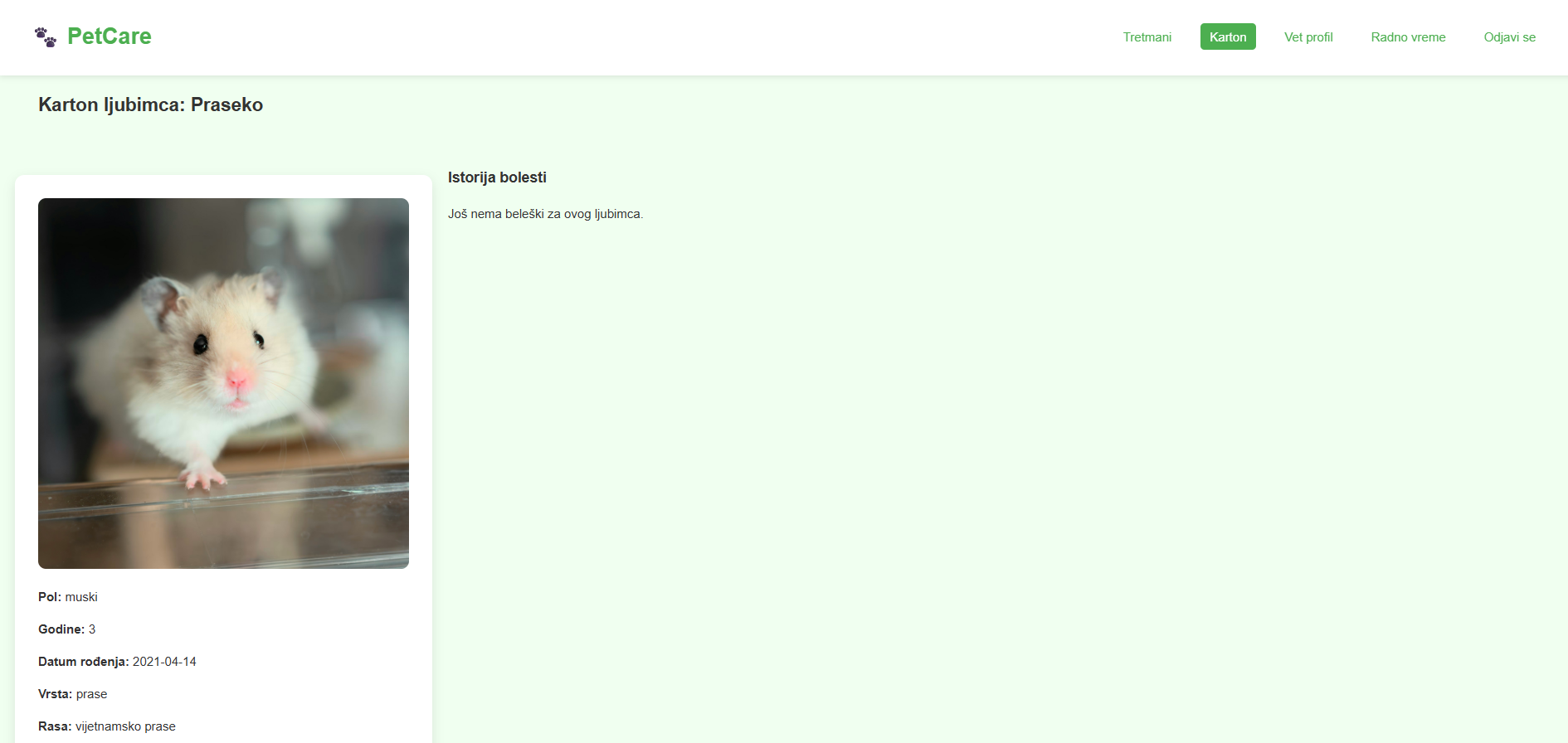
Sa kodom rezervacije koji korisnik da veterinaru, veterinar može da pristupi kartonu ljubimca, tako što će kod uneti u polje za tekst, zatim kliknuti dugme “Detalji”. Jedan kod važi samo za određeni termin, ne može se korisititi za ostale.

Otvara mu se stranica za upisivanje beleške o tretmanu. Kanije tu belešku mogu svi veterinari da vide, kao i vlasnik.

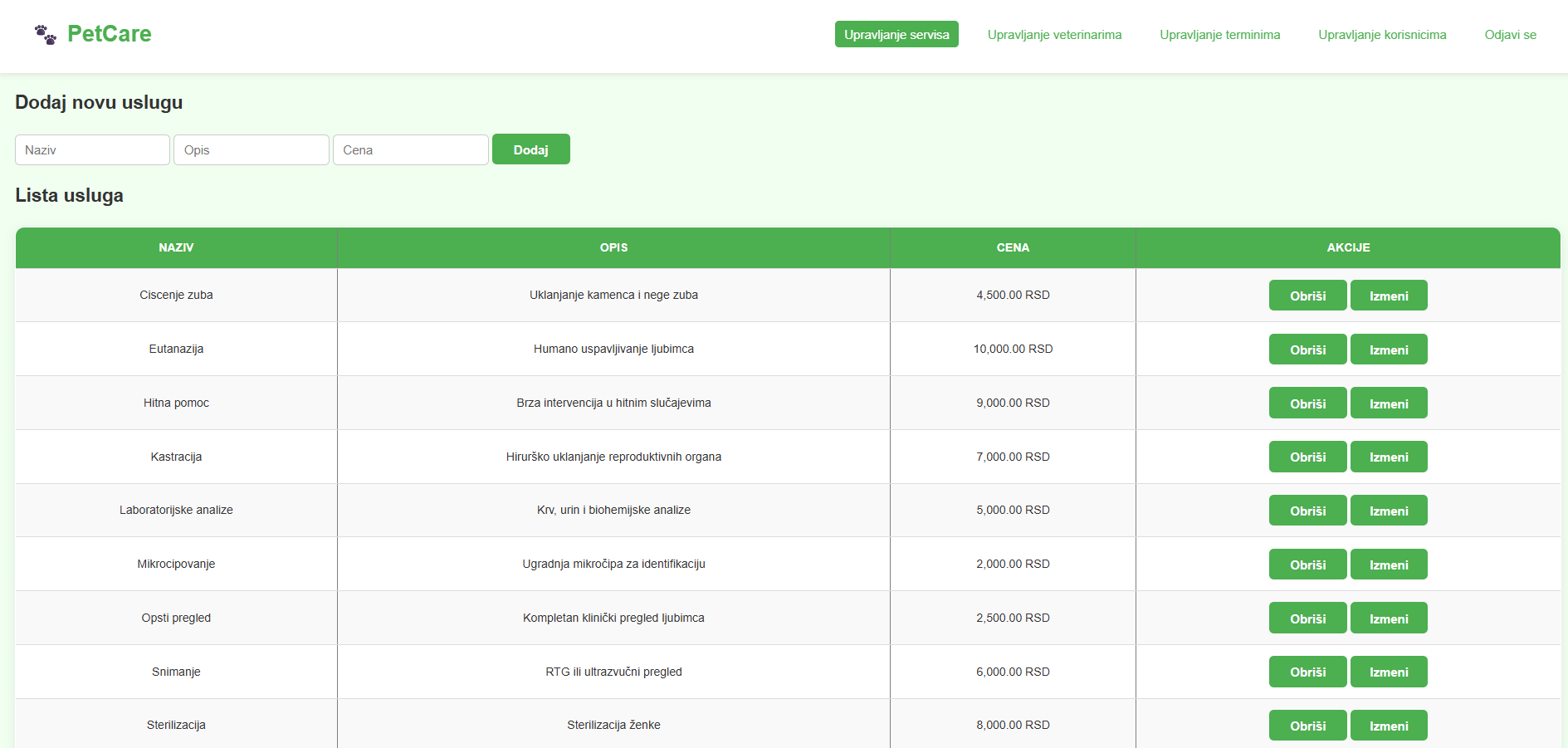
Klikom na dugme “Negativan poen” veterinar beleži da se korisnik nije pojavio na zakazani termin.



Stranica “Kartoni” sadrži elektronske kartone svih ljubimaca iz baze, od svakog korisnika.



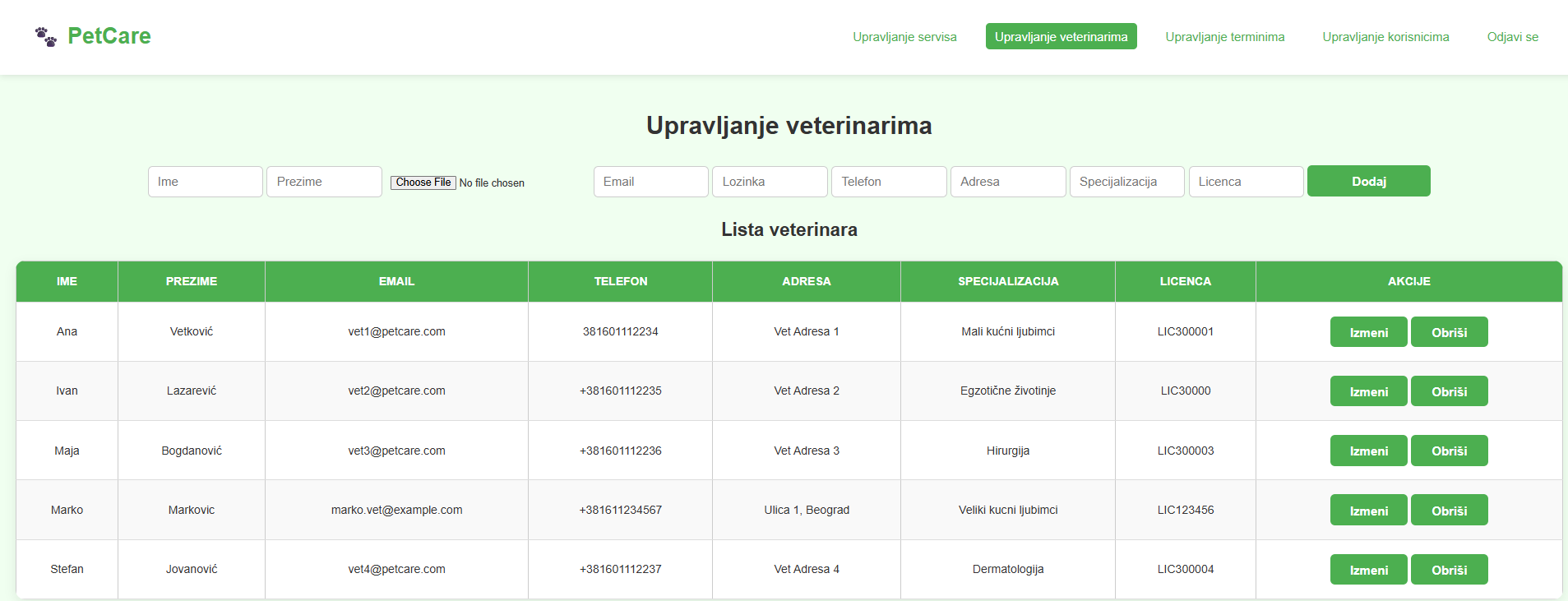
Klikom na karticu neke životinje veterianar ima uvid u istoriju bolesti životinje.



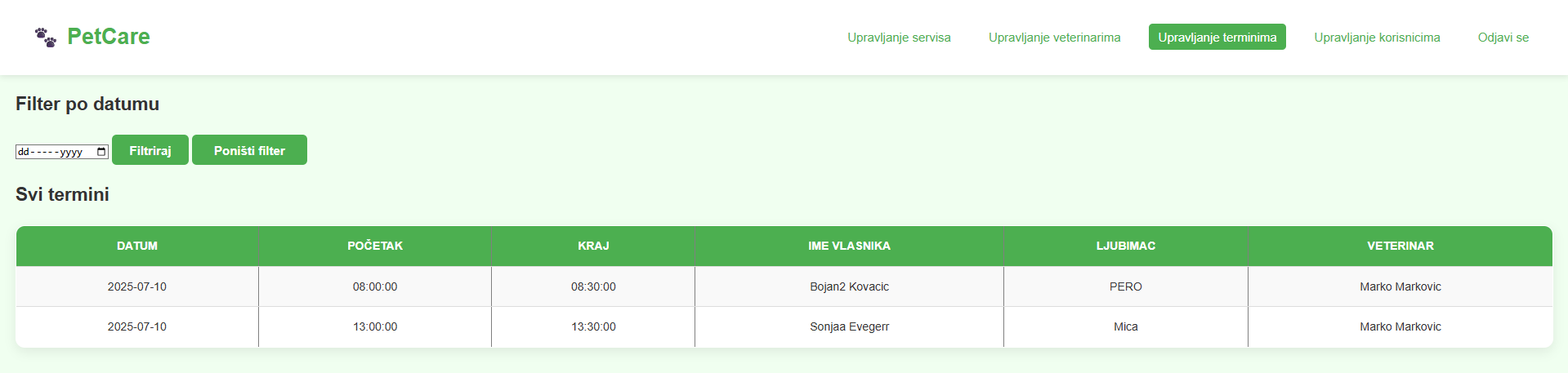
Četvrti nivo pristupa je admin pristup.

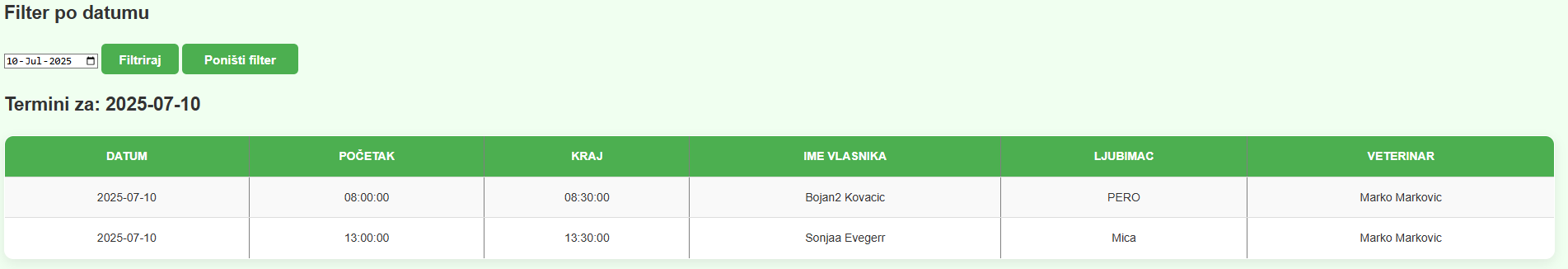
Admin ima 4 stranice za upravljanje podacima.

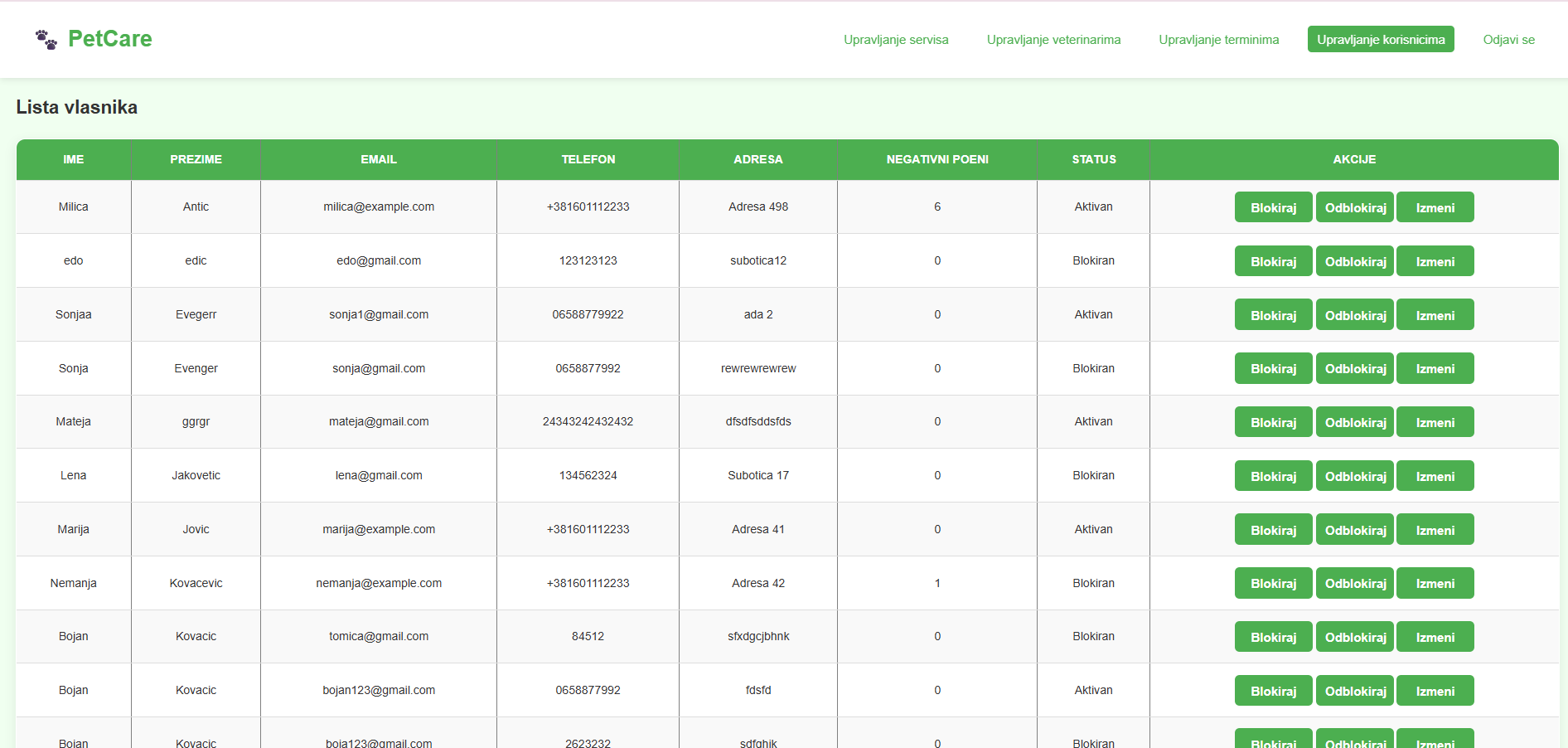
Prva stranica je “Upravljanje servisima”, gde admin može da doda novu uslugu ordinacije, da izmeni postojeću, kao i da je obriše.



Druga stranica je “Upravljanje veterinarima”, koja služi za dodavanje novog veterinara, izmenu podataka postojećeg i brisanje istog.

Treća stranica je stranica “Upravljanje tretmanima” gde admin može da vidi sve zakazane tretmane od svakog korisnika kod svakog veterinara.

Postoji mogućnost prikazivanja tretmana na osnovu datuma, tada admin bira datum u dateTimePicker-u i pritiska dugme “Filtriraj”. Filter se ponštava dugmetom “Poništi filter”.



Četvrta stranica je “Upravljanje korisnicima”, gde admin može da blokira, odblokira i obriše korisnika.

Klikom na dugme “Blokiraj” admin deaktivira korisniku nalog, klikom na “Odblokira” reaktivira nalog.



Sa dugmetom “Izmeni” menja podatke korisniku tako što mu se otvori forma iznad tabele.

Odjava korisnika, veterinara i admina se obavlja identično, samo se klikne link u navigaciji na kojem piše “Odjavi se”.

# Korišćena literatura

W1. <https://moodle2.vts.su.ac.rs/course/view.php?id=15>

W2. <http://www.w3schools.com/>