Web-aplikacija za veterinarske stanice

Šmuk, Karla

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:651172

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-23**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1347

WEB-APLIKACIJA ZA VETERINARSKE STANICE

Karla Šmuk

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1347

WEB-APLIKACIJA ZA VETERINARSKE STANICE

Karla Šmuk

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1347

Pristupnica:	**Karla Šmuk (0036540731)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Ivica Botički

Zadatak: **Web-aplikacija za veterinarske stanice**

Opis zadatka:

U okviru završnog rada potrebno je izraditi web-aplikaciju za veterinarske stanice. Web-aplikacija se treba sastojati od pregleda prijavljenih veterinarskih stanica, pregleda veterinara, pregleda i praćenja stanja i cijena medicinskih potrepština, modula za dodavanje vlasnika i njihovih ljubimaca, modula za dodavanje i praćenje dolazaka te izradu računa. U aplikaciji trebaju postojati različite uloge korisnika: administrator, veterinari i vlasnici ljubimaca. Administrator treba imati mogućnost dodavati veterinarske stanice te njihove veterinare. Veterinari trebaju imati mogućnost dodavati vlasnike i njihove ljubimce, evidentirati njihove dolaske te generirati račune. Vlasnici trebaju imati mogućnost pregledavati svoje ljubimce i njihovu evidenciju dolazaka u veterinarsku stanicu. Trebaju imati i mogućnost rezervacije termina gdje moraju navesti i kratki razlog dolaska. Prijavljeni i neprijavljeni korisnici trebaju moći vidjeti popis prijavljenih veterinarskih stanica.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod **3**

2. Teorijska pozadina **4**

2.1. Tehnologije 4

2.2. Obrasci uporabe 4

3. Arhitektura i dizajn sustava **6**

3.1. Dijagram komponenti 7

3.2. Korisnici sustava 8

3.2.1. Administrator 8

3.2.2. Veterinar 8

3.2.3. Vlasnik 8

3.3. Baza podataka 9

3.3.1. Dijagram baze podataka 9

3.3.2. Opis relacija 9

3.4. Poslužiteljska strana sustava 17

3.4.1. Rute 18

3.4.2. Ostalo 19

4. Pregled funkcionalnosti **22**

4.1. Početna stranica 22

4.2. Stranica za prijavu 22

4.2.1. Pogrešni podaci za prijavu 23

4.3. Stranica za unos jednokratne lozinke 23

4.3.1. Primjer emaila 24

4.4. Stranica za promijenu lozinke 24

4.5. Stranica za pregled profila 25
4.6. Stranica za pregled veterinarskih stanica 26
4.7. Stranica za pregled veterinara 26
4.7.1. Potvrda slanja emaila 27
4.8. Stranica za pregled tretmana 27
4.9. Stranica za pregled vlasnika 28
4.10. Stranica za pregled ljubimaca 29
4.10.1. Obrazac za stvaranje novog ljubimca 29
4.11. Stranica za pregled dolazaka i računa 30
4.11.1. Dolazak 31
4.11.2. Obrazac za stvaranje računa 31
4.12. Stranica za rezervaciju termina 32
4.12.1. Poruka pogreške 33
4.12.2. Rezervacija drugog termina 33
4.12.3. Prikaz rezerviranih termina 35
4.13. Stranica za pregled rezerviranih termina 36

5. Zaključak 37

Literatura 38

Sažetak 39

Abstract 40

1. Uvod

Razmišljajući o temi svoga završnog rada, došla sam na ovu ideju *Web-aplikacija za veterinarske stanice*. Kako i sama imam kućne ljubimce još nisam naišla na ovakav sustav u veterinarskim stanicama pa mi se činilo izazovno i interesantno riješiti taj problem u obliku web aplikacije kako bi i vlasnici ljubimaca u bilo kojem trenutku imali uvid u zdravstveno stanje svoga ljubimca.

Cilj ovoga završnog rada je oblikovati web aplikaciju koja bi pomogla u radu veterinarskih stanica i njihovih korisnika. Administrator može dodavati veterinarske stanice te njihove veterinare. Web aplikacija veterinarima dozvoljava unos novih vlasnika te njihovih ljubimaca, dodavanje svakog novog posjeta te račune, praćenje stanja zaliha te dostupnih tretmana. Vlasnici se mogu prijaviti u aplikaciju nakon što zatraže veterinara da im se na email pošalje jednokratna lozinka s pomoću koje se prijavljuju te postavljaju svoju novu lozinku. Nakon toga mogu pregledavati svoje ljubimce, sve posjete, račune te imaju mogućnost rezervacije termina u bilo kojoj veterinarskoj stanici zbog bolje organizacije veterinara.

U ovome radu prvo ću se osvrnuti na teorijsku pozadinu, opisati koje sam tehnologije koristila te ću izdvojiti najbitnije obrasce uporabe. Sljedeće ću opisati samu arhitekturu sustava, kako se odvija komunikacija između klijentske aplikacije i poslužitelja te poslužitelja i baze podataka. Nabrojat ću korisnike sustava te koje su njihove uloge u aplikaciji. Detaljno ću opisati bazu podataka i sve njene relacije te priložiti dijagram baze podataka. Na samom kraju ću izdvojiti sve komponente klijentske aplikacije, objasniti funkcionalnosti, navesti korisnike (administrator, veterinar ili vlasnik) koji mogu pristupiti pojedinoj stranici te kratko objasniti kako se sučelje koristi.

2. Teorijska pozadina

2.1. Tehnologije

Za razvoj klijentske aplikacije korišten je React [1] uz pomoć programskog jezika TypeScript [2]. Također, koristila sam biblioteku ChakraUI i TailwindCSS. Te dvije biblioteke znatno su mi ubrzale razvoj zbog jednostavnog korištenja.
Poslužitelj je izgrađen koristeći Node.js/Express.js [3] uz pomoć programskog jezika TypeScript.
PostgreSQL [4] baza podataka čuva sve potrebne podatke. Kako bih si lakše mogla predočiti bazu i njene podatke, koristila sam DBeaver [5].
Kao sustav za upravljanje kodom projekta izabrala sam GitHub. Za razvoj aplikacije koristila sam Visual Studio Code zbog njegove jednostavnosti i pametnog formatiranja koda te IntelliSense-a.

2.2. Obrasci uporabe

U ovome poglavlju izdvojila sam najbitnije obrasce uporabe.

- Prijava
  - Korisnik se uspješno prijavljuje emailom i lozinkom.
- Prijava jednokratnom lozinkom
  - Korisnik upisuje email i jednokratnu lozinku. Ako je jednokratna lozinka dobra preusmjerava ga se na promjenu lozinke. Ako je lozinka uspješno promijenjena, preusmjerava ga se na stranicu prijave.
- Promjena radnog vremena stanice
  - Korisnik veterinar može za bilo koji dan u tjednu promijeniti radno vrijeme.

- Dodavanje novog dolaska
  - Korisnik veterinar dodaje novi dolazak za ljubimca, upisuje tražene podatke. Naknadno može izmijeniti bilješku za dolaske.

- Dodavanje novog računa
  - Za određeni dolazak ima mogućnost dodavati više računa. Za svaki račun ima mogućnost dodavati tretmane te njihovu količinu.

- Rezervacija termina
  - Korisnik vlasnik odabire željenu veterinarsku stanicu te ljubimca za kojeg želi napraviti rezervaciju. Bitno je da korisnik ne može rezervirati termin pola sata prije radnog vremena. Također, rezervacije se mogu napraviti svakih 15 minuta. Ako se pokuša rezervirati termin, a 15 minuta prije ili kasnije već postoji neka rezervacija sustav će poslati pogrešku i sljedeći slobodni termin. Kako bi korisniku bila olakšana rezervacija, nakon odabira veterinarske stanice prikazuje se radno vrijeme kao podsjetnik.

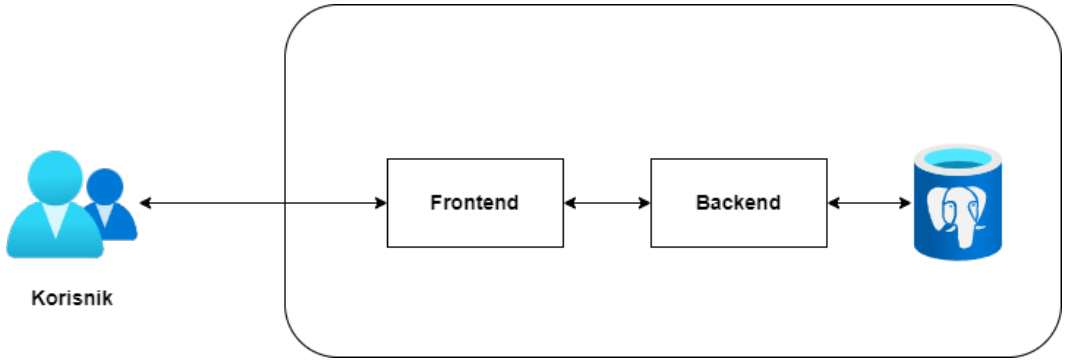
- Promjena statusa rezerviranog termina
  - Ako nam je korisnik vlasnik, on može otkazati termin. Ako nam je korisnik veterinar, on može potvrditi da se vlasnik pojavio te ga se preusmjerava na stranicu ljubimca za kojeg je termin rezerviran, može i obrisati rezervaciju ako se vlasnik ne pojavi.

- Promjena statusa ljubimca
  - Korisnik može označiti ljubimca kao nestalog ili ga može obrisati ako je uginuo.

3. Arhitektura i dizajn sustava

Arhitektura aplikacije sastoji se od tri dijela:

- Klijentska aplikacija (Frontend)
- Poslužitelj (Backend)
- Baza podataka



Slika 3.1. Arhitektura sustava

Korisnik komunicira s klijentskom aplikacijom putem web preglednika. Poslužitelj je REST jer zadovoljava njegove uvjete, a to su korištenje metoda HTTP-a (GET, POST, DELETE, PUT), korištenje imenica u URI-ju, a ne samo glagola, te slanje podataka u JSON formatu.

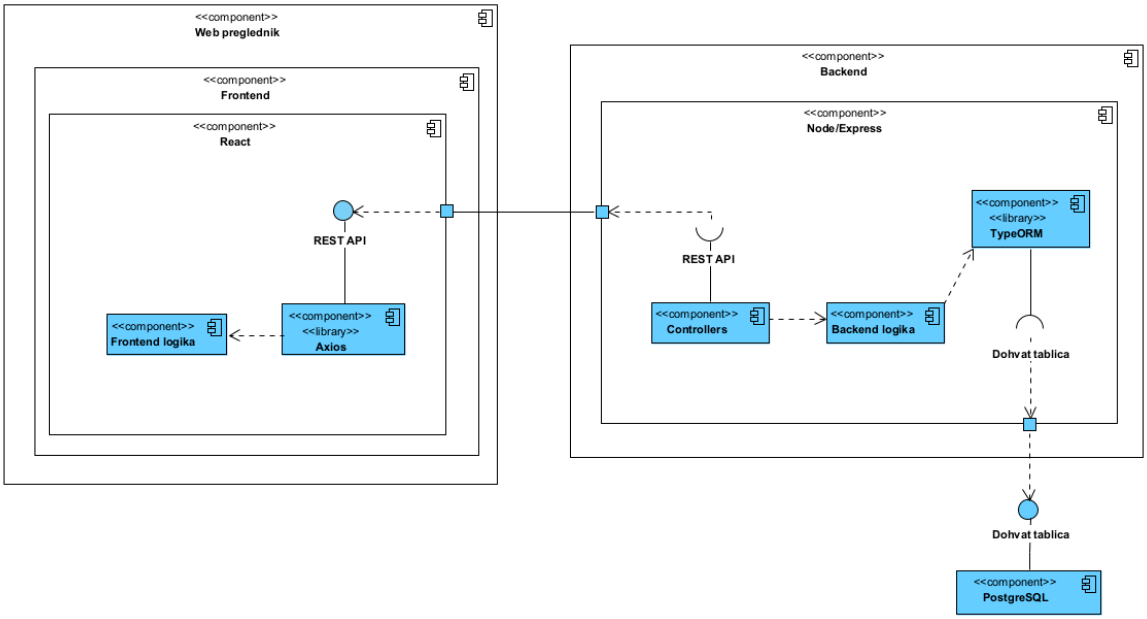
Klijentska aplikacija je odgovorna za prikazivanje podataka korisniku i slanje korisničkih zahtjeva poslužitelju. Ona komunicira s poslužiteljem putem REST API-ja.

Nakon što je poslužitelj zaprimio zahtjev obrađuje ga i ako je potrebno on komunicira s bazom pomoću TypeORM-a [6]. Nakon obrade zahtjeva šalje se odgovor koji klijentska aplikacija obradi i prikaže korisniku.

Baza podataka se pokreće pomoću Docker containera [7].

3.1. Dijagram komponenti

Na donjoj slici prikazan je dijagram komponenti koji prikazuje ovisnost komponenata. Aplikacija se sastoji od frontenda, backenda i baze podataka. Komponente frontend i backend međusobno komuniciraju putem REST API-ja, a baza podataka s backend-om pomoću TypeORM-a koji nam omogućava brze i jednostavne SQL upite. Korisnik pristupa aplikaciji preko web preglednika.



Slika 3.2. Dijagram komponenti

3.2. Korisnici sustava

Svaki korisnik ima mogućnost prijave, pregled profila te spremanje/mijenjanje slike profila.

3.2.1. Administrator

Administrator ima mogućnost pregledati sve veterinarske stanice, dodavati nove, uređivati osnovne podatke te ih brisati. Također, ima mogućnost za svaku veterinarsku stanicu pregledati veterinare, dodavati nove, uređivati podatke veterinara, brisati ih te poslati jednokratnu lozinku na email u svrhu prijave veterinara u aplikaciju.

3.2.2. Veterinar

Veterinar ima mogućnost pregledati sve veterinarske stanice, uređivati radno vrijeme svoje stanice. Također, ima mogućnost za svaku veterinarsku stanicu pregledati veterinare. Ima mogućnost upravljanja tretmanima, zalihama te pregled i upravljanje rezervacijama. Može dodavati nove vlasnike, njihove ljubimce, posjete te račune.

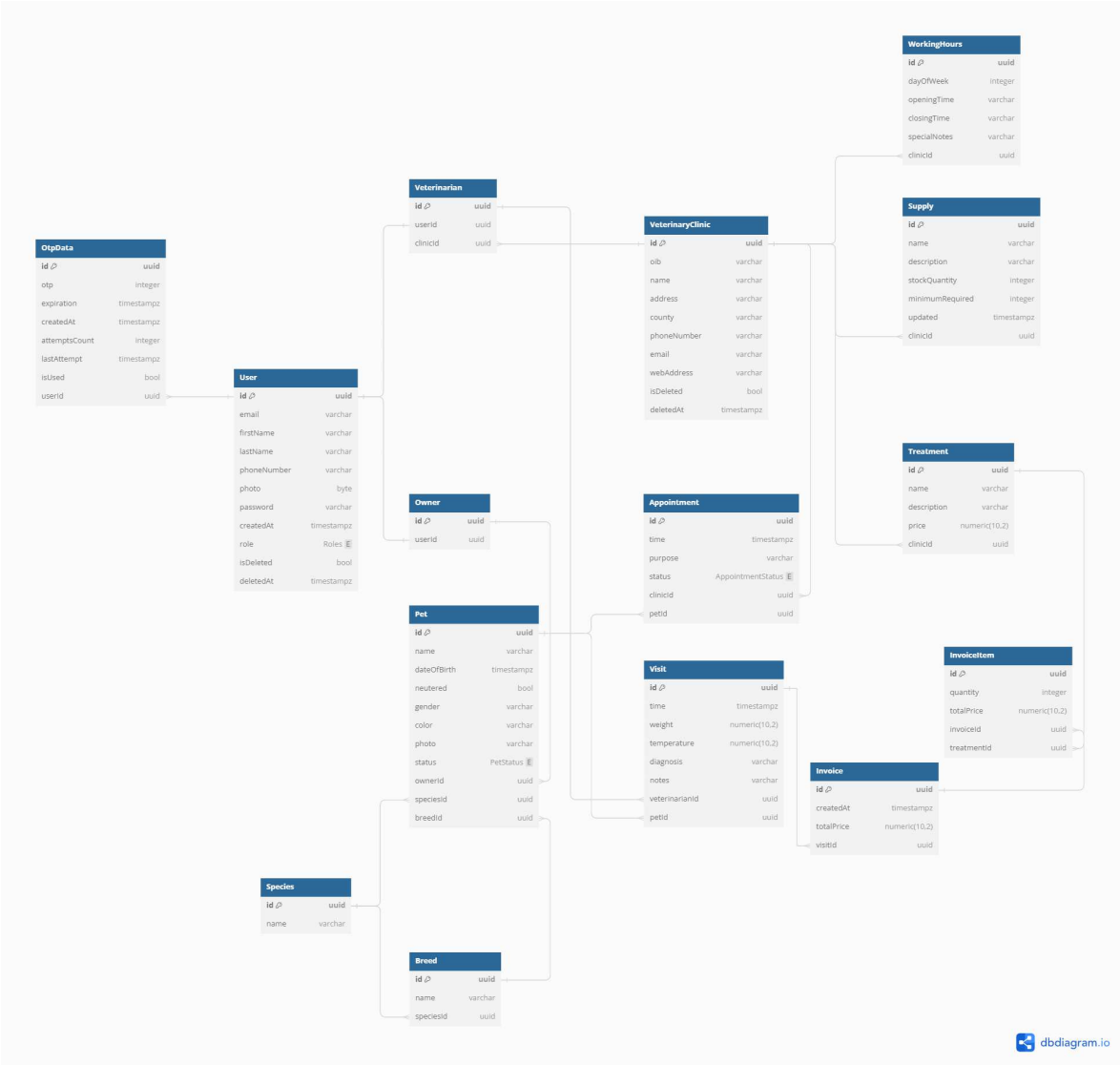
3.2.3. Vlasnik

Vlasnik ima mogućnost pregledati sve veterinarske stanice. Ima mogućnost pregledati svoje ljubimce, posjete te račune. Također, ima mogućnost rezervacije za određenu stanicu i ljubimca.

3.3. Baza podataka

Koristila sam relacijsku bazu podataka. Općenito, baza nam je potrebna kako bi pohranjivali potrebne podatke i brzo ih dohvaćali.

3.3.1. Dijagram baze podataka



Slika 3.3. Relacijski dijagram baze podataka

3.3.2. Opis relacija

User Ovaj entitet sadrži sve informacije o korisniku. Ovaj entitet je u vezi *One-to-Many* s entitetom OtpData te *One-to-One* s entitetima Veterinarian i Owner. Uloga korisnika definirana je enumeracijom UserRole: [Administrator, Veterinar, Vlasnik].

User		
id	UUID	primarni ključ
email	VARCHAR	email korisnika
firstName	VARCHAR	ime korisnika
lastName	VARCHAR	prezime korisnika
phoneNumber	VARCHAR	broj mobitela korisnika
photo	VARCHAR	id slike
password	HASH	lozinka korisnika
createdAt	TIMESTAMPZ	vrijeme stvaranja korisnika
role	UserRole	korisnikova uloga
isDeleted	BOOL	zastavica je li korisnik obrisan
deletedAt	TIMESTAMPZ	vrijeme brisanja korisnika

OtpData Ovaj entitet sadrži sve informacije o jednokratnoj lozinki korisnika. Ovaj entitet je u vezi *Many-to-One* s entitetom User.

OtpData		
id	UUID	primarni ključ
otp	INTEGER	jednokratna lozinka
expiration	TIMESTAMPZ	rok uporabe jednokratne lozinke
createdAt	TIMESTAMPZ	vrijeme stvaranja
attemptsCount	INTEGER	broj pokušaja
lastAttempt	TIMESTAMPZ	vrijeme zadnjeg pokušaja
isUsed	BOOL	zastavica je li iskorišten
userId	UUID	id korisnika

Veterinarian Ovaj entitet sadrži sve informacije o korisniku veterinaru. Ovaj entitet je u vezi *One-to-One* s entitetom User, *One-to-Many* s entitetom Visit te *Many-to-One* s

entitetom VeterinaryClinic.

Veterinarian		
id	UUID	primarni ključ
userId	UUID	id korisnika
clinicId	UUID	id veterinarske stanice

VeterinaryClinic Ovaj entitet sadrži sve informacije o veterinarskoj stanici. Ovaj entitet je u vezi *One-to-Many* s entitetima Veterinarian, WorkingHours, Supply, Treatment i Appointment.

VeterinaryClinic		
id	UUID	primarni ključ
oib	VARCHAR	oib stanice
name	VARCHAR	ime stanice
address	VARCHAR	adresa stanice
county	VARCHAR	županija u kojoj se nalazi stanica
phoneNumber	VARCHAR	broj mobitela stanice
email	VARCHAR	email stanice
webAddress	VARCHAR	web stanice
isDeleted	BOOL	zastavica je li stanica obrisana
deletedAt	TIMESTAMPZ	vrijeme brisanja stanice

WorkingHours Ovaj entitet sadrži sve informacije o radnom vremenu veterinarske stanice. Ovaj entitet je u vezi *Many-to-One* s entitetom VeterinaryClinic.

WorkingHours		
id	UUID	primarni ključ

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

WorkingHours		
dayOfWeek	INTEGER	broj dana
openingTime	VARCHAR	vrijeme otvaranja stanice
closingTime	VARCHAR	vrijeme zatvaranja stanice
specialNotes	VARCHAR	napomene
clinicId	UUID	id stanice

Supply Ovaj entitet sadrži sve informacije o zalihama veterinarske stanice. Ovaj entitet je u vezi *Many-to-One* s entitetom VeterinaryClinic.

Supply		
id	UUID	primarni ključ
name	VARCHAR	ime zalihe
description	VARCHAR	opis zalihe
stockQuantity	INTEGER	broj na stanju
minimumRequired	INTEGER	minimalni broj na stanju
updated	TIMESTAMPZ	vrijeme ažuriranja
clinicId	UUID	id stanice

Treatment Ovaj entitet sadrži sve informacije o tretmanima veterinarske stanice. Ovaj entitet je u vezi *Many-to-One* s entitetom VeterinaryClinic te *One-to-Many* s entitetom InvoiceItem.

Treatment		
id	UUID	primarni ključ
name	VARCHAR	ime tretmana
description	VARCHAR	opis tretmana

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Treatment		
price	NUMERIC(10,2)	cijena tretmana
clinicId	UUID	id stanice

Owner Ovaj entitet sadrži sve informacije o korisniku vlasniku. Ovaj entitet je u vezi *One-to-One* s entitetom User te *One-to-Many* s entitetom Pet.

Owner		
id	UUID	primarni ključ
userId	UUID	id korisnika

Pet Ovaj entitet sadrži sve informacije o ljubimcu. Ovaj entitet je u vezi *Many-to-One* s entitetima Owner, Species i Breed, u vezi *One-to-Many* s entitetima Appointment i Visit. Status ljubimca definiran je enumeracijom PetStatus: [Živ, Preminuo, Nestao].

Pet		
id	UUID	primarni ključ
name	VARCHAR	ime ljubimca
dateOfBirth	TIMESTAMPZ	datum rođenja ljubimca
neutered	BOOL	zastavica je li kastriran/sterilizirana
gender	VARCHAR	spol
color	VARCHAR	boja
photo	VARCHAR	id slike
status	PetStatus	status ljubimca
ownerId	UUID	id vlasnika
speciesId	UUID	id vrste
breedId	UUID	id pasmine

Species Ovaj entitet sadrži sve informacije o vrstama ljubimaca. Ovaj entitet je u vezi *One-to-Many* s entitetima Pet i Breed.

Species		
id	UUID	primarni ključ
name	VARCHAR	ime vrste

Breed Ovaj entitet sadrži sve informacije o pasminama ljubimaca. Ovaj entitet je u vezi *One-to-Many* s entitetom Pet te u vezi *Many-to-One* s entitetom Species.

Breed		
id	UUID	primarni ključ
name	VARCHAR	ime pasmine
speciesId	UUID	id vrste

Appointment Ovaj entitet sadrži sve informacije o rezervacijama vlasnika za ljubimca. Ovaj entitet je u vezi *Many-to-One* s entitetima Pet i VeterinaryClinic. Status rezervacije definiran je enumeracijom AppointmentStatus: [Zakazano, Dovršeno, Otkazano, Nepojavljivanje].

Appointment		
id	UUID	primarni ključ
time	TIMESTAMPZ	vrijeme rezervacije
purpose	VARCHAR	razlog rezervacije
status	AppointmentStatus	status rezervacije
clinicId	UUID	id vlasnika
petId	UUID	id vrste

Visit Ovaj entitet sadrži sve informacije o dolascima ljubimaca i vlasnika. Ovaj entitet je u vezi *Many-to-One* s entitetima Pet i Veterinarian te u vezi *One-to-Many* s entitetom

Invoice.

Visit		
id	UUID	primarni ključ
time	TIMESTAMPZ	vrijeme posjeta
weight	NUMERIC(10,2)	težina ljubimca
temperature	NUMERIC(10,2)	temperatura ljubimca
diagnosis	VARCHAR	dijagnoza
notes	VARCHAR	bilješke
veterinarianId	UUID	id vlasnika
petId	UUID	id vrste

Invoice Ovaj entitet sadrži sve informacije o računima. Ovaj entitet je u vezi *Many-to-One* s entitetom Visit te u vezi *One-to-Many* s entitetom InvoiceItem.

Invoice		
id	UUID	primarni ključ
createdAt	TIMESTAMPZ	vrijeme stvaranja računa
totalPrice	NUMERIC(10,2)	ukupna cijena računa
visitId	UUID	id posjeta

InvoiceItem Ovaj entitet sadrži sve informacije o tretmanima računa. Ovaj entitet je u vezi *Many-to-One* s entitetima Invoice i Treatment.

InvoiceItem		
id	UUID	primarni ključ
quantity	INTEGER	broj tretmana računa
totalPrice	NUMERIC(10,2)	ukupna cijena tretmana

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

InvoiceItem		
invoiceId	UUID	id vlasnika
treatmentId	UUID	id vrste

3.4. Poslužiteljska strana sustava

Poslužitelj se pokreće tako da se prvo poziva server.ts. On poziva app.ts koji se spaja s bazom podataka kako bi se mogli raditi upiti. Također, server.ts postavlja rutere koje ćemo koristiti za komunikaciju s klijentskom aplikacijom. Struktura razmještaja direktorija poslužitelja:

```
│ src
│ │
│ ├── config
│ ├── constants
│ ├── controllers
│ ├── middleware
│ ├── migrations
│ ├── model
│ │ │
│ │ ├── entity
│ │ └── requests
│ ├── routes
│ ├── app.ts
│ └── server.ts
├── .env.example
├── .eslintignore
├── .eslintrc.js
├── .gitignore
├── docker-compose.yml
├── package-lock.json
├── package.json
└── tsconfig.json
```

3.4.1. Rute

U ovoj tablici prikazat ću glavne rutere. Svaki ruter ima svoje rute koje koriste HTTP metode te se za svaku poziva njen controller koji obrađuje podatke i vraća odgovor.

/api/user	ruter koji definira krajnje točke za korisnike
/api/owner	ruter koji definira krajnje točke za vlasnike
/api/veterinarian	ruter koji definira krajnje točke za veterinare
/api/otp	ruter koji definira krajnje točke za jednokratne lozinke
/api/clinic	ruter koji definira krajnje točke za veterinarske stanice
/api/pet	ruter koji definira krajnje točke za ljubimce
/api/visit	ruter koji definira krajnje točke za dolaske
/api/invoice	ruter koji definira krajnje točke za račune
/api/auth	ruter koji definira krajnje točke za autentifikaciju
/api/appointment	ruter koji definira krajnje točke za rezervacije

Primjer ruta za user.router.ts:

```
import { Router } from "express";
import { createOwnerUser, createVetUser, deleteUser, getAllUsers, getUser,
    saveProfilePhoto, updateUser } from "../controllers/user.controller";
import { validateRequest } from "../middleware/validateRequest";
import { CreateUserDto } from "../model/requests/createUser.dto";
import { multer } from "../middleware/multer";
import { UpdateUserDto } from "../model/requests/updateUser.dto";

const router = Router();

router.post("/", validateRequest(CreateUserDto), createOwnerUser);
router.post("/:clinicId", validateRequest(CreateUserDto), createVetUser);
router.get("/", getAllUsers);
router.get("/:id", getUser);
router.put("/update", validateRequest(UpdateUserDto), updateUser);
router.delete("/delete/:userId", deleteUser);
router.post("/saveProfileImage/:userId", multer.single("file"), saveProfilePhoto);

export default router;
```

3.4.2. Ostalo

Migracije

Kako sam dodavala entitete baze i njihove veze, radila sam migracije kako bih osigurala da brisanjem tablica iz baze neće biti sve izgubljeno, već kako bi lako mogla opet stvoriti iste pokretanjem poslužitelja. Također, pomoću migracija sam napravila i inicijalne podatke za neke entitete. Za komunikaciju s bazom sam koristila TypeORM te je on bio zadužen i za izradu migracija. Kako bi migracije mogla generirati, pokretati ili poništiti, definirala sam naredbe unutar package.json te je taj dio koda prikazan u nastavku.

```
...
"scripts": {
  "build": "npx tsc",
  "start": "tsc && node ./dist/server.js",
  "dev": "nodemon --exec ts-node ./src/server.ts",
  "lint": "eslint . --ext .ts",
  "typeorm": "npm run build && ts-node ./node_modules/typeorm/cli.js -d src/config/db.
      ts",
  "migration:generate": "npm run typeorm -- migration:generate",
  "migration:run": "npm run typeorm -- migration:run",
  "migration:revert": "npm run typeorm -- migration:revert"
}
...
```

Slanje emaila korisnicima

Za slanje emaila korisniku ili veterinaru sam koristila nodemailer. Dio koda za slanje emaila je prikazan u nastavku.

```
...
const transporter = nodemailer.createTransport({
  service: process.env.MAIL_HOST,
  auth: {
    user: process.env.MAIL_USER,
    pass: process.env.MAIL_PASS
  }
});

const mailOptions = {
  from: 'VetConnect ' + process.env.MAIL_USER,
  to: email,
  subject: 'Jednokratna lozinka za pristup VetConnect',
  html:   `<p>Postovani,</p>
      <p>Vasa jednokratna lozinka za pristup je: <strong>${otp}</strong></p>`
}
...
```

Spremanje slika

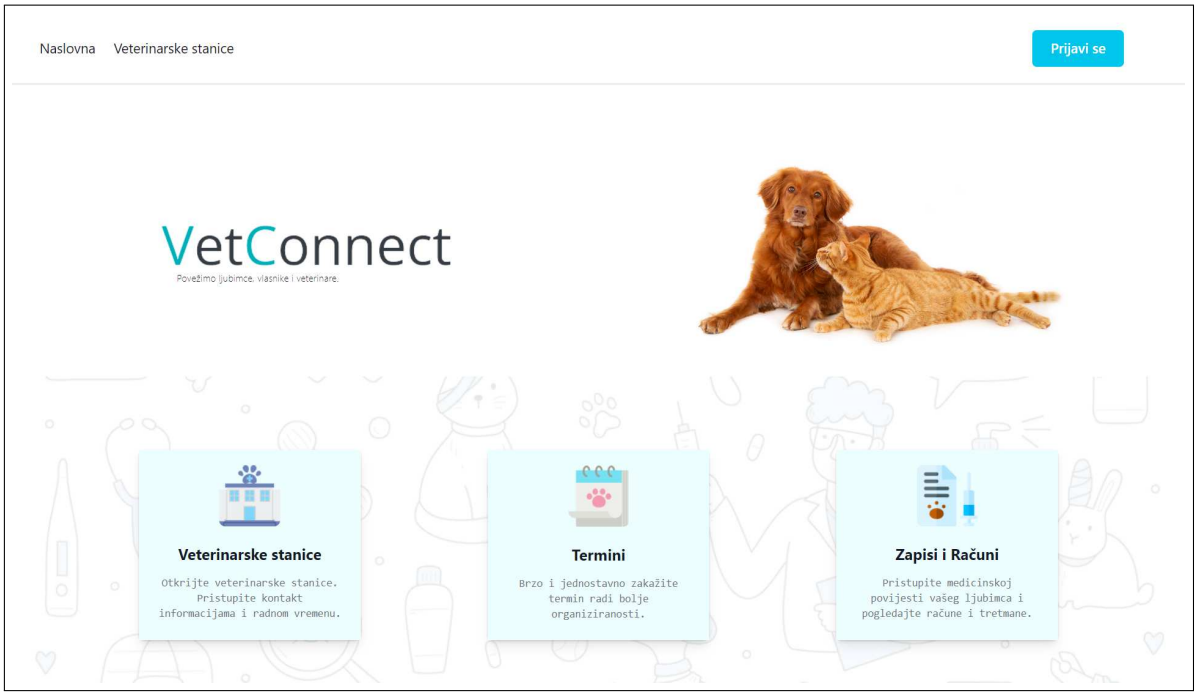
Slike sam odlučila spremati na Google Drive. To sam implementirala pomoću Google Drive API-ja [8] te je u nastavku prikazan kod za spremanje slike na drive. Nakon uspješnog spremanja kao odgovor sam dobila id slike kojeg sam spremila u bazu kako bih lako mogla dohvatiti sliku.

```
...
export const uploadToGoogleDrive = async (file: any, id: string, auth: any) => {
  const fileMetadata = {
    name: id,
    parents: [process.env.GOOGLE_DRIVE_FOLDER_ID!], // folder id
  };

  const bufferStream = new Readable({
    read() {
      this.push(file.buffer);
      this.push(null); // End of stream
    }
  });

  const media = {
    mimeType: file.mimetype,
    body: bufferStream
  };

  const driveService = google.drive({ version: "v3", auth });

  const response = await driveService.files.create({
    requestBody: fileMetadata,
    media: media,
    fields: "id"
  });

  return response.data;
};
...
```

4. Pregled funkcionalnosti

4.1. Početna stranica

Prikaz početne stranice za neprijavljenog korisnika. Neprijavljeni korisnik ima mogućnost pregledati sve veterinarske stanice ili se prijaviti.

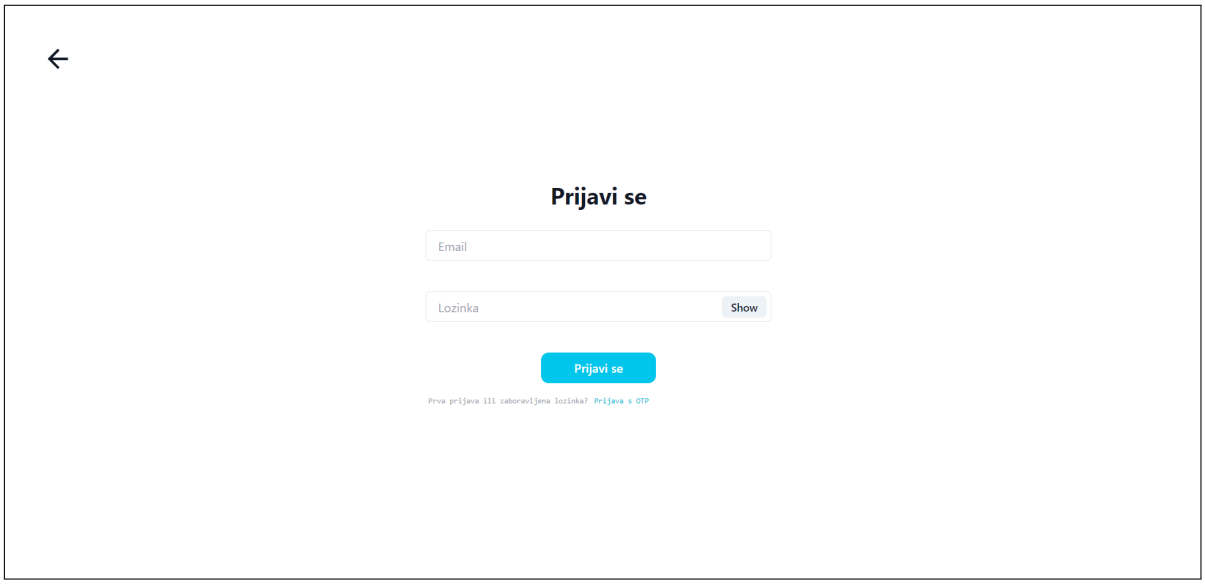


Slika 4.1. Početna stranica - neregistrirani korisnik

4.2. Stranica za prijavu

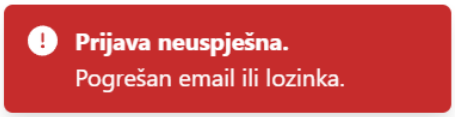
Na stranici za prijavu korisnik unosi svoj email i lozinku. Ako jedno ili oboje nije ispravno, korisniku će se prikazati pogreška (slika 4.3.). Ako je korisnik uspješno prijavljen, prikazuje se početna stranica za njegovu ulogu. Ako se korisnik prvi puta prijavljuje u sustav ili je zaboravio svoju lozinku te je na email dobio jednokratnu lozinku, može odabrati opciju *Prijava s OTP* te ga se preusmjerava na stranicu za verifikaciju (slika 4.4.

).



Slika 4.2. Stranica za prijavu

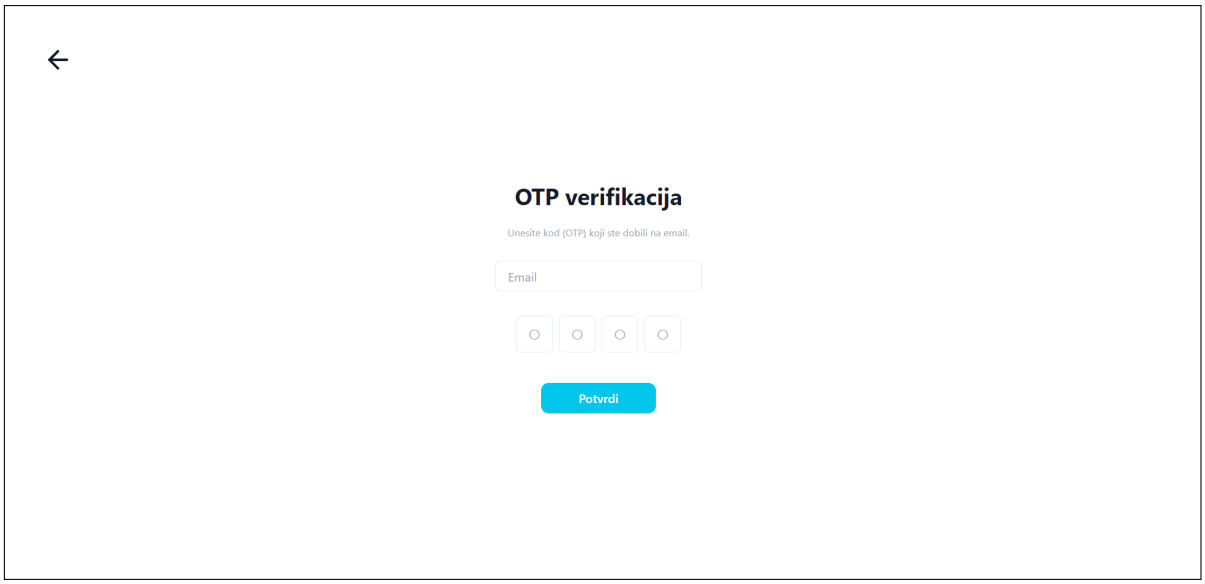
4.2.1. Pogrešni podaci za prijavu



Slika 4.3. Poruka pogreške

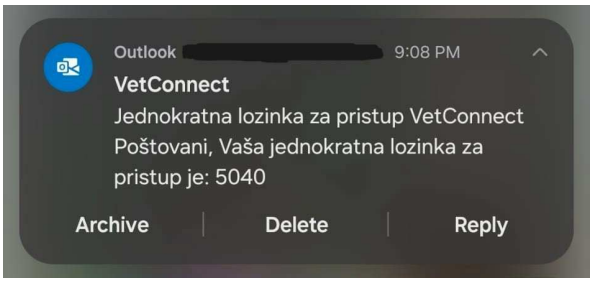
4.3. Stranica za unos jednokratne lozinke

Nakon dobivene jednokratne lozinke na email korisnik upisuje svoj email i lozinku. Ukoliko je unesena ispravna lozinka, ali je istekla, verifikacija neće biti uspješna. Također, ukoliko se neispravna lozinka unese tri puta, lozinka za tog korisnika više neće biti ispravna te će morati zatražiti novu lozinku. Ukoliko je unesen ispravan email i jednokratna lozinka korisnika će se preusmjeriti na stranicu za promjenu svoje lozinke (slika 4.6.)



Slika 4.4. Stranica za unos jednokratne lozinke

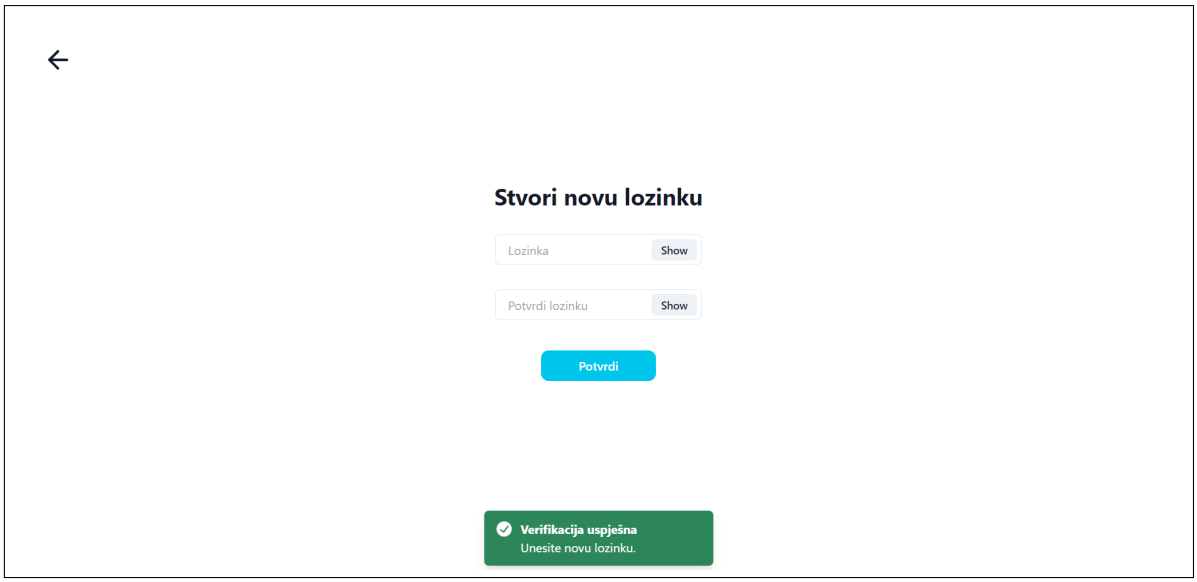
4.3.1. Primjer emaila



Slika 4.5. Email

4.4. Stranica za promijenu lozinke

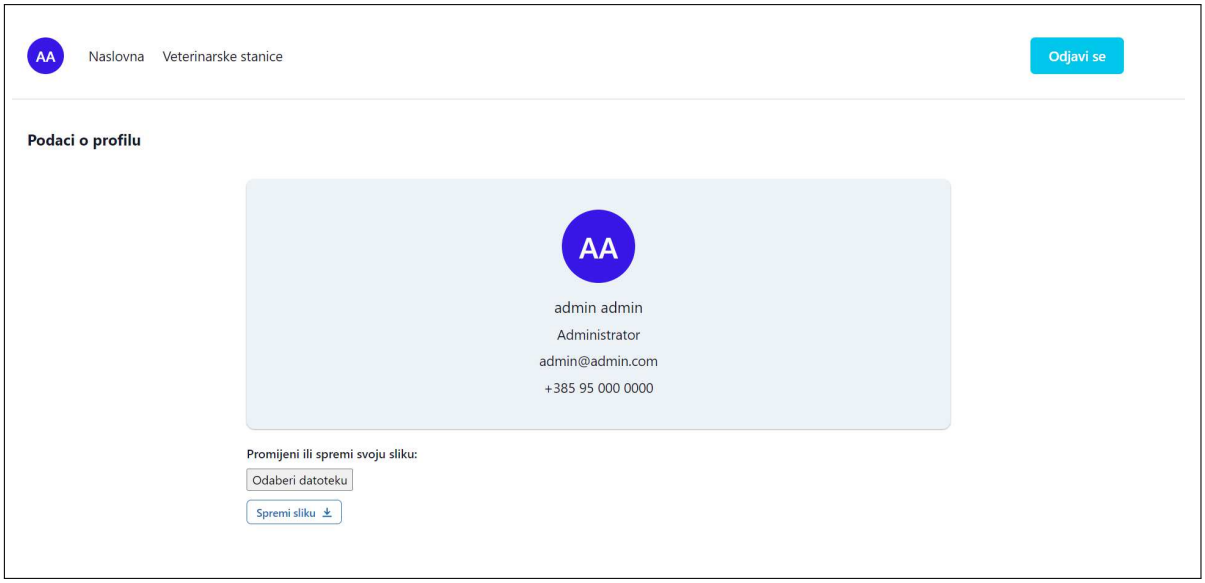
Nakon uspješne verifikacije korisnik može unijeti svoju lozinku s kojom će se moći prijaviti. Ako je lozinka uspješno promijenjena, korisnika se preusmjeruje na stranicu prijave.



Slika 4.6. Stranica za unos jednokratne lozinke

4.5. Stranica za pregled profila

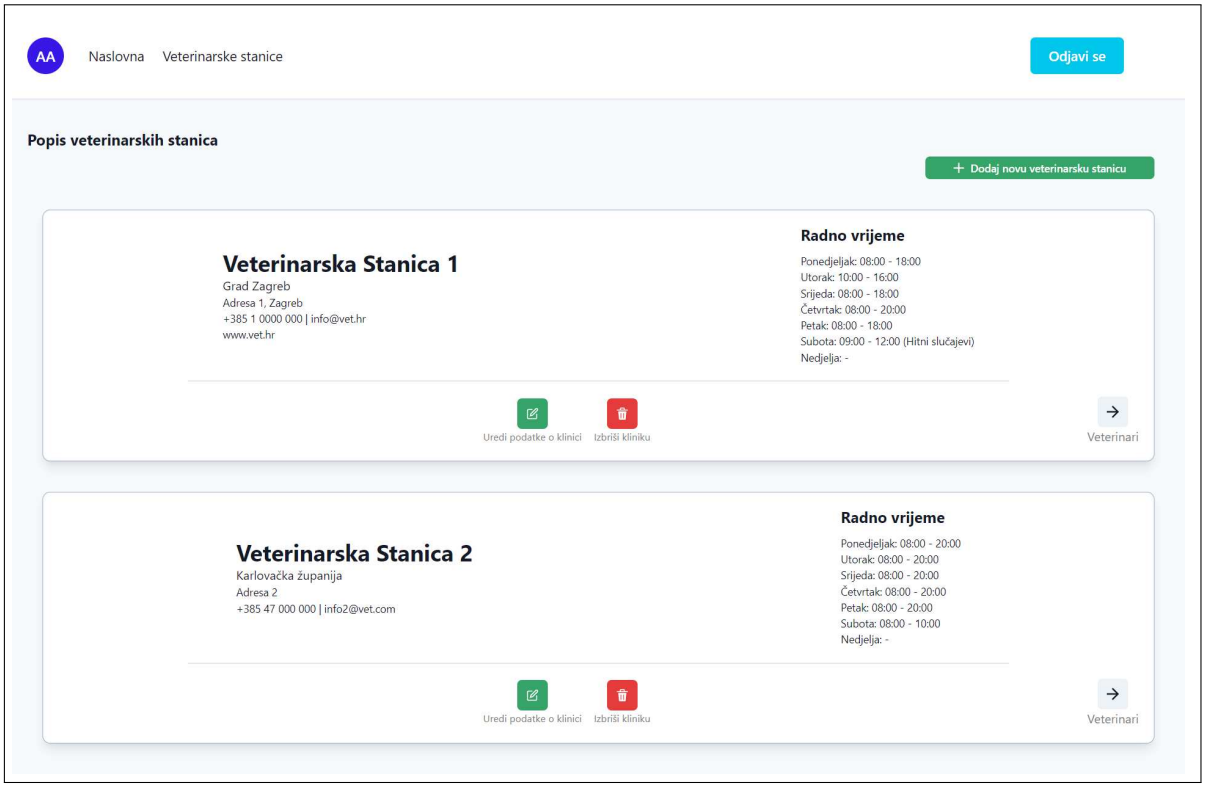
Svaki prijavljeni korisnik može pregledati svoje podatke profila te dodati/promijeniti sliku. Ostale podatke može promijeniti jedino korisnik s ulogom koji je zaslužan za tog korisnika (za veterinara je to administrator, a za vlasnika veterinar).



Slika 4.7. Stranica profila - Administrator

4.6. Stranica za pregled veterinarskih stanica

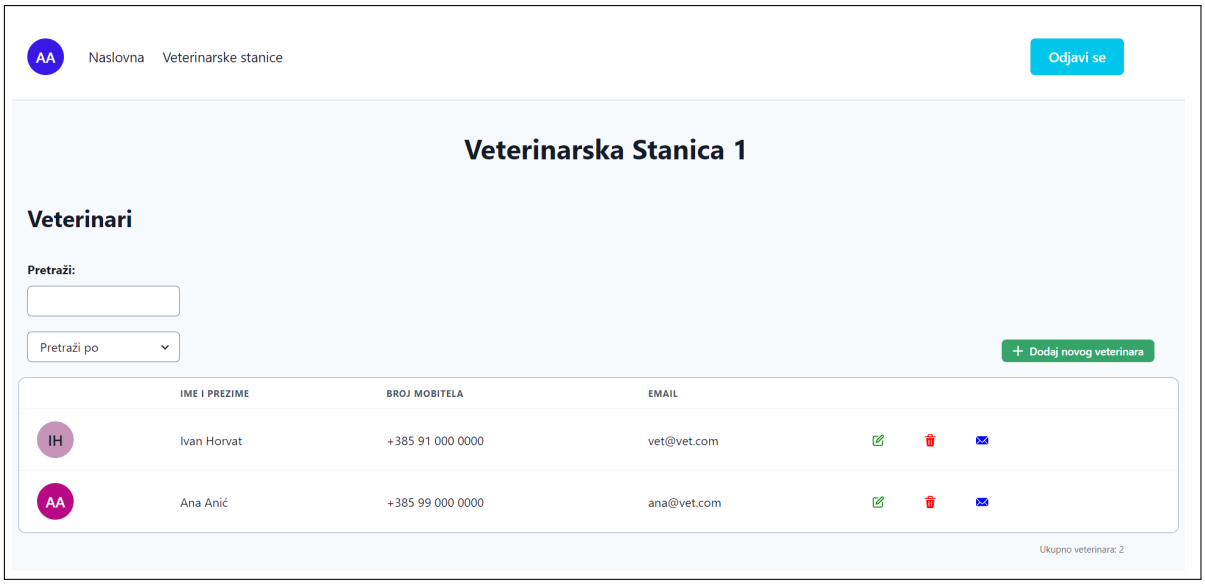
Na sljedećoj slici je prikazan pregled veterinarskih stanica za korisnika administrator. Svaki korisnik ima pristup stranici za pregled veterinarskih stanica. Administrator može dodavati nove veterinarske stanice, uređivati osnovne informacije stanice, brisati ih te pristupiti popisu veterinara. Veterinar ima mogućnost mijenjati radno vrijeme svoje veterinarske stanice te pristupiti popisu veterinara bilo koje veterinarske stanice. Vlasnik i neprijavljeni korisnik ne mogu obaviti niti jednu drugu akciju.



Slika 4.8. Stranica veterinarskih stanica - Administrator

4.7. Stranica za pregled veterinara

Na sljedećoj slici je prikazan pregled veterinara za pojedinu veterinarsku stanicu za korisnika administrator. Korisnici administrator i veterinar imaju pristup stranici za pregled veterinara. Administrator može dodavati nove veterinare, uređivati osnovne informacije o veterinaru, brisati ih te im poslati email s jednokratnom lozinkom za prijavu. Ako je email uspješno poslan administrator će dobiti poruku o uspješnom slanju (slika 4.10.).



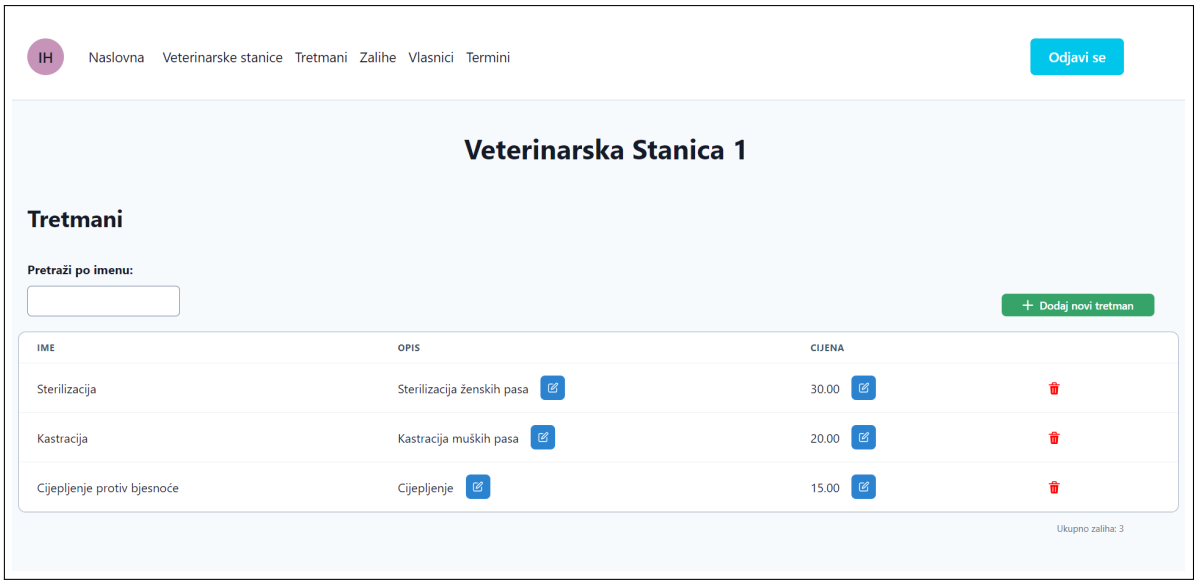
Slika 4.9. Stranica veterinara za pojedinu stanicu - Administrator

4.7.1. Potvrda slanja emaila

Slika 4.10. Potvrda slanja - Administrator

4.8. Stranica za pregled tretmana

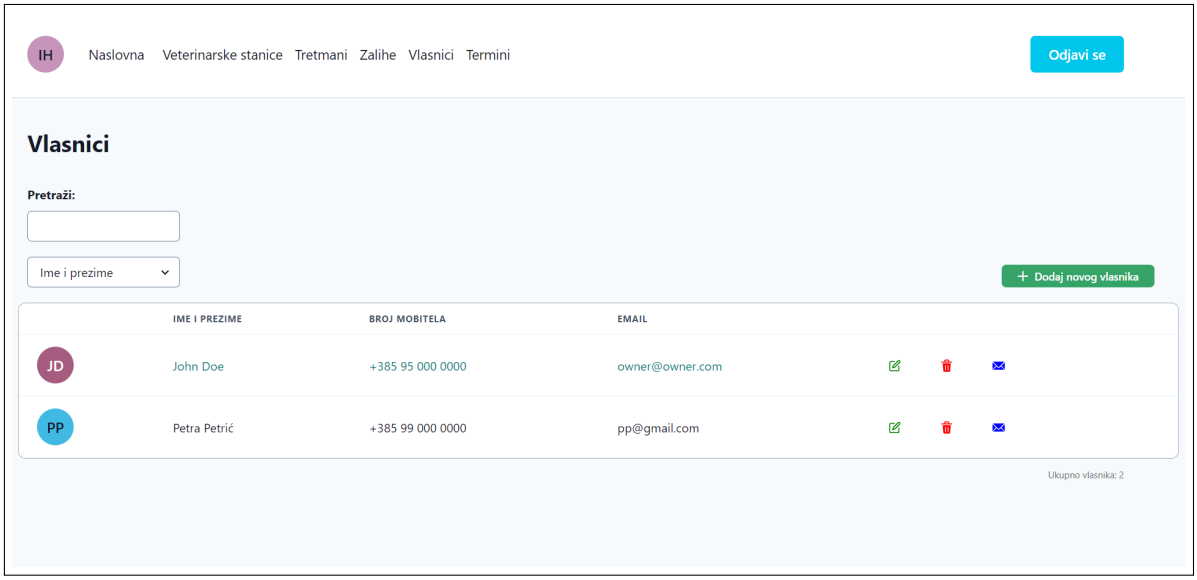
Stranici za pregled tretmana ima pristup samo veterinar i to za veterinarsku stanicu kojoj on pripada. On može dodavati nove tretmane te uređivati opis i cijenu ili ga obrisati. Također, slična stranica ovoj je i stranica zaliha. Ona se sastoji od popisa zaliha za tu veterinarsku stanicu. Veterinar može dodavati nove zalihe ili uređivati već postojeće (opis i trenutno stanje) te ih brisati.



Slika 4.11. Stranica tretmana - Veterinar

4.9. Stranica za pregled vlasnika

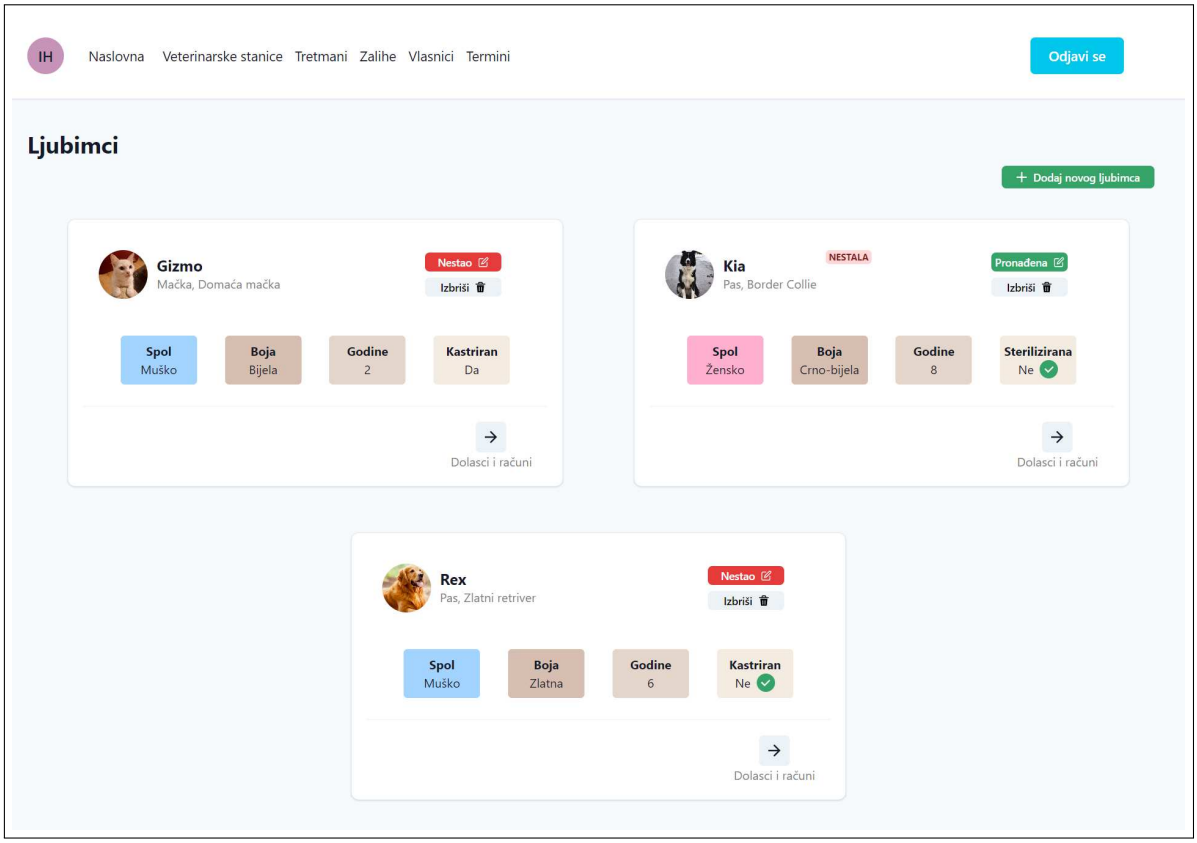
Svaki veterinar ima pristup stranici za pregled vlasnika. Može dodavati nove vlasnike, uređivati njihove podatke, brisati ih te poslati email sa jednokratnom lozinkom. Klikom na bilo kojeg vlasnika se preusmjerava na stranicu njegovih ljubimaca (slika 4.13.).



Slika 4.12. Stranica vlasnika - Veterinar

4.10. Stranica za pregled ljubimaca

Ovoj stranici imaju pristup veterinar i vlasnik. Veterinar može dodavati nove ljubimce (obrazac 4.14.) te raditi promijene na već postojećim ili prijeći na stranicu za dolaske i račune (slika 4.15.). Za svakog ljubimca veterinar može promijeniti ako je nestao što bi moglo ubrzati potragu za ljubicem, može ga izbrisati te može promijeniti status kastracije/sterilizacije. Vlasnik ima uvid u svoje ljubimce, može označiti ljubimca kao nestalog te može dodati/promijeniti sliku svojeg ljubimca.

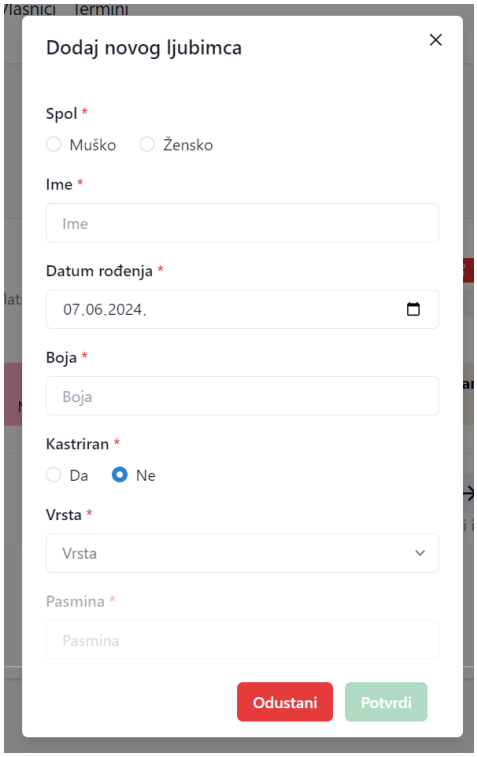


Slika 4.13. Stranica ljubimaca - Veterinar

4.10.1. Obrazac za stvaranje novog ljubimca

Ovaj obrazac se prikazuje veterinaru u slučaju da želi unesti novog ljubimca. Ovdje bih izdvojila odabir vrste i pasmine. Pasminu može odabrati isključivo kada je odabrao vrstu. Kod odabira vrste veterinaru će se ponuditi sve koje već postoje u bazi. Ako ne postoji točno ta koja mu treba, veterinar odabire opciju "Nema ponuđene vrste" te vrstu i pasminu upisuje sam. Ako je odabrao vrstu koja već postoji, za pasminu mu se prikažu one koje pripadaju toj vrsti. Također, ako pasmina koja mu treba ne postoji može ju sam

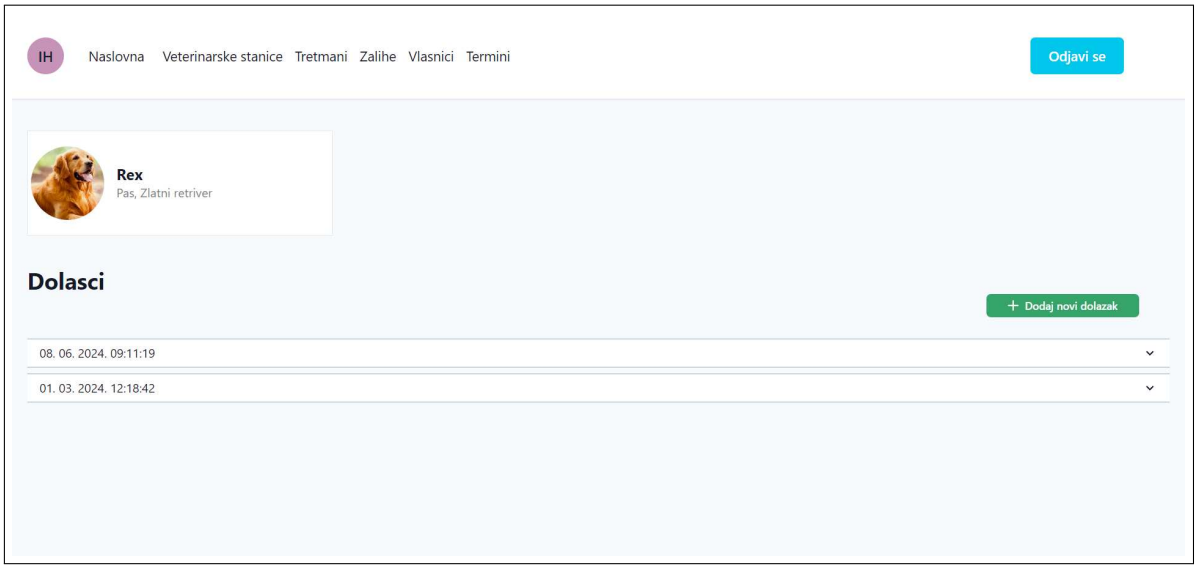
unijeti odabirom opcije "Nema ponuđene pasmine".



Slika 4.14. Stranica ljubimaca - Veterinar

4.11. Stranica za pregled dolazaka i računa

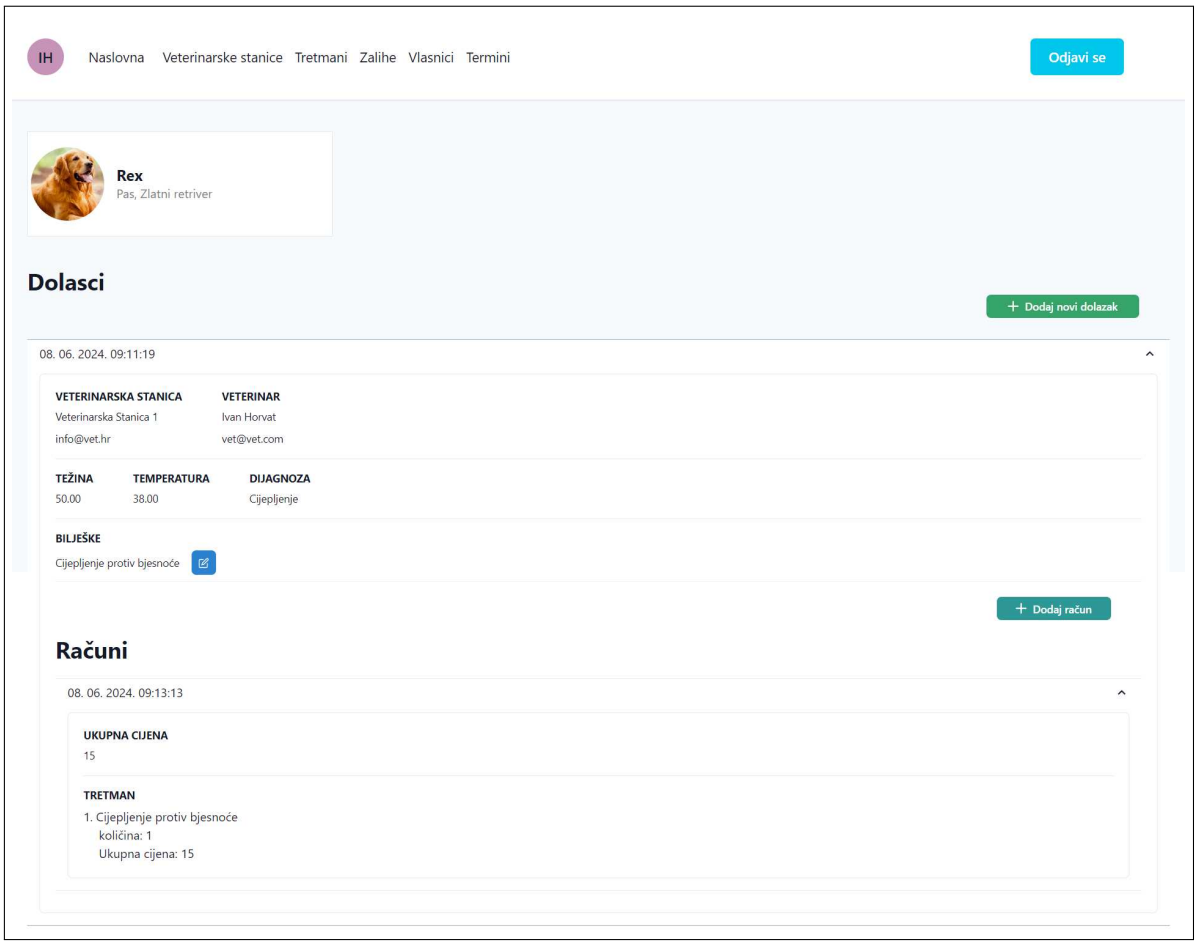
Ovoj stranici imaju pristup veterinar i vlasnik. Veterinar može dodavati nove dolaske, pregledati ili urediti postojeće te dodavati račune za njih. Vlasnik može samo pregledavati dolaske.



Slika 4.15. Stranica dolazaka - Veterinar

4.11.1. Dolazak

Na sljedećoj slici je prikazan detaljniji prikaz jednog dolaska.

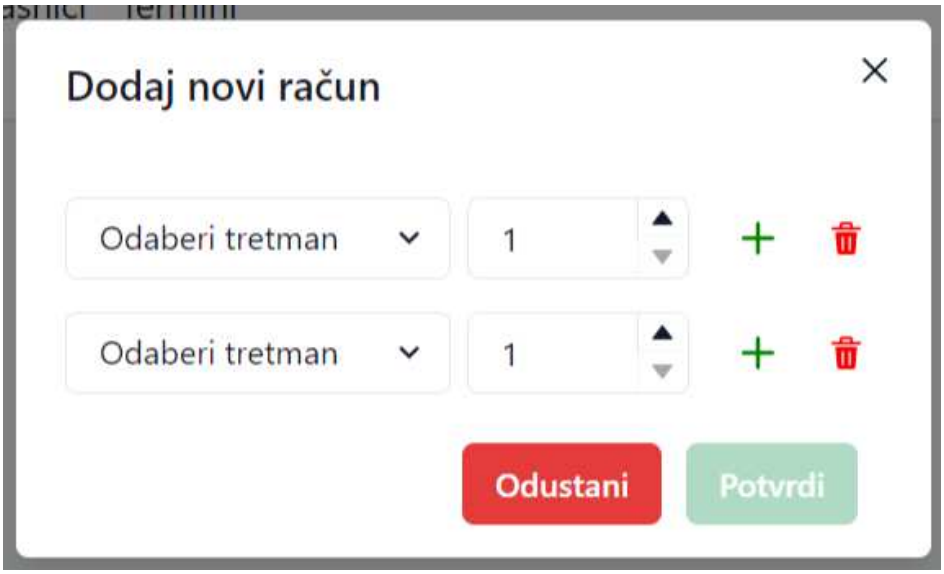


Slika 4.16. Stranica dolazaka - Veterinar

4.11.2. Obrazac za stvaranje računa

Veterinar ima mogućnost za svaki dolazak unijeti jedan ili više računa. To može učiniti na način da odabere tretman i količinu, ako želi dodati još tretmana može kliknuti na ikonu + te će se pojaviti još jedno polje za unos što je prikazano na drugoj slici. Također, ima mogućnost i micanja tretmana koje je unio prije nego što potvrdi račun.

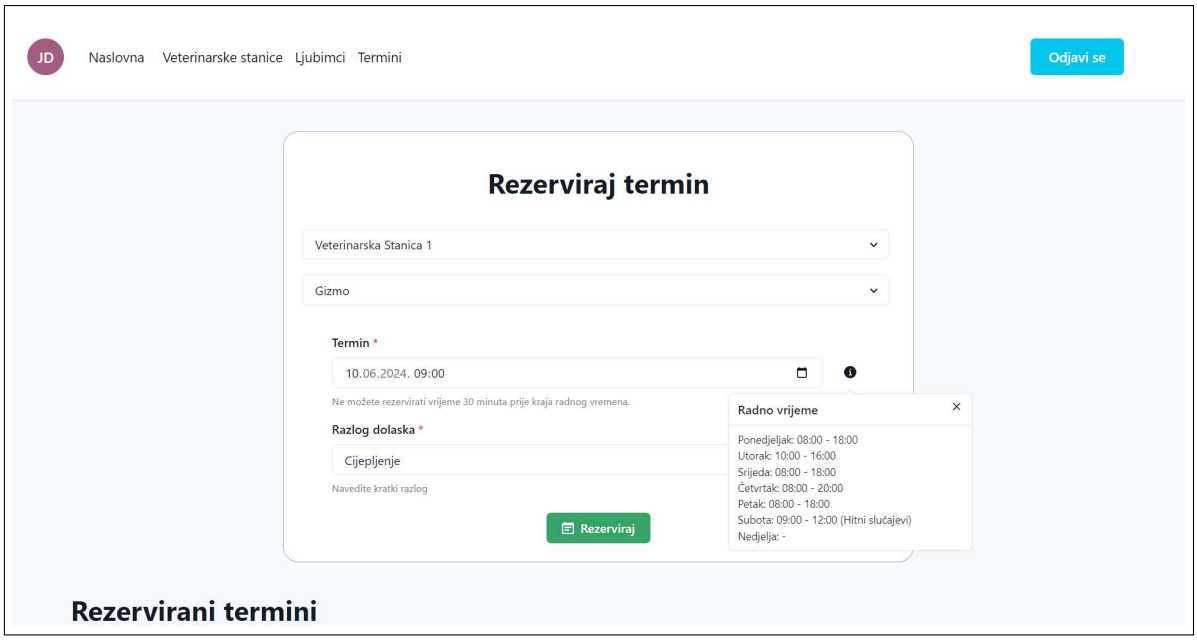
Slika 4.17. Stvaranje računa - Veterinar



Slika 4.18. Stvaranje računa - Veterinar

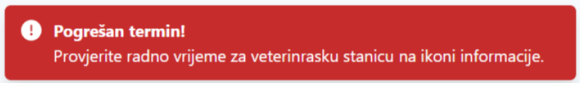
4.12. Stranica za rezervaciju termina

Ovoj stranici ima pristup vlasnik. Dostupan mu je obrazac za rezervaciju kojeg ispunjava na način da prvo odabire veterinarsku stanicu u kojoj želi napraviti rezervaciju te mu se prikazuje ikona informacije na koju može kliknuti i dobiti uvid u radno vrijeme koje služi kao podsjetnik. Sljedeće odabire za kojeg ljubimca želi rezervirati termin. Tek tada mu se dopušta odabir termina. Također, mora unijeti i kratki razlog dolaska. Ako vlasnik odabere vrijeme izvan radnog vremena prikazuje mu se pogreška (slika 4.20.).



Slika 4.19. Stranica za rezervaciju termina - Vlasnik

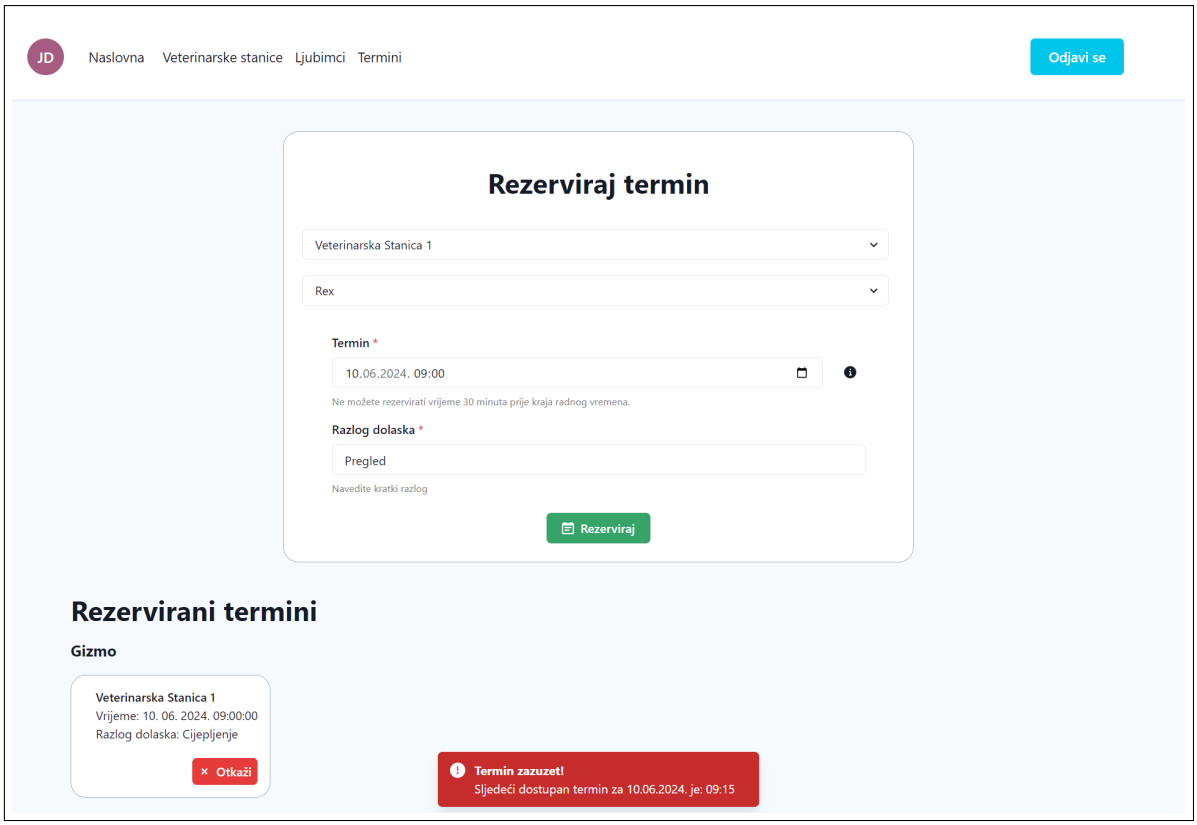
4.12.1. Poruka pogreške



Slika 4.20. Poruka pogreške

4.12.2. Rezervacija drugog termina

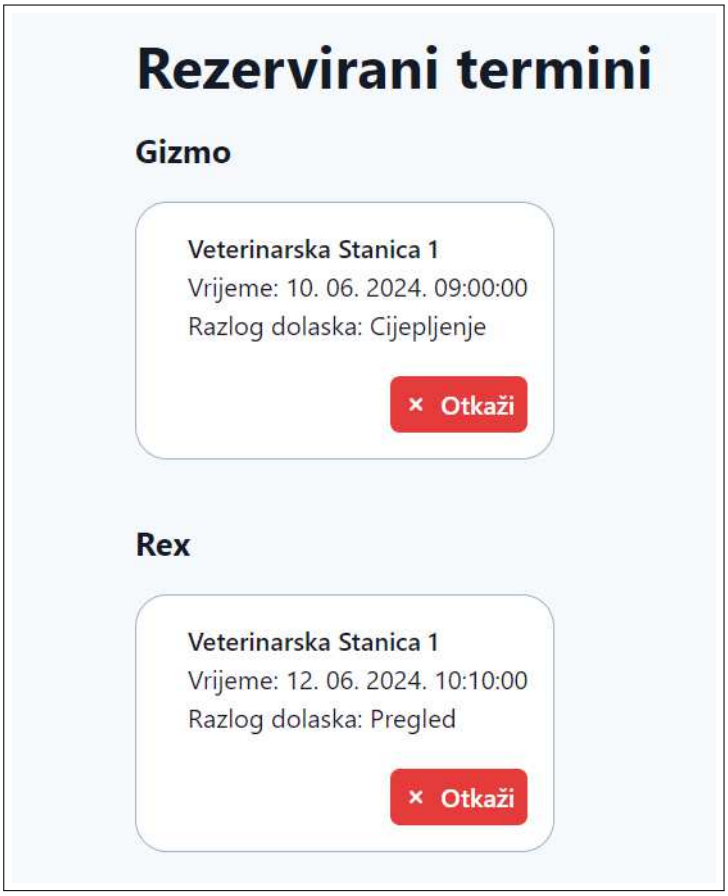
Na sljedećoj slici je rezervacija drugog termina. Odabrala sam istu veterinarsku stanicu kako bi mogla prikazati na koji način rezervaciju funkcionira. Za drugog ljubimca sam odabrala isto vrijeme kao i za prvog. Tada ćemo dobiti poruku pogreške da je termin rezerviran i odmah će nam reći koji je sljedeći dostupan termin kojeg možemo rezervirati. Svakih 15 minuta se može rezervirati novi termin.



Slika 4.21. Rezervacija drugog termina - Vlasnik

4.12.3. Prikaz rezerviranih termina

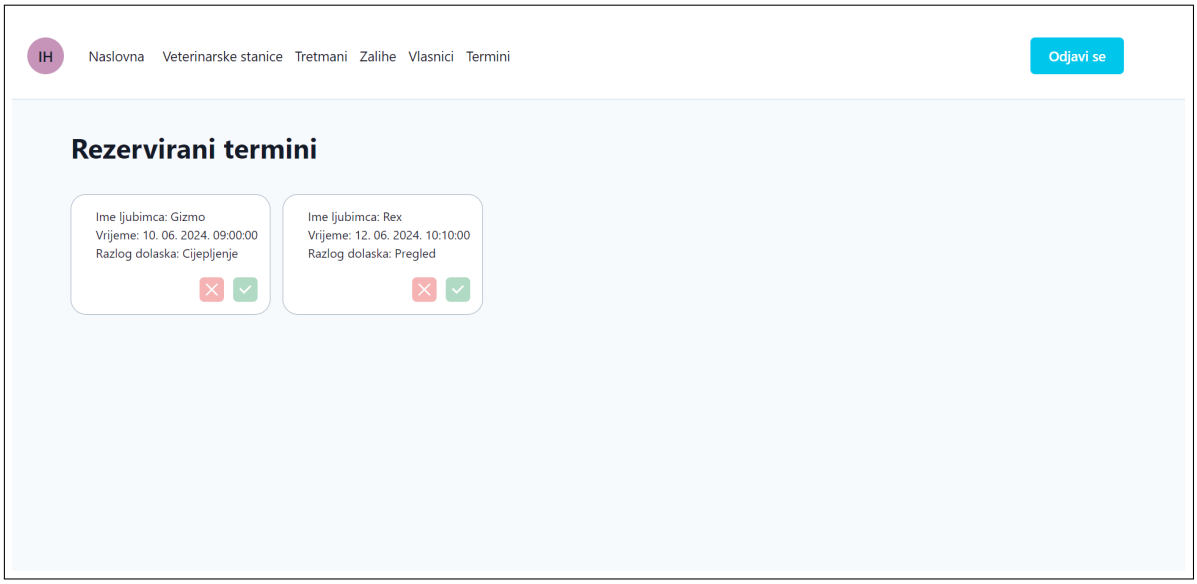
Nakon rezerviranih termina ispod obrasca za rezervaciju vlasnik može vidjeti sve svoje rezervacije. Ako želi može ih i otkazati.



Slika 4.22. Prikaz rezerviranih termina - Vlasnik

4.13. Stranica za pregled rezerviranih termina

Ovoj stranici ima pristup veterinar. Vidljive su mu sve rezervacije koje su poredane po vremenu. Kada dođe vrijeme rezervacije omogućuje mu se pritisak na gumbe. Lijevi gumb označava da se vlasnik sa ljubimcom nije pojavio, a desni da je. Klikom na desni gumb veterinara se preusmjerava na stranicu dolaska i računa tog ljubimca.



Slika 4.23. Stranica za pregled rezerviranih termina - Veterinar

5. Zaključak

U završnom radu napravljena je web aplikacija za veterinarske stanice. U početku je bio cilj osmisliti funkcionalnosti koje bi trebale biti prisutne u jednom takvom sustavu. Dalje je slijedilo definiranje baze podataka te sama implementacija poslužitelja i klijentske aplikacije. Ova aplikacija povezuje sve veterinarske stanice, veterinare i vlasnike kako bi sve bilo lako dostupno. Ovaj sustav je zamišljen kao centraliziran zato što svaki veterinar ima pristup medicinskoj dokumentaciji ljubimca iz neke druge veterinarske stanice. Vlasnik, ako želi, u bilo kojem trenutku može pristupiti svim podacima o svojim ljubimcima koje veterinari bilježe te računima i dolascima, to jest medicinskoj dokumentaciji za svakog ljubimca.

Smatram kako ova aplikacija ima još dosta prostora i potencijala za napredak. Tu bih izdvojila dodavanje chata između veterinara i vlasnika, plaćanje računa preko aplikacije, rezervacije termina i pregled rezervacija u obliku kalendara. Također, dizajn same aplikacije bi se sigurno mogao poboljšati.

Tijekom cijelog razvoja ove aplikacije naučila sam puno toga, a najviše rad u Reactu te korištenje već gotovih komponenti pomoću ChakraUI. To mi je znatno ubrzalo razvoj ove aplikacije kako bih mogla implementirati sve zamišljeno za ovaj rad.

Literatura

[1] React, “React”, Poveznica:https://react.dev/, online. Pristupljeno: svibanj 2024.

[2] TypeScript, “Javascript with syntax for types”, Poveznica:https://www.typescriptlang.org/, online. Pristupljeno: travanj 2024.

[3] Node.js, “Run javascript everywhere”, Poveznica:https://nodejs.org/en, online. Pristupljeno: travanj 2024.

[4] PostgreSQL, “The world’s most advanced open source database”, Poveznica:https://www.postgresql.org/, online. Pristupljeno: travanj 2024.

[5] Dbeaver, “Free universal database tool”, Poveznica:https://dbeaver.io/, online. Pristupljeno: travanj 2024.

[6] TypeORM, “Amazing orm for typescript and javascript”, Poveznica:https://typeorm.io/, online. Pristupljeno: travanj 2024.

[7] Docker, “Accelerated container application development”, Poveznica:https://www.docker.com/, online. Pristupljeno: travanj 2024.

[8] Google, “Google drive api overview”, Poveznica:https://developers.google.com/drive/api/guides/about-sdk, online. Pristupljeno: svibanj 2024.

Sažetak

Web-aplikacija za veterinarske stanice

Karla Šmuk

Cilj ovoga rada je izrada web-aplikacije za veterinarske stanice. Implementacija je započela analizom i razradom korisničkih zahtjeva te definiranjem baze podataka. Nakon toga započela je implementacija poslužitelja i klijentske aplikacije. Rezultat ovoga rada je web aplikacija koja veterinarima omogućava praćenje i upravljanje vlasnicima i ljubimcima te njihovim dolascima. Za veterinarsku stanicu kojoj pripadaju, veterinari mogu pratiti i upravljati zalihama, tretmanima te rezervacijama vlasnika. Vlasnicima omogućava praćenje svojih ljubimaca, dolazaka, računa te rezervacije termina. Administratoru omogućava praćenje i upravljanje veterinarskim stanicama i veterinarima. U pisanom dijelu objašnjeno je korisničko sučelje i njegovo korištenje.

Ključne riječi: veterinarska stanica; web aplikacija; React; ChakraUI; Node.js; TypeScript; PostgreSQL

Abstract

Web Application for Veterinary Clinics

Karla Šmuk

The aim of this thesis is to create a web application for veterinary clinics. The implementation began with the analysis and elaboration of user requirements and the definition of the database. After that, the implementation of backend and frontend started. The result of this thesis is web application that allows veterinarians to manage owners, their pets and visits. For the veterinary clinics to which they belong, veterinarians can manage supplies, treatments and owners appointments. It allows owners to view all the details about their pets, visits, invoices and appointments. The administrator has the ability to manage veterinary clinics and their veterinarians. The written part of this thesis explains the user interface and its use.

Keywords: veterinary clinic; web application; React; ChakraUI; Node.js; TypeScript; PostgreSQL