Web aplikacija za online rezervaciju termina kod zubara

Kovačević, Ivana

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:351076

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Diplomski sveučilišni studij Računarstvo

WEB APLIKACIJA ZA ONLINE REZERVACIJU TERMINA KOD ZUBARA

Diplomski rad

Ivana Kovačević

Osijek, 2020.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 07.12.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Ivana Kovačević
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-993R, 28.09.2020.
OIB studenta:	20640993979
Mentor:	Izv. prof. dr. sc. Krešimir Nenadić
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Mirko Köhler
Član Povjerenstva 1:	Izv. prof. dr. sc. Krešimir Nenadić
Član Povjerenstva 2:	Izv. prof. dr. sc. Ivan Aleksi
Naslov diplomskog rada:	Web aplikacija za online rezervaciju termina kod zubara
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak diplomskog rada:	Opisati proces rezervacije tremina kod zubara. Kreirati bazu podataka u koju će se zapisivati svi podaci web aplikacije. Navesti funkcionalnosti koje bi trebala imati web aplikacija za rezervaciju termina kod zubara. Opisati sve tehnologije koje će se koristit prilikom izrade web aplikacije. Izraditi web aplikaciju pomoću koje korisnici mogu rezervirati termin kod zubara i opisati postupak izrade. Predvidjeti najmanje dva korisnička profila u web aplikaciji - zaposlenik i klijent. Za izradu web aplikacije koristiti MVC i objektno orijentirani PHP. Usporediti izrađenu web aplikaciju sa sličnim rješenjima. (tema rezervirana: Ivana Kovačević)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene mentora:	07.12.2020.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 19.12.2020.

Ime i prezime studenta:	Ivana Kovačević
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-993R, 28.09.2020.
Turnitin podudaranje [%]:	23%

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za online rezervaciju termina kod zubara**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Krešimir Nenadić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD 1
1.1. Zadatak rada 1
2. PREGLED POSTOJEĆIH RJEŠENJA 2
2.1. Aspen Dental 2
2.2. Schedule an Appointment 3
2.3. Setmore Calendar 5
2.4. OpenCare 5
2.5. Dental Clinic 6
3. TEHNOLOGIJE KORIŠTENE ZA RAZVOJ WEB APLIKACIJE 8
3.1. Internet (Web) aplikacije 8
3.2. Arhitektura i princip rada web aplikacija 8
3.3. Internetski preglednici 10
3.4. Bootstrap 11
3.5. PHP 13
3.6. MySQL 14
3.7. MVC 14
4. STRUKTURA APLIKACIJE 18
4.1. Struktura baze podataka 18
4.2. Struktura web aplikacije 19
4.2.1. Razvoj sustava prijave u aplikaciju 20
4.2.2. Uloga liječnika 21
4.2.3. Uloga pacijenta 25
5. NAČIN KORIŠTENJA APLIKACIJE 27
6. ZAKLJUČAK 33
LITERATURA 34
SAŽETAK 35
ABSTRACT 36
ŽIVOTOPIS 37
PRILOZI 38

1. UVOD

U današnje vrijeme, veliki dio naše svakodnevice odvija se na internetu te samim time internet ima veliku ulogu u životu svakog pojedinca. Internet je postao svakodnevni oslonac i izvor odgovora na većinu pitanja koja korisnik pretražuje. Kako se velik dio naših života odvija na internetu, sve više i više nailazimo na neka rješenja koja nam olakšavaju naše svakodnevne obaveze, kao na primjer dogovaranje termina kod zubara. Umjesto dosadašnjih načina dogovaranja na principu odlaska u samu ordinaciju ili telefonskog dogovora, dobro će nam doći web aplikacija koja omogućuje dogovor unutar par klikova. Ovakva web aplikacije pacijentu omogućuje samostalni odabir datuma i vremena pregleda te olakšava komunikaciju između zubara i pacijenata te se iz navedenih razloga ovaj diplomski rad bavi upravo ovom tematikom: izgradnja web aplikacije za online rezervaciju termina kod zubara.

Drugo poglavlje ovoga rada sadrži pregled postojećih rješenja slične tematike. Za svaki primjer rješenja slične tematike opisan je postupak korištenja osnovnih dijelova aplikacije. U poglavlju tri ukratko su opisane i na primjeru pokazane tehnologije koje su korištene prilikom izrade ovog diplomskog rada. Četvrto poglavlje opisuje strukturu same web aplikacije. Prvo se opisuje struktura baze podataka, a nakon toga i struktura same aplikacije zajedno sa svim funkcionalnostima koje aplikacija nudi. U petom poglavlju na primjeru je prikazan način upotrebe izrađene web aplikacije. U zaključku je iznesen kratak osvrt na cjelokupan proces izrade web aplikacije i samog diplomskog rada.

1.1. Zadatak rada

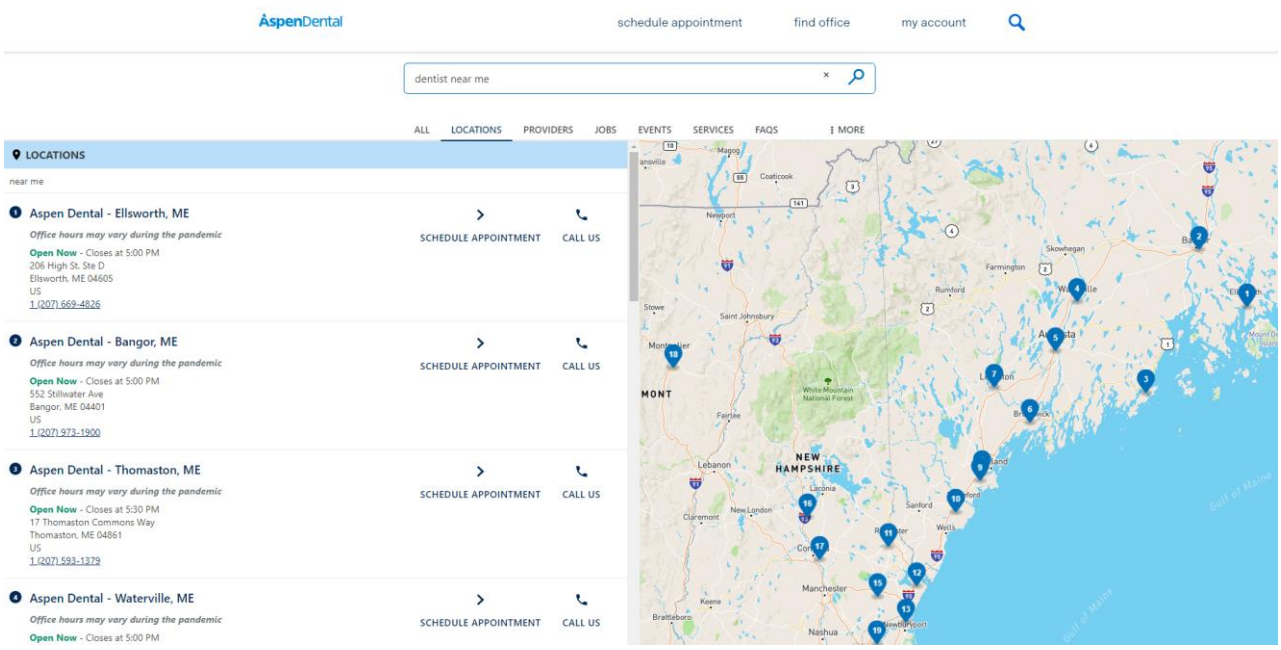
Opisati proces rezervacije termina kod zubara. Kreirati bazu podataka u koju će se zapisivati svi podaci web aplikacije. Navesti funkcionalnosti koje bi trebala imati web aplikacija za rezervaciju termina kod zubara. Opisati sve tehnologije koje će se koristit prilikom izrade web aplikacije. Izraditi web aplikaciju pomoću koje korisnici mogu rezervirati termin kod zubara i opisati postupak izrade. Predvidjeti najmanje dva korisnička profila u web aplikaciji - zaposlenik i klijent. Za izradu web aplikacije koristiti MVC i objektno orijentirani PHP. Usporediti izrađenu web aplikaciju sa sličnim rješenjima.

2. PREGLED POSTOJEĆIH RJEŠENJA

Danas na tržištu postoje razna rješenja vezana za zakazivanje termina kod zubara. Većina aplikacija predstavljena je kao neki oblik obrasca koje korisnik mora popunjavati. Aplikacije sadrže polja gdje korisnik unosi ime, prezime, telefonski broj i adresu elektroničke pošte. Neke aplikacije također korisnicima nude odabir razloga zakazivanja termina, kao na primjer redovan pregled, bol ili puknuće zuba. Kako je ovakav način zakazivanja termina pregleda kod zubara popularniji izvan naših granica, na obrascu nailazimo na pitanja o zdravstvenom osiguranju i struci. U nastavku su predstavljene neke aplikacije koje nude slična rješenja.

2.1. Aspen Dental

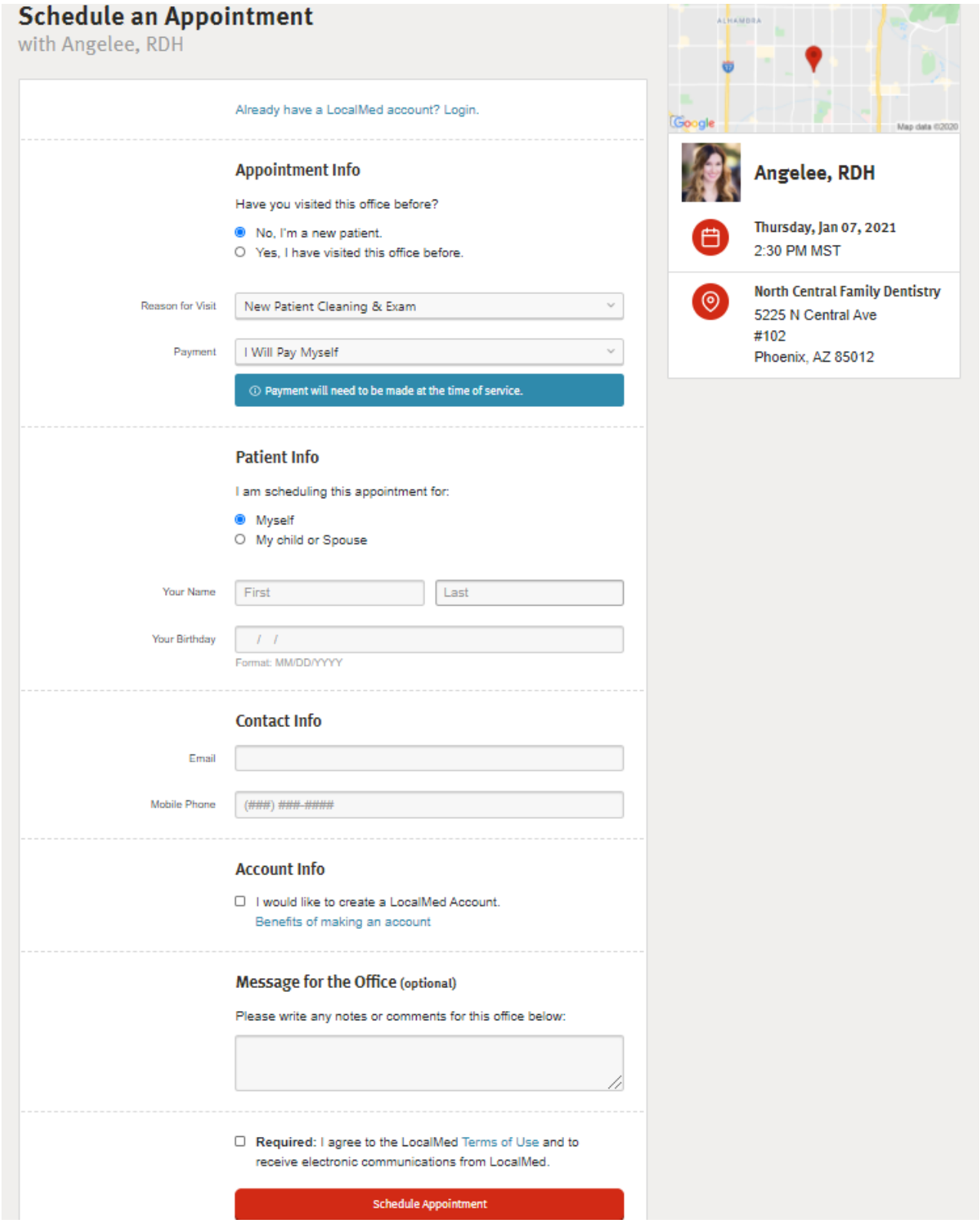
Aspen Dental je web aplikacija koja korisnicima omogućuje zakazivanje termina u samo nekoliko klikova. Za zakazivanje termina korištenjem ove aplikacije nije potrebna registracija. Na početku korištenja aplikacije korisnik mora unijeti osobne podatke (ime, prezime, adresu elektroničke pošte, poštanski broj, telefonski broj) nakon čega odabire razlog pregleda. Na temelju unesenog poštanskog broja aplikacija korisniku daje nekoliko mogućnosti koje može odabrati. Nakon odabira željenog zubara korisnik odabire željeno vrijeme i datum termina te zakazuje termin. Na slici 2.1. prikazan je izbor zubara na temelju unesenih podataka. Korištenje navedene aplikacije omogućeno je na američkom tržištu.



Sl. 2.1. Aspen Dental

2.2. Schedule an Appointment

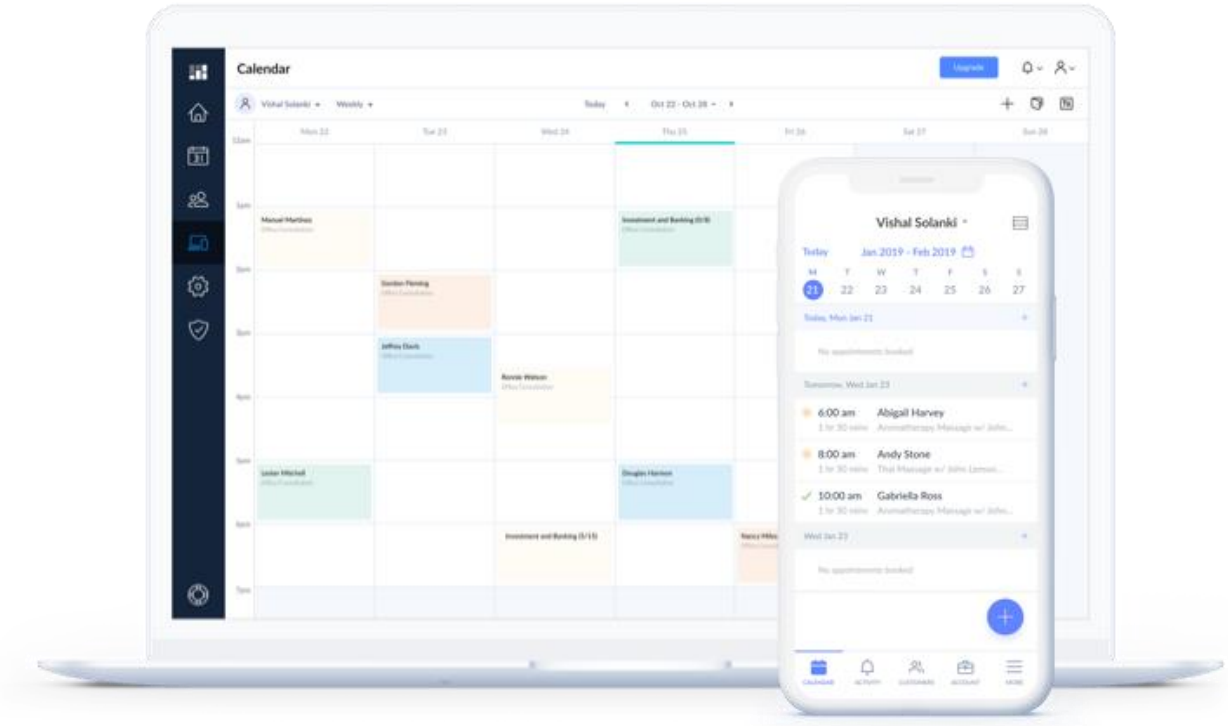
Schedule an appointment aplikacija je jednostavna za korištenje. Korisnik prvo unosi poštanski broj na temelju čega mu aplikacija daje prikaz najbližih zubara s prikazom slobodnih termina. Nakon što korisnik odabere zubara i termin, aplikacija traži popunjavanje nekoliko polja za unos osobnih podataka, prikazanih na slici 2.2. Nakon što korisnik unese potrebne podatke i klikne na gumb za zakazivanje termina potrebno je potvrditi unesenu adresu elektroničke pošte. Kada je adresa elektroničke pošte potvrđena, korisnik zakazuje svoj pregled. Potvrda zakazanog termina korisniku dolazi u poruci na unesenu adresu elektroničke pošte.



Sl. 2.2. Schedule an Appointment

2.3. Setmore Calendar

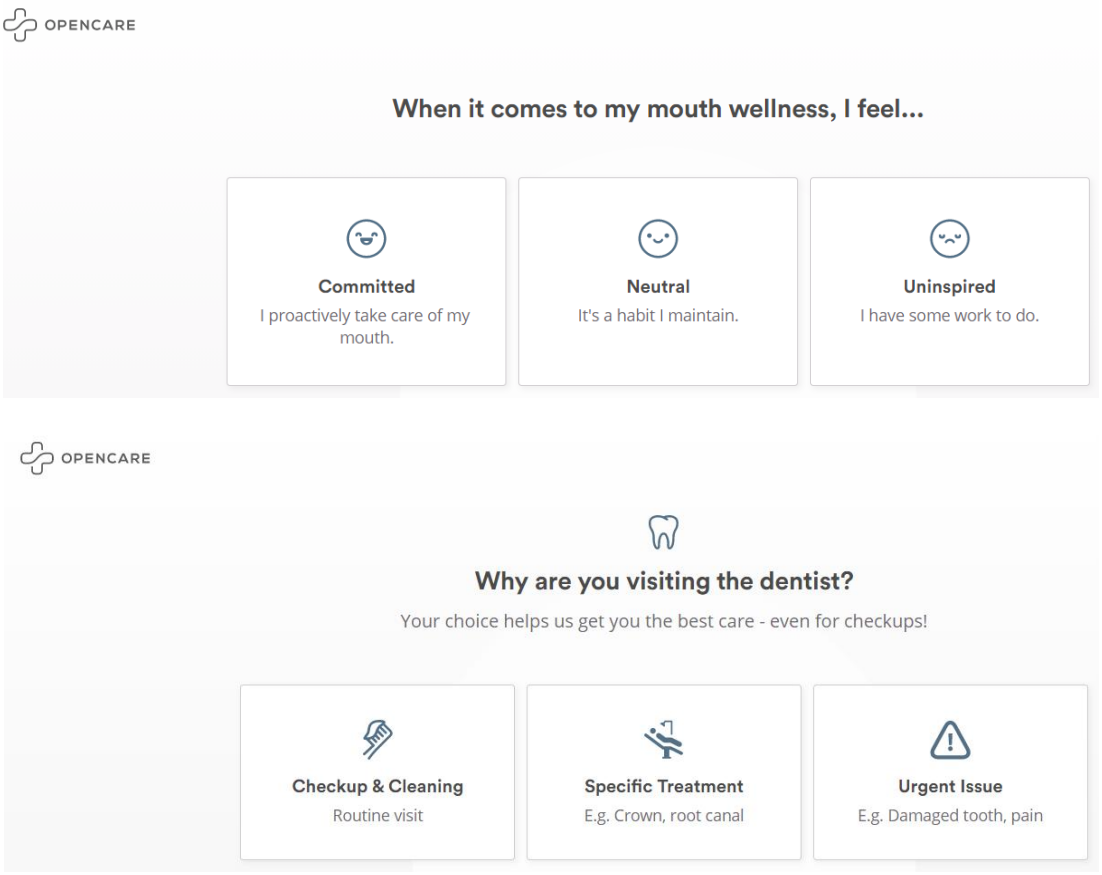
Aplikacija Setmore Calendar više je namijenjena zubarima i njihovim zaposlenicima nego samim pacijentima. Putem ove aplikacije zubari i njihovi zaposlenici imaju uvid u svoj tjedni i mjesečni raspored te mogu pratiti plaćanja pacijenata. Izgled tjednog rasporeda aplikacije Setmore Calendar prikazan je na slici 2.3.



Sl. 2.3. Setmore Calendar [1]

2.4. OpenCare

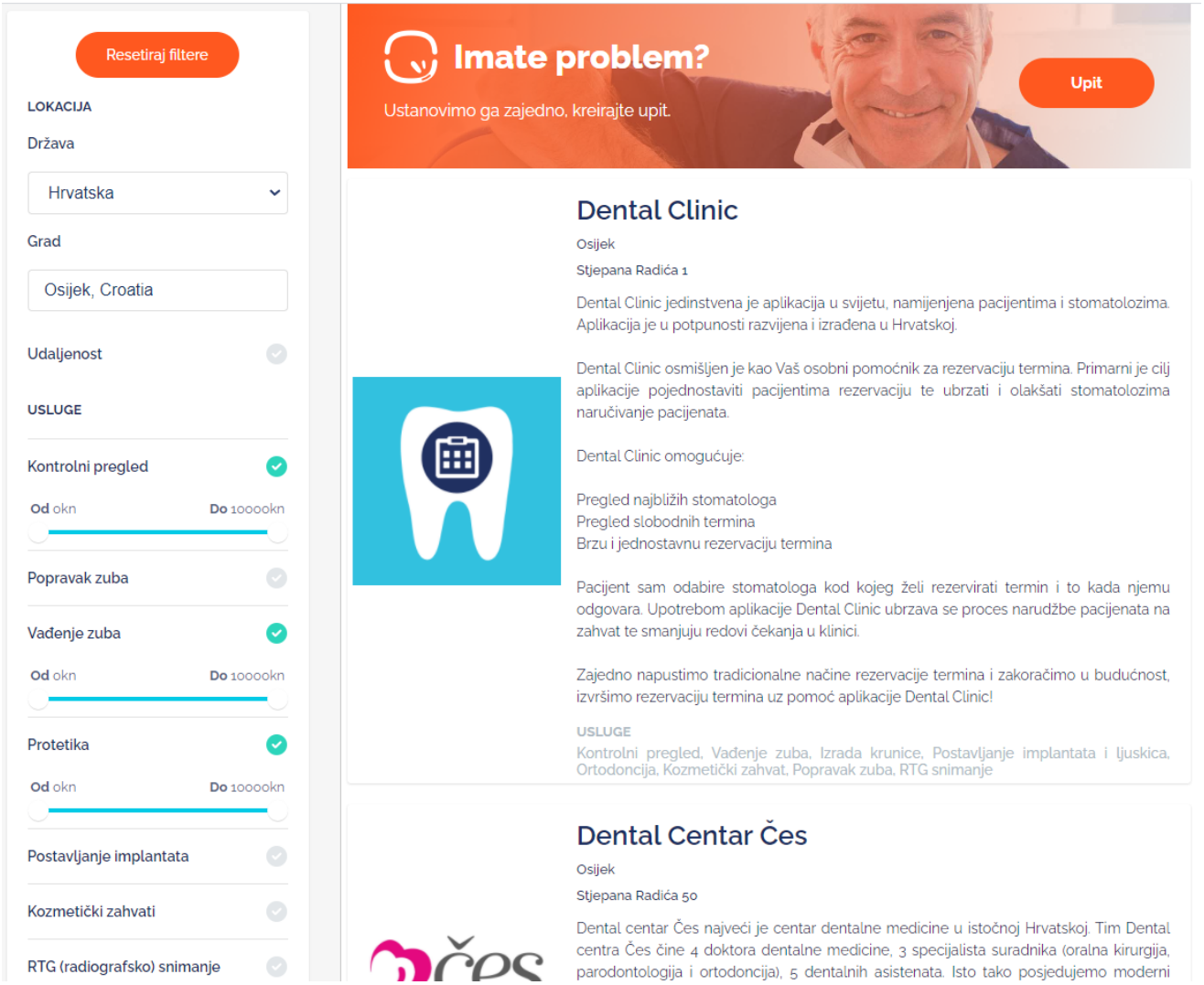
Aplikacija OpenCare pomaže korisnicima pronaći zubara koji najviše odgovara njihovim potrebama. Nakon što je unio svoju adresu, korisnik nastavlja rješavati kviz na temelju čijih odgovora aplikacija pronalazi najbližeg zubara koji zadovoljava potrebe pacijenta. Na slici 2.4. u nastavku prikazana su neka od pitanja na koja pacijent treba odgovoriti.



Sl. 2.4. OpenCare

2.5. Dental Clinic

Dental Clinic aplikacija je aplikacija namijenjena pacijentima i zubarima. Ovom aplikacijom pacijentima je pojednostavljena rezervacija termina, a zubarima ubrzano i olakšano naručivanje pacijenata. Aplikacija Dental Clinic korisnicima omogućuje pregled najbližih zubara, pregled slobodnih termina te brzu i jednostavnu rezervaciju termina. Korisnik sam odabire zubara kod kojeg želi zakazati termin i to kada njemu odgovara. Korištenjem filtera, aplikacija korisniku prikazuje zubare koji na unesenoj lokaciji obavljaju željenu uslugu, što je i prikazano na slici 2.5. Dental Clinic aplikacija je u potpunosti razvijena i izrađena u Hrvatskoj.



Sl. 2.5. Dental Clinic

3. TEHNOLOGIJE KORIŠTENE ZA RAZVOJ WEB APLIKACIJE

Razvoj aplikacije (engl. *Application development*) predstavlja proces izgradnje aplikacije. Pojam razvoja aplikacije obuhvaća sve faze izgradnje, od definiranja problema, odabira alata i tehnologija do same izrade aplikacije. Postoje različiti alati, tehnologije i načini izrade aplikacija s kojima se susrećemo prilikom izgradnje aplikacija.

3.1. Internet (Web) aplikacije

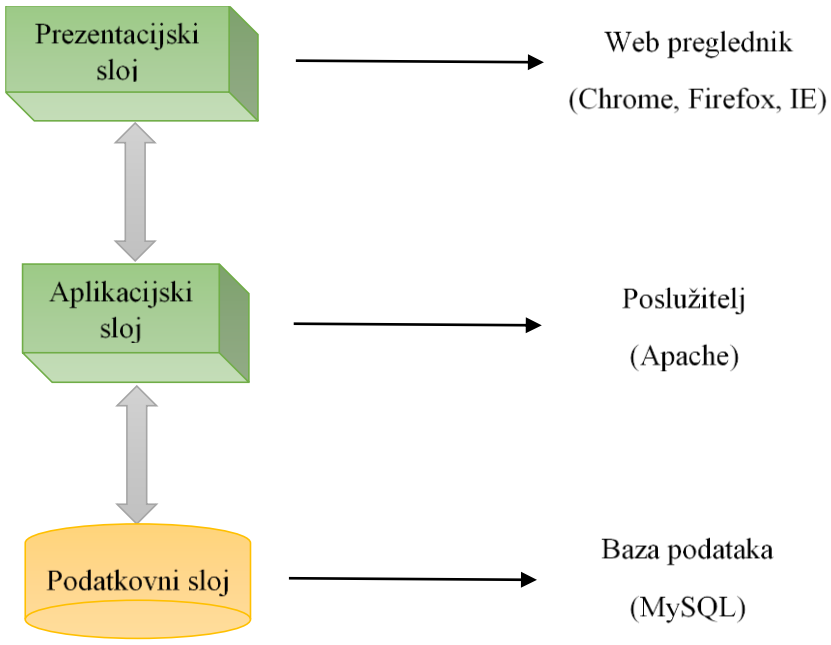
Prema definiciji web aplikacija je interaktivni računalni program izrađen s određenim web tehnologijama (HTML, CSS, JS) koji pohranjuje i manipulira podacima (CRUD) [2]. Web aplikaciju koristi tim ili pojedinačni korisnik za izvršavanje zadataka putem interneta. CRUD označava kraticu koja se nalazi u središtu razvoja web aplikacija. Dolazi od engleskih riječi za stvaranje (engl. *Create*), čitanje (engl. *Read*), ažuriranje (engl. *Update*) i brisanje (engl. *Delete*). Za pristup web aplikaciji potreban je internet preglednik. Web aplikacije često sadrže mehanizam za registraciju i prijavu korisnika.

Svoj nagli rast i sve veću popularizaciju za razvoj web aplikacije zahvaljuju činjenici da je internet danas dostupan na bilo kojem mjestu u bilo koje vrijeme, što istovremeno omogućuje neprestan pristup različitim web aplikacijama. U novije vrijeme web aplikacijama se pristupa putem mobilnih uređaja koji imaju internet preglednik. Za održavanje i ažuriranje web aplikacija nije potrebna posebna instalacija programske podrške, što je i jedna od glavnih prednosti web aplikacija. Za prijavu i korištenje web aplikacije u većini slučajeva se koristi forma gdje se od korisnik traži unos korisničkog imena, adrese elektroničke pošte i zaporke.

3.2. Arhitektura i princip rada web aplikacija

Osnovni princip rada svake web aplikacije polazi od načina komunikacije između klijenta i poslužitelja (engl. *server*), korištenjem internetske mreže. Glavni zadatak svake internet aplikacije je prijenos poruka i informacija, korištenjem nekih od komunikacijskih

protokola, od web poslužitelja preko korisničkog sučelja do krajnjeg korisnika. Arhitektura internet aplikacije prikazana je na slici 3.1.



Sl. 3.1. Arhitektura web aplikacija

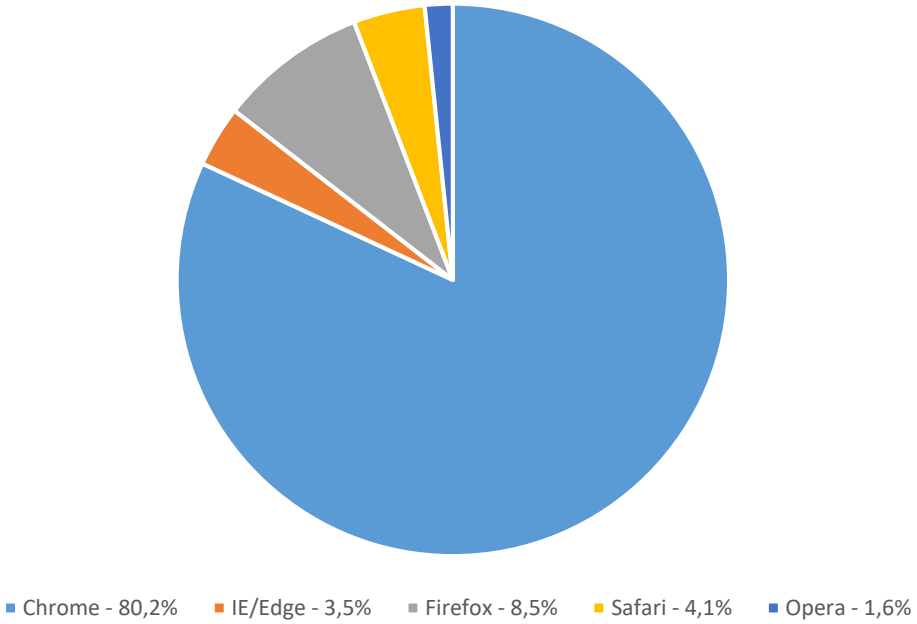
Web aplikacije podijeljene su u tri glavna sloja. Ovakva arhitektura još se naziva i troslojnom arhitekturom (engl. *Three-tiered architecture*). Prvi sloj je prezentacijski sloj (engl. *Presentation tier*) koji predstavlja prikaz informacija korisniku putem preglednika. Za prezentacijski sloj koristi se jedan od internet preglednika (MS Internet Explorer, Google Chrome i drugi). Drugi sloj je aplikacijski sloj (engl. *Application tier* ili *Middleware tier* ili *Busines Logic*). Ovaj sloj upravlja aktivnostima koje aplikacija treba izvršavati, odnosno bavi se obradom ili funkcioniranjem same aplikacije. Kako bi aplikacijski sloj obavljao svoju funkciju koriste se dinamičke web tehnologije (ASP, .NET, PHP i druge). Treći, odnosno posljednji sloj je podatkovni sloj (engl. *Data tier*) koji je zadužen za pohranu korisničkih podataka u bazu podataka te za prikaz istih na poslužitelju. Za normalnu funkciju podatkovnog sloja koriste se sustavi za upravljanje bazama podataka (SQL, MySQL). Kako bi aplikacija obavljala svoju zadanu funkcionalnost, prilikom razvoja aplikacije potrebno je pridržavati se standardiziranih internet protokola koji omogućuju međusobnu komunikaciju prethodno navedenih slojeva. [3]

Za prikaz sadržaja internetskim preglednikom, web aplikacija dinamički generira niz HTML dokumenata. Ukoliko se u korisničko sučelje dodaju dinamički

elementi, koriste se skriptni jezici koji se izvode na klijentskoj strani. Svaka pojedina stranica web aplikacije korisniku se prikazuje kao pojedini statični element. Unutar aplikacije te se stranice mogu interaktivno izmjenjivati. Za unos podataka koji se mogu spremati u bazu podataka i koje će aplikacija koristiti u budućnosti koriste se web forme koje su uključene u stranice.

3.3. Internetski preglednici

Kako bi uređaj mogao pristupiti sadržaju na internetu trebao bi imati instaliran internetski preglednik (engl. *Web browser*). Za komunikaciju između poslužitelja i klijenta najčešće se koristi HTTP protokol (engl. *HyperText Transfer Protocol*). HTTP protokol zapravo predstavlja skup pravila za prenošenje internet stranice od poslužitelja do korisnika, također korisnicima omogućuje prezentaciju i objavljivanje HTML dokumenta, odnosno internet stranica. Osnovna zadaća ovog protokola je omogućiti prenošenje informacija takvo da bi se te informacije na internetu svim korisnicima mogle prikazati na isti način, neovisno o uređaju i pregledniku koje korisnik koristi. Slika 3.2. prikazuje najpopularnije internet preglednike i postotke korištenja u svibnju 2020. godine [4].



Sl. 3.2. Najpopularniji internet preglednici u svibnju 2020. godine

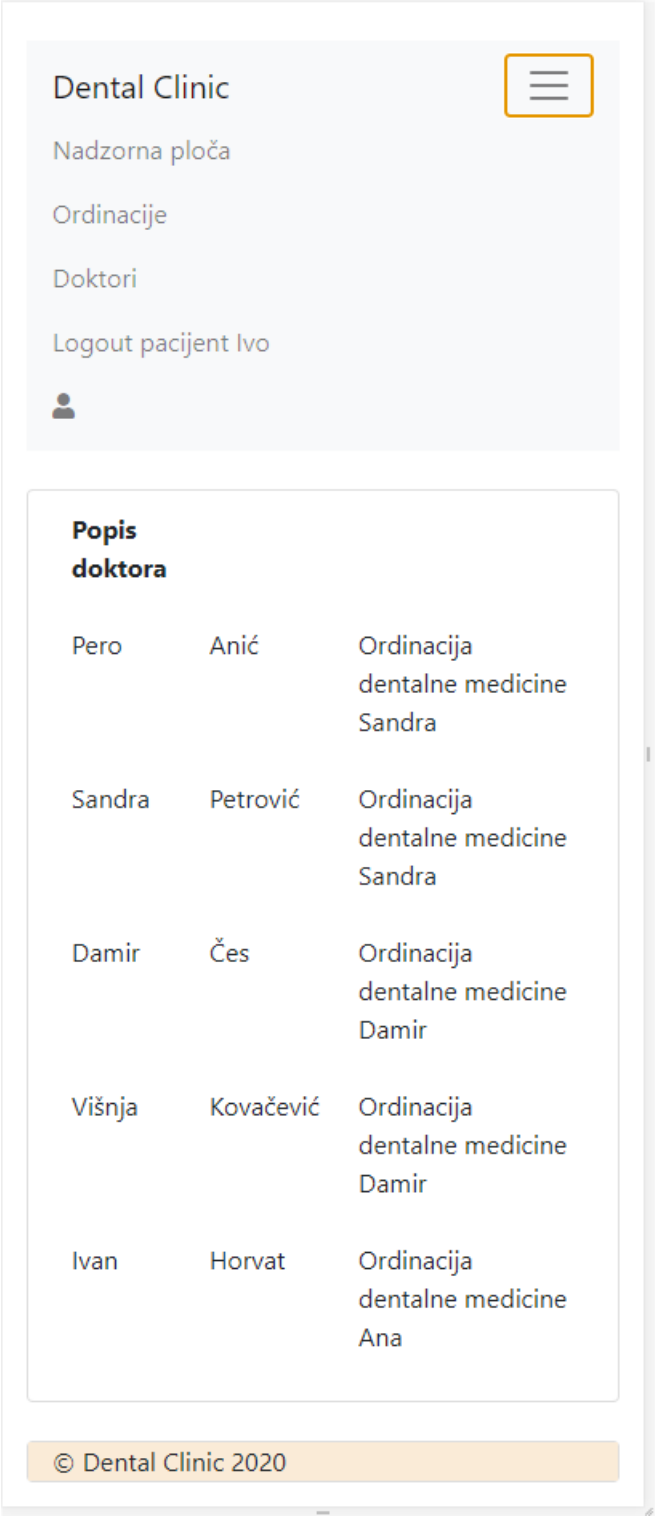
Važno je naglasiti da internet preglednici ne sudjeluju u izravnom generiranju i kreiranju sadržaja web aplikacije krajnjem korisniku. Iako internet preglednici služe samo za prikaz sadržaja krajnjem korisniku, vrlo su važan dio cjelokupne komunikacije između poslužitelja i klijenta koji komuniciraju putem internet protokola.

3.4. Bootstrap

Bootstrap je jedan od najčešće korištenih okvira za izradu responzivnih web stranica. Sadrži HTML i CSS temeljne predloške za zajedničke komponente korisničkog sučelja, kao što su gumbovi, padajući izbornici, forme, tablice, navigacije, upozorenja i slično.

Može se koristiti u sklopu bilo koje tehnologije na strani poslužitelja i na bilo kojoj platformi. Neke od njegovih prednosti su podržavanje odgovarajućeg dizajna i omogućavanje bržeg i lakšeg stvaranja odgovarajuće web aplikacije. Jedna od bitnijih značajki je responzivni dizajn koji omogućuje aplikaciji automatsko prilagođavanje različitim veličinama zaslona na kojima se web aplikacija koristi. [5]

Korištenjem Bootstrap okvira aplikacija je dobila svoj responzivni izgled što je vidljivo slikom 3.3. Prema navedenoj slici vidljivo je da je izgled prilagođen manjim dimenzijama ekrana, izbornik je poprimio novi izgled, a podaci su i dalje uredno i vidljivo prikazani. [6] Podaci su na stranici pregledno prikazani korištenjem Bootstrap tablice kao što je prikazano na slici 3.4. u nastavku. [7]



Sl. 3.3. Prikaz aplikacije na manjem zaslonu

```
Linija  Kod
 1:     <table class="table table-borderless">
 2:     <thead>
 3:     <tr>
 4:     <th scope="col">Popis doktora</th>
 5:     </tr>
 6:     </thead>
 8:     <tbody>
 9:     <?php foreach($entiteti as $entitet): ?>
10:     <tr>
11:     <td><?=$entitet->ime?></td>
12:     <td><?=$entitet->prezime?></td>
13:     <td><?=$entitet->iban?></td>
14:     <td><?=$entitet->naziv?></td>
15:     <td>
16:     </td>
17:     </tr>
18:     <?php endforeach; ?>
19:     </tbody>
20:     </table>
```

Sl. 3.4. Prikaz korištenja Bootstrap tablice

3.5. PHP

PHP (engl. *Open Source Server Side*) je skriptni programski jezik otvorenog koda koji se koristi za dinamičko generiranje HTML koda [8]. Pomoću PHP programskog jezika može se kreirati HTML stranica na serveru, prije nego što je ona poslana klijentu zajedno sa popunjenim dinamičkim sadržajem. Ovakvim načinom generiranja sadržaja klijent ima pristup samo čistom HTML kodu. PHP danas predstavlja jednu od najnaprednijih i najiskorištenijih *server side* skriptnih tehnologija. Statistički podaci govore kako je 90% postojećih web stranica pisano upravo PHP-om.

PHP ne ovisi niti o internetskom pregledniku niti o operacijskom sustavu korisnika web aplikacije. Svaka linija koda u PHP programskom jeziku završava znakom „ ; “ čime se odvajaju različite naredbe. Osnovna naredba za ispis tekst u PHP-u je *echo*. Za pohranjivanje različitih vrijednosti, brojeva i ostalih podatka koristimo varijable. Kako bismo iste varijable mogli koristiti više puta u istoj skripti, u PHP-u sve varijable trebaju započinjati znakom $. Kako bismo razdvojili PHP kod od HTML koda, koristimo znakove

<?php ?>. Na taj način govorimo serveru da se unutar ovih znakova nalazi PHP kod koji se izvršava prije HTML koda.

Upotreba PHP-a zahtijeva zadovoljavanje ovih pretpostavki:

- Instaliran Apache server na Windows ili Unix/Linux poslužitelju
- Instaliran PHP jezik na Windows ili Unix/Linux poslužitelju
- Instaliranu MySQL bazu na Windows ili Unix/Linux poslužitelju

3.6. MySQL

MySQL predstavlja sustav za upravljanje relacijskim bazama podataka. Najviše se koristi u suradnji s PHP programskim jezikom jer imaju dobru međusobnu integraciju i podržanost. PHP ne može samostalno pristupiti bazi podataka i informacijama o korisnicima te mu je za to potreban MySQL. Pojam upravljanje podacima odnosi se na skladištenje, pregled i ažuriranje podataka putem programskog jezika SQL (engl. *Structured Query Language*), dok pojam relacije predstavlja organizaciju podataka i njihovu međusobnu povezanost.

Postoje razni načini dizajniranja baze podataka u MySQL-u, a neki od najčešćih su kreiranje baze podataka pomoću posebnog alata phpMyAdmin ili kreiranje baze podataka direktno programskim kodom SQL jezika i PHP-a. phpMyAdmin aplikacija je popularna aplikacija otvorenog koda koja je besplatna za korištenje svim korisnicima.

3.7. MVC

MVC označava kraticu za engl. *Model-View-Controller*, a predstavlja arhitekturu koja se koristi za razvoj korisničkih sučelja web aplikacija. MVC arhitektura svoju popularnost je stekla sve većom popularnošću i korištenjem web aplikacija. Arhitektura je usklađena s primarnim principima funkcioniranja web aplikacija te posjeduje mogućnost razdvajanja podataka, korištenja programske logike te korisničko sučelje što omogućuje lakše održavanje aplikacije. Navedene karakteristike ističu MVC arhitekturu kao osnovnu arhitekturu za izradu web aplikacija.

MVC arhitektura sastoji se od tri osnovna dijela odnosno komponente:

1) Model – dinamička struktura podataka koja izravno upravlja podacima, logikom i pravilima aplikacije
2) Pogled (engl. *View*) – web stranica koja sadrži podatke modela
3) Kontroler (engl. *Controller*) – izvršava operacije na modelu, prima korisničke zahtjeve, šalje rezultat

Korištenje MVC arhitekture opisano je na primjeru pacijenta. Datoteka koja predstavlja model MVC arhitekture sadrži klasu Pacijent koja sadrži sve funkcije potrebne za rad sa bazom podataka (čitanje, unos i promjena podataka) napisane u PHP programskom jeziku sa kombinacijom SQL naredbi, što je prikazano slikom 3.5. u nastavku.

Linija ***Kod***

```
1:      <?php
2:      class Pacijent
3:      {
4           public static function ucitajSve()
5           {
6               $veza = DB::getInstanca();
7               $izraz = $veza->prepare("SELECT * FROM pacijent;");
8               $izraz->execute();
9               return $izraz->fetchAll();
10          }
```

Sl. 3.5. Model sa funkcijom čitanja podataka iz baze podataka

Nadalje, datoteka koja predstavlja pogled MVC arhitekture sadrži HTML kôd za prikaz web stranice. HTML kôd je prikazan slikom 3.6. u nastavku.

```
Linija  Kod
 1:     <table class="table table-borderless">
 2:     <thead>
 3:     <tr>
 4:     <th scope="col">Popis pacijenata</th>
 5:     </tr>
 6:     </thead>
 8:     <tbody>
 9:     <?php foreach($pacijenti as $pacijent): ?>
10:     <tr>
11:     <td><?=$entitet->ime?></td>
12:     <td><?=$entitet->prezime?></td>
13:     <td><?=$entitet->mbo?></td>
14:     <td>
15:     </td>
16:     </tr>
17:     <?php endforeach; ?>
18:     </tbody>
19:     </table>
```

Sl. 3.6. Pogled

Datoteka koja predstavlja kontroler MVC arhitekture sadrži klasu PacijentController sa funkcijama koje su potrebne za izvršavanje svih naredbi koje su zadane na modelu. Primjer kontroler datoteke vidljiv je na slici 3.7.

Linija ***Kod***

```
1:      <?php
2:      class PacijentController extends AutorizacijaController
3:      {
4:      public function novi()
5:          {
6:              if($_SERVER['REQUEST_METHOD']==='GET')
7:              {
8:                  $pacijent = new stdClass();
9:                  $pacijent->ime='';
10:                 $pacijent->prezime='';
11:                 $pacijent->mbo='';
                    $this-
12:     >novoView('Unesite tražene podatke!', $pacijent);
13:                 return;
14:             }
15:             $pacijent = (object)$_POST;
16:             Pacijent::dodajNovi($_POST);
                if(!$this-
17:     >kontrolaIme($pacijent,'novoView')){return;}
18:             $this->index();
19:     }
```

Sl. 3.5. Kontroler

4. STRUKTURA APLIKACIJE

U narednom poglavlju opisani su struktura baze podataka i struktura same aplikacije. Prikazane u osnove naredbe korištene prilikom izrade baze podataka kao i osnovne funkcionalnosti koje aplikacija pruža korisnicima.

4.1. Struktura baze podataka

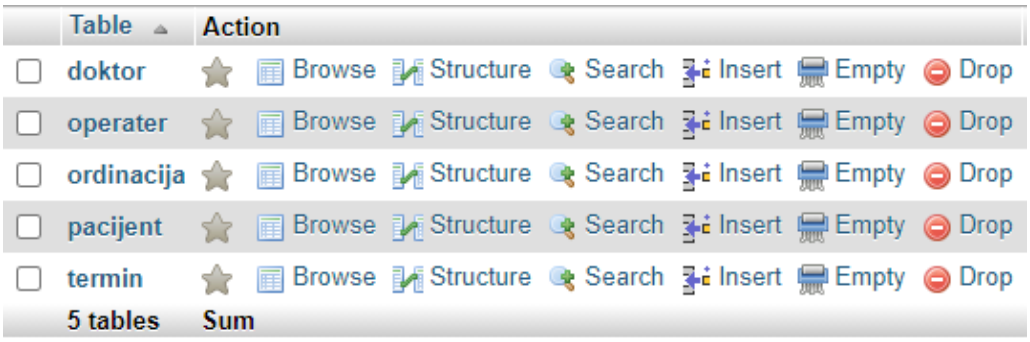
Za potrebe izrade web aplikacije izrađena je baza podataka korištenjem SQL programskog jezika. Prilikom izrade baze podataka korištene su osnovne SQL naredbe kao na primjer *create*, *update*, *alter table* za manipulaciju tablicama. Korištenje naredbe *create table* prikazano je slikom 4.1. u nastavku.

Linija ***Kod***

```
1:      create table ordinacija(
2:      sifra int not null primary key auto_increment,
3:      naziv varchar(50)
4:      );
6:      create table doktor(
7:      sifra int not null primary key auto_increment,
8:      ime varchar(30),
9:      prezime varchar(30),
10:     iban char(32),
11:     pregled datetime,
12:     ordinacija int not null
13:     );
```

Sl. 4.1. Primjer kôda za kreiranje tablica ordinacija i doktor

Baza podataka sastoji se od 5 tablica, prikazano na slici 4.2. u nastavku, pri čemu svaka tablica sadrži svoje atribute. Svaka tablica posjeduje atribut „sifra“ koji je ujedno i primarni ključ. Primarni ključ predstavlja atribut koji jedinstveno identificira svaki zapis u tablici. Vrijednost primarnog ključa mora biti jedinstvena vrijednost i ne smije sadržavati *NULL* vrijednost.

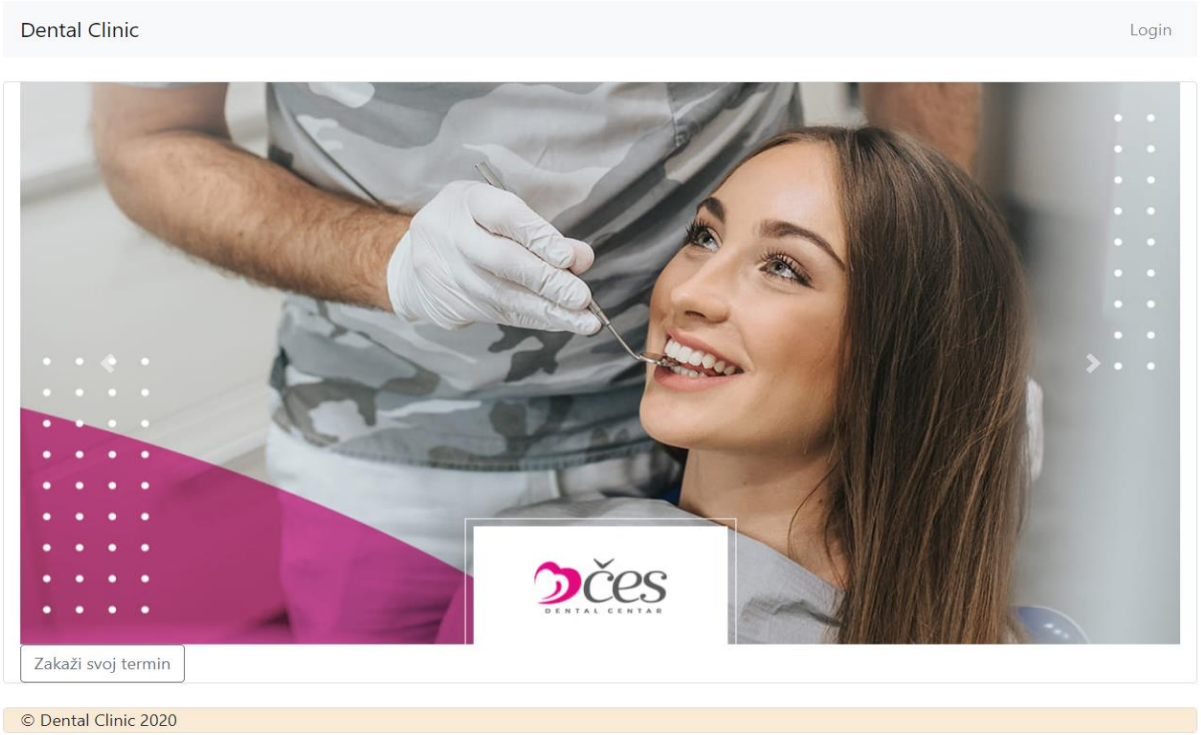


Sl. 4.2. Tablice u bazi podataka

4.2. Struktura web aplikacije

Struktura web aplikacije može se koristiti kroz dvije uloge rada: liječnik i pacijent. Prilikom pokretanja aplikacije otvara se početna stranica na kojoj su vidljive fotografije ordinacije i gumb za zakazivanje novog termina prikazano slikom 4.3. u nastavku.

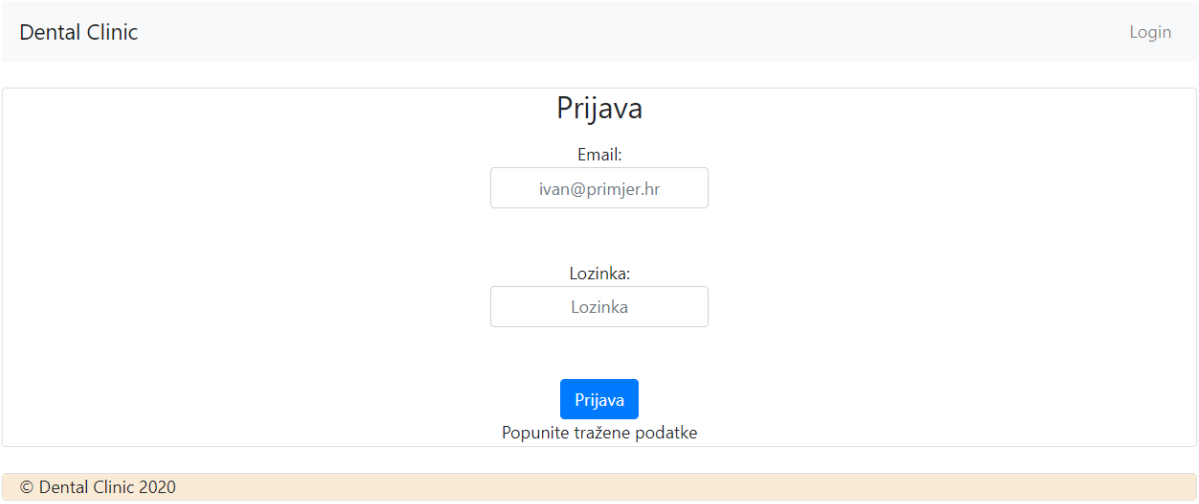
Korisnici koji nisu registrirani nemaju mogućnost prikaza ordinacija, liječnika niti termina. Za prikaz navedenih informacija korisnik mora biti prijavljen u aplikaciju.



Sl. 4.3. Početna stranica web aplikacije

4.2.1. Razvoj sustava prijave u aplikaciju

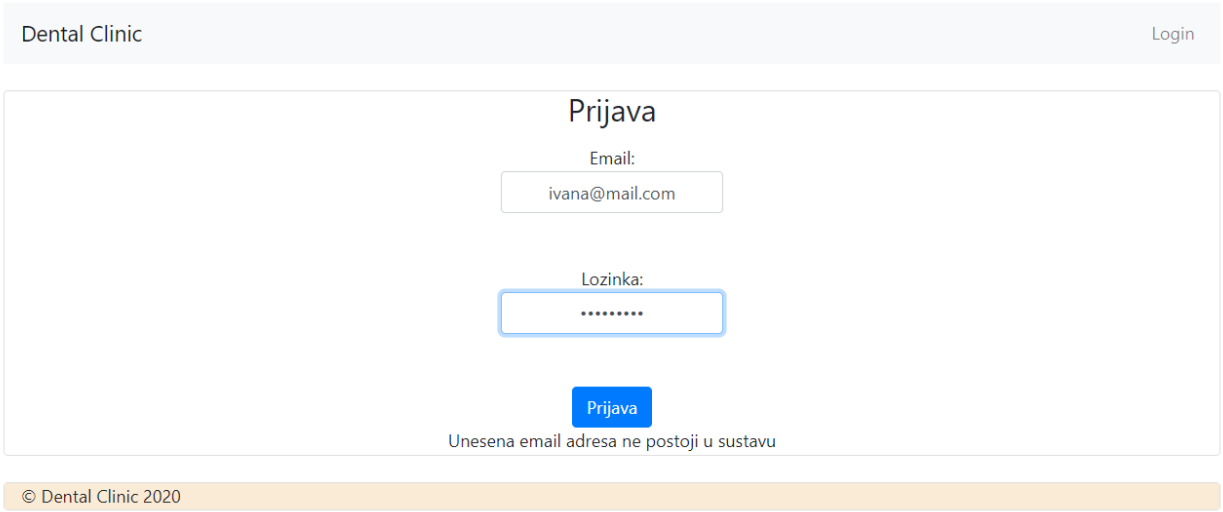
Prijava u sustav obavlja se pomoću forme za prijavu korisnika. Nakon što korisnik klikne na gumb prijava, uneseni podaci se šalju skripti putem HTTP POST metode. Slika 4.4. prikazuje formu za unos podataka koje je potrebno ispuniti za uspješnu prijavu korisnika.



Sl. 4.4. Forma za prijavu korisnika

Prilikom prijave u sustav postavljeni su uvjeti koji se moraju zadovoljiti za uspješnu prijavu korisnika. Neki od uvjeta su: zahtjev da email adresa u svom nazivu mora sadržavati znak @, zaporka obavezno mora biti unesena te provjera postojanja unesene email adrese u bazi podataka. Ukoliko neki od zahtjeva nije zadovoljen, sustav će, nakon što korisnik klikne na gumb Prijava, ispisati koji zahtjev nije zadovoljen, odnosno koja polja nisu ispravno unesena, što je prikazano slikom 4.5. u nastavku.

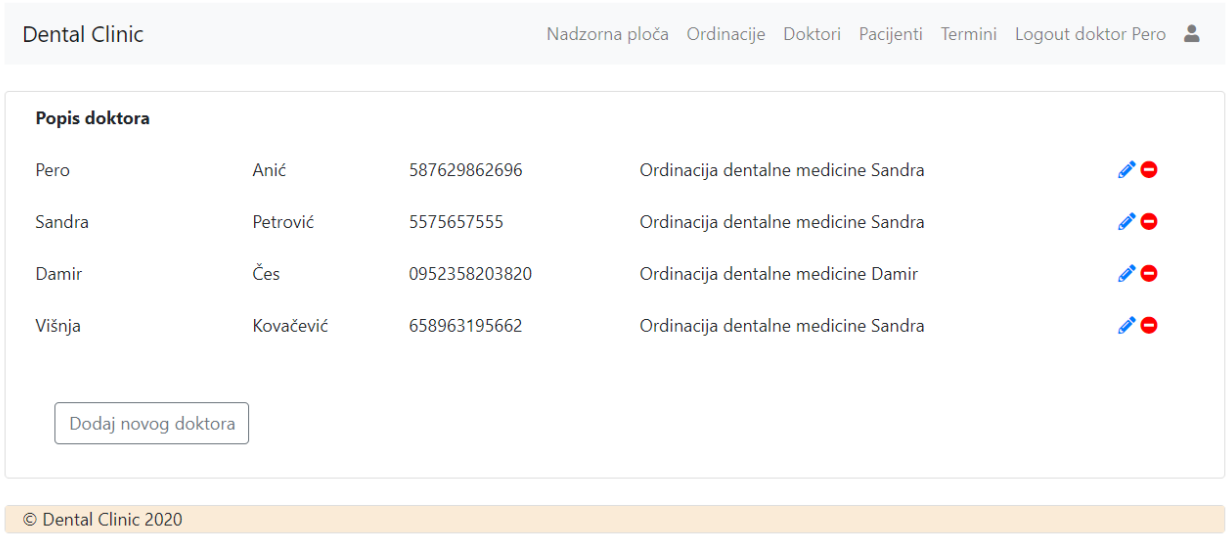
Ako se podaci uneseni u formi za prijavu korisnika, poklapaju s podacima u bazi podataka gdje se nalaze podaci o korisnicima sustava, korisnik pristupa aplikaciji sa svojom ulogom i koristi se pravima koje mu uloga omogućuje.



Sl. 4.5. Neuspješna prijava korisnika

4.2.2. Uloga liječnika

Nakon što se korisnik u sustav prijavi kao liječnik, otvara se početna stranica s dodatnim funkcionalnostima koje mu navedena uloga omogućuje. Korisnik s ulogom liječnika ima sljedeće funkcije: uvid u ordinacije, liječnika, pacijente i termine kao i manipulaciju istima.

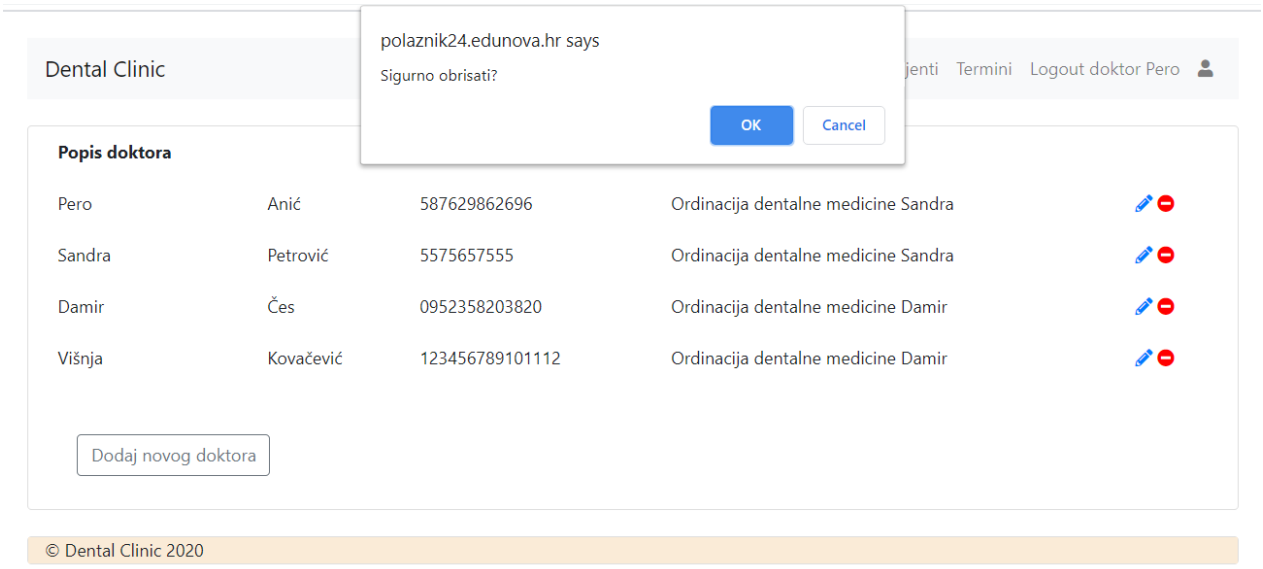


Sl. 4.6. Lista doktora sa pripadajućim ordinacijama

Tablični prikaz zaposlenih liječnika omogućuje preglednost prikaza. Tablica sadrži osnovne informacije o svakom liječniku, kao što su ime, prezime, IBAN te ordinacija u kojoj je zaposlen, što je prikazano slikom 4.6.

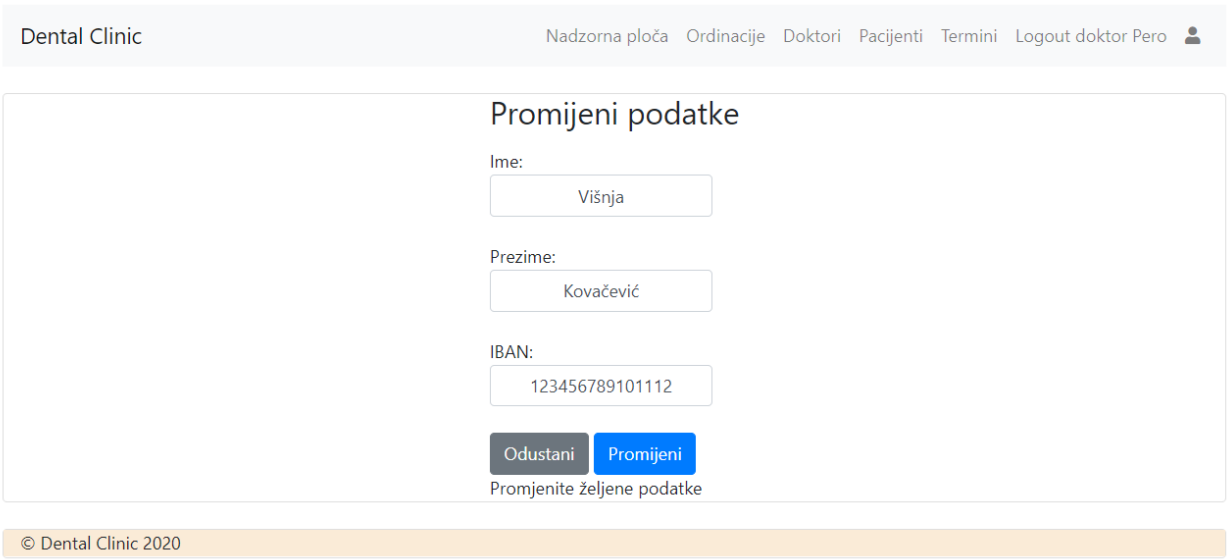
Osim uvida u osnovne informacije o članovima, uloga liječnika korisniku omogućuje dodatne funkcionalnosti poput brisanja pojedinog liječnika.

S obzirom da funkcija brisanja može biti osjetljiva, sustav će izbaciti poruku želimo li sigurno obrisati podatke, što je prikazano na slici 4.7. Nakon potvrde, klikom na ponuđeni gumb OK podatak se trajno briše iz baze podataka.



Sl. 4.5. Poruka upozorenja

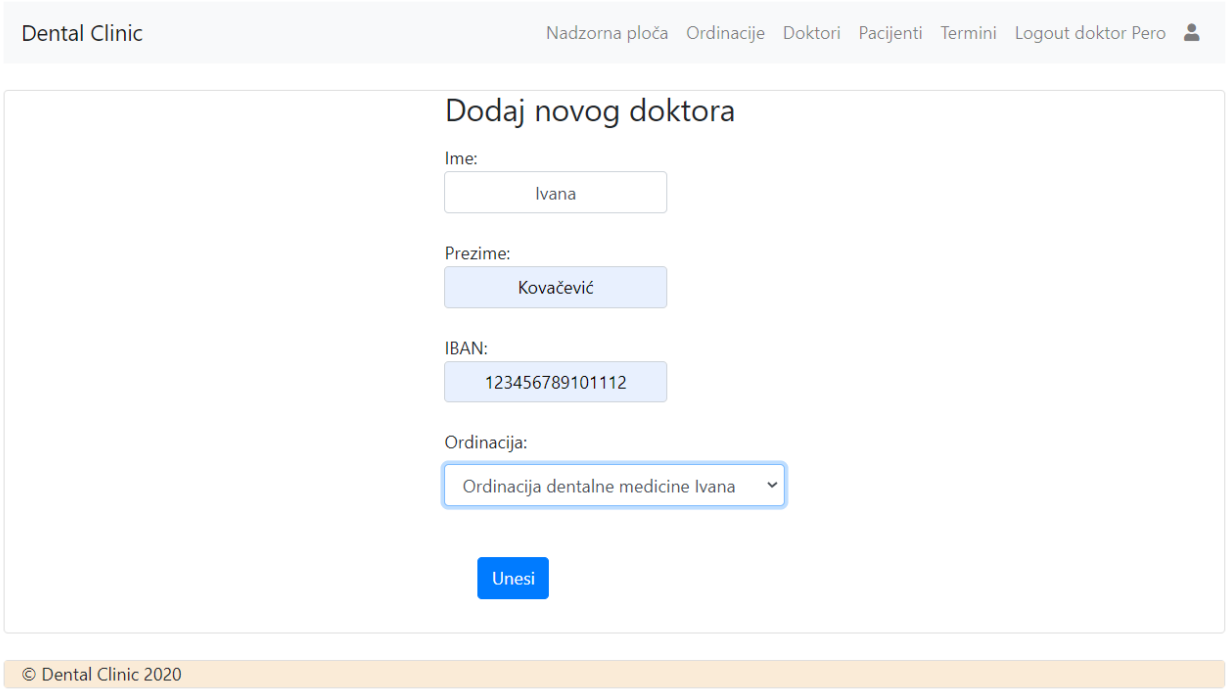
Još jedna od funkcionalnosti koju uloga doktora ima je mogućnost mijenjanja osnovnih podataka pojedinog doktora. Navedena funkcionalnost korisnu je omogućena putem forme prikazane na slici 4.8. u nastavku.



Sl. 4.8. Promjena postojećih podataka

Nakon što je korisnik promijenio željene podatke, klikom na gumb Promijeni, podaci se mijenjaju i u bazi podataka. Korisnik u svakom trenutku može odustati od promjene podataka te se klikom na gumb Odustani vraća na prethodnu stranicu, prikaz svih liječnika.

Osim brisanja i promjene postojećih podataka, uloga doktora korisniku omogućuje još jednu funkcionalnost, dodavanje novih podataka u bazu. Na slici 4.4. prikazan je gumb za dodavanje novog doktora. Klikom na navedeni gumb korisniku se otvara forma za dodavanje novih podataka, u ovom konkretnom slučaju, dodavanje novog liječnika, prikazana na slici 4.9. u nastavku.



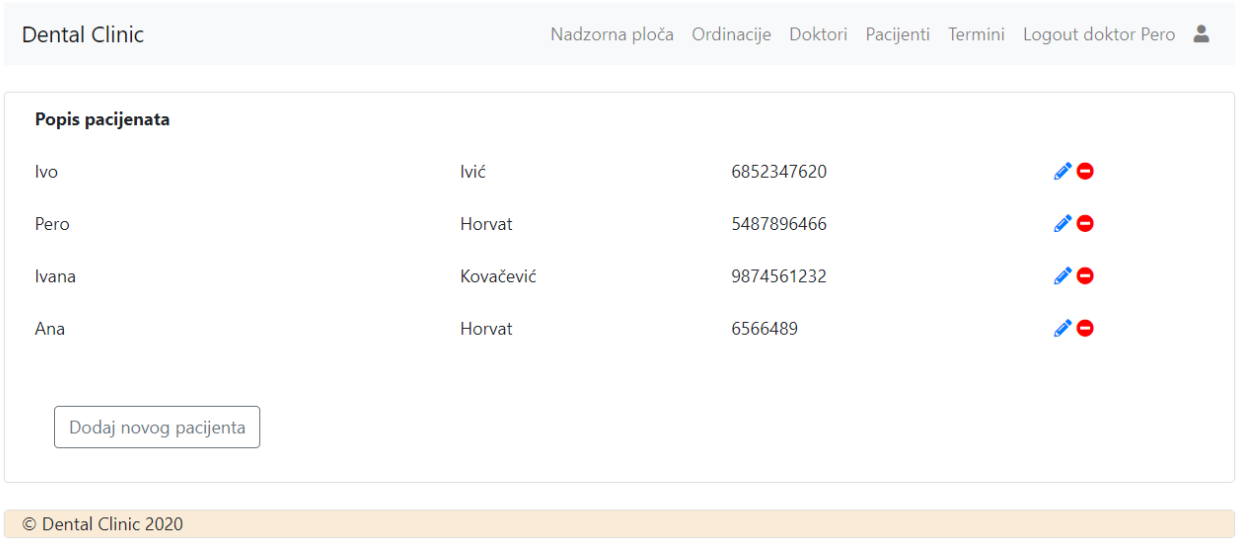
Sl. 4.9. Dodavanje novih podataka

Klikom na gumb Unesi, uneseni podaci spremaju se u bazu podataka te se korisnih vraća na prethodni zaslon sa popisom svih liječnika sa njihovim podacima, na kojemu se nalaze i upravo dodani podaci.

Na isti način kao i kod uređivanja podataka pojedinog liječnika, uloga liječnika korisniku omogućuje uređivanje podataka o pacijentima i ordinacijama, kao i brisanje istih i dodavanje novih. Korisnik može uređivati podatke klikom na ikonicu olovke, dok je brisanje omogućeno klikom na crveni kružić na kraju svakog retka tablice. Dodavanje novih podataka

omogućeno je klikom na gumb Dodaj novog pacijenta prikazano na slici 4.10. u nastavku, odnosno Dodaj novu ordinaciju.

Prethodno opisane funkcionalnosti za dodavanje, brisanje i uređivanje podataka za doktora, na isti način vrijede i za dodavanje, brisanje i uređivanje podatka za pacijenta i ordinaciju.



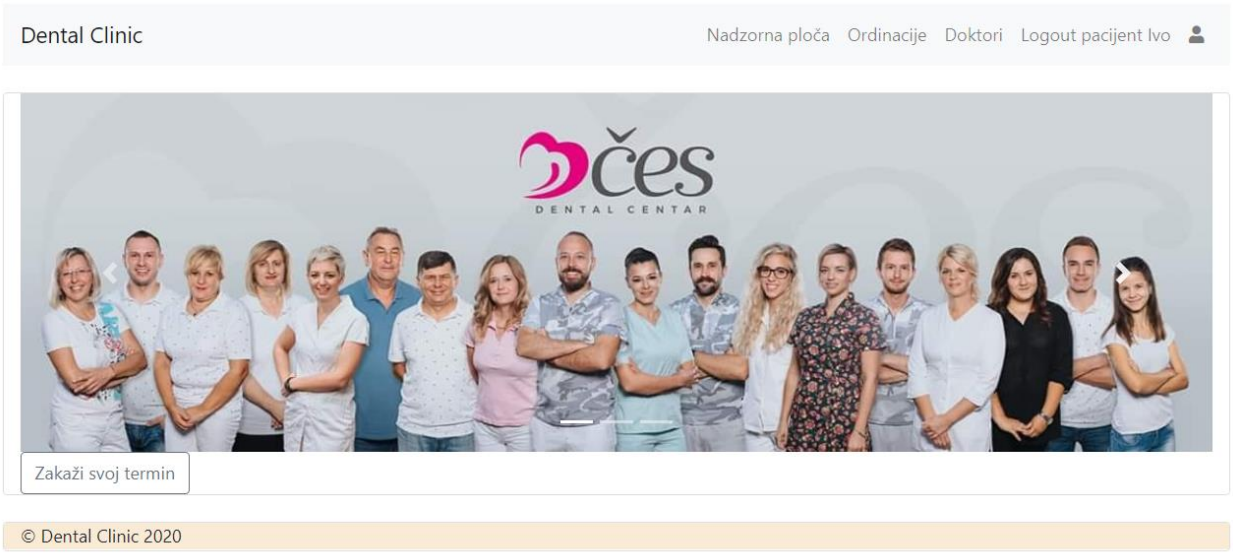
Sl. 4.10. Prikaz podataka o pacijentima sa ostalim funkcionalnostima

Osim navedenih funkcionalnosti, korisniku prijavljenom u sustav sa ulogom doktora omogućen je prikaz termina kao što je i prikazano na slici 4.11.

Sl. 4.11. Prikaz podataka o terminima pojedinih pacijenata sa pripadajućim doktorima

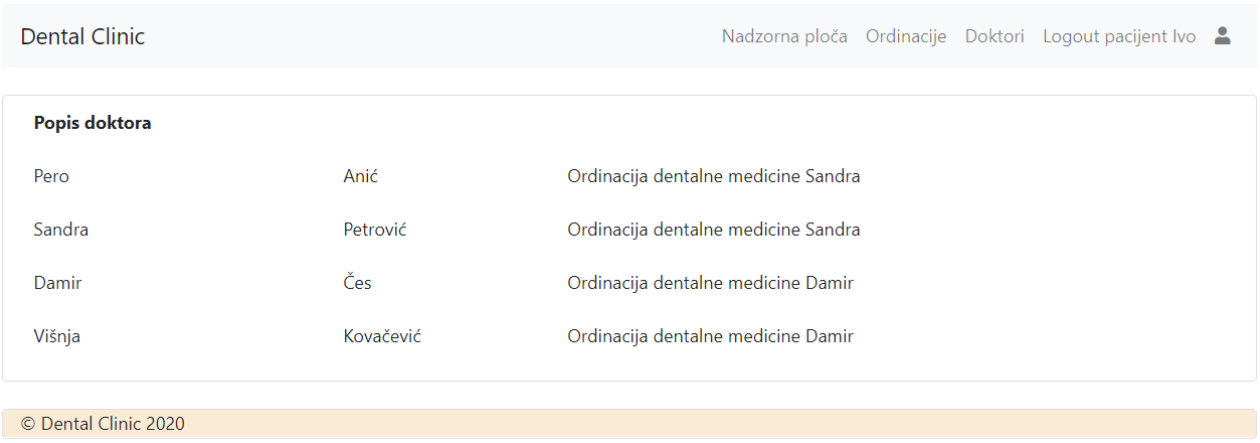
4.2.3. Uloga pacijenta

Nakon prijave u sustav sa ulogom pacijenta, korisniku se otvara početna stranica sa izbornikom koji mu nudi funkcionalnosti koje su dozvoljene samo njemu, prikazana slikom 4.12. u nastavku. Pacijent ima pravo vidjeti popis ordinacija i doktora te može zakazati novi termin pregleda.



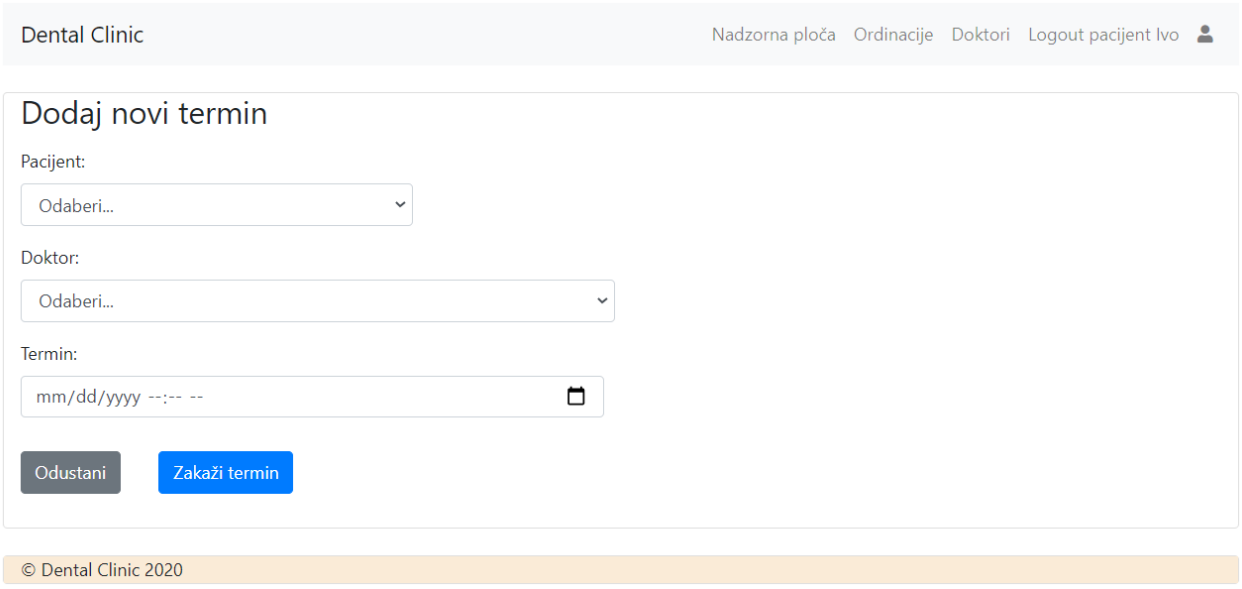
Sl. 4.12. Početna stranica web aplikacije – uloga pacijent

Klikom na polja Ordinacije ili Doktori u gornjem izborniku, korisniku se otvara popis ordinacija, odnosno liječnika. Tablični prikaz doktora prikazan je slikom 4.13. u nastavku.



Sl. 4.13. Tablični prikaz doktora – uloga pacijent

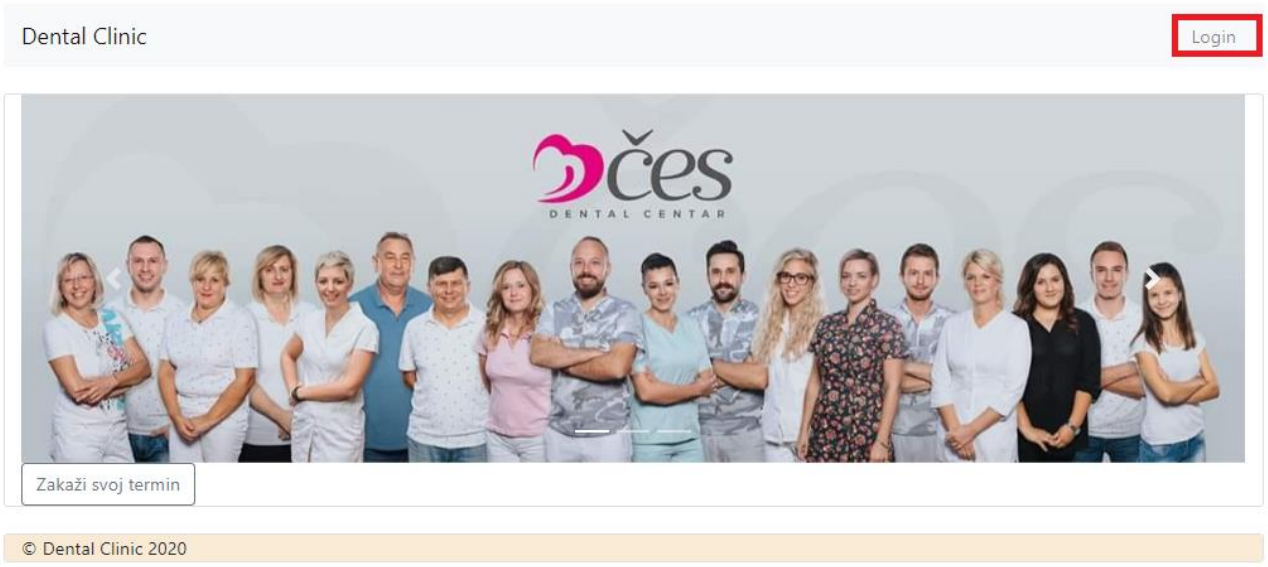
Na slici 4.12. vidimo kako se na početnoj stranici korisniku prikazuje gumb Zakaži svoj termin. Klikom na navedeni gumb korisniku otvara se stranica sa formom za unos novog termina sa svim potrebnim podacima. Prema slici 4.14. vidimo da se forma sastoji od dva padajuća izbornika i polja za odabir datuma i vremena pregleda. Padajući izbornici korisniku nude odabir pacijenta koji trenutno postoji u bazi podataka, što znači da samo pacijenti koji su već prethodno dodani u bazi podataka mogu zakazati svoj termin na ovaj način. Isto tako, korisnik iz padajućeg izbornika može odabrati doktora kod kojega želi zakazati svoj termin.



Sl. 4.14. Forma za unos novog termina

5. NAČIN KORIŠTENJA APLIKACIJE

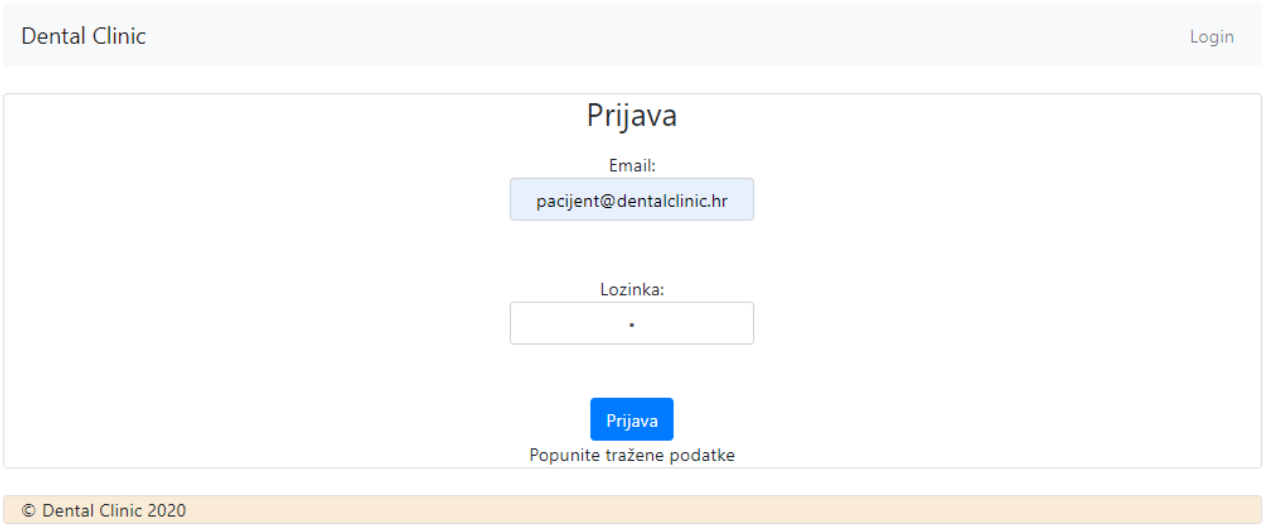
Kako bi rad ove aplikacije bio testiran, biti će prikazana prijava korisnika sa ulogom pacijenta, dodavanje novog termina, sa ulogom liječnika biti će prikazano dodavanje novog liječnika, lista zaposlenika, brisanje ordinacije, lista termina te prikaz dijela aplikacije gdje se mogu urediti podaci o pacijentu.



Sl. 5.1. Početna stranica aplikacije sa označenim gumbom za prijavu korisnika

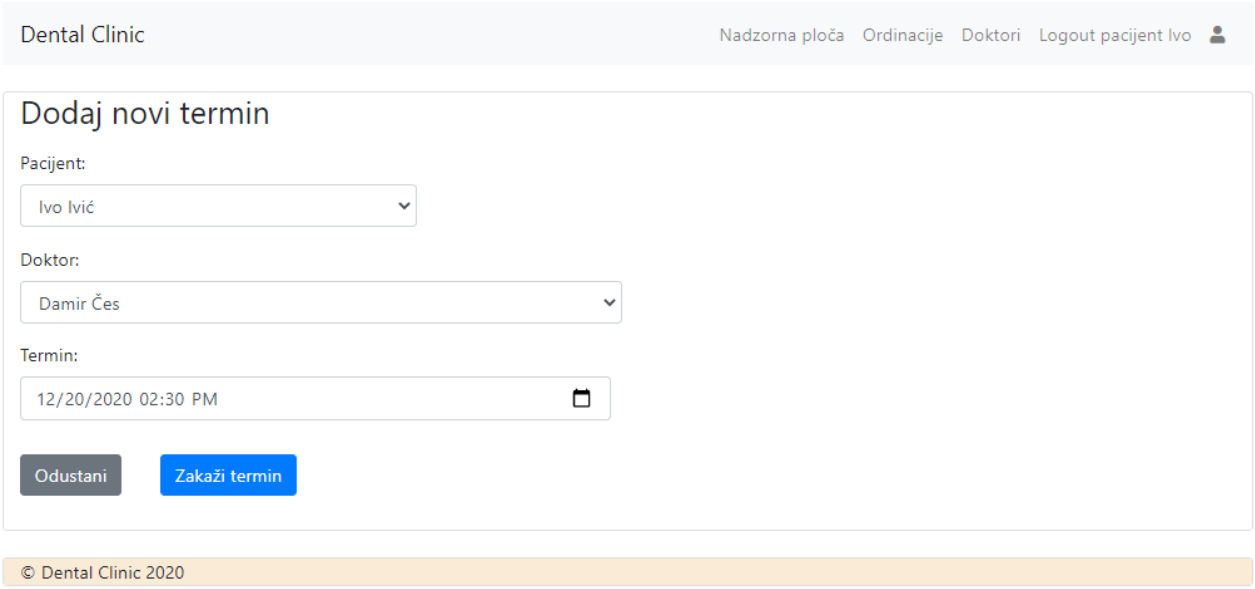
Na slici 5.1. prikazan je izgled početne stranice aplikacije na kojoj je vidljiv gumb za prijavu korisnika.

Nakon što je korisnik kliknuo na gumb za prijavu korisnika otvara se stranica sa formom za unos potrebnih podataka, što je i prikazano na slici 5.2.



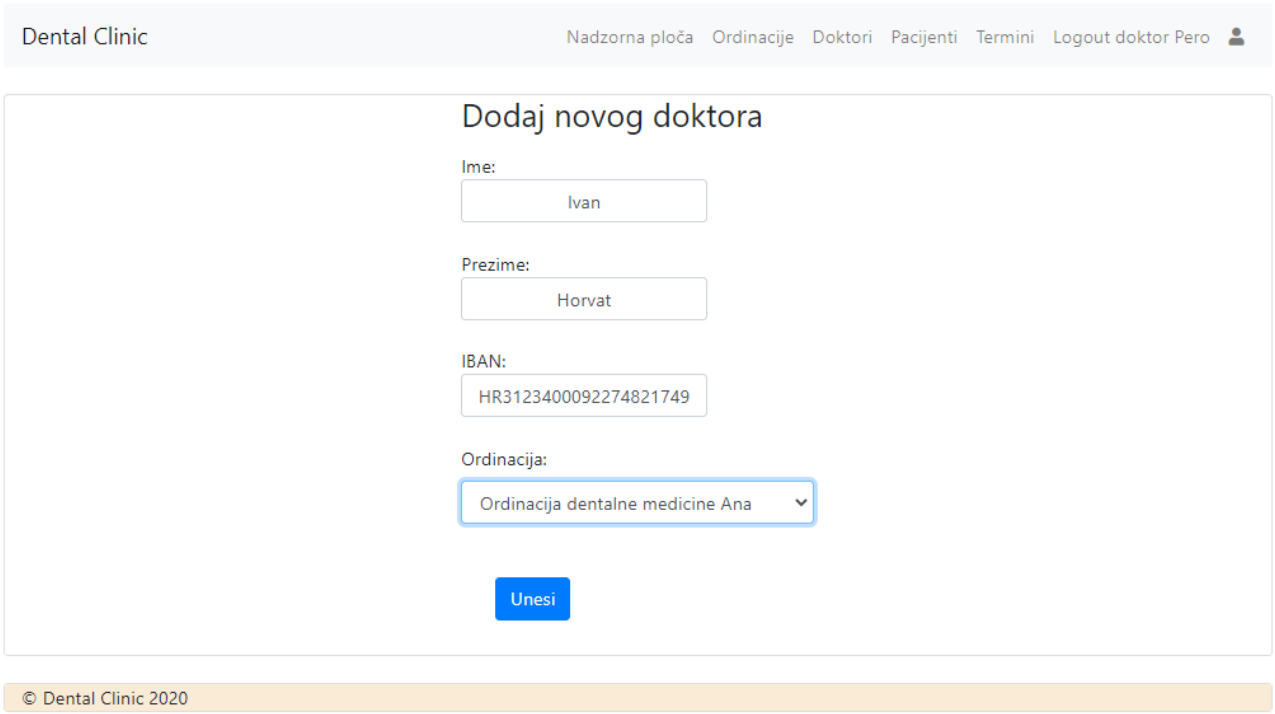
Sl. 5.2. Prijava korisnika

Ukoliko je korisnik dobro unio potrebne podatke ulazi u aplikaciju i može koristiti funkcionalnosti aplikacije koje mu njegova uloga omogućava. Glavna funkcionalnost koja mu je omogućena je zakazivanje termina pregleda, što je omogućeno klikom na gumb Zakaži svoj termin koji se nalazi na početnoj stranici aplikacije. Zakazivanje termina omogućeno je popunjavanjem forme koja sadrži polja za unos podataka, što je prikazano sliko 5.3. u nastavku.

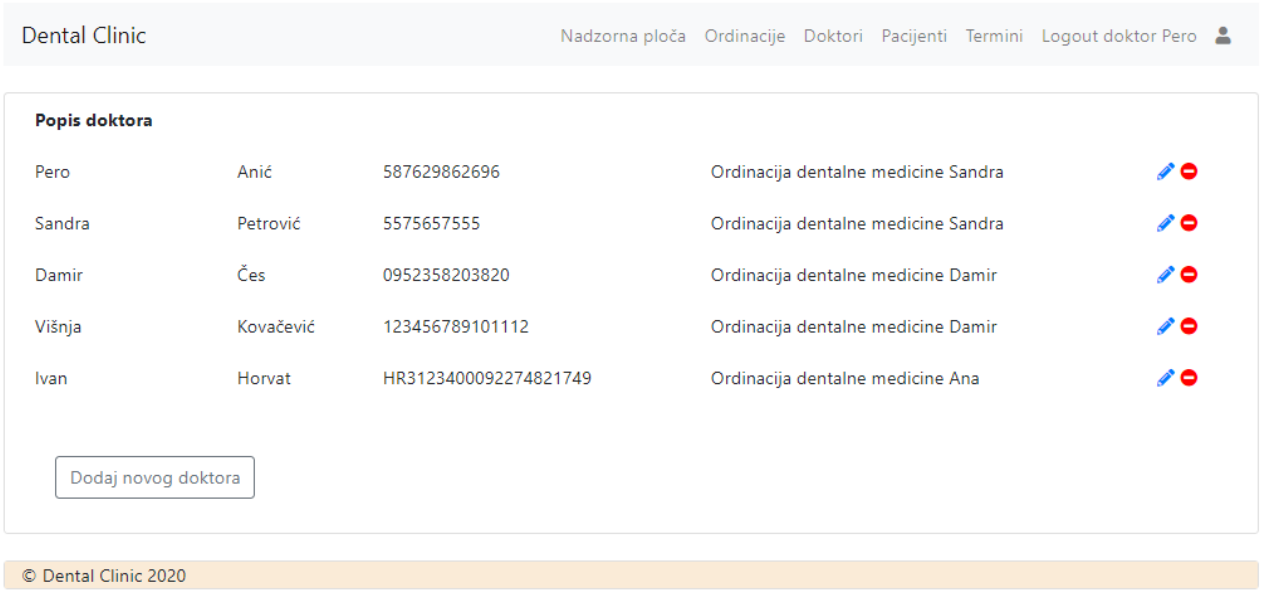


Sl. 5.3. Dodavanje novog termina

Korisnici prijavljeni s ulogom liječnika mogu vidjeti termin zakazan na prethodnoj slici. Ukoliko je korisnik prijavljen s ulogom liječnika omogućene su mu tzv. CRUD funkcije (engl. *Create, Read, Update, Delete*), odnosno može dodati nove doktore, pacijente i ordinacije, omogućen mu je prikaz podataka iz baze podataka koje također može obrisati u slučaju da mu nisu potrebni, može promijeniti podatke doktora, pacijenata i ordinacija. Slikom 5.4. u nastavku prikazano je dodavanje novog liječnika.



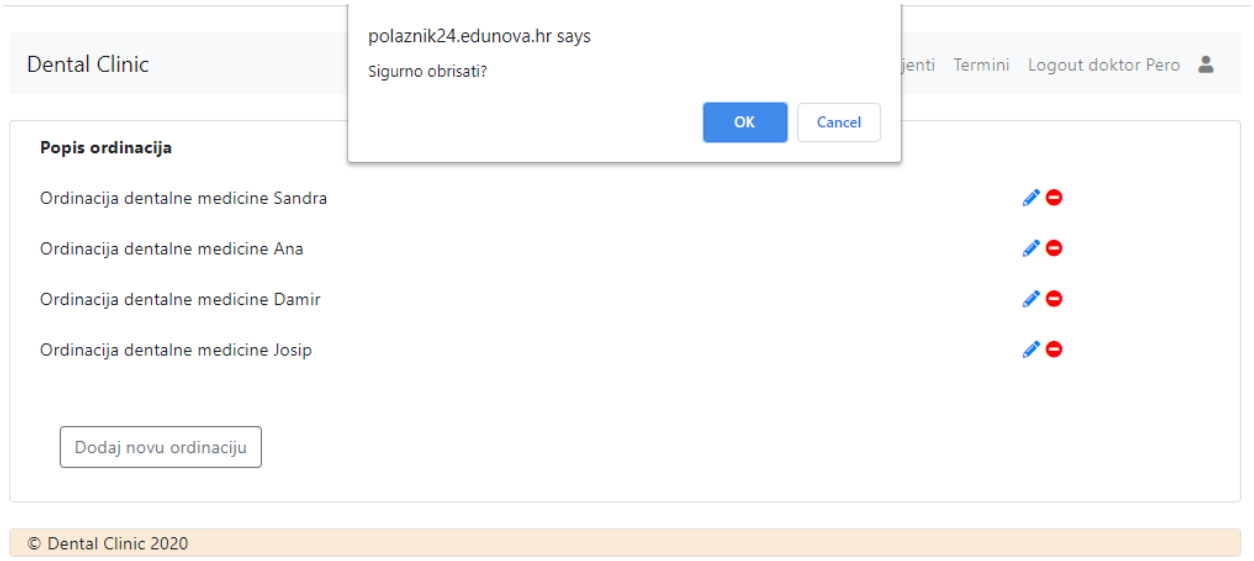
Sl. 5.4. Dodavanje novog liječnika



Sl. 5.5. Lista liječnika

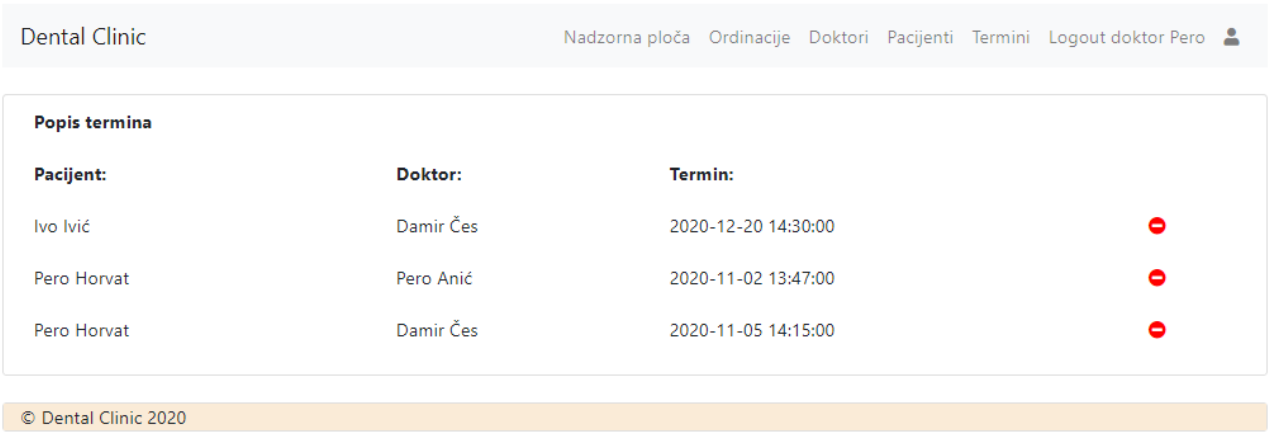
Prema slici 5.5. vidljivo je kako je upis u bazu podataka uspješno odrađen, novi podaci su prikazani na kraju liste.

Ako korisnik želi obrisati pojedini podatak to je omogućeno klikom na gumb za brisanje. Kako je brisanje rizična funkcija, korisnik mora potvrditi brisanje određenog podatka što je i prikazano slikom 5.6.



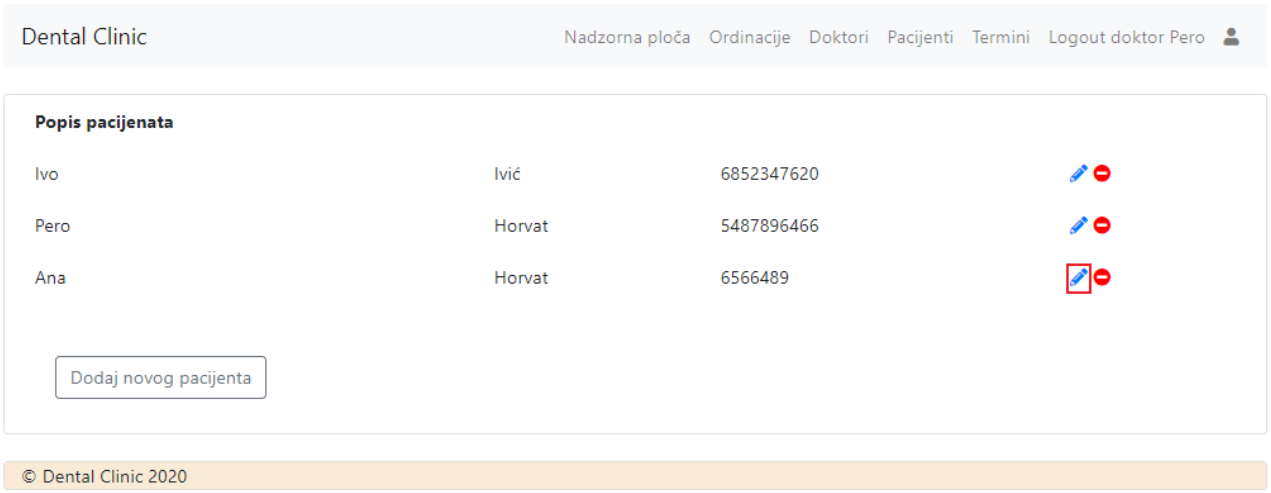
Sl. 5.5. Brisanje ordinacije

Slikom 5.3. prikazano je popunjavanje polja za unos podataka, dok na slici 5.6. u nastavku vidimo kako je termin uspješno zakazan sa unesenim podacima.

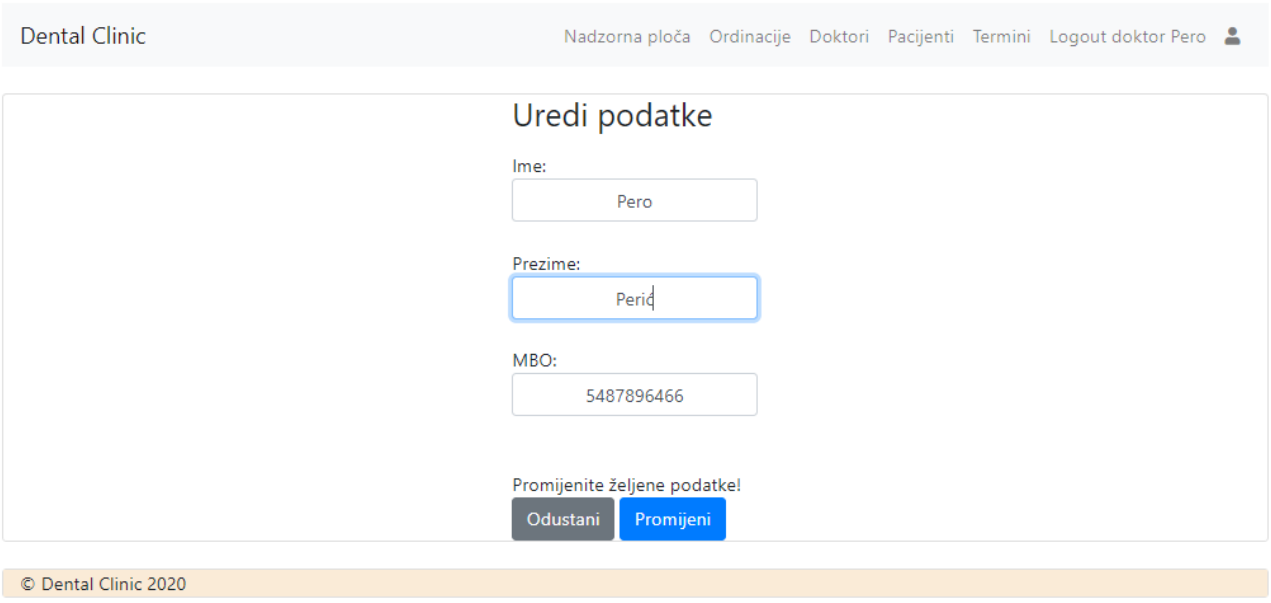


Sl. 5.6. Lista termina

Promjena podataka omogućena je klikom na gumb za uređivanje prikazanom na slici 5.7. u nastavku.



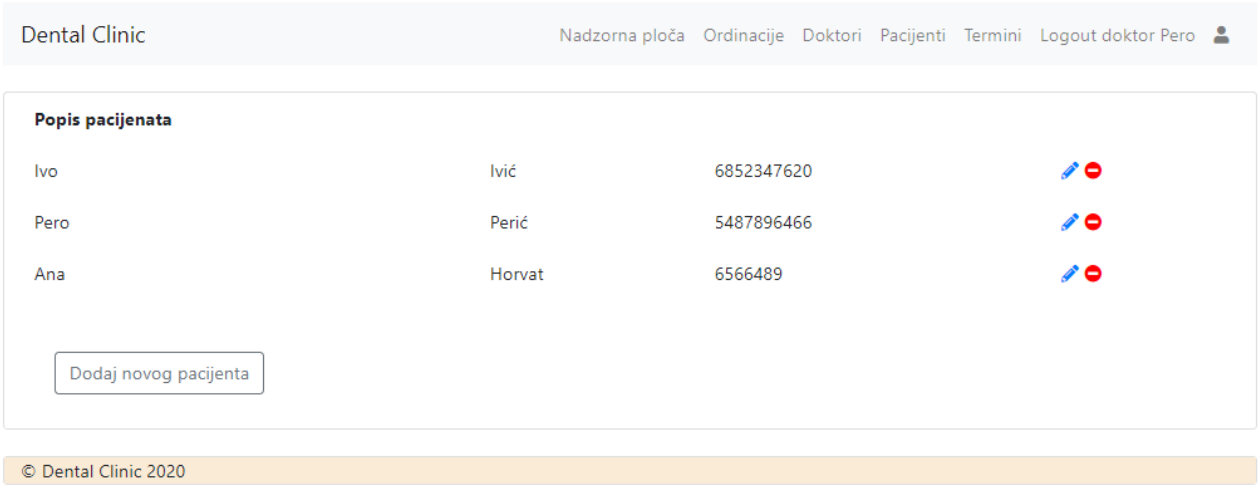
Sl. 5.6. Lista pacijenata sa označenim gumbom za uređivanje podataka



Sl. 5.8. Promjena podataka

Slikom 5.8. prikazan je unos novog prezimena pacijenta. Klikom na gumb Promijeni podaci će biti promijenjeni u bazi podataka.

Na slici 5.9. prikazana je uspješna promjena prethodno navedenih podataka u bazi podataka.



Sl. 5.9. Uspješna promjena podataka

6. ZAKLJUČAK

Cilj ovog diplomskog rada bila je praktična izrada web aplikacije za online rezervaciju termina kod zubara korištenjem MVC arhitekture. Ovom aplikacijom omogućeno je jednostavnije zakazivanje termina koje nam štedi dragocjeno vrijeme, ali i omogućuje jednostavniju organizaciju rada same stomatološke ordinacije.

S obzirom na to da su prilikom razvijanja aplikacije korištene različite tehnologije, izradom ove aplikacije stečeno je određeno praktično iskustvo. Aplikacija je izrađena u dva različita korisnička profila, gdje korisnici s obzirom na ulogu imaju određene mogućnosti koje mogu koristiti. Unutar ovog rada prikazana su teoretska znanja o izradi web aplikacija, koja su kasnije primijenjena na razvoj konkretnog primjera web aplikacije.

LITERATURA

[1.] https://storage.googleapis.com/setmore-website/v2/images/common-images/new/setmore-laptop-phone-devices.png

[2.] https://www.mozilla.org/en-US/firefox/browsers/what-is-a-browser/

[3.] https://searchsoftwarequality.techtarget.com/definition/3-tier-application

[4.] https://www.techadvisor.co.uk/test-centre/software/best-web-browsers-3635255/

[5.] https://getbootstrap.com/docs/4.5/getting-started/introduction/

[6.] https://getbootstrap.com/docs/4.0/components/navbar/

[7.] https://getbootstrap.com/docs/4.1/content/tables/

[8.] https://www.php.net/

[9.] https://www.budibase.com/blog/web-application-development/

[10.] http://www.mathos.unios.hr/wp/wp2009-10/P14_Web_aplikacije.pdf

[11.] https://www.w3schools.com/browsers/default.asp

[12.] https://www.w3schools.com/html/html_head.asp

[13.] https://www.w3schools.com/tags/tag_body.asp

[14.] https://www.w3schools.com/whatis/whatis_bootstrap.asp

[15.] https://www.codecademy.com/articles/mvc

[16.] https://developer.chrome.com/apps/app_frameworks

[17.] https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

[18.] https://www.tutorialspoint.com/sql/sql-overview.htm

SAŽETAK

U ovom radu izrađena je web aplikacija za online rezervaciju termina kod zubara. Aplikacija korisnicima omogućuje prijavu korisnika, unos novih podataka, brisanje i promjenu postojećih podatka. Opisane su tehnologije korištene za izradu ovog diplomskog rada kao i razvoj same aplikacije.

Aplikacija je izrađena kako bi se pacijentima olakšao postupak zakazivanja termina kod stomatologa, a stomatolozima pomogao prilikom organizacije rada.

Ključne riječi: baza podataka, CSS, HTML, MVC, PHP , web aplikacija,

ABSTRACT

Web application for online appointment reservation at the dentist

In this master thesis a web application for online appointment booking at the dentist was developed. The application allows users to log in, enter new data, delete and change existing data. The technologies used for the development as well as the development of the application itself are described in this thesis.

The application is designed to make it easier for patients to make an appointment with a dentist and to help dentists organize their work.

Keywords: CSS, database, HTML, MVC, PHP, web application,

ŽIVOTOPIS

Ivana Kovačević rođena je 4.4.1996. godine u Našicama. 2011. godine završila je Osnovnu školu Ante Starčevića u Viljevu. Nakon završetka osnovne škole upisuje opću gimnaziju u Srednjoj školi Donji Miholjac koju 2014. godine i završava. Iste godine na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija upisuje smjer računarstvo. Obranivši završni rad *Mobilna aplikacija za praćenje poljoprivrednih radova* pod mentorstvom prof.dr.sc. Gorana Martinovića, 2018. godine stekla titulu inženjerka računarstva. Iste godine, na istom fakultetu, upisuje diplomski studij računarstva, smjer Računalno inženjerstvo.

Potpis

_____

PRILOZI

Program web aplikacije nalazi se na priloženom CD-u.