



VEB APLIKACIJA ZA ONLAJN ZAKAZIVANJE KOD VETERINARA

A WEB APPLICATION FOR AN ONLINE APPOINTMENT AT THE VET

Sandra Martinović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – INŽENJERSTVO INFORMACIONIH SISTEMA

Kratak sadržaj – *Danas se sve više korisnika odlučuje za zakazivanje ili kupovinu preko interneta, kako bi uštedeli na svom vremenu. Kada je reč o zakazivanju pregleda kod doktora, ovakav način nije toliko uobičajen. Rešenje koje je predstavljeno u ovom radu, trebalo bi da prikaže primer kako bi mogao da izgleda jedan jednostavan sistem koji pruža onlajn zakazivanje za ljubimce. Veb aplikacija koja omogućava onlajn zakazivanje termina kod veterinara je implementirana koristeći tehnologije kao što su MySQL Workbench, Spring i Angular.*

Ključne reči: *sistemi elektronske uprave, e-zdravstvo, veterinar, onlajn zakazivanje*

Abstract – *Today, more and more users are choosing to make an appointment or shop online, in order to save their time. When it comes to scheduling a doctor's appointment, this method is not so common. The solution presented in this paper should set an example of what a simple online pet booking system might look like. A web application that enables making an online appointment with a vet is implemented using technologies such as MySQL, Spring and Angular.*

Keywords: *electronic government or e-government, e-health, vet or veterinarian, online appointment*

1. UVOD

Sa napredovanjem i razvijanjem tehnologija, povećava se i upotreba interneta, koji je postao jedan od ključnih vidova komunikacije, kako preko društvenih mreža, tako i preko sistema onlajn zakazivanja. Korišćenje interneta postaje rutina u obrazovnim ustanovama, različitim firmama i u različitim delatnostima, što dovodi do želje za proširivanjem onlajn prodaje, kupovine, zakazivanja i do sve veće potrebe za veb developerima (engl. *Web developer*).

Zakazivanje preko interneta evoluiralo iz dana u dan, bilo da je u pitanju zakazivanje pregleda kod lekara ili termina izrade lične karte, pasoša ili vozačke dozvole. S obzirom na to da doskoro ovakav način zakazivanja nije bio ustaljen u Srbiji, okolnosti su počele da se menjaju, i broj onlajn pacijenata i klijenata se povećao. U drugim zemljama je ovakav način funkcionisanja prisutan duži vremenski period i predstavlja popularan i jednostavan način dolaska do termina.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Teodora Lolić, docent.

Različiti faktori utiču na porast broja novih onlajn korisnika. Korišćenje interneta, kako među mladim uzrastom, tako i među starijim, abnormalno raste, pa shodno tome raste i veća zainteresovanost za svetom onlajn kupovine ili zakazivanja.

2. SISTEMI ELEKTRONSKE UPRAVE

Sistemi elektronske uprave omogućavaju da se administrativni poslovi izvrše od kuće, iz kancelarije ili sa neke druge lokacije, kako bi se uštedelo vreme, a poslovi obavili mnogo brže nego inače; na primer, čekanje u redu predstavlja spor način obavljanja poslova. Cilj elektronske uprave je stvaranje novog, dinamičnog odnosa između vlade i građana, a to će za rezultat imati jednostavniji pristup građana elektronskoj upravi.

Elektronsko poslovanje ili e-poslovanje (engl. *E-Business*) predstavlja poslovanje koje se obavlja posredstvom elektronskih sredstava komunikacije, što se danas svodi na upotrebu interneta. Pored same kupovine i prodaje preko interneta, e-poslovanje obuhvata i druge poslovne aktivnosti koje se obavljaju posredstvom interneta, kao što su servisiranje kupaca, učenje na daljinu, upravljanje lancem snabdevanja, upravljanje odnosima sa potrošačima ili upravljanje internim poslovnim procesima. Važan deo e-poslovanja je upotreba informaciono-komunikacionih tehnologija (engl. *Information and Communications Technology, IKT*) i digitalizacija, koja u savremenom poslovanju sve više postaje uslov opstanka na tržištu [1].

Elektronska trgovina ili e-trgovina (engl. *E-Commerce*) užo je pojam od e-poslovanja i odnosi se na poslove kupovine i prodaje koji se obavljaju posredstvom elektronskih sredstava komunikacije, tj. interneta. E-trgovina u užem smislu odnosi se samo na tehnički deo kupovine i prodaje robe preko interneta, dok se u širem smislu odnosi na sve ostale prateće aktivnosti kao što su komunikacija, distribucija ili servisiranje. U praksi se pojmovi e-poslovanja i e-trgovine često poistovećuju, pa je važno napomenuti da je e-poslovanje širi pojam od e-trgovine, te da kao takav ima i širi značaj [1].

Osim klasičnog e-poslovanja i e-trgovine, postoje i posebni oblici elektronskog poslovanja preko interneta kao što su e-bankarstvo (engl. *E-Banking*), e-marketing (engl. *E-Marketing*), e-uprava (engl. *E-Government*), e-učenje (engl. *E-Learning*) ili e-zdravlje (engl. *E-Health*), koji predstavljaju specifične oblike e-poslovanja [1].

Postoje različite vrste onlajn prodaje, kupovine ili zakazivanja, gde su među najpoznatijim oblicima elektronskog poslovanja sledeći odnosi između prodavaca i kupaca:

- *Business to Customer* (B2C),
- *Business to Business* (B2B),
- *Customer to Customer* (C2C).

Kada je reč o zdravstvu, e-zdravlje je novija oblast u sklopu medicinske informatike, koja se odnosi na zdravstvene usluge i informacije koje se isporučuju ili unapređuju putem interneta [2].

3. SISTEMATSKI PREGLED LITERATURE

U ovom radu dat je sistematski pregled literature na temu onlajn zakazivanje pregleda kod veterinara.

3.1. Protokol pregleda literature

Primarni zadatak ovog istraživanja bio je da se predstavi najoptimalnije rešenje, koje će biti prilagođeno korišćenjem sistema elektronske uprave. Za pretraživanje radova korišćena je indeksna baza Gugl akademik (engl. *Google Scholar*, <https://scholar.google.com/>).

3.2. Izvođenje sistematskog pregleda literature

Gugl akademik pruža jednostavan način za široku pretragu naučne literature. Moguće je pretraživati mnoge discipline i izvore: članke, teze, knjige, abstrakte i sudska mišljenja, od akademskih izdavača, profesionalnih društava, onlajn repozitorijuma, univerziteta i drugih veb lokacija [3].

3.3. Prikaz istraživanja

S obzirom na to da pregledom literature nije identifikovan velik broj radova koji se odnosi na temu onlajn zakazivanja kod doktora, doneta je odluka da se napravi sistem, koji će omogućiti da ljudi, koji imaju ljubimce, konkretno psa ili/i mačku, imaju mogućnost da zakažu termin kod željenog doktora.

Sve više korisnika odlučuje se za zakazivanje preko interneta kako bi uštedeli na vremenu i iskoristili ga na efikasniji način. Međutim, iako onlajn zakazivanje ima mnoge prednosti, postoje i mane kojih mnogi ljudi moraju biti svesni. Na primer, iznenađan prekid konekcije ili greška na samom sajtu mogle bi da naprave problem da se termin uopšte ne zakaže, a da korisnik o tome ne bude obavешten. Takođe, zloupotreba ličnih podataka putem interneta čest je problem sa kojim se mnogi ljudi susreću.

Kada je reč o zakazivanju pregleda kod doktora, ovakav način nije toliko uobičajen kod nas, pogotovo kada je reč o zakazivanju termina pregleda kod veterinara. Velikom broju korisnika i dalje je najsigurniji način obavljanje telefonskog poziva ili lični odlazak do veterinarske ordinacije kako bi zakazali termin pregleda kod doktora za svog ljubimca. Mali je broj onih ordinacija koje omogućavaju zakazivanje termina onlajn putem. Stoga, rešenje koje je predstavljeno u ovom radu, trebalo bi da prikaže jednostavan primer kako bi mogao da izgleda jedan sistem, koji korisnicima pruža zakazivanje termina kod doktora za svog ljubimca.

4. OPIS KORIŠĆENIH TEHNOLOGIJA I OKRUŽENJA

4.1. MySQL Workbench

Za razvoj relacionih baza podataka (engl. *Relational Database Management System*, RDBMS) veb aplikacije korišćen je grafički alat MySQL Workbench. To je

najpopularniji open sors (engl. *open source*) sistem za upravljanje bazama podataka na svetu. Ovaj alat pruža modeliranje podataka, kao i razvoj *Structured Query Language* (SQL), i dostupan je na Vindous (engl. *Windows*), Linuks (engl. *Linux*) i Mac ou-es eks (engl. *Mac OS X*) [4].

4.2. InteliJ

Za izradu bekind (engl. *backend*) dela aplikacije korišćen je InteliJ IDEA koji predstavlja inteligentno, integrisano, razvojno okruženje *Integrated Development Environment* (IDE).

Ovaj najkorišćeniji alat radi sa više programskih jezika, kao što je, na primer, Java programski jezik koji je korišćen za potrebe realizacije veb aplikacije onlajn zakazivanja termina kod veterinara. Još neki od podržanih programskih jezika su: Kotlin, Scala, PHP, HTML, CSS, XML ili Groovy [5,6].

4.3. Visual Studio Code

Za izradu frontend (engl. *frontend*) dela aplikacije korišćen je *Visual Studio Code* (VSC) koji predstavlja moćno razvojno okruženje koje olakšava pisanje koda za veb, mobilne aplikacije i klaud (engl. *cloud*) platforme za sve programere koji rade na različitim operativnim sistemima kao što su Vindous, Linuks i Mac ou-es. U okviru editora VSC-a omogućeno je istovremeno uređivanje više datoteka koje stoje jedna pored druge [7].

4.4. MySQL

MySQL predstavlja višekorisnički SQL sistem za upravljanje bazama podataka. Sistem radi kao server, obezbeđujući višekorisnički interfejs (engl. *interface*) za pristup bazi podataka. SQL je jezik koji se koristi za pisanje kverija (engl. *query*) unutar MySQL baze podataka. MYSQL baza podataka je popularna u razvoju veb aplikacija [4].

4.5. Spring framework

Spring frejmwork (engl. *Spring framework*) predstavlja radni okvir koji je pogodan za izgradnju Java aplikacije. Spring može da se koristi za izgradnju bilo koje aplikacije napisane na Java programskom jeziku, na primer za izgradnju veb aplikacije. Spring frejmwork je lagani (engl. *lightweight*) okvir koji se zasniva na principu tehnike *Inversion of Control* (IOC). Spring je kreirao Rod Džonson (engl. *Rod Johnson*) [8].

4.6. Angular

Angular predstavlja radni okvir koji služi kao platforma za razvoj klijentskih aplikacija. Angular je napisan na programskom jeziku TypeScript. Dizajniran je sa ciljem da razdvoji klijentsku stranu aplikacije od serverske strane aplikacije. Ovo omogućava rad na razvoju obe strane paralelno i omogućava ponovnu upotrebu obe strane. Angular okvir razvio je Gugl (engl. *Google*) [9].

5. POSTUPAK IMPLEMENTACIJE SISTEMA

Kako bi se predstavio uvid u sam zadatak, u ovom delu rada predstavljen je realni sistem i prikaz korisničkog interfejsa.

5.1. Realni sistem

Veb aplikacija onlajn zakazivanja veterinarima namenjena je svim korisnicima za onlajn zakazivanje termina kod doktora.

Veb aplikacija omogućava:

- registrovanje i logovanje korisnika,
- pregled zaposlenih doktora i prikaz njihove kratke biografije,
- pregled osnovnih informacija o veterinarskoj ambulanti,
- kreiranje profila za ljubimce,
- brisanje profila za ljubimce,
- zakazivanje termina kod doktora,
- slanje imejla prilikom zakazivanja termina,
- prikaz zakazanih termina, i
- prikaz liste svih korisnika koji imaju napravljen profil na sajtu.

Posetiocima veb aplikacije omogućen je pregled početne stranice, na kojoj se nalazi prikaz doktora koji rade u datoj veterinarskoj ordinaciji, kao i prikaz kratke biografije o svakom od njih. Prilikom registrovanja ili logovanja, korisnik unosi potrebne podatke. Nakon što se ulogovao, korisniku se pojavljuju dodatne mogućnosti u zavisnosti od njegove uloge. Ukoliko je u pitanju običan korisnik, on ima mogućnost prikaza osnovnih informacija o veterinarskoj ordinaciji, mogućnost kreiranja i brisanja profila za ljubimce, kao i opciju zakazivanja termina pregleda. Nakon što je zakazao termin, isti mu ostaje sačuvan i može bilo kada da mu pristupi. Takođe, korisniku stiže potvrda u vidu imejla da je uspešno zakazao termin. Ukoliko je ulogovan admin, dodatna opcija koja mu je dostupna jeste prikaz liste svih korisnika koji su registrovani na sajtu.

5.2. Postupak implementacije rešenja

U okviru ovog dela, biće prikazani neki delovi koda za beken i frontend deo aplikacije.

5.2.1. Bekend

Na Slici 1 prikazana je klasa koja predstavlja model *PetProfile*.

```
PetProfile.java
1 package project.online_vet.model;
2
3 import lombok.Data;
4
5 import javax.persistence.*;
6
7 @Data
8 @Entity
9 @Table(name = "pet_profile")
10 public class PetProfile {
11     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     private Integer petProfileId;
14     @Column()
15     private String petProfileName;
16     @Column()
17     private String petProfileType;
18     @Column()
19     private String petProfileBreed;
20     @Column()
21     private String petProfileGender;
22     @Column()
23     private String petProfileAge;
24     @Column()
25     private Float petProfileKg;
26     @Column()
27     private String petProfilePhoto;
28     @ManyToOne(cascade = CascadeType.MERGE)
29     @JoinColumn(name = "user_id", referencedColumnName = "user_id")
30     private User user;
31 }
```

Slika 1. Model *PetProfile* za beken deo aplikacije

Kako bi Spring znao sa kojim tabelama iz baze podataka bi trebalo da se poveže, bilo je potrebno iznad same klase staviti anotaciju *@Table* nakon koje se navodi tačan naziv tabele iz baze. Takođe, anotacija *@Entity* naznačava da je u pitanju entitet, dok anotacija *@Data* omogućava generisanje get i set metoda u pozadini, što znatno skraćuje kod.

5.2.2. Frontend

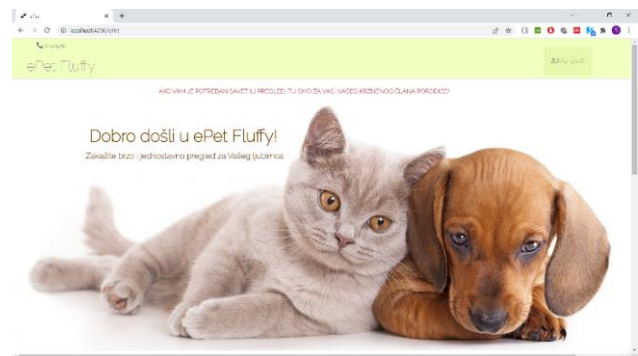
Kreiranje modela vrši se i u ovom delu aplikacije. Primer modela za *User*, prikazan je na Slici 2. Ostali fajlovi koji su kreirani za modele su: *appointment.model.ts*, *auth.model.ts*, *calendar.model.ts*, *pet-profile.model.ts*, *role.model.ts*, *user.model.ts* i *vet.model.ts*.

```
TS user.model.ts U X
vet-project > src > app > model > TS user.model.ts > ...
1 import { Role } from "../model/role.model";
2
3 export class User {
4     public userId: number;
5     public username: string;
6     public userSurname: string;
7     public userUsername: string;
8     public userGender: string;
9     public userEmail: string;
10    public userPassword: string;
11    public roleId: Role;
12
13    constructor(username: string, userSurname: string, userUsername: string, userGender: string,
14                userEmail: string, userPassword: string, roleId: Role) {
15        this.username = username;
16        this.userSurname = userSurname;
17        this.userUsername = userUsername;
18        this.userGender = userGender;
19        this.userEmail = userEmail;
20        this.userPassword = userPassword;
21        this.roleId = roleId;
22    }
23
24 }
```

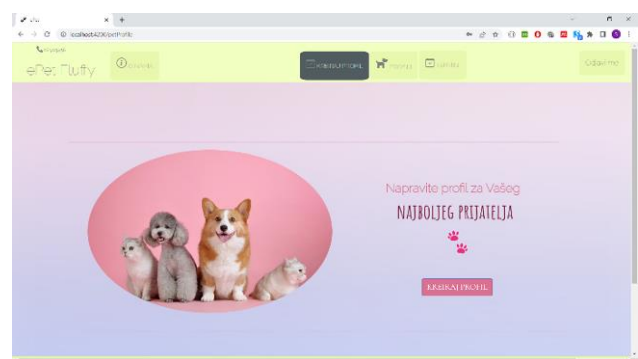
Slika 2. Model *User* za frontend deo aplikacije

5.3. Prikaz funkcionalnosti zadatka

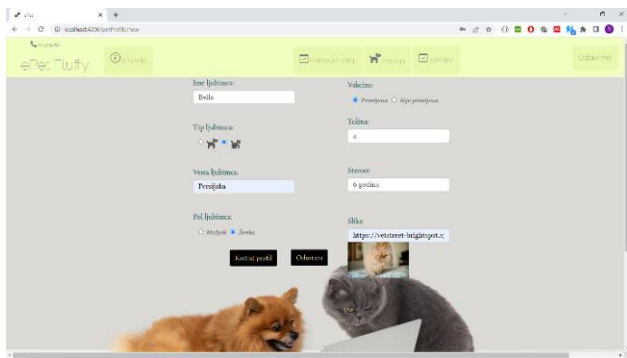
Na narednim slikama biće predstavljen sam izgled veb aplikacije sa nekim od implementiranih funkcija (Slika 3, Slika 4, Slika 5, Slika 6).



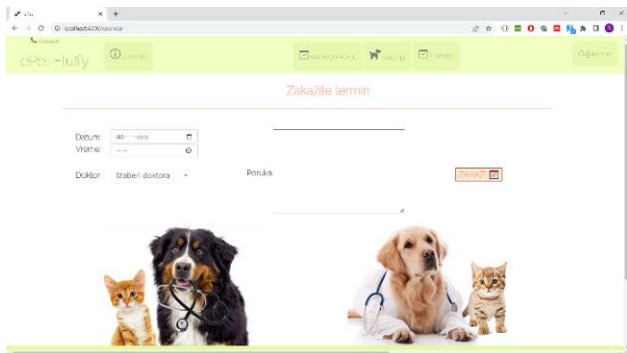
Slika 3. Izgled početne stranice



Slika 4. Kartica „Kreiraj profil“



Slika 5. Forma za kreiranje profila za ljubimca



Slika 6. Forma za zakazivanje termina

6. ZAKLJUČAK

S obzirom na to da primeri veb aplikacija, koje se odnose na temu onlajn zakazivanja pregleda kod veterinara, nisu, u velikoj meri, istraživani i implementirani, primarni zadatak ovog rada bio je da se predstavi optimalno i *user-friendly* rešenje, koje se odnosi na datu temu.

Korišćenjem najpopularnijih tehnologija, kao što su Spring i Angular, implementirana je data veb aplikacija za onlajn zakazivanje pregleda kod veterinara. U okviru ovog rada obuhvaćen je opis, kao i razvoj samog projekta. Na samom početku, predstavljen je uvod u datu temu i u sisteme elektronske uprave, a zatim sistematski pregled literature. Naredni deo rada odnosi se na opis korišćenih tehnologija i okruženja u kojem je implementiran projektni zadatak, dok je na samom kraju dokumenta predstavljen opis i postupak implementacije rešenja.

Kako bi aplikacija bila dovedena do savršenstva, poželjno bi bilo implementirati određena unapređenja u okviru iste. Unapređenje koje bi moglo biti od značaja jeste uvođenje mogućnosti ostavljanja komentara i ocena na sajtu, dok bi neke manje modifikacije mogle da se odnose na izmene u samom interfejsu projekta. S obzirom na to da je sigurnost korisnika na prvom mestu, bilo bi poželjno da se prilikom registracije uradi dodatna autentifikacija, pomoću, na primer, imejla ili broja telefona.

Nakon završene implementacije veb aplikacije i napisanog master rada, rad na ovakvom projektu bio je, većinski, primenjivanje znanja koje je stečeno u dosadašnjem fakultetskom obrazovanju, kao i sticanje novog znanja prikupljanjem adekvatnih informacija iz literature i sa interneta.

7. LITERATURA

- [1] "Domen," [Online]. Available: <https://www.domen.rs/sr-latn/zasto-je-elektronsko-poslovanje-vazno-za-vasu-organizaciju>.
- [2] C. Ilioudis and G. Pangalos, "A Framework for an Institutional High Level Security Policy for the Processing of Medical Data and their Transmission through the Internet," 2001.
- [3] "Google Scholar," [Online]. Available: <https://scholar.google.com/intl/en/scholar/about.html>.
- [4] "Šta je MySQL?," [Online]. Available: <https://cubes.edu.rs/sr/37/obuke-i-kursevi/sta-je-mysql>.
- [5] H. O. Assumpção, Getting started with IntelliJ IDEA, Packt Publishing, 2013.
- [6] J. Krochmalski, IntelliJ IDEA Essentials, Packt Publishing, 2014.
- [7] A. D. Sole, Visual Studio Code Distilled Evolved Code Editing for Windows, macOS, and Linux-Apress, 2019.
- [8] R. B. Craig Walls, (In Action series) Spring in Action, Manning, 2005.
- [9] "What is AngularJS?," [Online]. Available: <https://docs.angularjs.org/guide/introduction>

Kratka biografija:



Sandra Martinović rođena je u Novom Sadu 1997. godine. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Inženjerstvo informacionih sistema odbranila je 2020. godine. Master akademske studije upisala je iste godine iz iste oblasti.
Kontakt: sandramartinovic.sm@gmail.com