**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

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

**PROJEKAT**

**iz predmeta Web Programiranje**

**KANDIDATI MENTOR**

**Mihajlo Baranovski dr Zlatko Čović**

**Milana Sabovljev**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SUBOTICA 2024. god.**

**SADRŽAJ**

[OPIS ZADATKA 1](#_Toc20668)

[STRUKTURA BAZE PODATAKA 4](#_Toc11184)

[REALIZACIJA ZADATKA 8](#_Toc13639)

[OPIS FUNKCIONALNOSTI 9](#_Toc26460)

[KORIŠĆENA LITERATURA 1](#_Toc18942)6

# 

# OPIS ZADATKA

**Veterinarska ordinacija**

**Opis projekta**:

Cilj projekta je da se kreira web aplikacija-sajt za zakazivanje termina u veterinarskoj ordinaciji.

Aplikacija razlikuje četiri nivoa pristupa: gost, registrovan/prijavljen korisnik, veterinar-radnik i administrator.

**Gost**

• može da pogleda ponuđene informacije na sajtu

• može da pregleda spisak svih veterinara sa njihovim fotografijama i uslugama za koje su specijalizovani

• može da izvrši registraciju na sajtu Registrovan/prijavljen korisnik

• može da rezerviše termin za tretman kod izabranog veterinara (maksimalno mesec dana unapred)

• može da promeni svoje profilne podatke (lozinka, ime, prezime, broj telefona)

• može da dodaje, izmeni i obriše podatke za svoje kućne ljubimce (ime, godina rođenja, vrsta, rasa, fotografija)

• može da izmeni, pregleda i obriše svoje podatke o rezervacijama

• može da pogleda informacije o tretmanima svog ljubimca (elektronski karton)

• može da otkaže rezervisani termin (najmanje do 4 sata pre rezervisnog termina)

• može da zahteva promenu lozinke (zaboravljena lozinka)

**Veterinar**

• može da unese informacije o svakom tretmanu

• može da pregleda elektronski karton za svakog kućnog ljubimca

• može da unese, promeni i obriše svoje termine kada radi u ordinaciji

**Administrator**

• može da unosi, menja i briše podatke o uslugama (naziv i cena)

• može da unosi, menja i briše podatke o veterinarima (povezati veterinare sa uslugama)

• može da pregleda informacije o svim terminima

• može da zabrani/odobri korisniku pravo prijavljivanja na sistem

• može da na osnovu odabira datuma pregleda termine za taj datum. Svi podaci ostaju u bazi podataka.

Registraciju korisnika mora biti urađena na bezbedan način i obaveznim slanjem aktivacionog linka putem e-maila. Slanje linka koristiti i kod zahteva za promenu lozinke. Ne sme se dozvoliti registracija više korisničkih naloga sa istom e-mail adresom. E-mail adresa mora biti jedinstvena i ona predstavlja korisničko ime.

Prijavljeni korisnik bira svog ljubimca, nakon toga vrši odabir veterinara, usluge, datuma i termina za tretman. Za svakog veterinara treba da postoji informacije o imenu, specijalizaciji, uslugama koje pruža i o slobodnim terminima.

Pre rezervacije termina obavezno proveravati da li je termin zauzet kao i da li se preklapaju vremenski periodi sa postojećim rezervacijama. Korisnik može da otkaže rezervaciju termina ali najmanje 4 sata pre početka termina tretmana. Nakon uspešne rezervacije slati e-mail korisniku koji je rezervisao termin sa kôdom rezervacija. Ovaj kôd prikazuje korisnik prilikom dolaska u ordinaciju.

Veterinar na osnovu prikazaog kôda otvara novi zapis u elektronskom kartonu gde unosi informacije o tretmanu i o ceni tretmana. Kôd treba da bude poveznik sa terminom, korisnikom i kućnim ljubimcem. Ukoliko se korisnik ne pojavi u datom terminu, veterinar evidentira to. Korisnik za nepojavljivanje dobija 1 negativan poen.

U okviru administracije treba da postoji opcija za prikaz korisnika i njihovih negativnih poena. Na osnovu broja negativnih poena administrator može da blokira korisničke naloge.

Administratorski deo zaštiti upotrebom sesija (PHP). Sve korisničke lozinke „hešovati“ bcrypt algoritmom.

Kreirati bazu podataka pod imenom vet i unutar nje tabele koje će zadovoljiti sve funkcionalnosti projekta.

**Zahtevi i Smernice**:

• Pored navedenih obaveznih funkcionalnosti studenti mogu dodati i neku svoju

funkcionalnost koja će po njihovom mišljenju unaprediti projekat.

• Upotreba spoljašnjeg fajla sa JavaScript kodom je obavezna. Programski kod

(promenljive, funkcije, objekte...) pisati na engleskom jeziku, koristiti komentare u

kodu, držati se uputstva koje je dato u posebnom fajlu coding\_style\_guide\_sr.pdf

• Obavezna upotreba spoljašnjeg fajla sa CSS kodom.

• Pristupne parametre za povezivanje sa MySQL serverom definisati u spoljašnjem fajlu db\_config.php. Za rad sa MySQL bazom koristiti PDO ekstenziju unutar PHP jezika. Deo PHP programskog koda mora biti objektno orijentisan.

• Projekat treba da bude multiplatformski (Responsive) i da bude prilagođen računarima i mobilnim uređajima.

• U okviru projekta obavezno treba koristiti sledeće tehnike i tehnologije: HTML, CSS, JavaScript, AJAX ili Fetch API, JSON, Bootstrap, PHP i MySQL. Za slanje mailova koristiti PHPMailer klasu (https://github.com/PHPMailer/PHPMailer).

• Programski deo koji koristi AJAX ili Fetch API tehniku mora da manipuliše sa podacima iz baze podataka. Preporuka za korišćenje AJAX ili Fetch API tehnika: provera validnosti podataka, nabavljanje podataka iz baze podataka, registracija i provere kod registracije.

• Na stranicama gde se radi validacija podataka obavezno uraditi validaciju na klijentskoj i serverskoj strani.

• Projekat je potrebno postaviti na školski web server. Svaki pojedinac ili tim će dobiti svoje pristupne parametre putem e-maila.

• Upotreba ostalih tehnologija, biblioteka i API-a je opciona.

• Koristiti Trello sistem unutar tima i dodeliti pristup predmetnom nastavniku (prowebing@gmail.com i helenatmanojlovic@gmail.com). Tokom rada na projektu definisati zadatke i dodeliti ih članovima tima. Potrebne liste u Trello sistemu: Additional Documents (Info), Brainstorming, To do, Doing, Testing, Done, To be Archived. Opise zadataka možete pisati na srpskom jeziku.

• Svaki tim ili pojedinac treba da koristi Git VCS sistem (Version Control System) i zajedno sa njim neko hosting rešenje (GitHub, BitBucket, GitLab...). Dodeliti pristup predmetnom nastavniku (chole@vts.su.ac.rs, helena@vts.su.ac.rs). Tokom rada na projektu potrebno je postavljati programski kod na svoj Git repozitorijum.

• Projekat se predaje u elektronskom obliku (programski kod, .sql fajlovi, i projektna dokumentacija). Dokumentacija se izrađuje prema datom šablonu. Sve je potrebno postaviti na svoj Git repozitorijum.

• Tokom realizacije projekta veliki naglasak treba da bude na sigurnosti i proveri unetih podataka.

# STRUKTURA BAZE PODATAKA

CREATE TABLE `admin` (

  `id\_admin` int(11) NOT NULL,

  `firstname` varchar(20) NOT NULL,

  `lastname` varchar(20) NOT NULL,

  `email` varchar(50) NOT NULL,

  `password` varchar(255) NOT NULL,

  `active` smallint(1) NOT NULL *DEFAULT* 0,

  `registration\_token` char(40) NOT NULL,

  `registration\_expires` datetime *DEFAULT* NULL

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `done\_procedures` (

  `id\_done\_procedure` int(11) NOT NULL,

  `id\_reservation` int(11) NOT NULL,

  `diagnosis` varchar(70) NOT NULL,

  `prescribed\_medication` varchar(70) NOT NULL,

  `pet\_condition` varchar(65) NOT NULL,

  `procedure\_duration` time NOT NULL,

  `procedure\_date` date NOT NULL

)

CREATE TABLE `pet` (

  `id\_pet` int(11) NOT NULL,

  `id\_user` int(11) NOT NULL,

  `id\_vet` int(11) *DEFAULT* NULL,

  `pet\_name` varchar(50) NOT NULL,

  `age` smallint(2) *DEFAULT* NULL,

  `species` varchar(50) NOT NULL,

  `breed` varchar(50) *DEFAULT* NULL,

  `photo` varchar(255) *DEFAULT* NULL

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `reservations` (

  `id\_reservation` int(11) NOT NULL,

  `id\_pet` int(11) NOT NULL,

  `id\_service` int(11) NOT NULL,

  `id\_vet` int(11) NOT NULL,

  `reservation\_date` date NOT NULL,

  `reservation\_time` time NOT NULL,

  `treatment\_price` int(11) NOT NULL,

  `code` varchar(255) NOT NULL,

  `code\_verified` smallint(1) NOT NULL *DEFAULT* 0,

  `reservation\_added` datetime NOT NULL *DEFAULT* current\_timestamp(),

  `service\_duration` time NOT NULL,

  `cancel` smallint(1) NOT NULL *DEFAULT* 0

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `services` (

  `id\_service` int(11) NOT NULL,

  `service\_name` varchar(70) NOT NULL,

  `service\_description` varchar(255) NOT NULL,

  `photo` varchar(255) *DEFAULT* NULL,

  `service\_duration` time NOT NULL,

  `price` int(11) *DEFAULT* NULL,

  `discount` int(2) *DEFAULT* NULL,

  `service\_date` date *DEFAULT* NULL,

  `service\_time` time *DEFAULT* NULL

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `users` (

  `id\_user` int(11) NOT NULL,

  `user\_fname` varchar(25) NOT NULL,

  `user\_lname` varchar(50) NOT NULL,

  `user\_phone` int(20) NOT NULL,

  `age` int(3) *DEFAULT* NULL,

  `photo` varchar(70) *DEFAULT* NULL,

  `user\_email` varchar(75) NOT NULL,

  `user\_password` varchar(255) NOT NULL,

  `active` smallint(1) NOT NULL *DEFAULT* 0,

  `registration\_token` char(40) NOT NULL,

  `registration\_expires` datetime *DEFAULT* NULL,

  `is\_banned` smallint(1) *DEFAULT* 0,

  `forgotten\_password\_token` char(40) *DEFAULT* NULL,

  `forgotten\_password\_expires` datetime *DEFAULT* NULL,

  `registration\_date` datetime *DEFAULT* current\_timestamp() ON UPDATE current\_timestamp(),

  `negative\_points` int(1) NOT NULL *DEFAULT* 0

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `vet` (

  `id\_vet` int(11) NOT NULL,

  `vet\_fname` varchar(30) NOT NULL,

  `vet\_lname` varchar(40) NOT NULL,

  `vet\_email` varchar(70) NOT NULL,

  `vet\_password` varchar(255) NOT NULL,

  `vet\_phone` int(20) NOT NULL,

  `photo` varchar(80) *DEFAULT* NULL,

  `specialization` varchar(40) NOT NULL,

  `city` varchar(40) NOT NULL,

  `active` smallint(1) NOT NULL *DEFAULT* 0,

  `registration\_token` char(40) NOT NULL,

  `registration\_expires` datetime *DEFAULT* NULL,

  `is\_banned` smallint(1) *DEFAULT* 0,

  `forgotten\_password\_token` char(40) *DEFAULT* NULL,

  `forgotten\_password\_expires` datetime *DEFAULT* NULL,

  `registration\_date` datetime NOT NULL *DEFAULT* current\_timestamp() ON UPDATE current\_timestamp(),

  `work\_start` time(4) NOT NULL,

  `work\_end` time(4) NOT NULL

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `vet\_email\_failures` (

  `id\_vet\_email\_failure` int(11) NOT NULL,

  `id\_user` int(11) NOT NULL,

  `message` varchar(255) NOT NULL,

  `date\_time\_added` datetime NOT NULL,

  `date\_time\_tried` datetime *DEFAULT* NULL

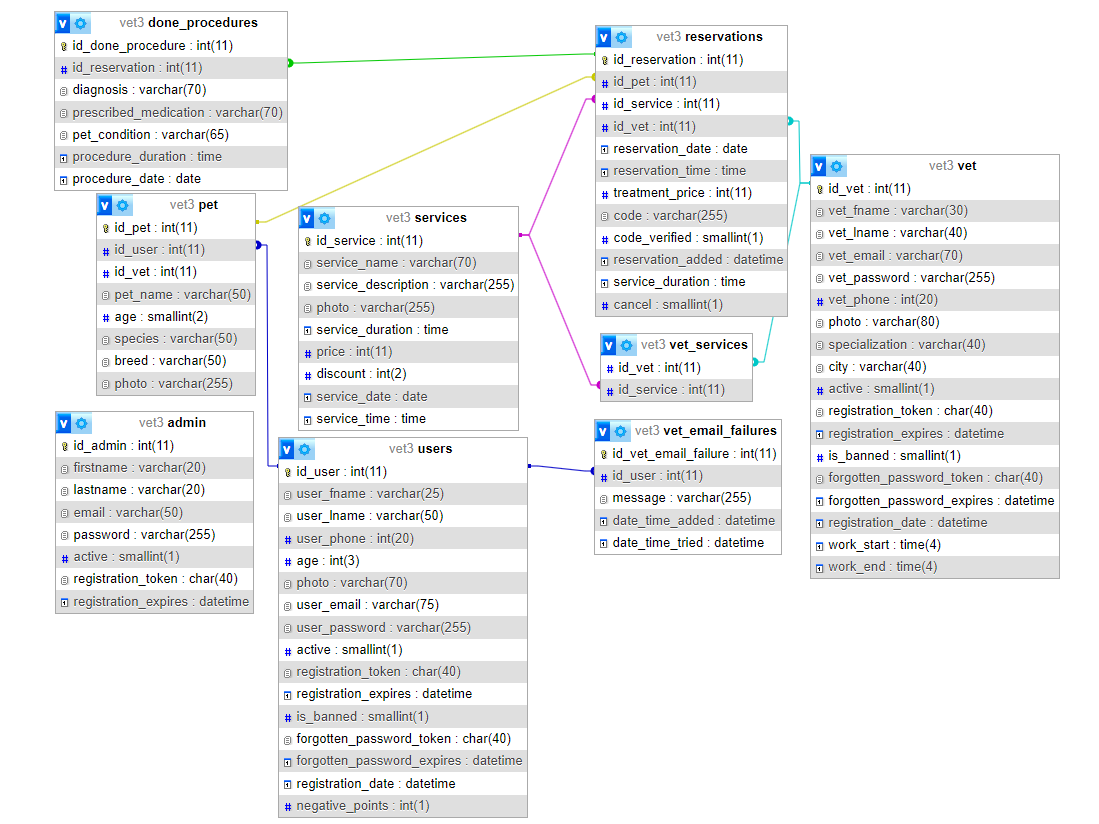
) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;

CREATE TABLE `vet\_services` (

  `id\_vet` int(11) NOT NULL,

  `id\_service` int(11) NOT NULL

) ENGINE=InnoDB *DEFAULT* CHARSET=utf8mb4 *COLLATE*=utf8mb4\_general\_ci;



# REALIZACIJA ZADATKA

**db\_config.php**



**signin.php** **login.php forgotten\_password.php**

**edit\_profile.php edit\_user.php edit\_vet.php**

**functions.php script.js style.css**

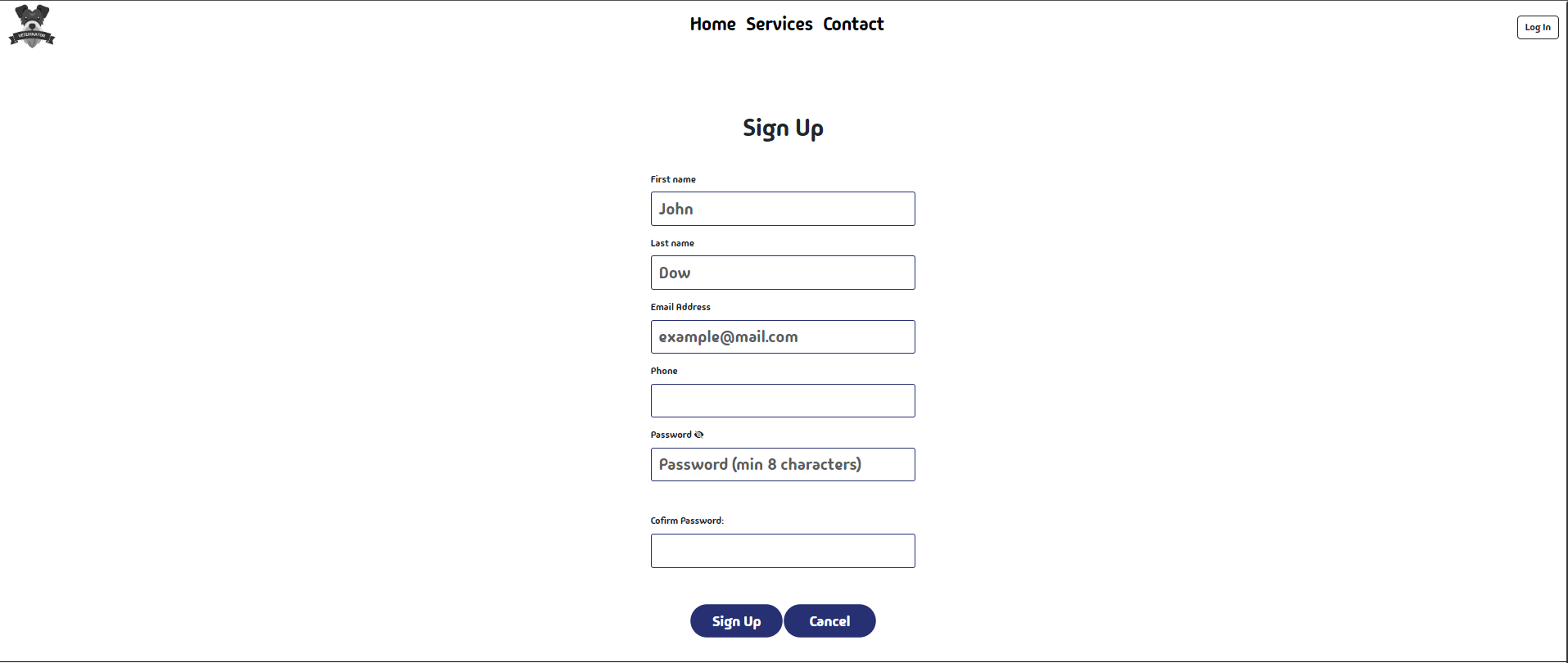
**reservations.php reservations.js getResevations.php**

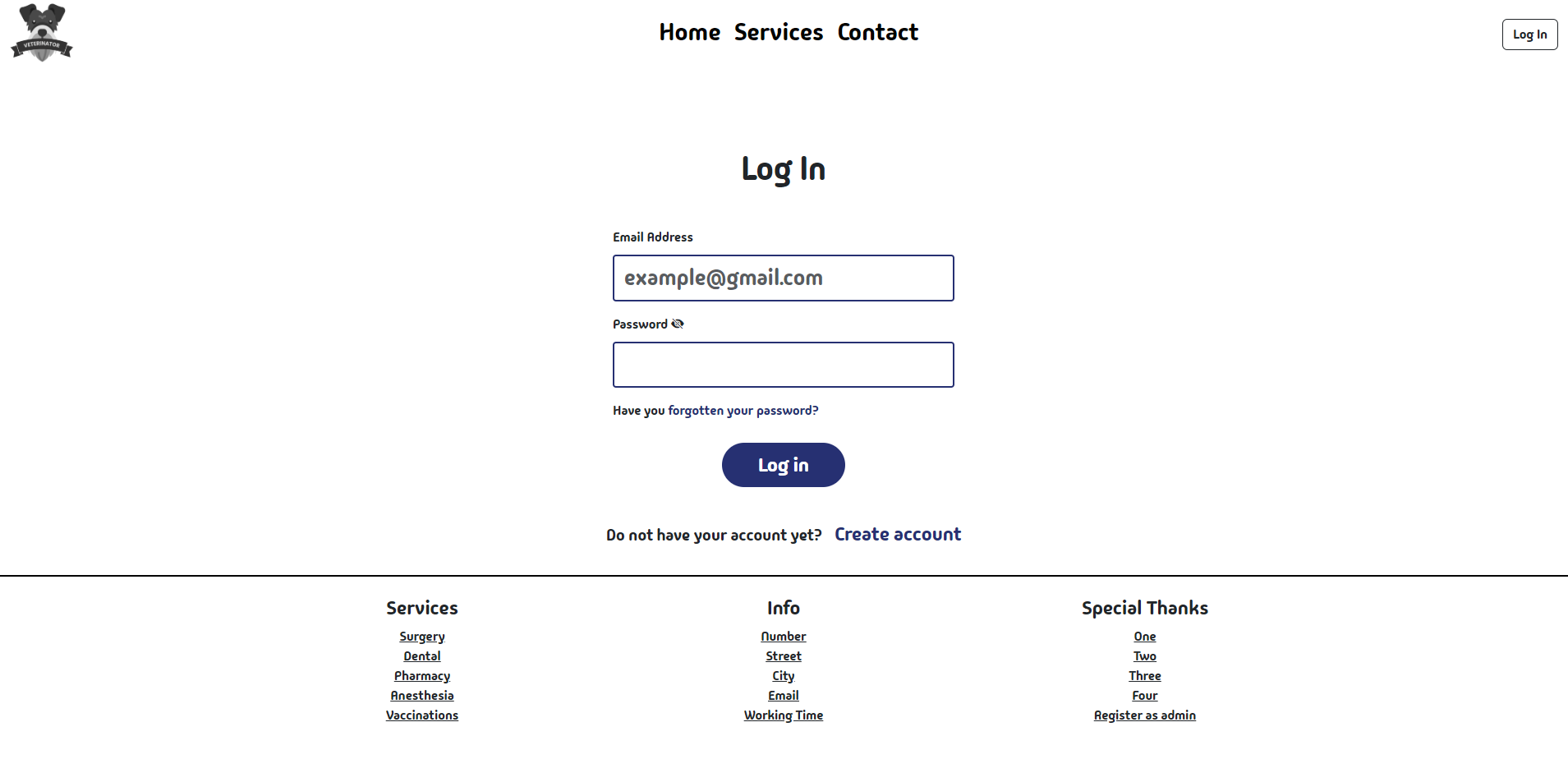
**admin\_service\_f.php admin\_service\_desc.php services\_description.js**

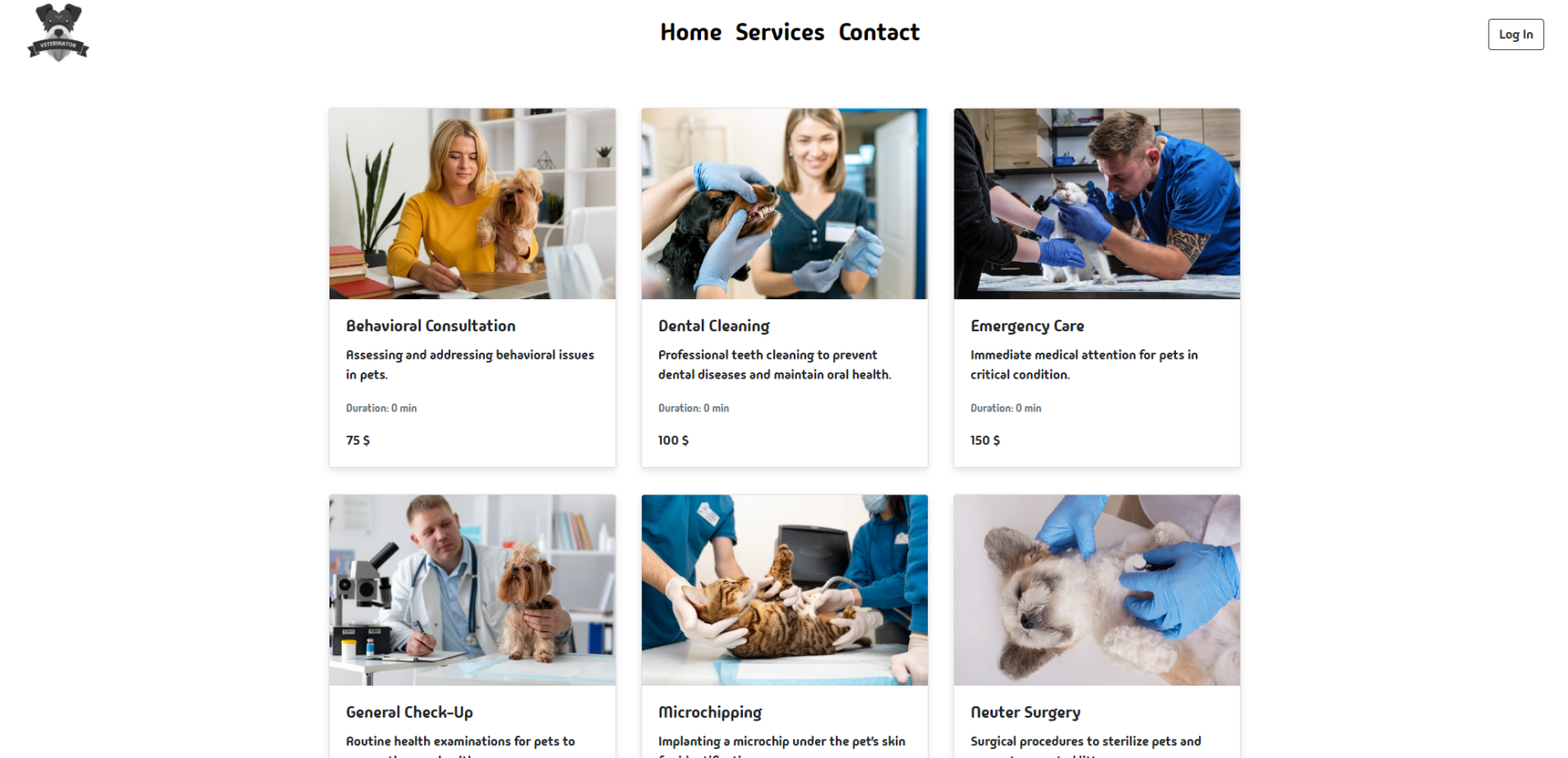
# OPIS FUNKCIONALNOSTI



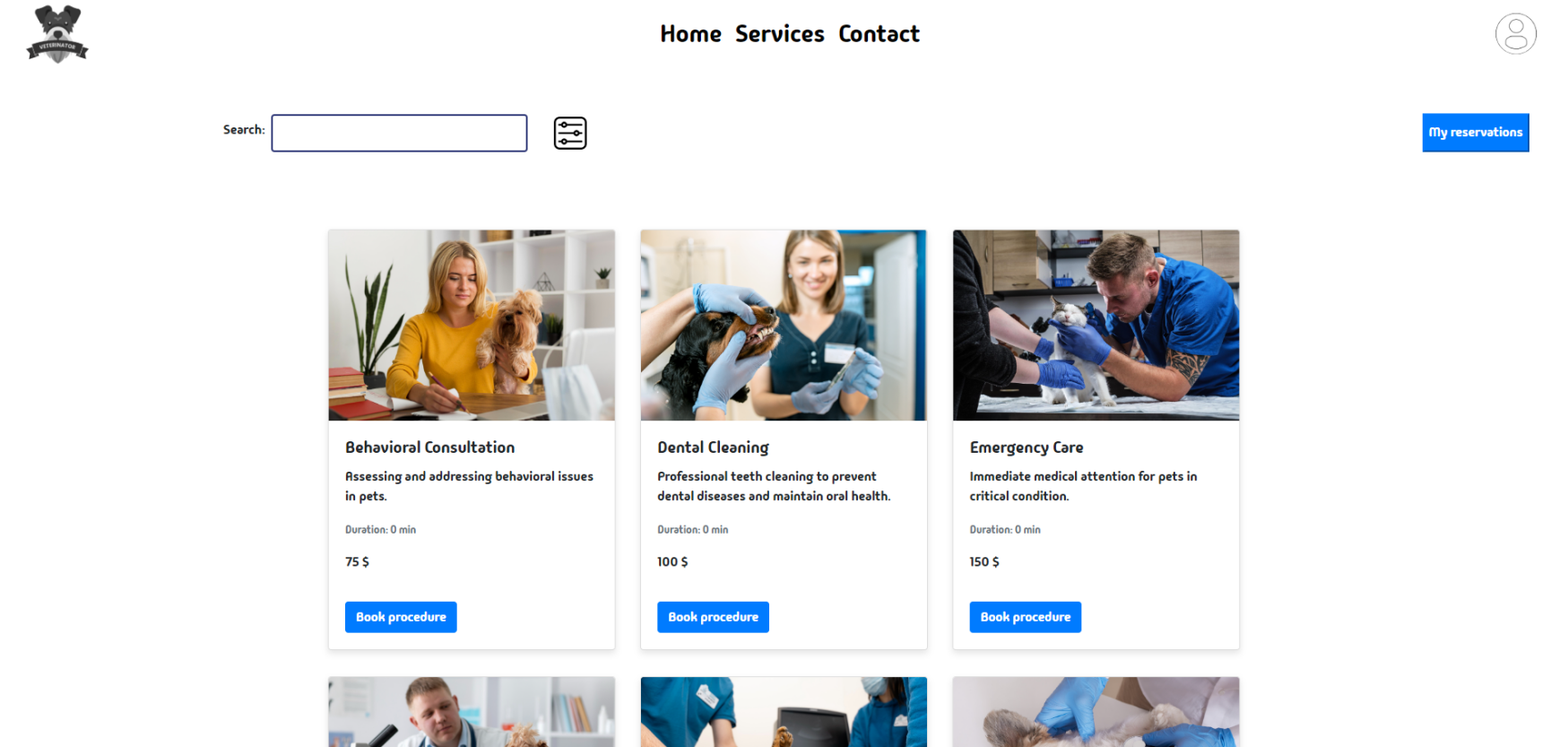
Register stranica za registraciju novih korisnika!



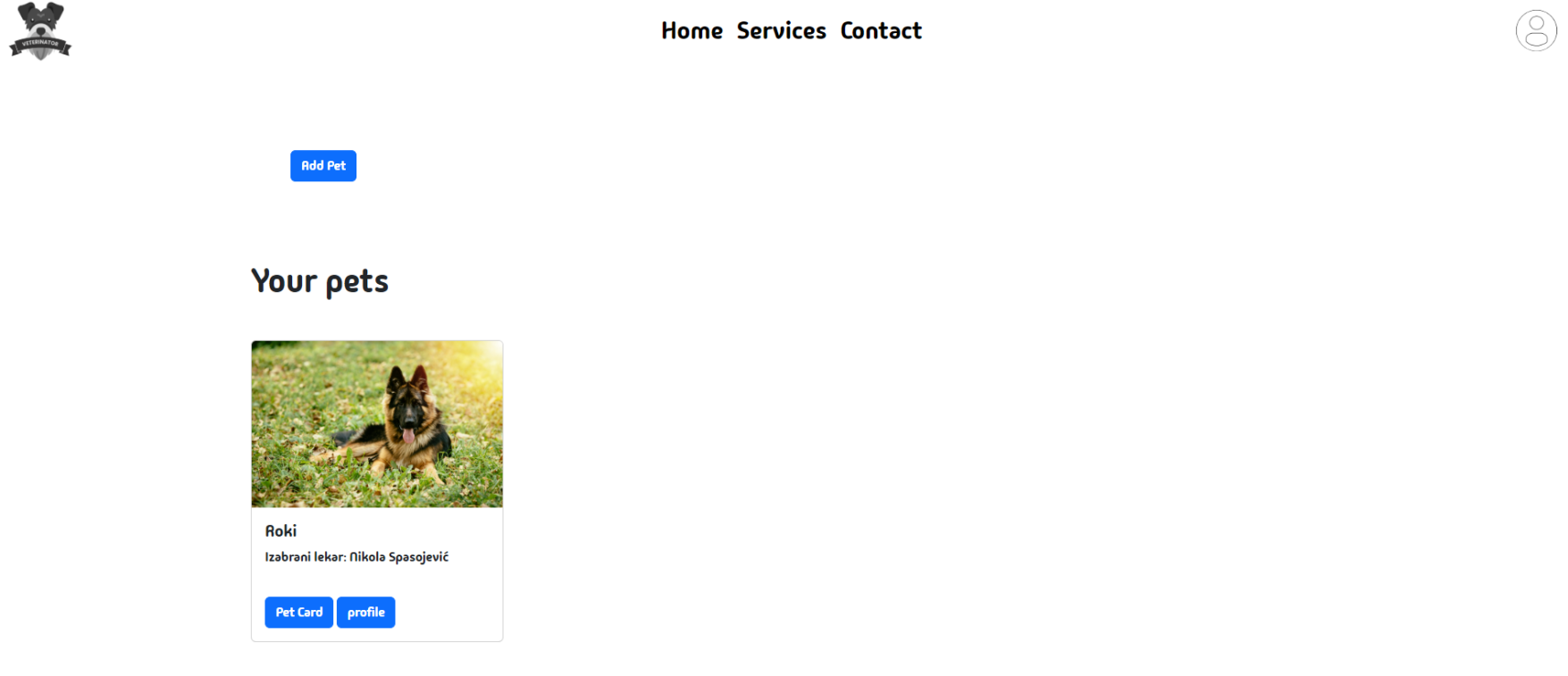
Login stranica za ulogovanje postojećih/novih korisnika!



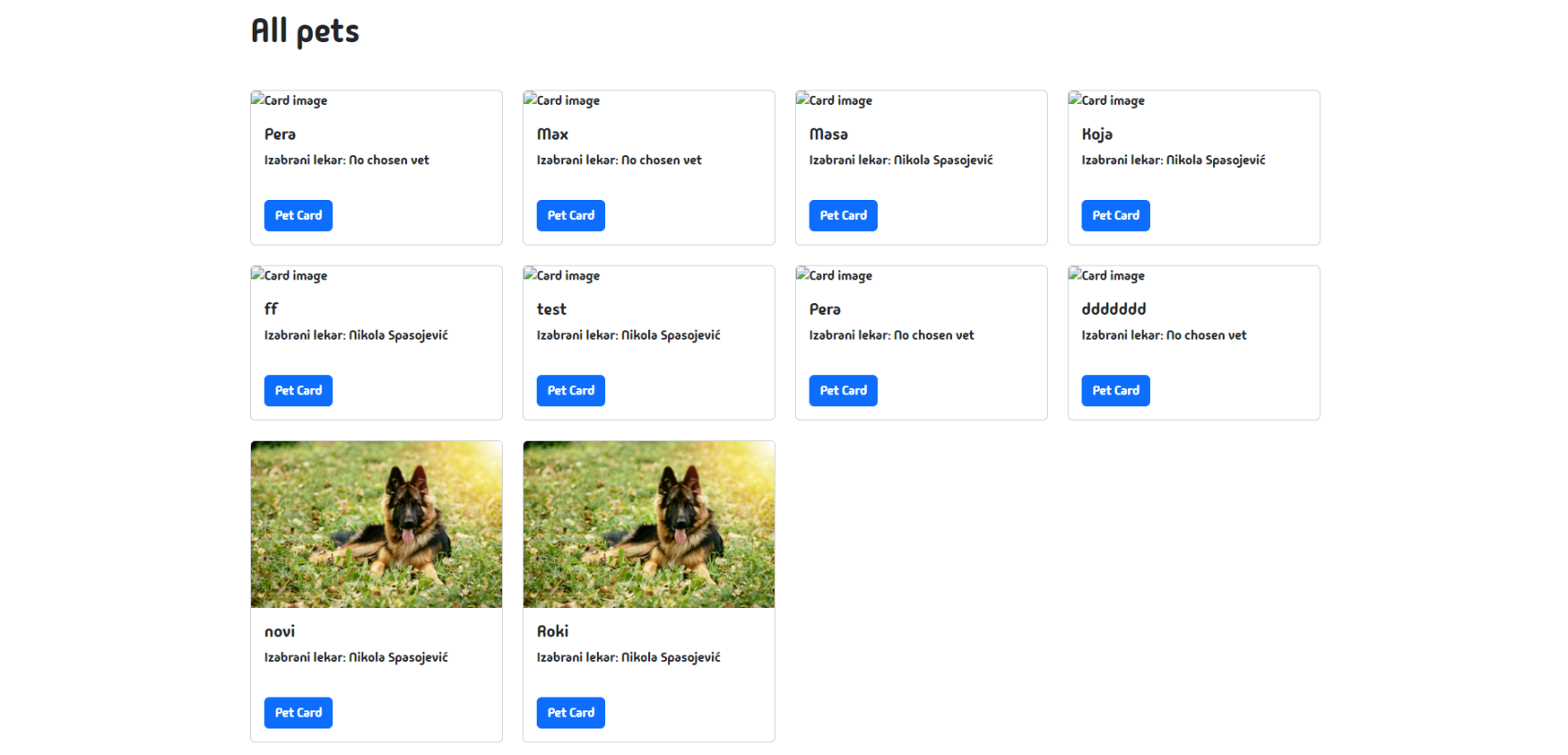
Lista usluga kada korisnik nije ulogovan!



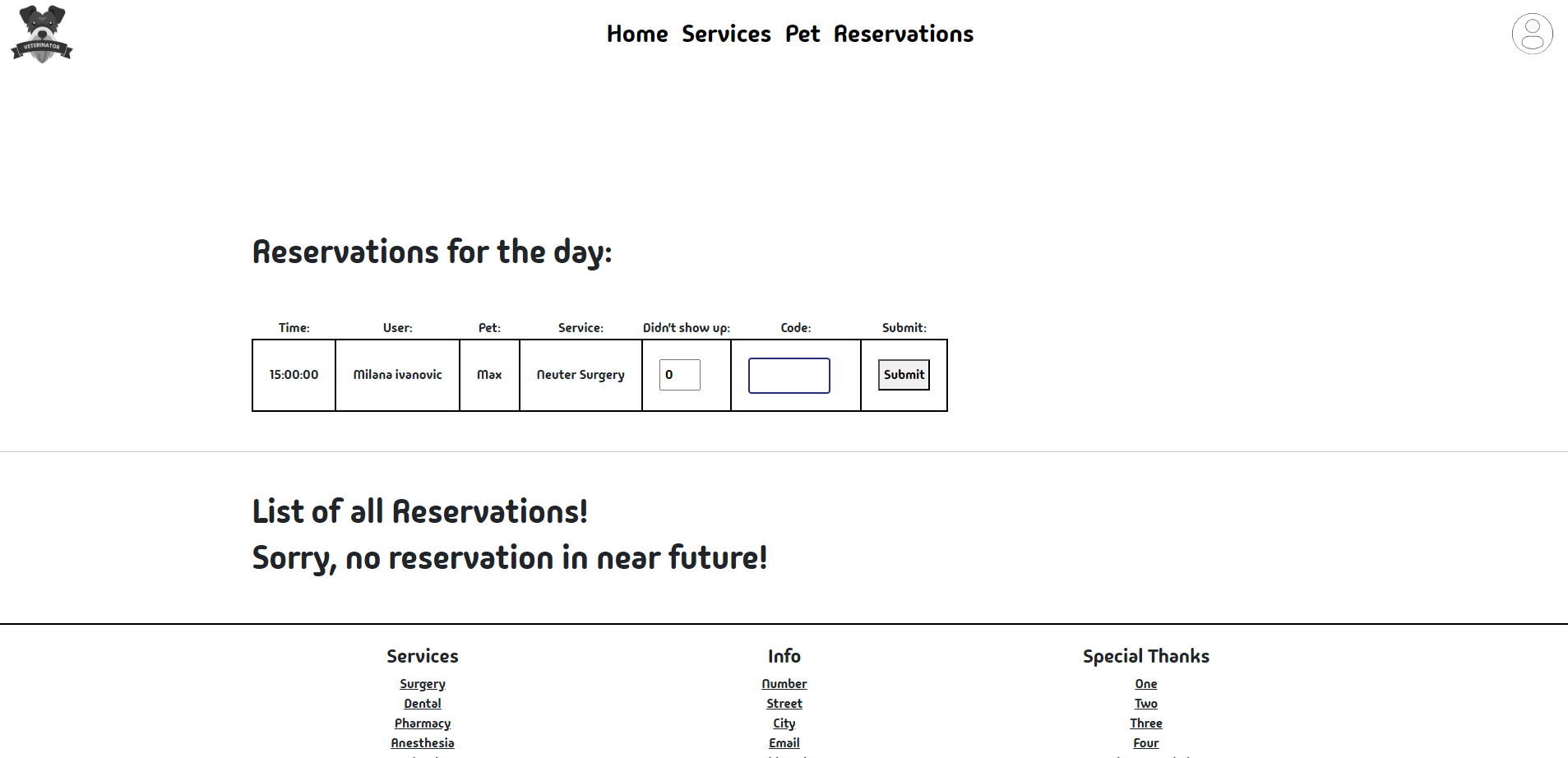
Lista usluga kada je korisnik ulogovan!



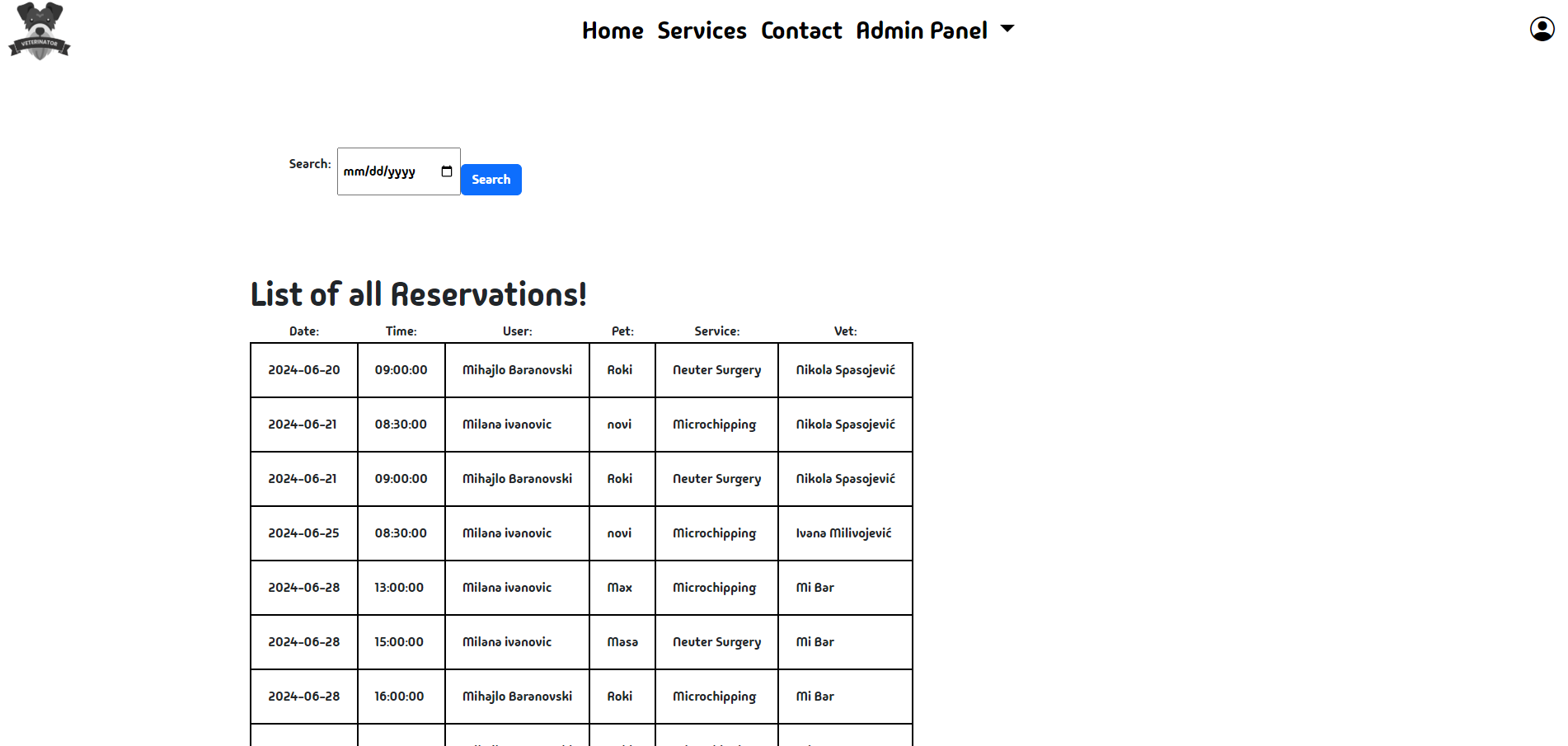
Lista korisničkih ljubimaca!



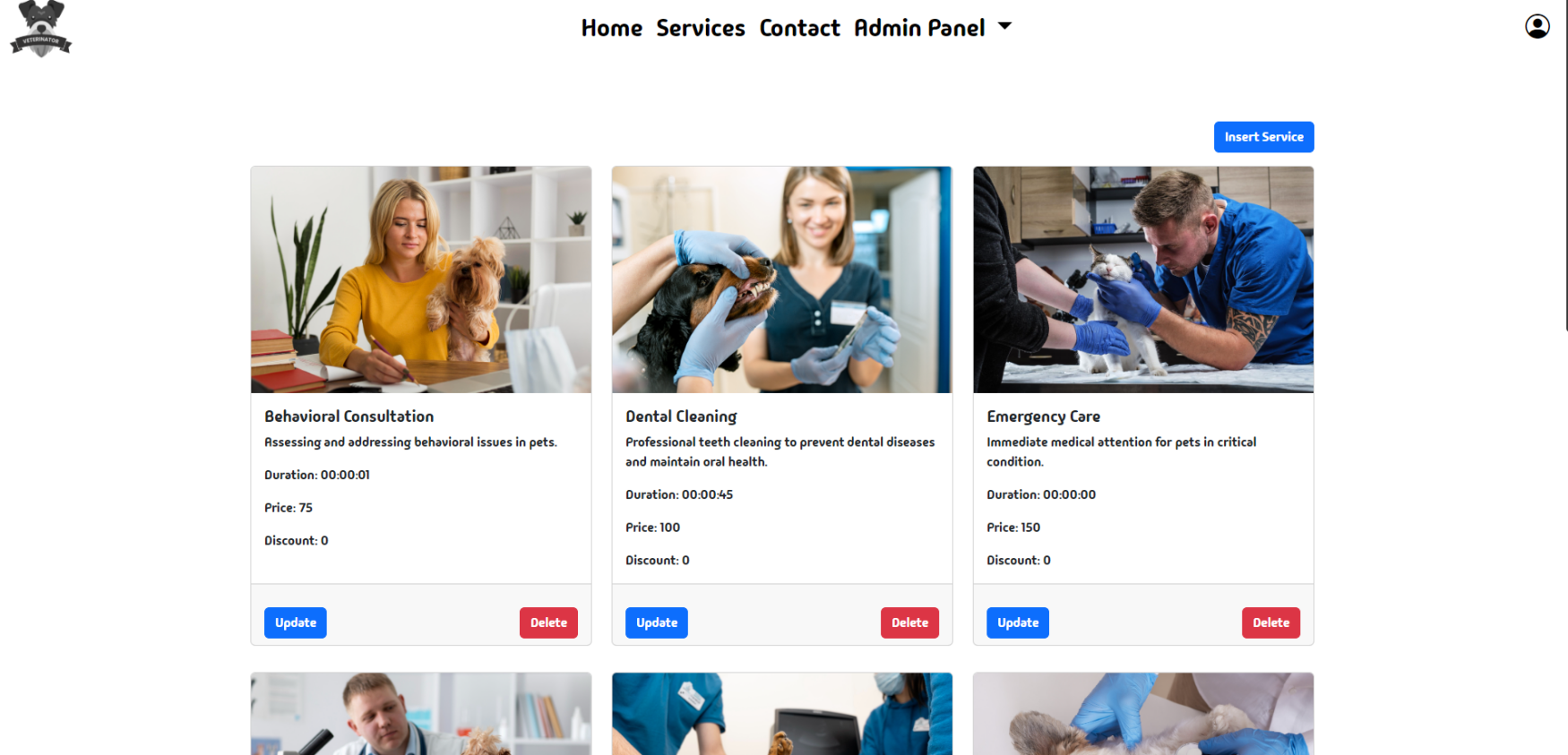
Lista ljubimaca određenog veterinara!



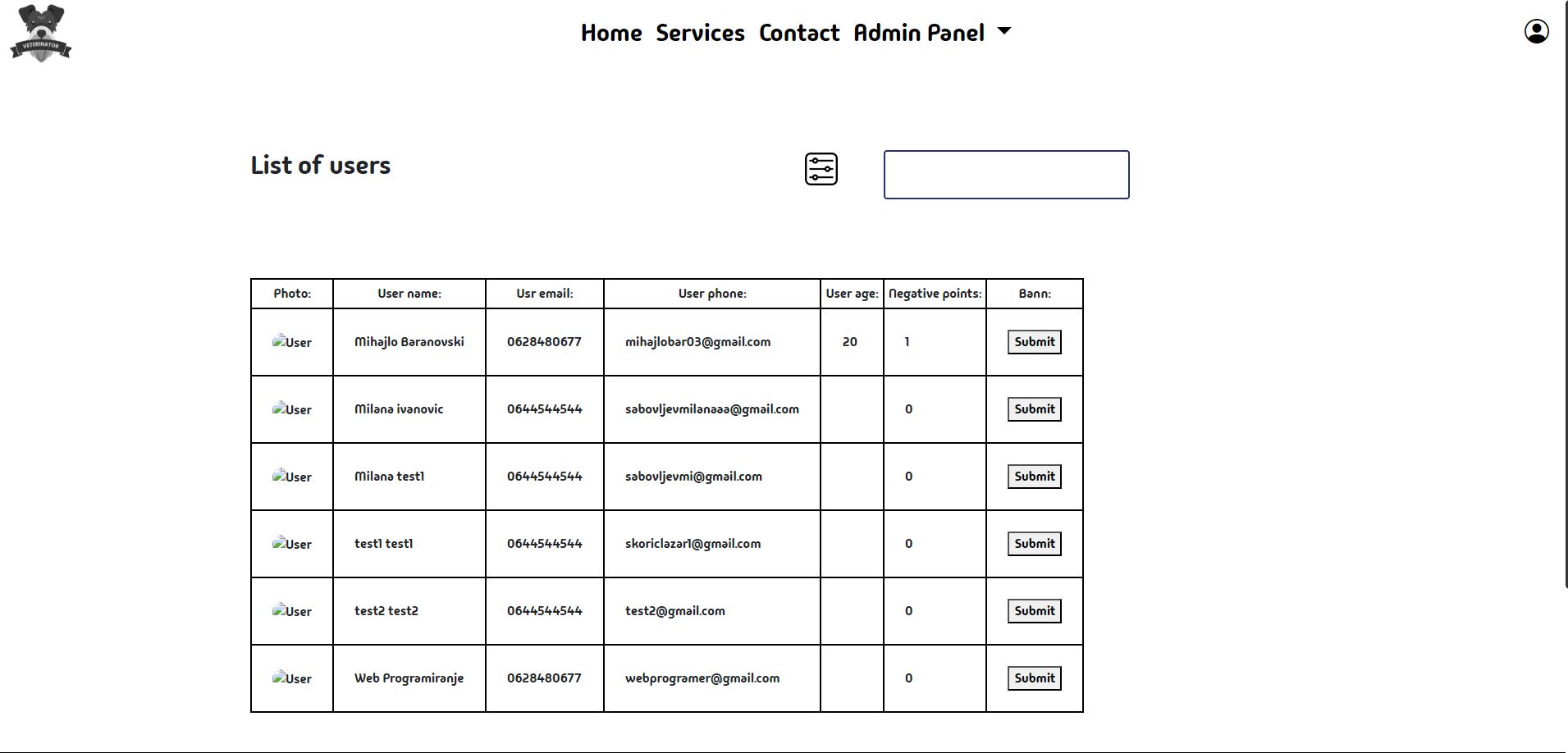
Rezervacije koje veterinar ima!



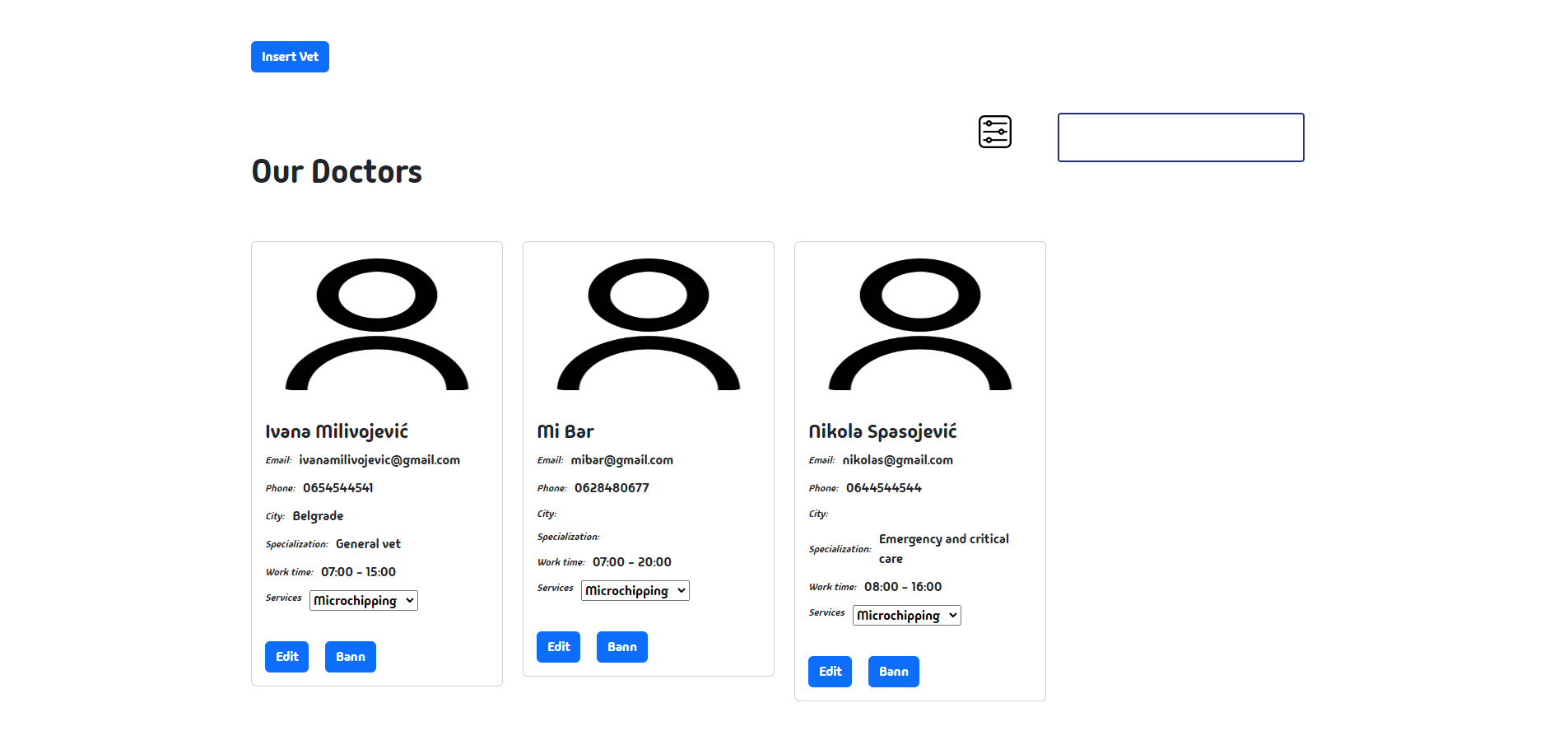
Lista svih rezervacija unutar Admin Panel-a!



Lista usluga unutar Admin Panel-a!



Lista svih korisnika unutar Admin Panel-a!



Lista svih veterinara unutar Admin Panel-a!

# KORIŠĆENA LITERATURA

1. [https://people.vts.su.ac.rs/~chole/wp/](https://people.vts.su.ac.rs/~chole/wp/primeri-peldak/)
2. <http://www.w3schools.com/>
3. <https://stackoverflow.com/>
4. <https://www.php.net/>