

Aplikacija za upoznavanje partnera

Tomić, Mario

Master's thesis / Diplomski rad

2019

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://um.nsk.hr/um:nbn:hr:200:944419>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

Sveučilišni studij

APLIKACIJA ZA UPOZNAVANJE PARTNERA

Diplomski rad

Mario Tomić

Osijek, 2019.

SADRŽAJ

1. UVOD	1
1.1. Zadatak diplomskog rada	1
2. IDEJA ZA STVARANJE APLIKACIJE	2
2.1. Ideja korištenja aplikacije.....	2
3. PRIMIJENJENE TEHNOLOGIJE I ALATI	3
3.1. Angular.....	3
3.2. HTML.....	4
3.3. CSS.....	4
3.4. NGX Bootstrap.....	4
3.5. ASP.NET core	5
3.6. Entity framework.....	5
3.7. Visual studio code	5
3.8. Clouinary	5
3.9. Postman	5
4. REALIZACIJA APLIKACIJE.....	6
4.1. Baza podataka.....	6
4.2. Izrada web aplikacije.....	8
4.3. Izrada Angular aplikacije	9
4.4. Zaštita podataka.....	10
4.5. Dodavanje vanjskih komponenti u aplikaciju	12
4.6. Navigiranje u aplikaciji	13
4.7. Angular komponente	14
4.8. Dodavanje mogućnosti učitavanja fotografija.....	15
4.9. Dodavanje funkcionalnosti sviđanja	17
4.10. Slanje i primanje poruka.....	17
5. TESTIRANJE I IZGLED WEB APLIKACIJE.....	19

5.1. Testiranje backend dijela aplikacije	19
5.2. Izgled web aplikacije	20
5.3. Mobilni prikaz aplikacije.....	28
6. ZAKLJUČAK	30
LITERATURA	31
SAŽETAK.....	32
ABSTRACT	33
ŽIVOTOPIS	34
PRILOZI.....	35

1. UVOD

Tehnologija konstantno napreduje, te je dosegla visoku razinu i prostranjenost. Jedan od primjera su pametni telefoni, tableti i računala. Sve populacije od najmlađih do najstarijih koriste barem jedan od tih uređaja. Svaki od navedenih uređaja ima mogućnost povezivanja na internet i mnoge druge funkcionalnosti koje nam olakšavaju komunikaciju i mnoge druge aktivnosti. Samim time, većina populacije ne koristi klasične i staromodne pristupe za pronalazak partnera kao što su odlasci na grupne spojeve na slijepo, dogovorena upoznavanja i upoznavanja preko poznanika, već svojeg partnera pokušavaju pronaći putem interneta. Iz tog razloga postoje mnoge web aplikacije koje omogućavaju registraciju, stvaranje vlastitog profila, učitavanje slika i mnoge druge funkcionalnosti, kako bi korisnici lakše pronašli odgovarajućeg partnera. Ovo je aplikacija koja će korisnicima pružiti sve navedene i još mnoge druge funkcionalnosti. Osim što bi bila korisna za sve korisnike koji traže partnere, ovu aplikaciju moguće je koristiti i za komunikaciju, upoznavanje novih prijatelja i stjecanje popularnosti.

Budući da je riječ o web aplikaciji, za izradu su korišteni odgovarajući programski jezici i alati koji su nužni za izradu iste. Za izgled i responzivnost aplikacije korišteni su HTML, CSS i Angular 6, a pozadinski kod je pisan u C# jeziku na ASP.NET Core platformi.

1.1. Zadatak diplomskog rada

Aplikacija za upoznavanje partnera koristit će ASP.NET core Web API, Entity framework i Angular tehnologiju. Aplikacija će sadržavati mogućnost registracije, autentifikacije tijekom prijave, korisnici će moći uređivati svoj profil, također će moći učitavati svoje slike, slati poruke i označivati osobe sa sviđa mi se.

2. IDEJA ZA STVARANJE APLIKACIJE

Mnoge društvene mreže korištene su za upoznavanje i komunikaciju, ali kada je riječ o pronalasku partnera, neki od korisnika su poprilično sramežljivi. Iz tog razloga im je vrlo izazovno na takvim mrežama tražiti partnere jer znaju da ne koriste svi korisnici društvenu mrežu iz tog razloga, nego možda samo za lakšu komunikaciju, saznavanje novosti ili online druženja s prijateljima. Zbog toga je ideja napraviti aplikaciju koja služi isključivo za upoznavanje partnera. Trenutno već postoje mnoge slične aplikacije i ova ideja nije ništa novo, ali mnoge od njih naplaćuju svoju članarinu na mjesečnoj bazi, naplaćuju osnovne funkcionalnosti koje bi zapravo trebale biti besplatne ili reklamiraju razne oglase koji korisnicima nisu zanimljivi. Ova aplikacija bi za sad bila u potpunosti besplatna, a pružala bi osnovne funkcionalnosti potrebne za uspješnu komunikaciju i pronalazak partnera. Ako bi aplikacija postala popularna, ideja je ubaciti dodatnu funkcionalnost za korisnike kojima je teško pronaći partnera, da se češće prikazuju traženoj skupini partnera. To bi zapravo bio jedini način zarade i ako bi veliki broj korisnika počeo koristiti aplikaciju, vjerojatno bi to bilo dovoljno za troškove održavanja aplikacije, dodatan razvoj i dodavanje novih funkcionalnosti i mogućnosti.

2.1. Ideja korištenja aplikacije

Aplikacija bi se za sada koristila isključivo za upoznavanje partnera i komunikaciju. Iz tog razloga, korisnicima je potrebno omogućiti registraciju i prijavu na aplikaciju. Iako je karakter poprilično bitan, mnogim korisnicima je također bitan i izgled partnera, pa je zbog toga potrebno omogućiti učitavanje i uklanjanje slika, te postavljanje slike na profil korisnika. Profil će također moći sadržavati neke osnovne elemente kao što su datum rođenja, spol i vlastiti opis. Te funkcionalnosti bi trebalo biti moguće editirati u svakom trenutku. Nakon što se korisnici registriraju i prijave, aplikacija bi im trebala pomoći u pronalasku partnera. Iz tog razloga ideja je implementirati filtere kako bi korisnik mogao lakše pronaći osobe određenog spola, u rasponu određene dobi i to sve sortirano u odnosu na neke postavke kao što su npr. posljednja aktivnost korisnika.

3. PRIMIJENJENE TEHNOLOGIJE I ALATI

Web aplikacija napravljena je pomoću više odvojenih tehnologija kao što su C# koji je primijenjen za „backend“ dijelove aplikacije pomoću kojih se povezuje baza podataka s aplikacijom, HTML jezik koji je namijenjen za prezentaciju i opisivanje hipertekstualnih dokumenata i prikazivanje navedenih sadržaja na stranici, te CSS koji se brine o stilu stranice. Za glavne funkcionalnosti stranice i responzivnost uglavnom se pobrinuo Angular 6 web framework.

3.1. Angular

Angular [1] je *okvir* (eng. framework) koji služi za razvijanje aplikacija napravljen 2009. godine od tvrtke Google. Prva verzija zvala se AngularJS, ali već od druge verzije naziv je samo Angular. Kratica JS govori nam da je riječ o JavaScript okviru za razvijanje web aplikacija. Uklanjanjem „JS“ kratice, Google inicira da se uz razvijanje web aplikacija, mogu razvijati mobilne i desktop aplikacije, što čini Angular univerzalnim okvirom. Angular je razvijen u programskom jeziku TypeScript, kojeg je razvio Microsoft. Cilj Angulara je popraviti nedostatke JavaScript programskog jezika, zbog toga se nekada u literaturama TypeScript naziva nadskup JavaScript-a. Najznačajnije nadopune su otkrivanje grešaka tijekom prevođenja koda i uvoženje objektno- orijentiranog mehanizma programiranja.

Osnovna primjena Angular-a je za razvijanje jednostraničnih web aplikacija, ali uz korištenje Angular Universal-a moguće je i razvijati višestranične web aplikacije. Brojni mehanizmi koji su ranije razvijeni odbačeni su zbog lakšeg korištenja prilikom rada s Angular-om, ali i zbog unaprjeđenja performansi aplikacije.

Angular 6 razvijen je u svibnju 2018. godine, 5 mjeseci nakon svojeg prethodnika Angular-a 5. Najznačajnija poboljšanja u odnosu na prijašnju verziju su dodavanje *komandnog sučelja* (eng. Command Line Interface), dodavanje funkcionalnosti za razvoj komponenata i ažuriranje paketa Angular materijala. Ova verzija dakle pruža ugrađene funkcionalnosti za animaciju, http servise, materijale kao što su automatsko nadopunjavanje naredbi, navigaciju, meni alatne trake i slično.

3.2. HTML

HTML [2] je skraćena za prezentacijski jezik za izradu web stranica (eng. HyperText Markup Language). Zapravo je riječ o opisnom jeziku koji sadrži poveznice (eng. link) i oznake (eng. tag). HTML je najrasprostranjeniji opisni jezik i gotovo svaka web stranica ga sadrži, a ujedno je i besplatan. HTML je prepoznatljiv po tome što je veoma jednostavan za korištenje te ga je lako savladati, jer se koristi isključivo za opis hipertekstualnih dokumenata. Kako bi prikazao svoj sadržaj, HTML koristi internetske preglednike. Svi internetski preglednici podržavaju HTML i pri tome se nastoji postići da je izgled stranice jednak na svim preglednicima. HTML datoteka je zapravo uobičajena tekstualna datoteka, a ekstenzija te datoteke je „.html“ ili „.htm“.

Prva verzija HTML-a razvijena je 1991. godine, dok je trenutno najnovija verzija HTML 5 koja je trenutačno standardna. U odnosu na prijašnje verzije, značajno unaprjeđenje je reprodukcija video sadržaja na stranicama bez korištenja Adobe flash playera ili Microsoft SilverLighta, sposobnost upravljanja funkcionalnostima stranice pomoću tipkovnice i mnoge druge mogućnosti.

3.3. CSS

CSS [3] je skraćena za CascadingStyleSheets, a koristi se za definiranje stilova pomoću kojih se određuje izgled HTML elemenata. Pojavom HTML-a, elementi za definiranje stilova ubacivali su se direktno na elemente, ali je vrlo brzo uočena potreba za stilskim jezikom koji će odvojiti prikazivanje sadržaja i oblikovanje stila sadržaja. Prednost dodavanja CSS stilskog jezika je kontroliranje stilova na više HTML stranica istim kodom, čime se smanjuje dupliciranje koda. Pristupanje stilovima obavlja se pomoću HTML oznaka. CSS datoteka je također uobičajena tekstualna datoteka i ekstenzija ove datoteke je „.css“.

3.4. NGX Bootstrap

Ngx-bootstrap [4] je javno dostupan alat otvorenog koda za lakšu izradu web aplikacija, koji sadrži mnoštvo različitih predložaka koji su načinjeni od HTML-a, CSS-a i JavaScripta. Korištenjem ngx-bootstrapa olakšava se izrada web aplikacija jer se koriste već gotovi predlošci s uređenim stilom, te se manje vremena provodi u razvijanju izgleda stranice. Ngx-bootstrap se koristi isključivo za front-end dio aplikacije, te je prilagođen i za mobilni prikaz i ima svoju biblioteku s komponentama. Razlika između uobičajenog bootstrapa i ngx-bootstrapa je ta što je ngx bootstrap prilagođen za angular aplikacije te ga je lakše implementirati u iste.

3.5. ASP.NET core

ASP.NET core [5] je besplatna inačica web okvira ASP.NET-a, s poboljšanim performansama razvijena od strane Microsofta i zajednice programera. Koristi se za izradu modernih internet aplikacija baziranih na *oblaku* (eng. cloud). Radi se o modularnom okviru koji radi na .NET okviru, Windows platformi i .NET core platformi. Bez obzira na to što je novi okvir, ASP.NET core ima visoki stupanj kompatibilnosti s ASP.NET MVC okvirom te podržava mnoštvo aplikacija razvijenih u različitim verzijama koje prethodna verzija ASP.NET ne može podržati.

3.6. Entity framework

Entity framework [6] je okvir koji se koristi za objektno relacijsko mapiranje koji omogućuje korisnicima .NET okvira rad i manipulaciju podacima iz baze podataka korištenjem objekata. Korištenjem ovog okvira uklanja se potreba za korištenjem standardnog koda za manipulaciju podataka iz baze. Entity framework sadrži code first funkcionalnost koja omogućava stvaranje tablica u bazi podataka u odnosu na modele i klase napisane u C# programskom jeziku.

3.7. Visual studio code

Visual studio code [7] je besplatni alat za uređivanje koda, koji se može pokrenuti na svim platformama, a služi za razvijanje aplikacija. Razvijen je od strane Microsoft-a, i vrlo je prilagodljiv, što znači da dopušta korisnicima izmjenu tema, prečaca na tipkovnicama, instaliranje raznih dodataka i mnoge druge mogućnosti. Visual studio code je trenutno najpopularniji alat za razvijanje aplikacija i koristi ga oko 50% programera u svijetu.

3.8. Cloudinary

Cloudinary [8] je američka tvrtka koja nudi rješenja za spremanje slika i videozapisa na oblak. Registracija je besplatna i postoji limit korištenja usluga, te kada se on prekorači daljnje usluge se naplaćuju ovisno o paketu koji korisnik odabere. Za ovu aplikaciju, cloudinary je korišten za učitavanje fotografija koje će korisnici koristiti i pregledavati unutar aplikacije. Fotografije će se spremati na oblak, dok će oblak nuditi linkove na kojima se fotografije nalaze.

3.9. Postman

Postman [9] je alat koji je razvijen za testiranje aplikacije prilikom njenog razvoja. Ovim alatom se bez korisničkog sučelja mogu testirati POST, PUT, GET, DELETE i mnogi drugi HTTP pozivi funkcija, te dobiti kompletan odgovor u odnosu na parametre koje smo postavili.

4. REALIZACIJA APLIKACIJE

U trenutnom poglavlju opisani su načini i logika razmišljanja kod rješavanja problema i realizacije aplikacije. Prikazan je i tok izrade aplikacije te razvoj korak po korak. Kroz slike će biti pokazano sučelje i izgled komponenti, te njene funkcionalnosti i na koji način su implementirane.

4.1. Baza podataka

Za spremanje podataka potrebno je imati odgovarajuću bazu podataka. U ovoj aplikaciji baza podataka napravljena je korištenjem code first [10] metode i Entity frameworka. Svaki model napravljen u aplikaciji koja se upotrebljava za spremanje podataka code first metodom uz pomoć entity frameworka omogućuje izradu odgovarajuće baze podataka. Slika .4.1 prikazuje model fotografije korisnika pomoću kojega će biti stvorena tablica u bazi podataka. Potrebno je nakon stvaranja modela u model napisati svojstva koje će sadržavati, kako će biti povezan s ostalim klasama i modelima tako što će sadržavati svojstvo nekog drugog modela.

```
using System;

namespace DatingApp.API.Models
{
    10 references
    public class Photo
    {
        4 references
        public int Id { get; set; }
        4 references
        public string Url { get; set; }
        0 references
        public string Description { get; set; }
        0 references
        public DateTime DateAdded { get; set; }
        11 references
        public bool IsMain { get; set; }
        3 references
        public string PublicId { get; set; }
        0 references
        public User User { get; set; }
        1 reference
        public int UserId { get; set; }
    }
}
```

Slika .4.1. Model fotografije

Nakon definiranja svih modela odrađena je migracija te je odrađen pregled u datoteku migracije koja nam daje informacije o tome kako će izgledati baza podataka koja će nastati ili na koji način će migracija utjecati i izmijeniti bazu podataka. Datoteka migracije korištena je i za definiranje primarnih i stranih ključeva te dodatne postavke vezane za bazu podataka kao što je svojstvo neke varijable da može biti prazna ili će zahtijevati neku vrijednost. Slika 4.2 pokazuje postavke migracije za dodavanje stupca u tablicu fotografija. Nakon odrađene migracije, potrebno je u komandnom prozoru odraditi nadogradnju baze podataka naredbom „dotnet update database“. Nakon toga, baza je stvorena i spremna za korištenje.

```
using Microsoft.EntityFrameworkCore.Migrations;

namespace DatingApp.API.Migrations
{
    0 references
    public partial class AddedPublicID : Migration
    {
        0 references
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.AddColumn<string>(
                name: "PublicId",
                table: "Photos",
                nullable: true);
        }

        0 references
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropColumn(
                name: "PublicId",
                table: "Photos");
        }
    }
}
```

Slika 4.2. Prikaz dodavanja stupca u tablicu fotografija pomoću migracije

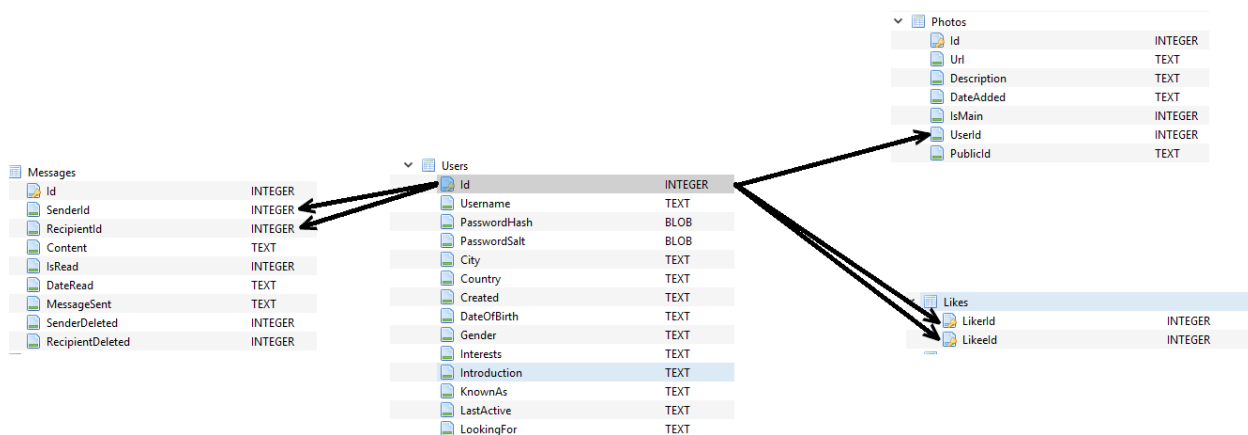
Baza podataka se sastoji od četiri međusobno povezane tablice:

Tablica korisnika (User): ova tablica sadrži sve podatke o korisniku, te se ID iz ove tablice koristi kao strani ključ u svim ostalim tablicama.

Tablica slika (Photos): ova tablica sadrži podatke o slici kao što su url slike i parametar je li slika glavna ili nije, datum kada je dodana te svoj osobni ID. ID korisnika naveden je kao strani ključ, te na osnovu ID-ja korisnika znamo kojem korisniku pripada fotografija.

Tablica poruka(Messages): u ovoj tablici se nalaze podaci o poruci kao što je sadržaj poruke, je li poruka pročitana i kada, kada je poruka poslana i je li poruka obrisana od strane primatelja ili pošiljatelja. Ako je poruka obrisana od strane jednog ili nijednog od njih, poruka ostaje u bazi podataka. Ako je poruka obrisana od pošiljatelja i primatelja, poruku također brišemo i iz baze. ID pošiljatelja i ID primatelja strani su ključevi tablice korisnika, te tako znamo koji korisnik je poslao poruku i kojem korisniku poruka treba stići.

Tablica svidanja(Like): u ovoj tablici nalazi se samo ID primatelja i ID pošiljatelja oznake „svidi mi se“, te ta dva strana ključa zajedno čine primarni ključ tablice.



Slika 4.3 Dijagram baze podataka

4.2. Izrada web aplikacije

Najprije je bilo potrebno napraviti Web API projekt. Za ovu aplikaciju odabran je ASP.NET Core Web API koji će omogućiti da aplikacija ima u potpunosti odvojen backend od frontend djela aplikacije. Nakon pozicioniranja u mapu u kojoj aplikacija treba biti smještena, naredbom „dotnet new webapi“ stvorena je nova aplikacija koja će služiti za backend dio realizacije rješenja. Slika 4.4 prikazuje datoteke i mape koje su stvorene navedenom naredbom, a sadržava osnovne datoteke i mape potrebne za početak razvoja

Name	Date modified	Type	Size
Controllers	9/16/2019 22:29	File folder	
obj	9/16/2019 22:29	File folder	
Properties	9/16/2019 22:29	File folder	
wwwroot	9/16/2019 22:29	File folder	
appsettings.Development.json	9/16/2019 22:29	JSON File	1 KB
appsettings.json	9/16/2019 22:29	JSON File	1 KB
DatingApp.csproj	9/16/2019 22:29	Visual C# Project File	1 KB
Program.cs	9/16/2019 22:29	Visual C# Source File	1 KB
Startup.cs	9/16/2019 22:29	Visual C# Source File	2 KB

Slika 4.4. Prikaz web api datoteka

Nakon toga je bilo potrebno stvoriti osnovne modele i kontrolere za aplikaciju. Prvo je napravljen model korisnika koji je sadržavao neke osnovne informacije kao što su ime korisnika, nadimak korisnika, datum rođenja, opis korisnika, grad i država u kojoj živi te kada je korisnik napravio korisnički račun. Ovaj model je služio za osnovne informacije i s vremenom i razvitkom aplikacije postepeno će se dodavati nova svojstva u njega. Zatim je napravljen kontroler za korisnika u koji su smještene funkcije i metode za dohvaćanje korisnika i njegovih podataka.

4.3. Izrada Angular aplikacije

Za izradu Angular aplikacije aplikacije potrebno je bilo instalirati Angular CLI [11] (Comand line interface) pomoću naredbe „npm install -g @angular/cli“. Nakon instalacije, naredbom „ng new DatingApp-SPA“ stvorena je nova angular aplikacija s također osnovnim početnim datotekama(Slika 4.5).

Name	Date modified	Type	Size
e2e	3/18/2019 19:52	File folder	
node_modules	9/11/2019 14:45	File folder	
src	3/18/2019 19:52	File folder	
.editorconfig	3/18/2019 19:52	EDITORCONFIG File	1 KB
.gitignore	3/18/2019 19:52	Text Document	1 KB
angular.json	4/1/2019 14:42	JSON File	4 KB
package.json	9/11/2019 14:46	JSON File	2 KB
package-lock.json	9/11/2019 14:46	JSON File	367 KB
README.md	3/18/2019 19:52	MD File	2 KB
tsconfig.json	3/18/2019 19:52	JSON File	1 KB
tslint.json	3/26/2019 10:42	JSON File	3 KB

Slika 4.5 Prikaz početnih datoteka Angular aplikacije

4.4. Zaštita podataka

Aplikacija za upoznavanje partnera mora biti pouzdana i jamčiti sigurnost podataka. Uz to, mora imati i osnovne funkcionalnosti kao što su registriranje korisnika, prijavljivanje korisnika na svoj račun, prikaz i izmjenu osobnog profila i mogućnost razmjene poruka između korisnika aplikacije. Za registraciju i prijavu korisnik mora odabrati korisničko ime i lozinku, te je lozinku potrebno pohraniti na siguran način. Ako bismo lozinku pohranili na uobičajeni način spremanjem lozinke u bazu podataka, osoba koja ima pristup bazi podataka imala bi i pristup svim korisničkim računima. Ovaj pristup rješenju je vrlo nesiguran. Iz tog razloga, prilikom registracije, korisnikovu lozinku je potrebno šifrirati pomoću algoritma i spremi u bazu podataka u obliku beznačajnih znakova(Slika 4.6).



Slika 4.6. Šifriranje lozinke SHA512 algoritmom

Međutim, to također nije dovoljna zaštita zato što će isti algoritam za istu lozinku stvoriti jednaki niz znakova što također nije poželjno. Zbog toga se u algoritam dodaje određeni „začin“, što omogućuje da nakon primjene algoritma ne možemo prepoznati dvije iste lozinke, čineći lozinku puno sigurnijom(Slika 4.7).



Slika 4.7. Prikaz primjene algoritma sa začinom za dvije iste lozinke

Za prijavu korisnika koji je registriran, korišten je JSON Web Token [12], koji koristi algoritam za šifriranje svojeg sadržaja. Sadržaj tokena sastoji se od zaglavlja, tijela i tajnog dijela. U zaglavlju se nalazi tip tokena i algoritam pomoću kojeg je šifriran, u tijelu tokena se nalaze podaci o korisniku, njegov identifikacijski broj, te podaci o tokenu kao što je vrijeme do isteka rada tokena. Tajni dio tokena sadrži tajnu lozinku koja se koristi za šifriranje i je spremljen na server i nikada zapravo nije prikazan korisniku iz sigurnosnih razloga. Slika 4.8 prikazuje izgled tokena kada je šifriran i njegovu strukturu nakon dešifriranja.

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.f6B5EvaqVVvKv3ASWdMJPLZs8G4AIK8J2iQXtsbcS3w
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

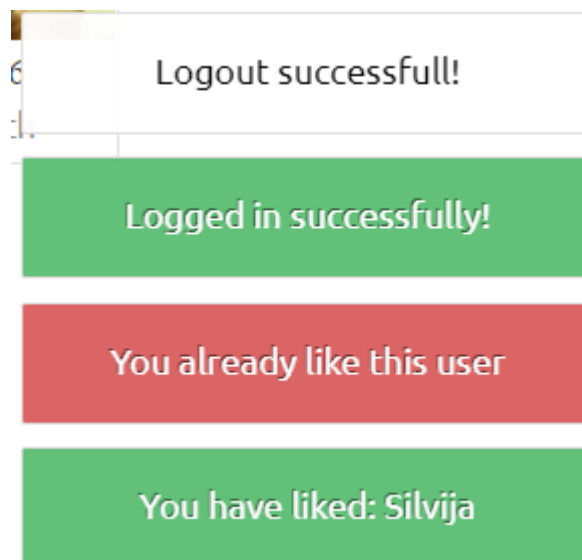
```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Tajna riječ za kodiran
)  secret base64 encoded
```

Slika 4.8 Izgled šifriranog i dešifriranog tokena [13]

4.5. Dodavanje vanjskih komponenti u aplikaciju

Korištenje vanjskih komponenti može biti vrlo korisno i značajno povećati brzinu izrade aplikacije. Nema smisla pisati kod koji je već netko drugi napisao, te gubiti vrijeme ako možemo iskoristiti postojeće biblioteke i komponente za realizaciju istog cilja. Zbog toga je potrebno dodati neke već gotove i testirane komponente u projekt radi lakše realizacije.

Za prikaz obavijesti u aplikaciji, korišten je Alertifyjs [14], koji nam omogućuje responzivni prikaz grešaka ili uspješno obavljene akcije u odgovarajuće uređenom stilu. Ova funkcionalnost nam je potrebna kako bismo korisnika obavijestili da je uspio u nekoj akciji kao što je registracija, prijava, odjava ili slanje oznake „sviđa mi se“ korisniku, te ovisno o akciji primjenjuje se odgovarajuća boja obavijesti. Slika 4.9 prikazuje nekoliko obavijesti koje ova komponenta prikazuje u aplikaciji.



Slika 4.9 Prikaz Alertifyjs obavijesti

Kako bismo lakše dizajnirali sučelje u aplikaciji te dodali responzivnost i funkcionalnosti poput onih koje bootstrap nudi, korišten je ngx-bootstrap. Razlog korištenja ovog dodatka za komponente u ovoj aplikaciji prvenstveno je taj što je ngx-bootstrap prilagođen za angular te nudi mnoštvo gotovih primjera za primjerice dropdown komponente, buttone i mnoštvo ostalih funkcionalnosti korištenih u ovoj aplikaciji.

Kao što je slučaj i kod ngx-bootstrapa, također je korišten i bootswatch dodatak za lakše dizajniranje boja i teme sučelja. Ovaj dodatak nam pruža mnoštvo CSS stilova koje možemo direktno dodavati u html, a jedino što trebamo poznavati su nazivi klasa i što one rade kako bismo ih pravilno primijenili. Informacije o klasama za izradu ove aplikacije pronađene su na službenoj stranici bootswatch-a.

4.6. Navigiranje u aplikaciji

Iako je moguće organizirati navigiranje na puno načina, za ovu aplikaciju korištena je typescript datoteka „routes.ts“ (Slika 4.10) u kojoj će se spremati sve rute stranica kojima će aplikacija moći pristupati. Na ovaj način je puno lakše organizirati pristup stranicama, te putanju koja će učitavati te stranice. Dakle prilikom upisivanja rute u pretraživač, ova datoteka će redom tražiti odgovara li ta ruta nekoj od navedenih u njenom sadržaju, te odabrati prvu navedenu rutu i preusmjeriti aplikaciju na istu ako odgovara. Također ako napisana ruta ne postoji, korisnik će biti preusmjeren na početnu stranicu aplikacije.

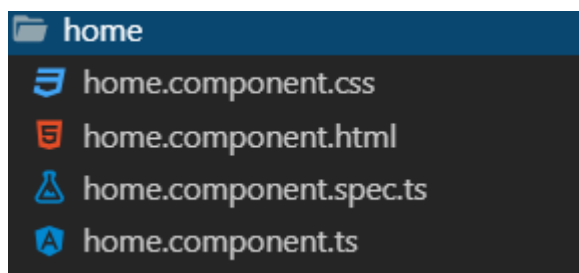
```
export const appRoutes: Routes = [
  {path: '', component: HomeComponent},
  {
    path: '',
    runGuardsAndResolvers: 'always',
    canActivate: [AuthGuard],
    children: [
      {path: 'members', component: MemberListComponent, resolve: {users: MemberListResolver}},
      {path: 'members/:id', component: MemberDetailComponent,
        resolve: {user: MemberDetailResolver}},
      {path: 'member/edit', component: MemberEditComponent,
        resolve: {user: MemberEditResolver}, canActivate: [PreventUnsavedChanges]},
      {path: 'messages', component: MessagesComponent, resolve: {messages: MessagesResolver}},
      {path: 'lists', component: ListsComponent, resolve: {users: ListsResolver}},
    ]
  },
  {path: '**', redirectTo: '', pathMatch: 'full'},
];
```

Slika 4.10 routes.ts datoteka

Ovim pristupom je također bilo potrebno za svaku novu stranicu aplikacije stvarati novu komponentu.

4.7. Angular komponente

Komponente u angularu (Slika 4.11) obično sadrže četiri datoteke, te svaka od njih ima određenu ulogu. HTML datoteka koristi se za prikaz sadržaja, CSS datoteka za oblikovanje prikaza sadržaja, TS datoteka za upravljanjem ponašanja stranice dok se SPEC.TS datoteka koristi za testiranje.



Slika 4.11 Komponenta u Angularu

AppComponent početna je komponenta modula. Za ovu komponentu specifično je to što ne pruža neke značajne funkcionalnosti nego sadrži router komponentu koja je zadužena za navigiranje u aplikaciji i prikaz drugih komponenti.

HomeComponent je komponenta početne stranice te pruža funkcionalnosti kao što odabir korisnika želi li se registrirati ili prijaviti u aplikaciju.

RegisterComponent je komponenta koja omogućava registraciju korisnika. Prilikom registracije u ovoj je komponenti sadržana sva potrebna validacija i komunikacija s backend dijelom aplikacije za spremanje podataka.

NavComponent je komponenta koja služi za prijavu korisnika te spremanje tokena i ostalih podataka o korisniku preko autentifikacijskog servisa. Ova komponenta također služi za brisanje tokena, korisnika i ostalih podataka iz lokalne memorije prilikom odjave korisnika sa stranice.

ListsComponent je komponenta koja služi za prikazivanje korisnika koji su označili prijavljenog korisnika s oznakom „sviđa mi se“, te također pomoću ove komponente možemo i prikazati sve osobe koje smo mi označili tom oznakom. Komponenta također sadrži navigiranje pomoću dodatnih stranica ako se radi o većem broju korisnika.

Members mapa sadrži **member-card**, **member-detail**, **member-edit**, **member-list**, **member-messages** i **photo-editor** komponente. Ove komponente vezane su za vlastiti profil korisnika te

svaka od komponenata omogućuje dio funkcionalnosti kako bi korisnik mogao uređivati svoj profil, pregledati profil, pregledati slike koje je učitao, učitati nove slike, postaviti sliku kao početnu te se dopisivati s drugim korisnicima.

MessagesComponent je komponenta koja isključivo služi za pregled i brisanje nepročitanih, poslanih i primljenih poruka. Funkcionalnost ove komponente razlikuje se od **member-messages** komponente tako da se u ovoj komponenti mogu brisati poruke, a u **member-messages** komponenti mogu slati poruke. Klikom na bilo koju poruku koja se nalazi u ovoj komponenti, aplikacija će preusmjeriti korisnika na **member-messages** komponentu te otvoriti tab za dopisivanje s korisnikom na čiju smo poruku kliknuli.

4.8. Dodavanje mogućnosti učitavanja fotografija

Jedna od nužnih funkcionalnosti koje aplikacija za upoznavanje partnera mora imati je mogućnost dodavanja i upravljanja fotografijama. Samim time, nameće se problem spremanja fotografija, koje zahtijevaju značajno više prostora nego uobičajeni podaci kao što su ime, lozinka i opis korisnika. Baza podataka pokazala se kao loše rješenje za spremanje podataka zbog toga što bi aplikacija značajno izgubila na performansama, te bi se baza podataka vrlo brzo napunila ako bi se radilo o velikoj količini korisnika i fotografija.

Nakon eliminiranja opcije spremanja fotografija u bazu podataka, preostale mogućnosti su spremanje fotografija na disk ili na oblak. ako bismo odabrali spremanje na lokalnom disku ili spremanje na memoriji web servera, vrlo brzo bi nam ponestalo memorije ako bismo imali puno korisnika. Zbog navedenih razloga odabrana je opcija spremanja fotografija na oblak, dok se samo link slike sprema u bazu podataka. Ovu opciju je složenije bilo implementirati ali je najbolja kada su u pitanju performanse i proširivanje prostora u slučaju da aplikacija postane popularna i počne ju koristiti puno korisnika.

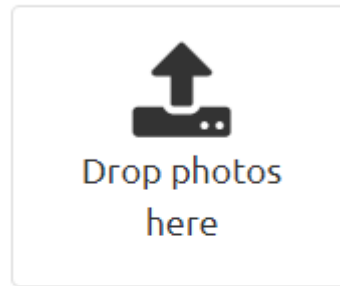
Na službenoj stranici Cloudinary oblaka nalazi se dokumentacija za učitavanje fotografija koja je prilagođena za .NET okruženje te su neke funkcije za učitavanje i uklanjanje fotografija implementirane na taj način. Također je bilo potrebno podesiti i podatke za spajanje na Cloudinary oblak, te je taj dio obavljen u datoteci „appsettings.json“, gdje smo pohranili i tajni ključ za šifriranje tokena (Slika 4.12).

```
{...} appsettings.json ×
DatingApp.API > {...} appsettings.json > {} Logging > {} LogLevel > abc Default
1 {
2   -- "AppSettings": -- {
3     -- "Token": -- "██████████"
4   -- },
5   -- "ConnectionStrings": --
6   -- {
7     -- "DefaultConnection": "Data Source=DatingApp.db"
8   -- },
9   -- "Logging": --
10  -- {
11    -- "LogLevel": -- {
12      -- "Default": -- "Warning"
13    -- }
14  -- },
15  -- "AllowedHosts": -- "*",
16  -- "CloudinarySettings": -- {
17    -- "CloudName": -- "dotkor",
18    -- "ApiKey": -- "418734813923934",
19    -- "ApiSecret": -- "████████████████████████████████████████"
20  -- }
21 }
```

Slika 4.12 Prikaz podataka za spajanje na Cloudinary

Nakon implementacije mogućnosti za učitavanje fotografija na backendu potrebno je bilo napraviti odgovarajuće sučelje za učitavanje fotografija na frontend dijelu aplikacije. Za ovaj problem odabrali smo rješenje ng2-file-upload [15] biblioteke dostupne na internetu. Uz mnoštvo opcija za učitavanje datoteka, odabrana je opcija dodavanja zone za „drag and drop“ datoteka, kao i opcija dodavanja tipaka za učitavanje jedne i više datoteka (Slika 4.13). Opcija „drag and drop“ odabrana je zbog jednostavnosti učitavanja fotografija s računala, dok je opcija s tipkom odabrana kako bi korisnici koji pristupaju aplikaciji putem mobilnog uređaja također mogli učitavati fotografije.

Add Photos



Multiple

Choose Files

Single

Choose File

Slika 4.13 Sučelje za dodavanje fotografija

4.9. Dodavanje funkcionalnosti sviđanja

Mnogi korisnici aplikacija za spojeve ne žele odmah slati poruku osobi koja im se sviđa. Korisnike zanima je li interes obostran prije nego što stupe u kontakt, ali bi svejedno trebali nekako dati do znanja drugome korisniku da su zainteresirani. Zbog toga je dodana funkcionalnost oznake „sviđa mi se“. Realizacija funkcionalnosti odvija se tako da pomoću tipke korisnik može označiti drugog korisnika s oznakom „sviđa mi se“, te se ID oba korisnika sprema u bazu. Nakon toga se na stranici pod kategorijom liste prikazuju svi korisnici koji su nas označili oznakom „sviđa mi se“ i tipkom možemo prebaciti na sve korisnike koje smo mi označili oznakom „sviđa mi se“. Pod pretpostavkom da će neki korisnici imati jako puno oznaka „sviđa mi se“, implementirana je i navigacija kroz stranicu koja dohvaća određen broj oznaka „sviđa mi se“ po stranici.

4.10. Slanje i primanje poruka

Jedna od najbitnijih funkcionalnosti za lakše upoznavanje korisnika je razmjena poruka. Bez ove funkcionalnosti sve ostale stvari ne bi imale smisla jer se korisnici ne bi mogli dogovoriti

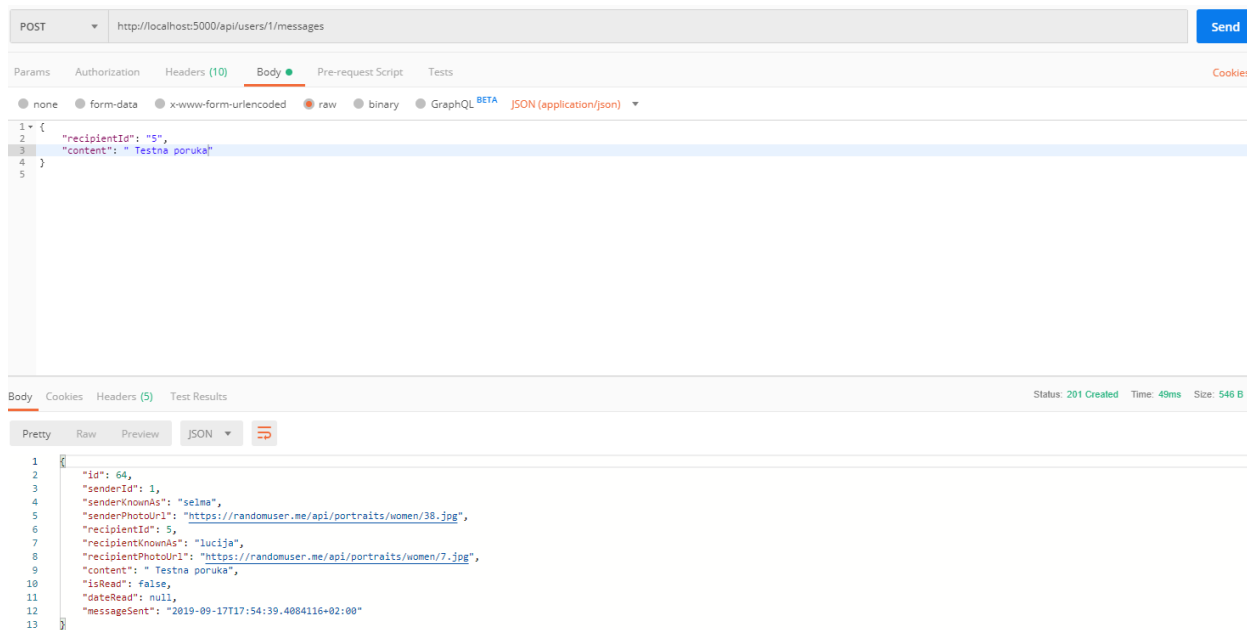
oko nalaženja ili upoznati malo bolje. Iz tog razloga potrebno je bilo implementirati ovu funkcionalnost na pristupačan i funkcionalan način. Spremanje poruka odvija se slično kao i spremanje oznake „sviđa mi se“, sprema se ID pošiljatelja i primatelja, dok se u ovom slučaju sprema i sadržaj i ostali parametri. Svaka poruka ima svoj ID kako bi se moglo točno tu poruku selektirati i obrisati od strane primatelja ili pošiljatelja. Model poruke također sadrži i svojstvo je li primatelj pročitao poruku.

5. TESTIRANJE I IZGLED WEB APLIKACIJE

U ovom poglavlju bit će prikazano kompletno sučelje aplikacije, kao i mogućnosti i funkcionalnosti aplikacije koje su dostupne krajnjem korisniku. Također će biti opisan postupak koji je korišten prilikom testiranja backend funkcija prije implementacije istih u frontend dio aplikacije

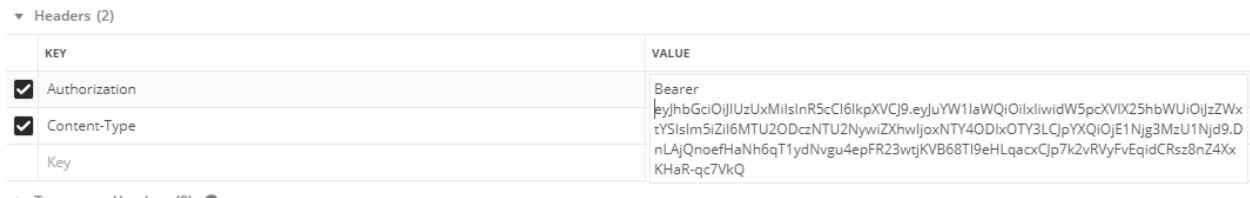
5.1. Testiranje backend dijela aplikacije

Kada razvijamo backend dio aplikacije, zna se desiti da nismo sigurni hoće li dio koda ili neka funkcija očekivano raditi. Mnoge greške i propuste ne možemo otkriti dok ne testiramo rad aplikacije i te je vrlo čest slučaj da tek nakon testiranja i prikaza odgovora aplikacije shvatimo gdje smo pogriješili i ispravimo grešku. Uz visual studio debugger, korištena je i aplikacija postman pomoću koje je testirana svaka nova funkcionalnost prije početka implementiranja frontend dijela jer nema smisla razvijati frontend dio aplikacije te tamo tražiti grešku ako je greška u backend djelu. Korištenjem postman aplikacije svaki zahtjev za obavljanje funkcije testiran je s odgovarajućim parametrima prije početka izrade korisničkog sučelja. Slika 5.1 pokazuje testiranje slanja korisnika, te odgovor aplikacije s podacima koji će biti stvoreni u bazi podataka.



Slika 5.1 Prikaz testiranja slanja poruka

Za svako testiranje u aplikaciji postman potrebno je postaviti odgovarajuće parametre. Gotovo sve funkcionalnosti osim registracije korisnika zahtijevaju da korisnik bude prijavljen u aplikaciju, pa je zbog toga prije testiranja potrebno izgenerirati token za autorizaciju korisnika, te ga upisati u polje za autorizaciju (Slika 5.2). Token dobivamo kao odgovor testiranja prijave korisnika te ga kopiramo u parametre svakog sljedećeg testa kako bi zaobišli autorizaciju.

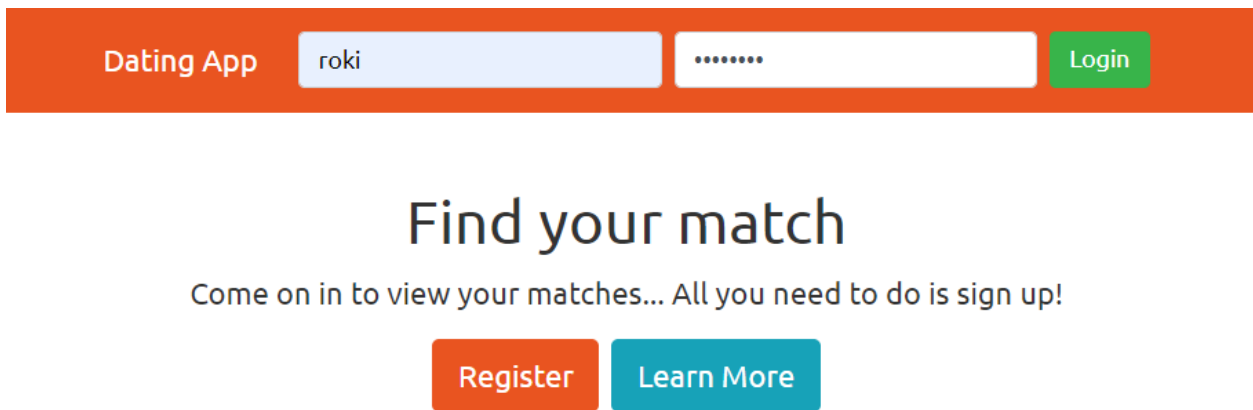


KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Bearer
<input checked="" type="checkbox"/> Content-Type	eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJ1bW1laWQiOiIxliwidW5pcXVIX25hbWUiOiJzZWxtYSlm5iZiI6MTU2ODczNTU2NywiZXhwIjoxNTY4ODIxOTY3LCJpYXQiOiE1Njg3MzU1Njd9.DnLajQnoeffHaNh6qT1ydNvgu4epFR23wtjKVB68TI9eHLqacxCjp7k2vRyFvEqidCRsz8nZ4XxKHAR-qc7VkJQ
Key	

Slika 5.2 Prikaz unosa tokena za autorizaciju prije testiranja

5.2. Izgled web aplikacije

Početna stranica (Slika 5.3) je dizajnirana jednostavno i kompaktno. Korisnik ima mogućnost unosa korisničkog imena i lozinke ako se želi prijaviti u aplikaciju. Ako korisnik nema račun, pritiskom na tipku za registraciju može napraviti novi račun.



Slika 5.3 Početna stranica aplikacije

Stranica za registraciju korisnika(Slika 5.4) dio je početne stranice, te je na ovoj stranici moguć unos podataka i registracija korisnika. Na početku registracije nema nikakvih upozorenja, ali nakon što pokušamo izmijeniti neko polje i prebacimo se na sljedeće, ako uneseni podaci ne odgovaraju zahtjevima aplikacije, aplikacija će onemogućiti pritisak tipke za registraciju, zacrveniti polje koje ne zadovoljava kriterije aplikacije i ispisati ispod polja koji su kriteriji prekršeni.

Dating App Username Password Login

Sign Up

I am a: Male
 Female

Dario

Known as ✖
Known as is required

09/08/1995

City ✖
City is required

Country ✖
Country is required

... ✖
Password must be at least 4 characters

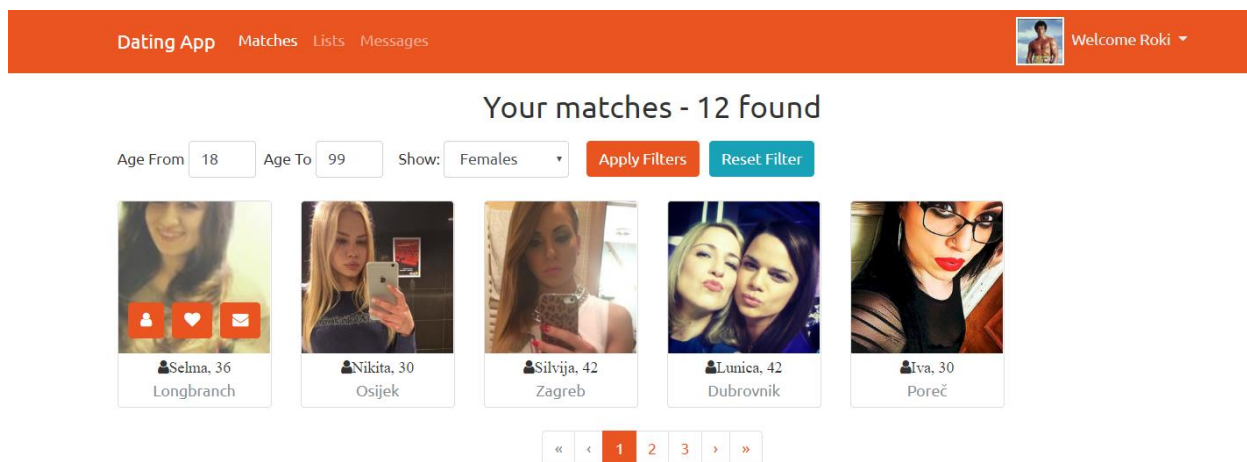
• ✖
Passwords must match

Register Cancel

Slika 5.4 Prikaz registracije korisnika

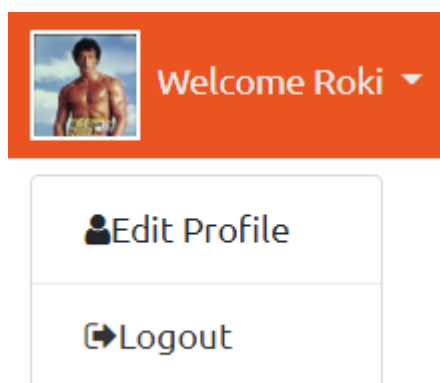
Nakon registracije korisnika, korisnik je automatski prijavljen u aplikaciju i preusmjeren na početnu stranicu.

Kada je korisnik prijavljen, početna stranica je stranica s pronađenim potencijalnim korisnicima za upoznavanje(Slika 5.5). Na ovoj stranici možemo filtrirati korisnike aplikacije, klikom na korisnika otići na profil, poslati mu poruku ili označiti korisnika s oznakom „sviđa mi se“. Ako nam se nakon filtriranja prikaže više korisnika, na dnu imamo popis stranica, a prikazujemo pet korisnika po stranici



Slika 5.5 Stranica pronađenih korisnika

Korisnik također u bilo kojem trenutku dok je prijavljen može kliknuti na padajući izbornik i odabrati opciju uređivanja vlastitog profila ili odjave s aplikacije(Slika 5.6).

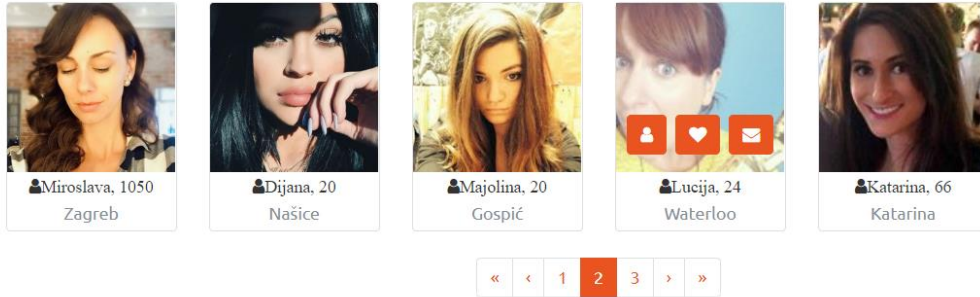


Slika 5.6 Padajući izbornik za odjavu i uređivanje profila

Stranica s listom korisnika(Slika 5.7) pruža uvid u korisnike koji su naš profil označili s oznakom „sviđa mi se“ i korisnike koje smo mi označili tom oznakom. Za navigaciju između ove dvije opcije postoje dvije tipke, dok za navigaciju popisa stranica korisnika također imamo izbornik na dnu stranice.

Members who I've Liked : 12

Members who like me Members who I like








Slika 5.7 Lista korisnika

Ako pokušamo označiti nekog korisnika sa „sviđa mi se“ kojeg smo već ranije označili istom, responzivna notifikacija će nam javiti da smo tog korisnika već označili oznakom „sviđa mi se“.

Pregled nepročitanih poruka obavlja se na stranici poruka (Slika 5.8), gdje imamo tri tipke. Prva je automatski odabrana i prikazuje nepročitane poruke, dok druge dvije tipke prikazuju sve primljene poruke i sve poslane poruke. Na ovoj stranici je također moguće brisati poruke, dok to nije moguće odlaskom na nečiji profil pod tabom poruka.

Unread Inbox Outbox

Message	From / To	Sent / Received	
Da haha	 Nikita	a day ago	Delete
Donji grad. Odakle si ti ?	 Nikita	a day ago	Delete
Pozdrav !	 Nikita	a day ago	Delete
Može! kada i gdje?	 Silvija	a day ago	Delete
Hvala ti ! nisi ni ti loš	 Silvija	a day ago	Delete

« < 1 2 > »

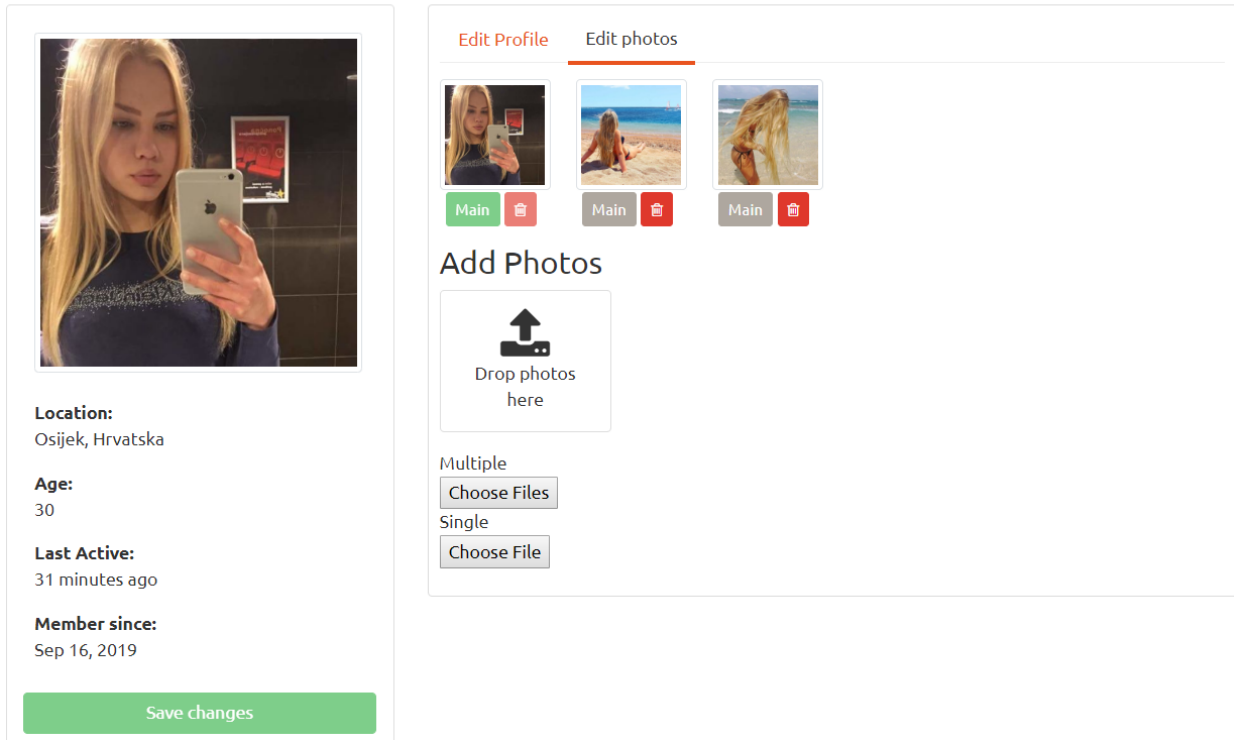
Slika 5.8 Poruke

Ako korisnik želi izmijeniti informacije koje je naveo prilikom registracije, u padajućem izborniku može odabrati izmjenu profila(Slika 5.9). Na ovoj stranici su editabilna polja kao što su opis korisnika, grad i država u kojoj živi i ostale informacije.

Slika 5.9 Izmjena profila

Uz izmjenu profila, sljedeći tab nudi dodavanje i brisanje fotografija(Slika 5.10), te postavljanje jedne od učitanih fotografija primarnom.

Your Profile




The screenshot shows a user's profile page. On the left, there is a large profile picture of a woman with long blonde hair holding a smartphone. Below it, the following information is displayed: **Location:** Osijek, Hrvatska; **Age:** 30; **Last Active:** 31 minutes ago; **Member since:** Sep 16, 2019. At the bottom of this section is a green button labeled "Save changes".

On the right, there is a "Edit photos" section. It has two tabs: "Edit Profile" and "Edit photos", with "Edit photos" being the active tab. Below the tabs are three photo thumbnails. The first thumbnail shows the same woman as the profile picture and has a green "Main" button and a red trash icon. The second and third thumbnails show her on a beach and have grey "Main" buttons and red trash icons. Below the photos is an "Add Photos" section with a large upload icon and the text "Drop photos here". At the bottom of this section are two options: "Multiple" with a "Choose Files" button, and "Single" with a "Choose File" button.

Slika 5.10 Manipulacija fotografijama

Korisnik ne može obrisati fotografiju koja je postavljena kao primarna, tako da ako želimo obrisati sve fotografije, moramo učitati neku nevažnu sliku i postaviti je kao primarnu kako bismo mogli obrisati sliku koja je prethodno bila primarna.

Klikom na profil nekog korisnika, prikazuje se korisnički profil s nekoliko tabova. Zadani tab koji se prvi učitava je tab s osnovnim informacijama korisnika(Slika 5.11). Ovdje možemo saznati opis korisnika i informaciju što korisnik traži i očekuje.



Location:
Lola, Zimbabwe

Age:
45

Last Active:
3 days ago

Member since:
Mar 4, 2017

Like Message

About lola Interests Photos Messages

Description

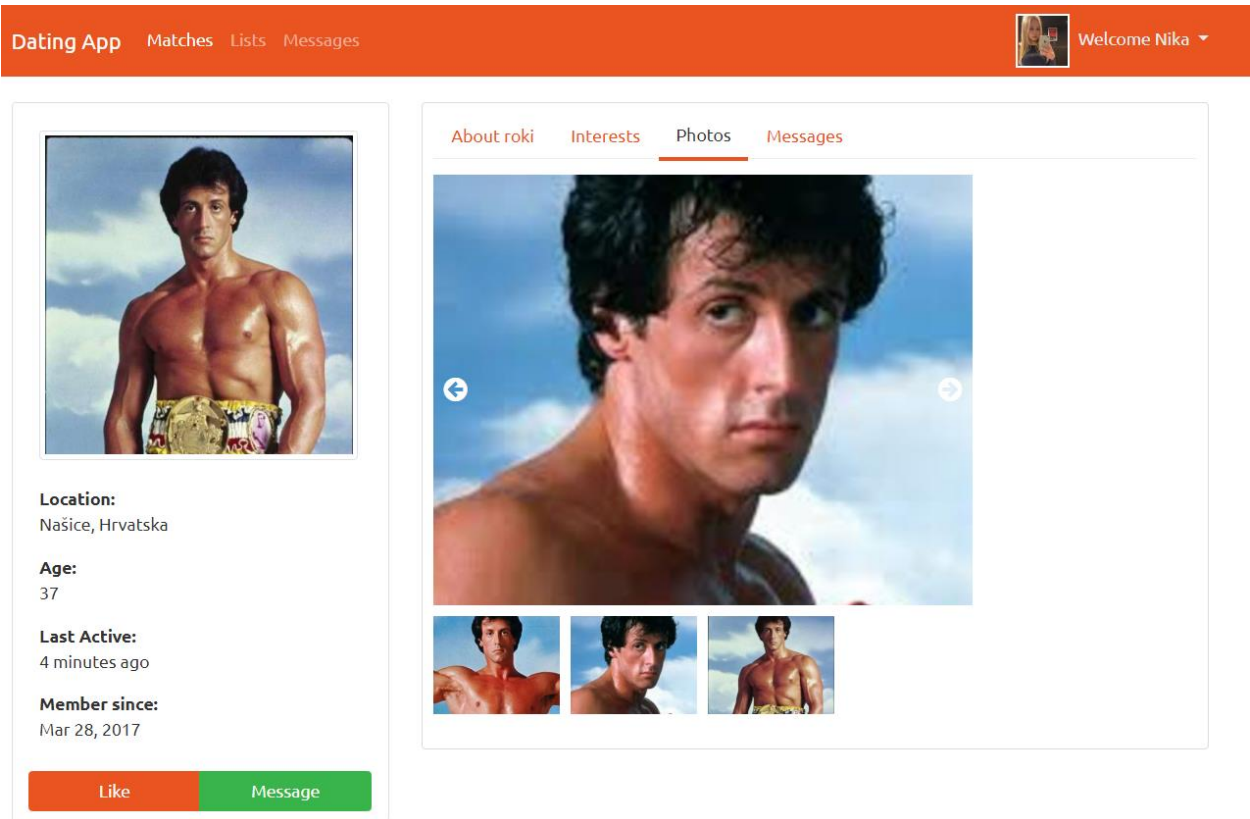
Eiusmod est magna tempor eu quis commodo dolor. Id cupidatat ut mollit do. Voluptate cillum magna est do ut consectetur id nulla est. Tempor exercitation consequat sint officia in deserunt ipsum labore mollit labore exercitation exercitation.

Looking For

Minim consequat incididunt aliqua sint adipisicing anim reprehenderit qui. Eiusmod et veniam do nulla elit dui tempor ut commodo nisi exercitation adipisicing minim dui. Commodo non do in sit pariatur reprehenderit nisi.

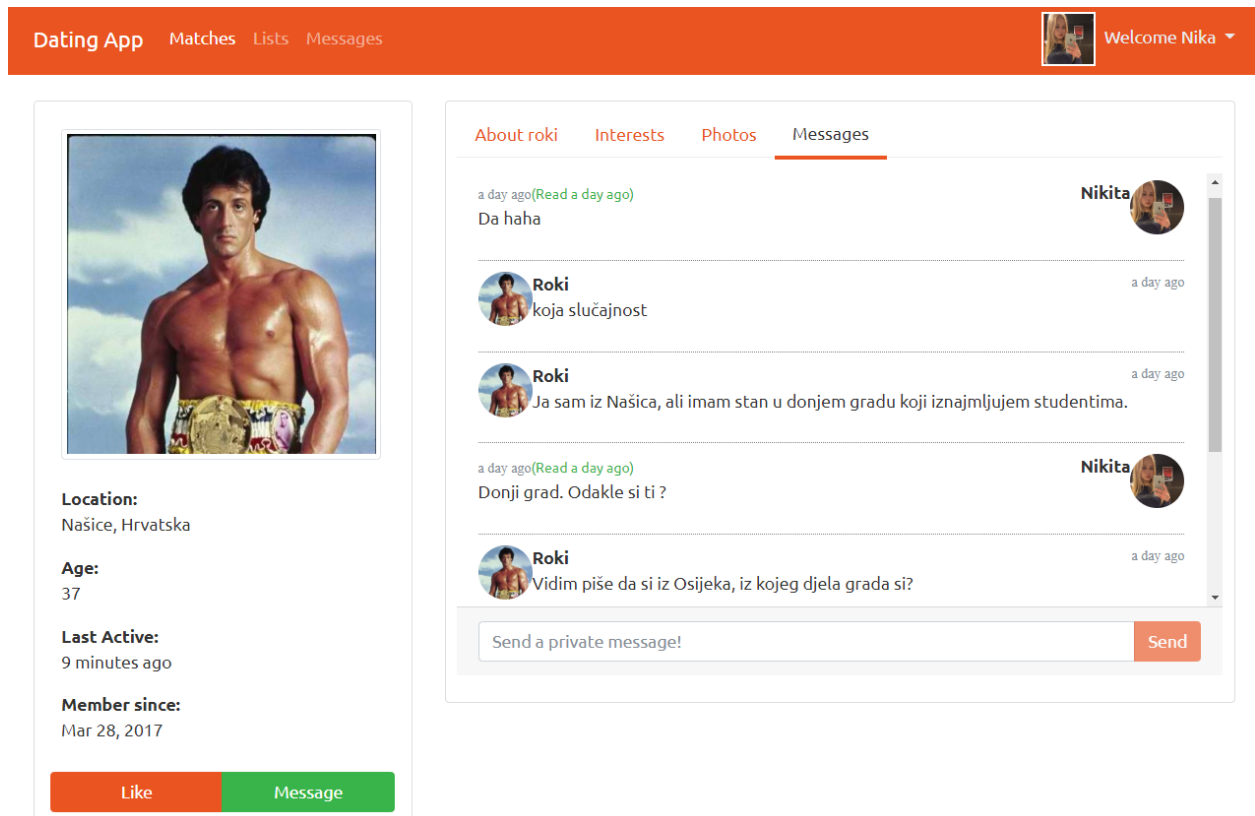
Slika 5.11 Tab opisa korisnika

Sljedeći tab predstavlja korisnikove interese, koji su također u tekstualnom obliku. Treći tab je namijenjen za pregled fotografija korisnika. U njemu se nalazi galerija slika koje možemo listati ako korisnik kojemu gledamo profil ima više od jedne slike(Slika 5.12).



Slika 5.12 Tab galerije fotografija

I posljednji tab i onaj najkorisniji je tab za dopisivanje s korisnikom(Slika 5.13). U ovom tabu se prikazuje razgovor između prijavljenog korisnika i korisnika čiji je profil posjećen. Svaka poruka je kronološki poredana te sadrži informacije je li korisnik pročitao poruku, kada je poslana i pored poruke je slika pošiljatelja.



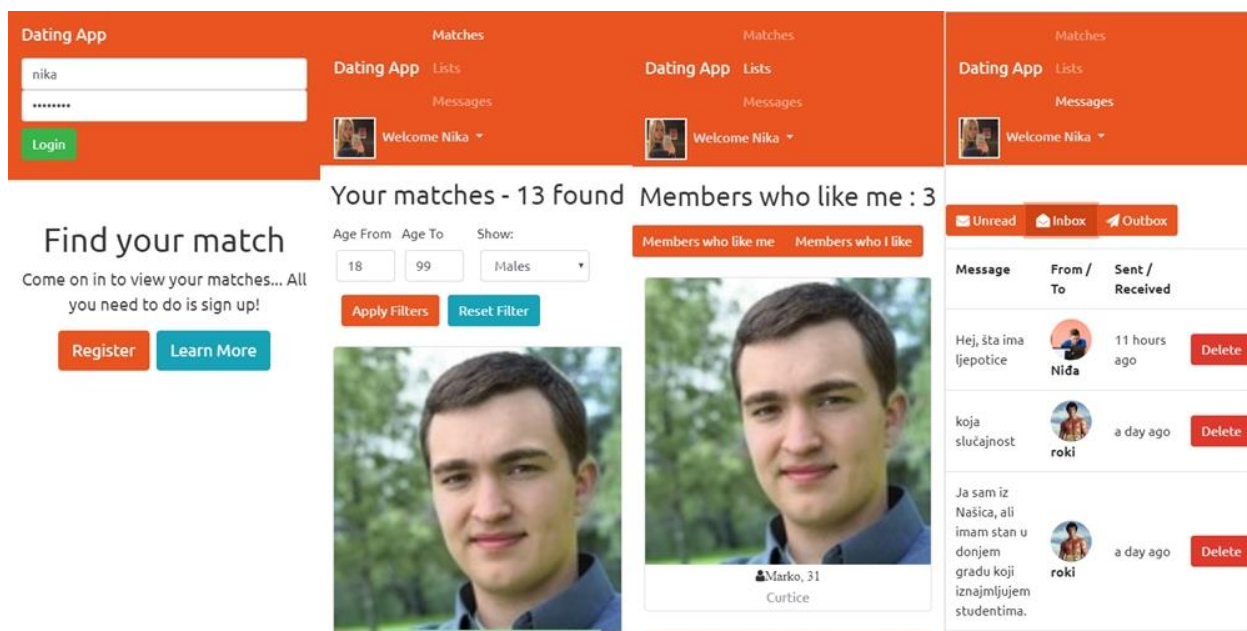
Slika 5.13 Tab poruka

Kada pošaljemo poruku, poslana poruka se responzivno kronološki prikaže na vrhu. Za razliku od ostalih aplikacija, u ovoj aplikaciji su najnovije poruke na vrhu razgovora, a postoji i traka za pomicanje sa strane kako bismo mogli pročitati i starije poruke. Na dnu taba nalazi se polje za upis poruke te tipka za slanje iste.

Ova stranica također sadrži informacije o lokaciji, starosti korisnika, informaciji kada je korisnik zadnji put bio aktivan te kada je napravio korisnički račun. Na dnu također postoje dvije tipke koje služe za oznaku „sviđa mi se“ i slanje poruka. Pritiskom na dugme poruke, korisnik se prebacuje na tab s porukama.

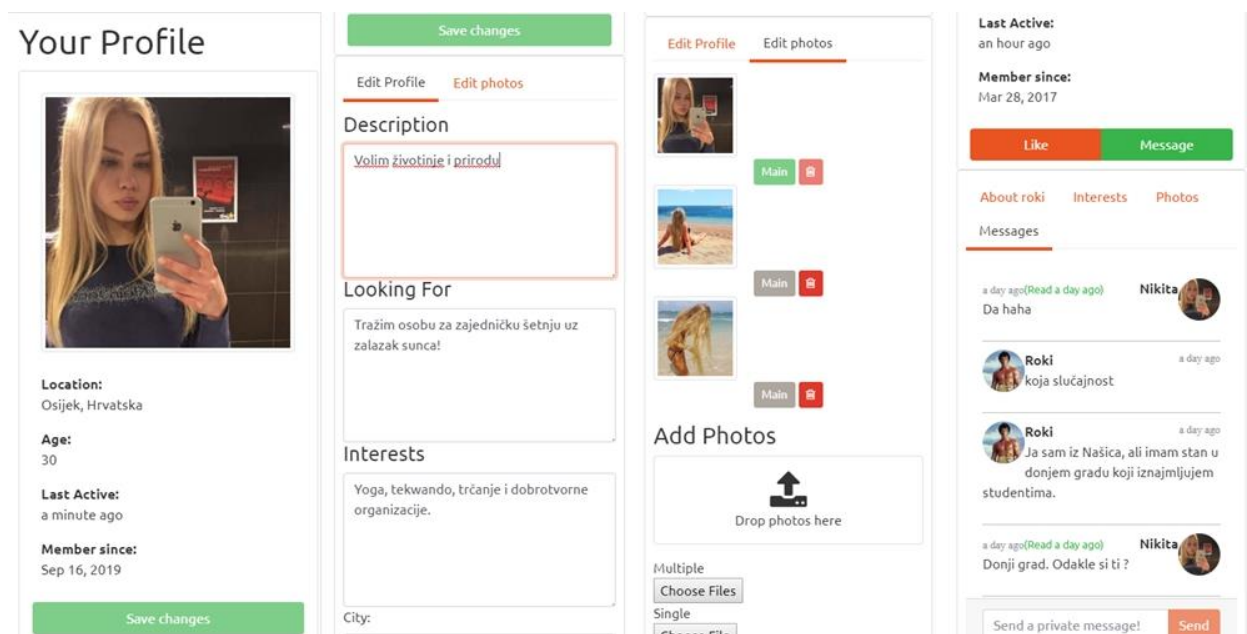
5.3. Mobilni prikaz aplikacije

U današnje vrijeme puno više korisnika koristi svoj mobilni uređaj za surfanje, komunikaciju i korištenje aplikacija poput ove. Iz tog razloga potrebno je bilo ovu aplikaciju prilagoditi za mobilni prikaz(Slika 5.14)



Slika 5.14 Mobilni prikaz primarnih stranica

Dopisivanje je također moguće preko mobilnog uređaja, kao i uređivanje profila i dodavanje fotografija. Treba samo pomicati stranicu dok ne dođemo do djela koji želimo koristiti.



Slika 5.15 Mobilni prikaz uređivanja profila, dodavanja fotografija i dopisivanja

6. ZAKLJUČAK

U ovom diplomskom radu razvijena je web aplikacija za upoznavanje partnera. Cilj web aplikacije bio je pružiti korisnicima barem osnovne funkcionalnosti i alate za komunikaciju kako bi korisnici mogli upoznati jedni druge i komunicirati jedni s drugima u što boljem okruženju uz što bolje performanse i izgled. Razvijena je aplikacija s korisničkim sučeljem koje u potpunosti sadržava funkcionalnosti kao što su registracija korisnika, prijava korisnika u aplikaciju, komunikaciju između korisnika, uređivanje podataka korisnika, učitavanje slika korisnika te filtriranje drugih korisnika za prikaz. Aplikacija ima zadovoljavajuću razinu zaštite podataka i korisničkih računa, te je namijenjena za veliku količinu korisnika. Aplikacija je u potpunosti funkcionalna, ali ima poprilično puno prostora za nadogradnju aplikacije.

LITERATURA

- [1] – Angular, [https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework)) (posljednji pristup – rujan 2019.)
- [2] HTML, <https://www.w3schools.com/html/> (posljednji pristup – rujan 2019.)
- [3] CSS, <https://www.w3schools.com/css/> (posljednji pristup – rujan 2019.)
- [4] NGX-BOOTSTRAP, <https://valor-software.com/ngx-bootstrap/#/> (posljednji pristup – rujan 2019.)
- [5] ASP.NET Core, <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2> (posljednji pristup – rujan 2019.)
- [6] Entity Framework, <https://docs.microsoft.com/en-us/ef/> (posljednji pristup – rujan 2019.)
- [7] Visual Studio Code, https://en.wikipedia.org/wiki/Visual_Studio_Code (posljednji pristup – rujan 2019.)
- [8] Cloudinary, <https://cloudinary.com/documentation> (posljednji pristup – rujan 2019.)
- [9] Postman, <https://www.getpostman.com/about-postman> (posljednji pristup – rujan 2019.)
- [10] Code first, <https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/new-database> (posljednji pristup – rujan 2019.)
- [11] Angular CLI, <https://angular.io/cli/> (posljednji pristup – rujan 2019.)
- [12] JWT Token, <https://jwt.io/introduction/> (posljednji pristup – rujan 2019.)
- [13] JWT izgled tokena, <https://jwt.io> (posljednji pristup – rujan 2019.)
- [14] AlertifyJS, <https://alertifyjs.com/> (posljednji pristup – rujan 2019.)
- [15] Ng2-file-upload, <https://valor-software.com/ng2-file-upload/> (posljednji pristup – rujan 2019.)

SAŽETAK

Tema ovog diplomskog rada je web aplikacija za upoznavanje partnera. Svrha aplikacije je pružiti korisniku jednostavno korisničko sučelje s bogatim funkcionalnostima i zadovoljavajućim performansama kako bi korisnik što lakše uspio pronaći partnera. Aplikacija omogućuje korisniku većinu tih funkcionalnosti i pruža zadovoljavajuće iskustvo, te se može koristiti za upoznavanje partnera ako dosegne veliki broj registracija korisnika na aplikaciju. Angular programski okvir u suradnji s ASP.NET Core platformom pokreće aplikaciju koja je prvobitno razvijena za računala, ali podržava i pristup preko pretraživača mobilnog uređaja.

Ključne riječi: Angular, Aplikacija za upoznavanje partnera, ASP.NET Core

ABSTRACT

The subject of this final paper is a Dating application. The purpose of the application is to provide the user with a simple user interface with rich functionality and satisfactory performance to make it easier for the user to find a partner. The application provides the user with most of these features and provides a satisfying experience and can be used to meet partners if they reach large number of user registrations on the application. The Angular framework, in conjunction with the ASP.NET Core platform, runs an application that was originally developed for computers but also supports access through a mobile browser.

Key words: Angular, ASP.NET Core, Dating application

ŽIVOTOPIS

Mario Tomić rođen je 29.2.1996. u Virovitici. Osnovnu školu pohađao je u Đulovcu. Nakon završene Osnovne škole, upisuje se u Tehničku školu Daruvar i završava četverogodišnji smjer Računarstvo. Na Otvorenom učilištu Algebra polaže ispit i stječe Microsoft Certifikat iz područja Windows Operating System Fundamentals. Obrazovanje nastavlja na Fakultetu elektrotehnike i informacijskih tehnologija u Osijeku, na preddiplomskom studiju Računarstva. Nakon uspješnog završetka preddiplomskog studija računarstva na Fakultetu elektrotehnike i informacijskih tehnologija, dobiva zvanje prvostupnik (baccalaureus) inženjer računarstva (univ. bacc. ing. comp.) te upisuje diplomski studij Računarstva, na smjeru DRD- informacijske i podatkovne znanosti. Stručnu praksu je odrađivao u tvrtki GDI, koja se bavi pružanjem programskih rješenja za upravljanje resursima, imovinama i operacijama u telekomunikacijama, energetici, javnoj upravi, poljoprivredi i prehrambenoj industriji, prirodnim resursima, okolišu, zaštiti i sigurnosti te komercijalnim industrijama. Nakon odrađene prakse postao je zaposlenik tvrtke GDI na poziciji software programera.

Mario Tomić,

Potpis: _____.

PRILOZI

Prilog 1. Datoteka pisanog dijela rada (docx i pdf)

Prilog 2. Programski kod aplikacije