Web aplikacija za istraživanje javnog mnijenja

Bagarić, Petar

Master's thesis / Diplomski rad

2018

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:200:252902

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-04**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

WEB APLIKACIJA ZA ISTRAŽIVANJE JAVNOG MNIJENJA

Diplomski rad

Petar Bagarić

Osijek, 2018.

Sadržaj

1. UVOD 3
   1.1. Zadatak diplomskog rada 3
2. KORIŠTENE TEHNOLOGIJE 4
   2.1. .NET framework 4
   2.2. Entity framework 4
   2.3. Angular 4 5
   2.4. HTML 5
   2.5. CSS 5
   2.6. Bootstrap 6
3. PROGRAMSKO RJEŠENJE 7
   3.1. Entity framework – *code first* pristup 7
   3.2. Repozitoriji 10
   3.3. Ubacivanje ovisnosti 10
   3.4. Autentikacija i autorizacija 11
      3.4.1. JSON web tokeni 11
      3.4.2. Testiranje JWT 12
4. KORIŠTENJE WEB APLIKACIJE 16
   4.1. Korisnik s funkcijom *ispitivača* 18
   4.2. Korisnik s funkcijom *ispitanika* 24
   4.3. Korisnik s funkcijom *administratora* 26
5. ZAKLJUČAK 28

LITERATURA 29

SAŽETAK 30

ABSTRACT 31

ŽIVOTOPIS 32

PRILOZI 33

1. UVOD

Živimo u vremenu gdje je postalo vrlo zahtjevno donijeti jednostavnu odluku pri nekakvoj kupovini ili odabiru bilo kakvog proizvoda, a jedan od glavnih razloga je to što ih je jako puno. Uz svaki proizvod postoji još veliki broj drugih koji su gotovo pa identični, te je teško pronaći koji bi najbolje zadovoljio naše potrebe, uz pristupačnu i razumnu cijenu. Kompanije pokušavaju konkurirati drugima pokušavajući napraviti što je moguće bolji proizvod, a pri tome im uvelike može pomoći povratna informacija korisnika. Navedeno je samo jedna od koristi anketa, koje danas igraju veliku ulogu u istraživanjima društva, marketinga, službenih statistika i slično. Iz ovog razloga potrebno je imati dobar sustav za izradu i popunjavanje anketa. Ankete koje se popunjavaju online su brže, jednostavnije i jeftinije nego tradicionalne ankete i lako se može prikupiti velik broj odgovora u kratkom vremenu. Cilj ovog rada je izraditi web aplikaciju koja će omogućiti jedan takav sustav, gdje će izrada anketa biti brza i efikasna, a njihovi rezultati biti vizualno prikazani.

U prvom dijelu detaljnije će biti opisane tehnologije korištene pri izradi programskog rješenja.

U drugom dijelu rada biti će opisano programsko rješenje, opisujući korištene tehnologije i njihovu primjenu kronološkim slijedom kako je tekla izrada programskog rješenja.

Treći dio sadrži upute i način na koji se koristi realizirana web aplikacija za svaku vrstu korisnika koju aplikacija razlikuje.

Na kraju dolazi zaključak u kojem će biti osvrt na ciljeve postavljane u ovom radu, postignutim programskim rješenjem, te njegovim mogućim unaprijeđenjima.

1.1. Zadatak diplomskog rada

Izraditi sustav za istraživanje javnog mnijenja koji omogućava izradu anketa, njihovo popunjavanje kao i analizu prikupljenih podataka. Sustav treba razlikovati tri vrste korisnika: administratora, ispitivača i ispitanika. Osim same prijave korisnika, sustav treba omogućiti i izradu anketa, pregled dostupnih anketa, ispunjavanje anketa te grafički prikaz rezultata prema različitim kriterijima.

2. KORIŠTENE TEHNOLOGIJE

2.1. .NET framework

.NET Framework je okruženje za razvoj softvera razvijen od strane tvrtke Microsoft koji se prvenstveno koristi na Microsoft Windowsu. Sadrži veliku biblioteku klasa koja se zove „FrameworkLock Library“ (FCL) i pruža jezičnu univerzalnost (svaki jezik može koristiti kod koji je napisan na drugim jezicima) na nekoliko programskih jezika. Programi napisani za .NET Framework izvršavaju se u softverskom okruženju (za razliku od hardverskog okruženja) pod nazivom Common Language Runtime (CLR), aplikacijski virtualni stroj koji pruža usluge kao što su sigurnost, upravljanje memorijom i upravljanje iznimkama. FCL i CLR zajedno čine .NET Framework.[1]

Osnovu .NET-a predstavlja .NET Framework. To je sustav koji nadograđuje mogućnosti samog operativnog sustava. Radi se o posebnoj infrastrukturi koja programerima nudi gotova rješenja i funkcionalnosti da bi ubrzala i pojednostavila razvoj aplikacija svih vrsta i oblika. [1]

2.2. Entity framework

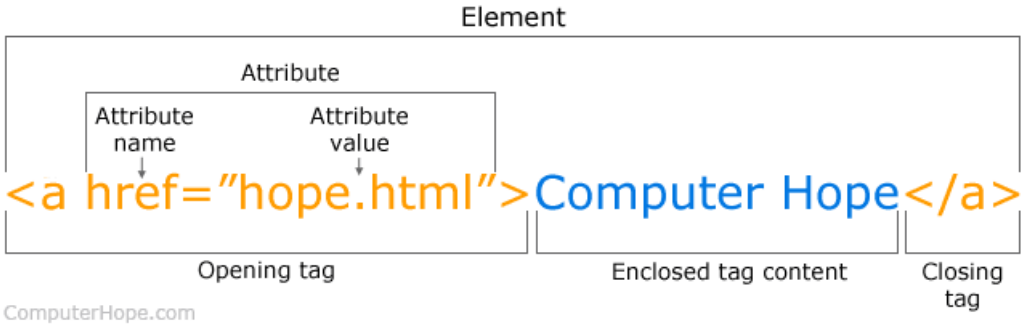
Razvojna okolina entiteta (engl. e*ntity framework*) je set tehnologija u ADO.NET-u koji podržava razvijanje podatkovno orijentiranih softverskih aplikacija.[2] Arhitekti i programeri podatkovno orijentiranih aplikacija su obično imali problema s potrebom postizanja dva jako različita cilja. Trebali su modelirati entitete, veze i poslovnu logiku problema koji su rješavali, te su također morali koristiti podatkovne mehanizme za spremanje i dohvaćanje podataka. Podaci su mogli biti rasprostranjeni na više sustava za spremanje, svaki sa svojim protokolom, čak i aplikacije koje su radili s jednim sustavom za spremanje su morale balansirati zahtjeve sustava za spremanje i zahtjeva za pisanje učinkovitog i održivog programskog koda. Razvojna okolina entiteta omogućava programerima raditi s podacima u obliku domensko specifičnih objekata i svojstava, bez razmišljanja o tablicama baze podataka i gdje se podaci spremaju. S razvojnom okolinom entiteta programeri mogu raditi s većim razinama apstrakcije s podacima i mogu izrađivati podatkovno orijentirane aplikacije s manje koda nego u tradicionalnim aplikacijama.

2.3. Angular 4

Angular 4 je JavaScript razvojna okolina za razvijanje web aplikacija i aplikacija u JavaScript-u, HTML i TypeScript, koji je nadskup JavaScript-a. Angular pruža ugrađene mogućnosti za animacije, http servise i materijale koji imaju značajke poput automatskog završavanja teksta, navigacije, menija itd. Kod se piše u Typescriptu, koji se prevodi u JavaScript i prikazuje na web pregledniku. Najčešće se koristi za izrađivanje dinamičnih web aplikacija, uz korištenje standardnog HTML-a i CSS-a.

2.4. HTML

HTML je računalni jezik koji se koristi za kreiranje web stranica. Osnova je svih web stranica na Internetu. HTML je kratica za HyperText Markup Language, gdje HyperText predstavlja hiperveze tj. konekcije do ostalih stranica. Svaka web stranica na internetu je napisana korištenjem neke od verzija HTML-a. U ovom projektu korištena je verzija HTML 5.



Sl. 2.1. Primjer HTML taga [3]

2.5. CSS

CSS je kratica od Cascading Style Sheets, te predstavlja stilski jezik koji se koristi za oblikovanje i uređivanje dokumenata napisanih u prezentacijskim jezicima, od kojih je najčešće korišten HTML.. CSS je nezavisan od HTML-a i obično se razdvaja od njega kako bi bilo lakše održavati stranice i dijeliti ga među stranicama. CSS skripta definira pravila stiliziranja nekog HTML elementa ili skupine istih elemenata preko njihovih tzv. identifikatora. Nakon definiranja identifikatora slijede vitičaste zagrade unutar kojih se navodi željeni stil tog elementa ili skupine elemenata. Primjer se može vidjeti na slici 2.2., gdje je u CSS skripti definiran željeni stil za HTML tag <h1>.

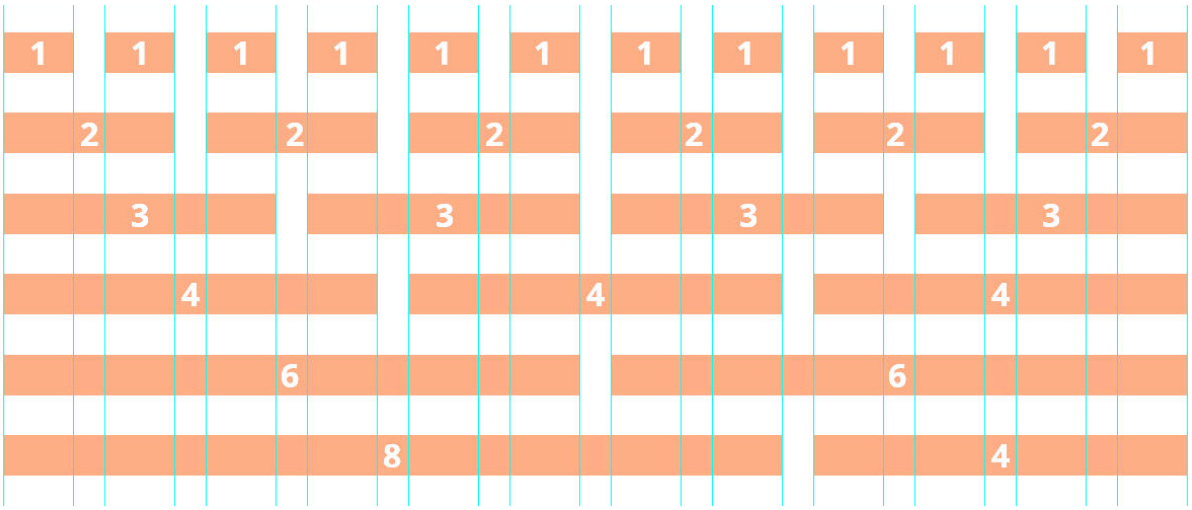
```
h1 {
    color: white;
    text-align: center;
}
```

Sl. 2.2. Primjer CSS koda

2.6. Bootstrap

Bootstrap je razvojna okolina na klijentskoj strani za HTML, CSS i JavaScript koji se koristi za razvoj mobilnih i responzivnih web stranica. Pod responzivnim se misli na to da će web stranice biti optimizirane da stanu na ekran uređaja koji se koristi, bilo to desktop računalo, tablet ili mobitel.

Jedan od ključnih dijelova Bootstrap razvojnog okruženja je njegov sustav rešetke (engl. *grid*). Rešetka predstavlja okvir stranice koristeći više redova i stupaca u koje se umeće sadržaj. Kako se uređaj na kojem se pregledava stranica povećava, tako se rešetka prilagođava i skalira do 12 stupaca, a isto vrijedi i za smanjivanje stranice, rešetka će se proporcionalno prilagoditi.

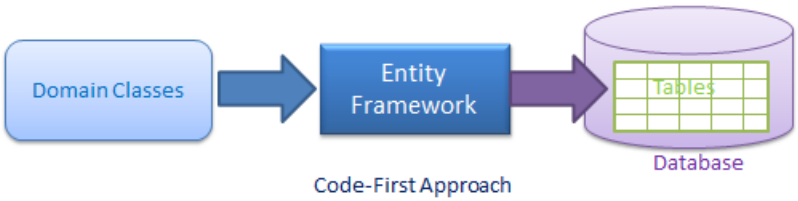


Sl. 2.3. Prikaz rešetke s 12 stupaca kod Bootstrap-a

3. PROGRAMSKO RJEŠENJE

3.1. Entity framework – *code first* pristup

Kod „prvo kod“ (*engl. code first*) pristupa je naglasak na domenskim modelima aplikacije i kreiranju klasa koje odgovaraju domenskim entitetima, umjesto da se prvo dizajnira cijela baza podataka i onda posebno klase koje odgovaraju dizajnu baze. *Code first* pristup namijenjen je za bazu koja ne postoji, već će ju on kreirati. On se također može koristiti ako je baza podataka prazna, te će tada *code first* pristup dodati odgovarajuće tablice u nju. Model koji se definira kodira se u C# klasama. Ako je potrebna nekakva dodatna konfiguracija, ona se može naznačiti na klasama ili svojstvima koristeći odgovarajuće atribute. *Code first* pristup je također jako pogodan ako se modeli mijenjaju, jer on automatski ažurira tablice u bazi. Ovo se može postići korištenjem *code first* migracija i one su bile korištene više puta u ovom projektu, budući da se baza podataka mijenjala kroz cijelu izradu. Instalacija razvojnog okruženja za entitete je vrlo jednostavna, te se ona može odraditi vrlo lako pomoću upravitelja nuget paketa (engl. *nuget package manager*).



Sl. 3.1. „Code-first“ pristup [4]

Kreiranje tablica u „code first“ pristupu se vrši kroz dodavanje C# klasa u kojima su opisani stupci za koje se želi da ta tablica sadrži, gdje se za svaki stupac naznačuje kakvoga je on tipa. Svi ovi stupci predstavljaju jedan redak u tablici. Primjer ovakve klase je prikazan na slici 3.2.

```
public class Answer
{
    public Guid Id { get; set; }
    public Guid? QuestionOptionId { get; set; }
    public Guid QuestionId { get; set; }
    public string AspNetUserId { get; set; }

    public string Text { get; set; }

    public virtual Question Question { get; set; }
    public virtual QuestionOption QuestionOption { get; set; }
}
```

Sl. 3.2. Primjer code first klase

Kontekst klasa predstavlja virtualnu reprezentaciju baze podataka u obliku jednog objekta. Korištenjem operacija nad tim objektom tj. kontekstom baze se automatski događaju promjene i u samoj bazi podataka. U konstruktoru same klase se obično navodi ime veze koju ASP.NET koristi za povezivanje s bazom. Primjer kontekst klase prikazan je na slici 3.3.

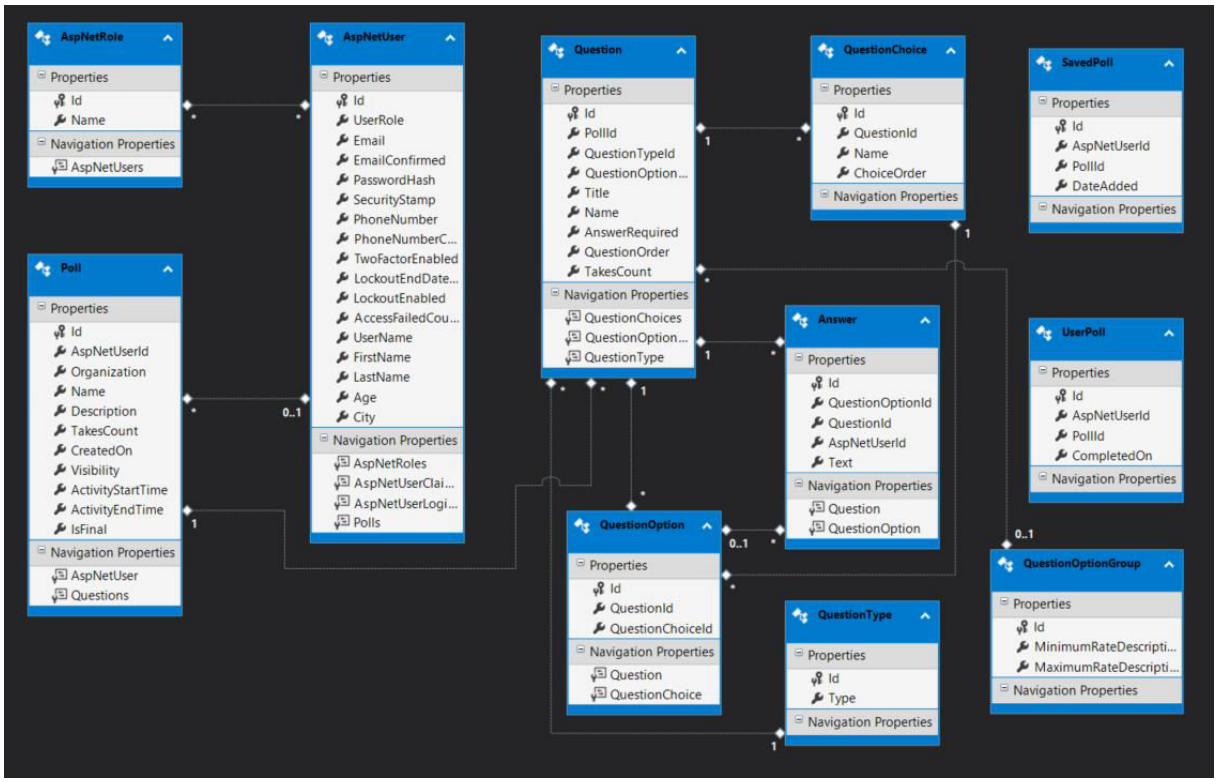
```
public class SurveyDatabaseContext : DbContext, ISurveyContext
{
  static SurveyDatabaseContext()
  {
    Database.SetInitializer<SurveyDatabaseContext>(null);
  }

  public SurveyDatabaseContext()
    : base("Name=SurveyConnString") { }

  public DbSet<AspNetRole> AspNetRoles { get; set; }
  public DbSet<AspNetUserClaim> AspNetUserClaims { get; set; }
  public DbSet<AspNetUserLogin> AspNetUserLogins { get; set; }
  public DbSet<AspNetUser> AspNetUsers { get; set; }
  public DbSet<Poll> Polls { get; set; }
  public DbSet<Answer> Answers { get; set; }
  public DbSet<Organization> Organizations { get; set; }
  public DbSet<Question> Questions { get; set; }
  public DbSet<QuestionOption> QuestionOptions { get; set; }
  public DbSet<QuestionChoice> QuestionChoices { get; set; }
  public DbSet<QuestionOptionGroup> QuestionOptionGroups { get; set; }
  public DbSet<UserPoll> UserPolls { get; set; }
  public DbSet<SavedPoll> SavedPolls { get; set; }

  protected override void OnModelCreating(DbModelBuilder modelBuilder)
  {
    modelBuilder.Configurations.Add(new AspNetRoleMap());
    modelBuilder.Configurations.Add(new AspNetUserClaimMap());
    modelBuilder.Configurations.Add(new AspNetUserLoginMap());
    modelBuilder.Configurations.Add(new AspNetUserMap());
    modelBuilder.Configurations.Add(new AnswerMap());
  }
}
```

Sl. 3.3. Primjer kontekst klase



Sl. 3.4. Model baze

Objašnjene tablice sa slike 3.4.

AspNetRole – ova tablica sadrži funkcije korisnika, a u ovoj aplikaciji su to *administrator*, *ispitivač* i *ispitanik*

AspNetUser – u ovoj tablici su sadržani registrirani korisnici. Povezana je s tablicom AspNetRole, te se tako može provjeriti funkcija pojedinog korisnika

Poll – tablica s anketama, povezana vezom *jedan-prema-više* s tablicom Question, budući da jedna anketa može imati više pitanja

Question – tablica s pitanjima. Povezana vezom *jedan-prema-više* s opcionalnom tablicom QuestionChoice jer jedno pitanje može imati više ponuđenih izbora (kod *radiogroup* i *checkbox* pitanja)

QuestionChoice – ponuđeni izbori kod pitanja koja su tipa *radiogroup* i *checkbox*

QuestionType – tablica s vrstama pitanja, a to su *text*, *radiogroup*, *checkbox* i *rating*

QuestionOption – tablica koja povezuje tablice Question i QuestionChoice

QuestionOptionGroup – tablica s opcionalnim grupama, služi za pohranjivanje donje i gornje granice odgovora kod *rating* pitanja

SavedPoll – u ovoj tablici se spremaju ankete koje su privremene, tj. one koje nisu konačne, već služe kako bi korisnik nastavio s njihovim izrađivanjem u budućnosti

UserPoll – tablica koja služi za prepoznavanje da li je korisnik već ispunio neku anketu

3.2. Repozitoriji

Repozitorij je dizajn uzorak koji se koristi kako bi se razdvojio sloj koji sadrži poslovnu logiku od sloja koji pristupa bazi tako što nijedan sloj neće znati kako ovaj drugi upravlja podacima, niti gdje i kako se oni zadržavaju [5]. Korištenjem repozitorija mogu se sakriti detalji o tome kako se podaci spremaju ili dohvaćaju s njihovog izvora. Povezivanje s bazom i upravljanje objektima koji sadrže podatke izvodi se kroz metode koje su definirane u repozitoriju. Zbog ovoga nije potrebno pozivati dio koda koji je usko vezan za uspostavljanje konekcije s bazom podataka i slično.

Prednosti korištenja repozitorija u ASP.NET su sljedeći:

- Organizacija aplikacije je jednostavnija i ne postoji izravna ovisnost između poslovnog okvira i okvira koji pristupa podacima
- Jednostavnije je ponovno upotrijebiti kod koji pristupa bazi podataka jer je taj kod upravljan s jednim ili više repozitorija
- Poslovni modeli se mogu testirati neovisno o okviru s bazom podataka

3.3. Ubacivanje ovisnosti

Ubacivanje ovisnosti (engl. *Dependency injection*) je dizajn uzorak u softverima koji omogućava razvijanje koda koji nije previse ovisan o samom sebi. Uz ubacivanje ovisnosti se smanjuje uska povezanost između komponenti. Također se još zove i inverzija kontrole (*IoC*), budući da mijenja normalan tijek upravljanja za razliku od proceduralnog programiranja [6]. To znači da u proceduralnom programiranju dio koda koji koristi neki drugi dio koda je u nadzoru nad cijelim procesom. Zna za svaki djelić koda, svaku metodu koju koristi. Zbog ovoga je vrlo lako moguće da će znati i za neke druge detalje implementacije koju taj kod koristi. Korištenjem inverzije kontrole neće doći do ovoga. Još jedna prednost inverzije kontrole je što se može izolirati kod pri izradi unit testova.

```
private IQuestionRepository QuestionRepository;

public QuestionController(IQuestionRepository _QuestionRepository)
{
    this.QuestionRepository = _QuestionRepository;
}
```

Sl. 3.5. Primjer ubacivanja ovisnosti kroz konstruktor

3.4. Autentikacija i autorizacija

Autentikacija je proces potvrđivanja nečije akreditacije, kao sto su korisničko i lozinka, kako bi se provjerio nečiji identitet. Sustav odlučuje da li je taj netko zapravo taj netko čiju akreditaciju koristi. U javnim i privatnim mrežama sustav autenticira korisnikov identitet preko njegove lozinke kojom se prijavi. Obično se vrši s korisničkim imenom i lozinkom, no u novije vrijeme se kombiniraju s razno raznim faktorima autentikacije.

Autorizacija se događa nakon što je identitet uspješno autenticiran u sustavu, koji nam odmah dodjeljuje mogućnost pristupa svim resursima kao što su informacije, podaci, baze podataka itd. Ukratko, to je proces odlučivanja da li autenticrani korisnik ima pravo pristupa određenim resursima i do koje granice.

3.4.1. JSON web tokeni

JSON web token (JWT) je standard koji definira kompaktan način za siguran protok informacija između dviju strana u pogledu JSON objekta. Ove informacije mogu biti provjerene i sigurne jer su digitalno potpisane. JWT mogu biti potpisani tajnim (s HMAC algoritmom) ključem ili s tajnim/javnim parom ključeva koristeći RSA ili ECDSA. [7]

Potpisani tokeni mogu provjeriti integritet identiteta koji on sadrži, dok enkriptirani tokeni sakrivaju identitet od ostalih strana. Kada su tokeni potpisani koristeći par tajnih/javnih ključeva, taj potpis također osigurava da samo strana koja ima privatni ključ je ona koja ga je potpisala.

JWT se sastoji od tri dijela koji su jedan od drugog odvojeni točkama. Oni su zaglavlje, korisni sadržaj (engl. *payload*) i potpis.

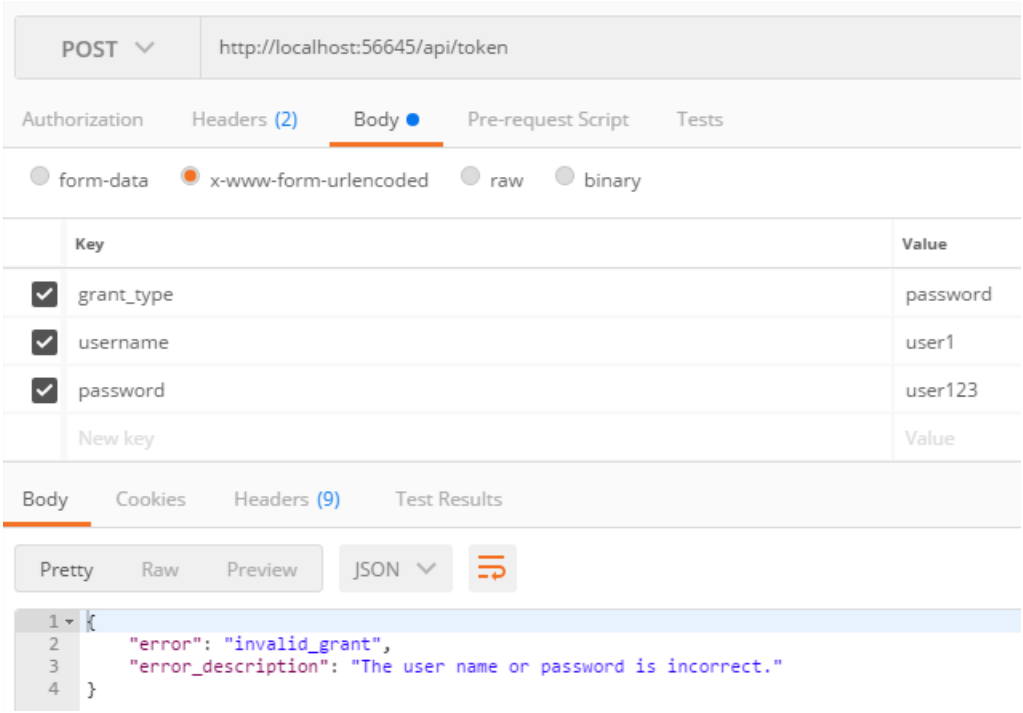
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Sl. 3.6. Izgled JSON web tokena

Pri autentikaciji, kada se korisnik uspješno prijavi koristeći svoje podatke, poslužitelj će vratiti JWT. Budući da tokeni sadrže te podatke u enkriptiranoj verziji, ne bi se trebali predugo zadržavati u sustavu. Kada korisnik želi pristupiti nekoj ruti ili resursi, on treba poslati JWT, obično u autorizacijskom zaglavlju koristeći „Bearer“ shemu. Poslužiteljeve zaštićene rute će zatim provjeriti da li je JWT u autorizacijskom zaglavlju ispravan i postoji, i ako je, korisniku će biti dopušten pristup zaštićenim resursima. Prednost JWT je što je zapis u JSON-u koji je jednostavniji i nije opširan kao XML, te kada je enkodiran manje prostora zauzima, a samim time je dobar izbor za korištenje u HTML i HTTP okruženju. JWT je korišten na globalnoj razini, a posebno se koristi na mobilnim platformama.

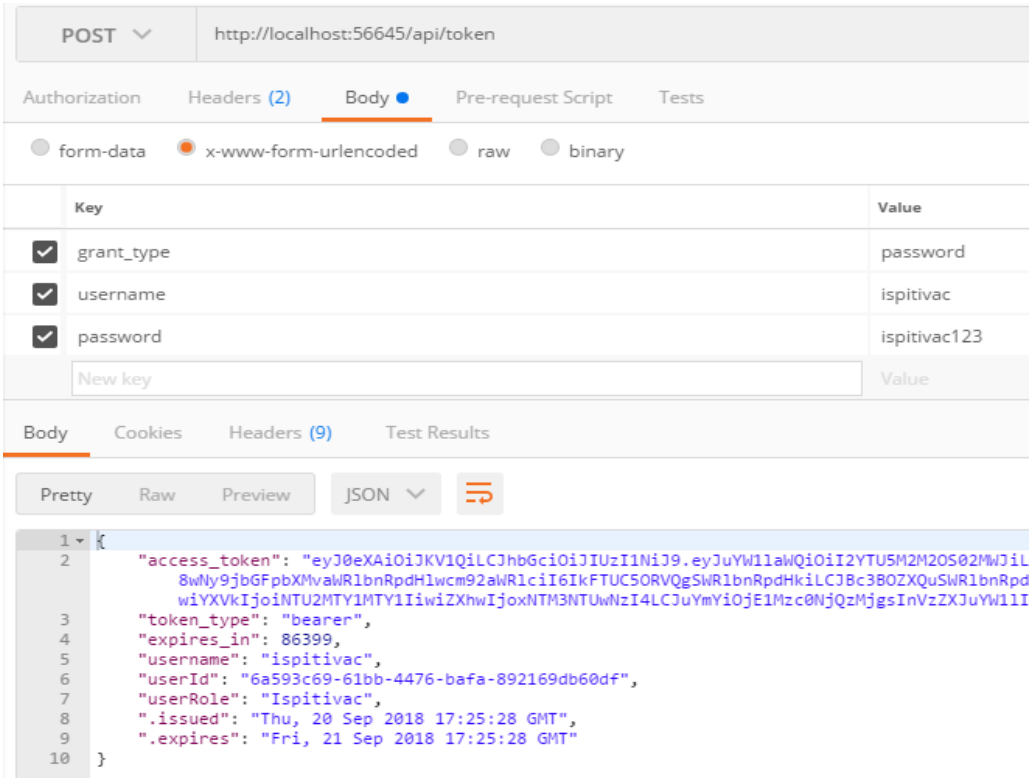
3.4.2. Testiranje JWT

Za testiranje metoda u web API upraviteljima korišten je program Postman. Postman je jednostavan alat slanje zahtjeva i primanje odgovora preko API-ja koji ima svoje grafičko sučelje kako bi se mogli organizirati i spremati testovi. Ovim alatom će biti testirana JWT implementacija na poslužiteljskoj strani. Postman funkcionira tako da se odabere HTTP metoda koja će se koristiti, te destinacija na koju se šalje zahtjev. Za početak će biti poslan POST zahtjev *na /api/token* kako bismo zatražili JWT od poslužitelja. U tijelu zahtjeva je potrebno odabrati x-www-form-urlencoded i staviti iduće „key-value“ vrijednosti: „grant_type: password“, te korisničko ime i lozinku računa s kojeg se šalje zahtjev. Ako su korisnički podaci ispravni, poslužitelj će vratiti valjani JWT.



Sl. 3.7. Zahtjev za JWT s neispravnim korisničkim podacima

Na slici 3.7. može se vidjeti primjer slanja zahtjeva gdje su korisnički podaci neispravni.



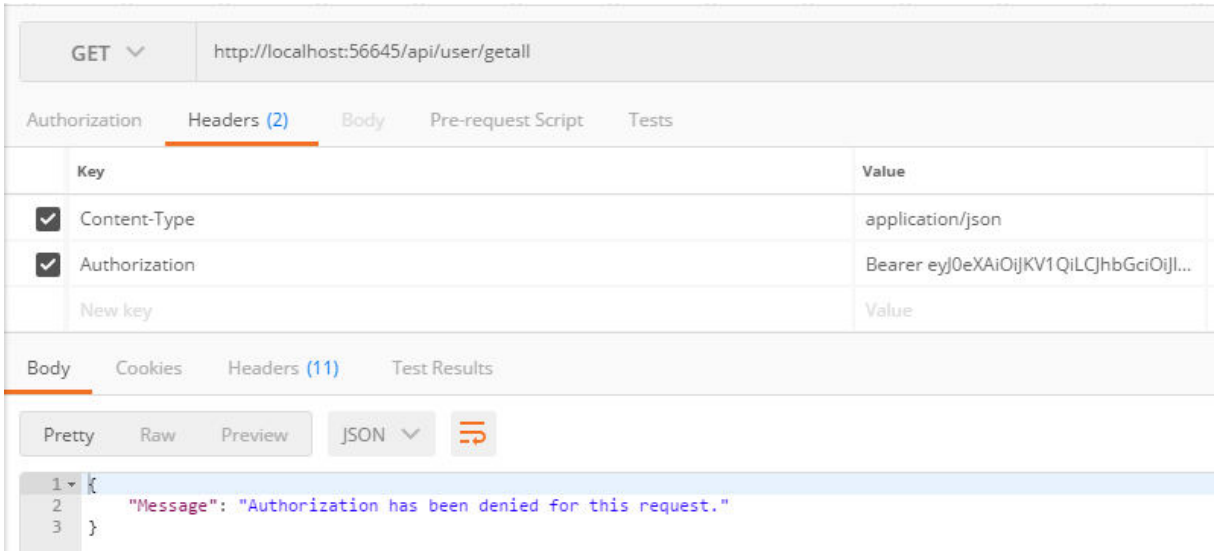
Sl. 3.8. Zahtjev za JWT s ispravnim korisničkim podacima

Na slici 3.8. je vidljivo da unošenjem ispravnih korisničkih podataka i tijela zahtjeva on uspješno prolazi, te poslužitelj vraća token, uz još neke vrijednosti koje su posebno specificirane. Ovaj token je jedinstven za korisnika čiji su podaci korišteni i traje onoliko dugo koliko je označeno na poslužiteljskoj strani, što je u ovom slučaju 1 dan. Nakon isteka 24 sata ovaj token više neće biti valjan i potrebno će biti zatražiti novi token. Token također ovisi i o ulozi korisnika koji ga zatražuje, te npr. token kojega šalje korisnik s funkcijom *ispitanika* neće biti prihvaćen pri pozivanju metoda koje može koristiti samo korisnik s funkcijom *ispitivača*. Ovo se odrađuje tako što se iznad metode u web upravitelju stavlja atribut [Authorize] unutar kojega se navodi koje su funkcije autorizirane za korištenje te metode.

```
[Authorize(Roles = "Admin")]
[Route("getall")]
[HttpGet]
public async Task<HttpResponseMessage> GetAll()
```

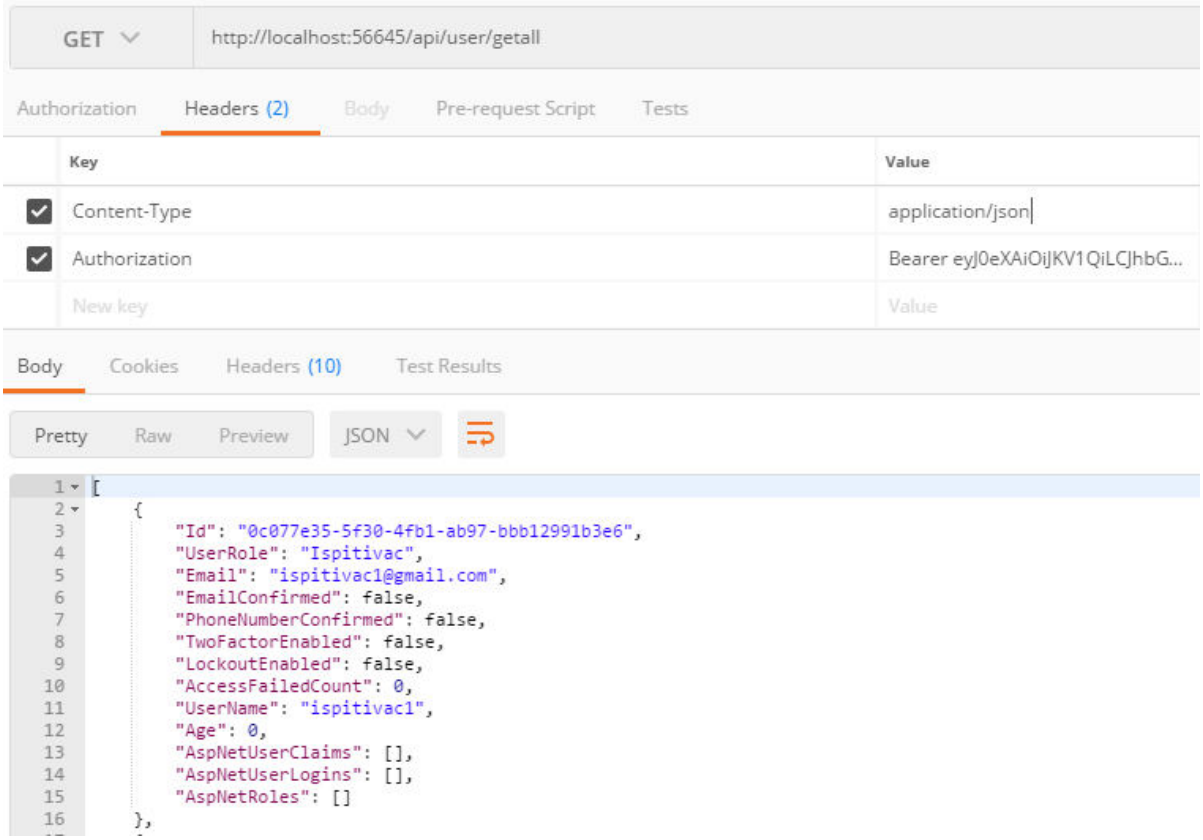
Sl. 3.9. Autorizirana metoda

Kako bi se u Postmanu testirala metoda sa slike 3.9., potrebno je pri zahtjevu na destinaciju ove metode u zaglavlje zahtjeva staviti token koji je dobijen u koraku prije. Budući da je to token kojega je zatražio korisnik s funkcijom *ispitivača*, a sa slike 3.9. je vidljivo da ovu metodu mogu koristiti samo korisnici s funkcijom *administratora*, ovaj zahtjev će biti odbijen, što se može vidjeti na slici 3.10.



Sl. 3.10. Zahtjev odbijen zbog neispravnog tokena

Da bi se uspješno pozvala spomenuta metoda, potrebno je poslati u zaglavlju zahtjeva poslati token kojega je zatražio korisnik s funkcijom *administratora*. Ako se ponovi postupak gdje se zatražuje token, no ovaj put s podacima korisnika *administrator*, i taj token stavi u zaglavlje zahtjeva, zahtjev će uspješno proći i biti će vraćeni podaci koje ta metoda vraća u JSON formatu, što su u ovom slučaju podaci sa svim registriranim korisnicima. Uspješno vraćeni podaci prikazani su na slici 3.11.

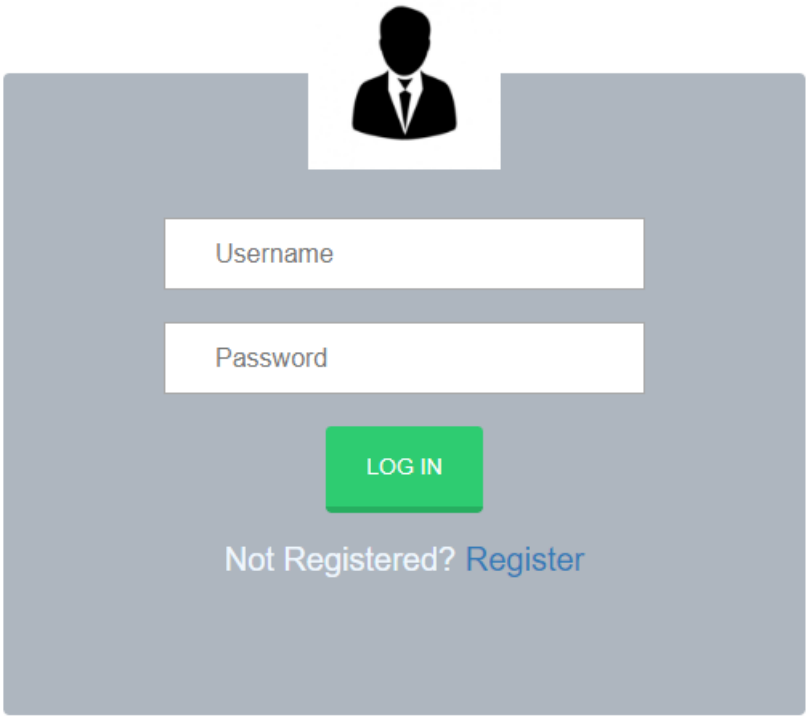


Sl. 3.11. Zahtjev uspješan

4. KORIŠTENJE WEB APLIKACIJE

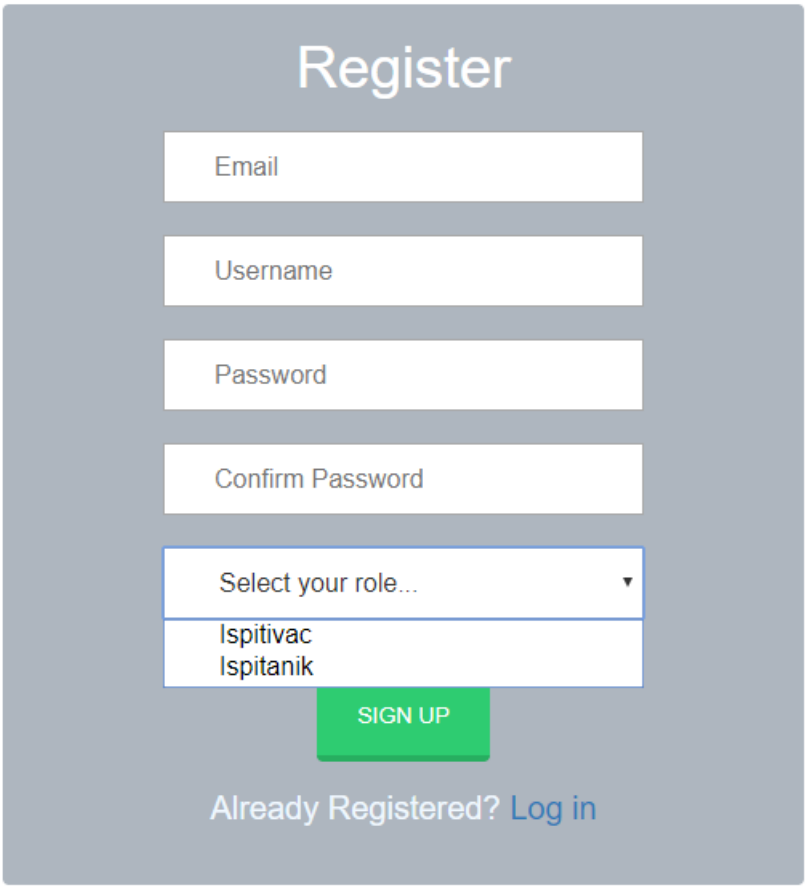
Aplikacija razlikuje tri vrste korisnika. Korisnik koji je prijavljen s funkcijom *ispitanika* ima mogućnost pregleda svih anketa ako su one vidljive, te popunjavanja istih ako im je status označen kao *active*. Korisnik prijavljen s funkcijom *ispitivača* ima mogućnost izrade anketa. Nakon izrade ankete korisnik može pregledati sve ankete koje je izradio, kao i njihove rezultate. Korisnik prijavljen s funkcijom *administratora* ima mogućnost uvida u sve ankete te može promijeniti njihovu vidljivost, tj. hoće li određena anketa biti vidljiva korisnicima koji ih popunjavaju. Ovakva vrsta korisnika također može vidjeti sve registrirane korisnike i uklanjati ih.

Prilikom prve posjete aplikaciji (ili ako korisnik nije prijavljen) korisnik će biti preusmjeren na stranicu, koja je prikazana na slici 4.1., gdje je nužno prijaviti se kako bi mogao koristiti funkcionalnosti aplikacije. Ako korisnik nema izrađen račun, moći će ga izraditi pritiskom na „Register“ na stranici za prijavu.



Sl. 4.1. Stranica za prijavu

Na stranici gdje korisnik vrši registraciju, prikazanoj na slici 4.2., treba unijeti uobičajene podatke koje se traže prilikom registracije novih korisnika: email, korisničko ime, lozinku, potvrđenu lozinku, no još je potrebno odabrati hoće li korisnik izraditi račun s funkcijom *ispitivača* ili *ispitanika*.

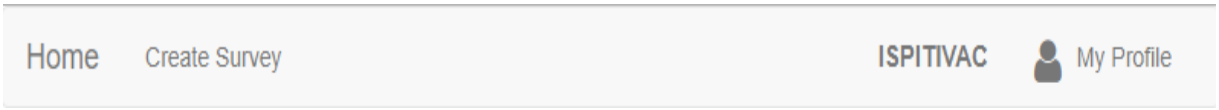


Sl. 4.2. Stranica za registraciju

Nakon popunjavanja svih podataka, aplikacija će na poslužiteljskoj strani pregledati sva korisnička imena i e-mailove i usporediti ih s onime što je korisnik unio u formu izrade novog računa. Ako se ispostavi da to korisničko ime i e-mail nisu zauzeti, registracija je uspješno provedena i korisnik se preusmjeruje na stranicu za prijavu gdje se može prijaviti s podatcima kojima se registrirao.

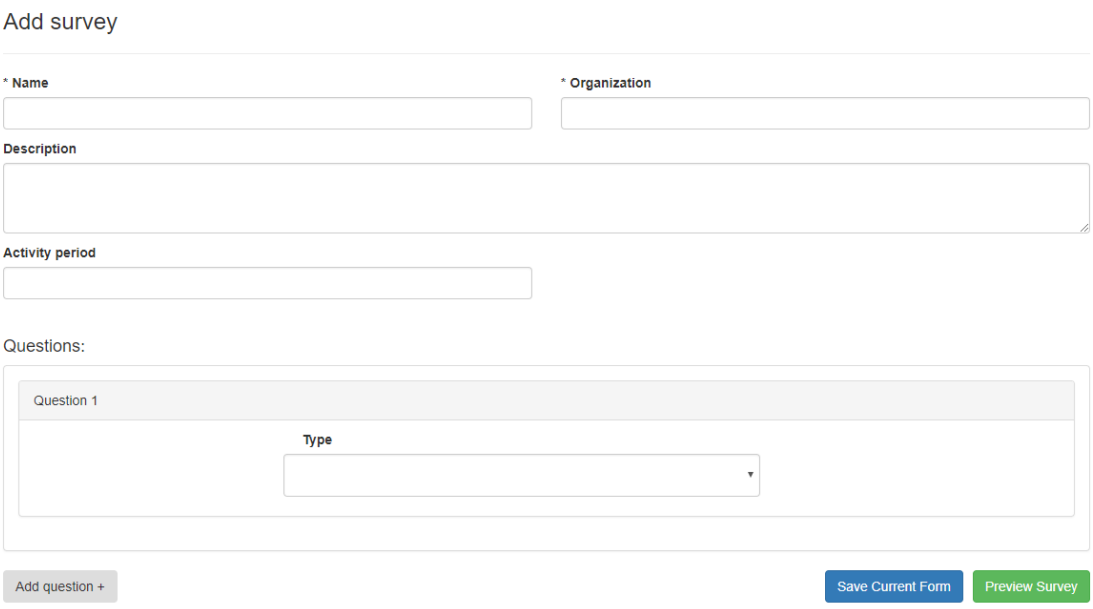
4.1. Korisnik s funkcijom *ispitivača*

Korisnik prijavljen kao *ispitivač* vidjeti će zaglavlje aplikacije kao na slici 4.3.



Sl. 4.3. Zaglavlje aplikacije kod ispitivača

Pritiskom na „Create Survey“ korisnik će biti navigiran na formu gdje može izraditi svoju vlastitu anketu, a ona je prikazana na slici 4.4.

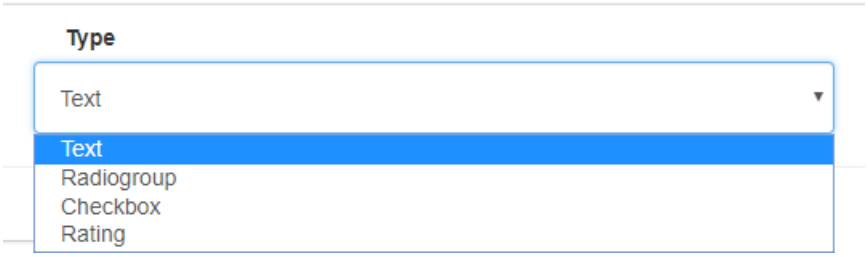


Sl. 4.4. Forma za izradu ankete

Korisnik zatim odabire ime pod kojim će anketa biti vidljiva, ime organizacije koja provodi anketu, nekakav kratki opis o čemu je anketa i koja joj je svrha, te period u kojemu će biti

anketa biti aktivna. Dakle, ako je trenutni datum npr. 10.09. i korisnik odabere da će anketa biti aktivna u periodu od 20.09. do 30.09., to znači da se anketa neće moći popunjavati sve dok datum ne bude 20.09. i do tada će njen status biti označen kao *inactive*. Isto tako, nakon 30.09., ova će anketa biti označena kao *expired* i također se neće moći popunjavati. U ostalim slučajevima status ankete će biti *active*.

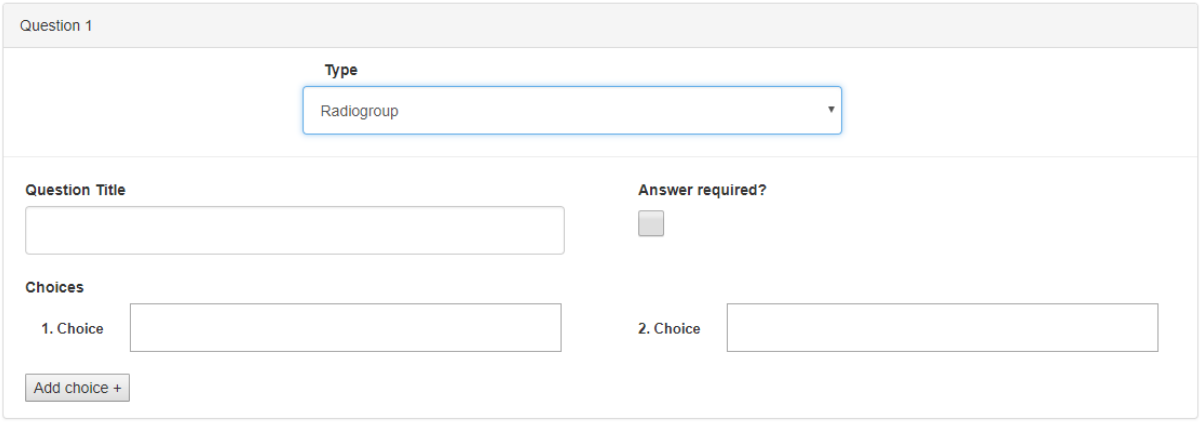
Korisnik zatim prelazi na dodavanje pitanja u svoju anketu. Prvo što odabire kod dodavanja svakog pitanja je vrsta pitanja. Postoje 4 vrste pitanja: *text*, *radiogroup*, *checkbox* i *rating* pitanje, a padajući izbornik s izborom pitanja može se vidjeti na slici 4.5.



Sl. 4.5. Vrste pitanja

Text pitanje – potrebno je samo napisati tekst pitanja, te korisnik na ovo pitanje odgovara unosom proizvoljnog odgovora

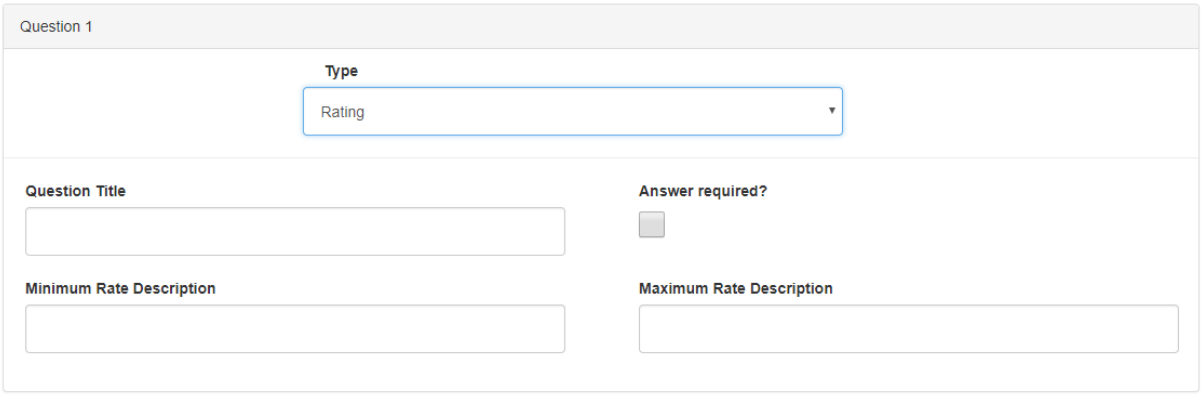
Radiogroup pitanje – odabirom ovakvog pitanja, uz tekst pitanja, potrebno je još dodati ponuđene odgovore, kojih je minimalno dva. Korisnik na ovakvo pitanje odgovara odabirom jednog od ponuđenih odgovora. Primjer je na slici 4.6.



Sl. 4.6. „Radiogroup“ pitanje

Checkbox pitanje – sve je isto kao kod radiogroup pitanja, jedina razlika je to što korisnik na ovakvo pitanje može odabrati više ponuđenih odgovora

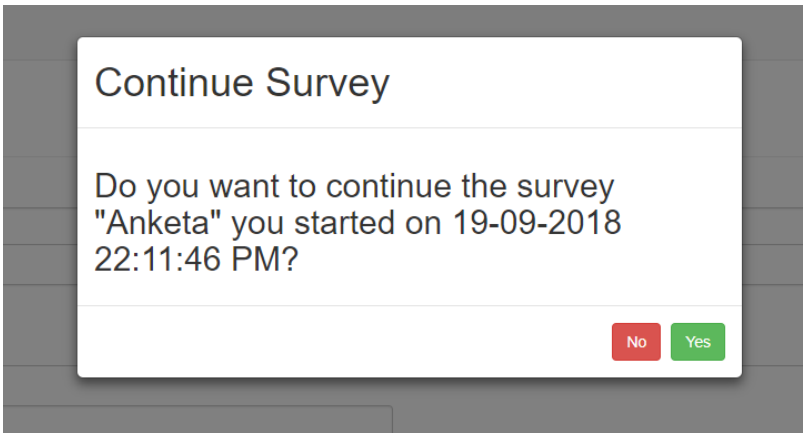
Rating pitanje – kod ovakvog pitanja je potrebno navesti što će predstavljati donju i gornju granicu odgovora, budući da se na ovo pitanje odgovara odabirom broja od 1 do 5. Primjer dodavanja ovakvog pitanja je na slici 4.6.



Sl. 4.7. Rating pitanje

Kod svakog pitanja je također moguće odabrati da li je nužno odgovori na to pitanje, a to se može stavljanjem kvačice na „Answer required?“ kvadratić.

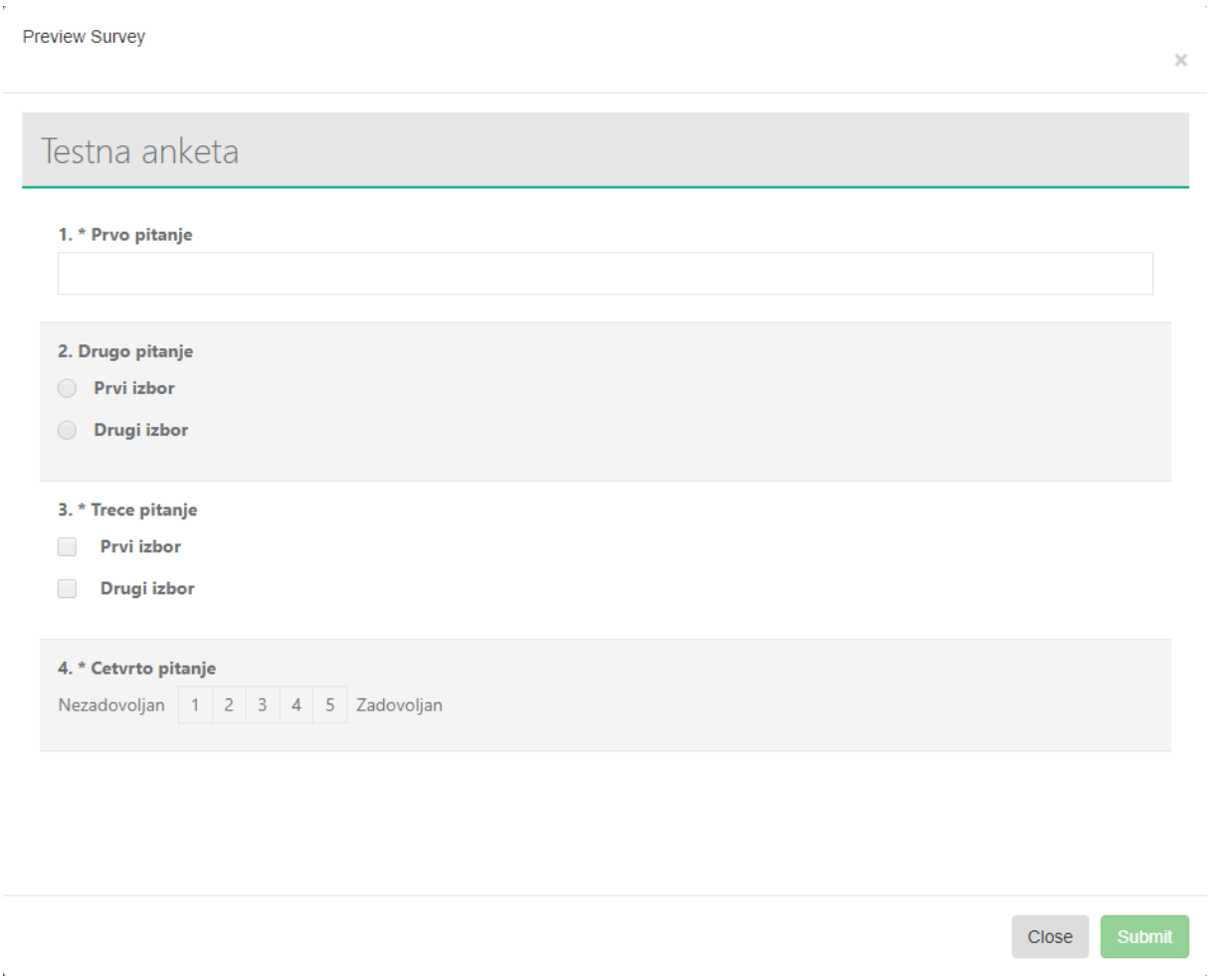
Ako korisnik ne stigne dovršiti anketu ili jednostavno želi nastaviti s njenom izradom nekad u budućnosti, korisnik može pritiskom na „Save Current Form“ spremiti trenutnu anketu. U tom slučaju će korisnik kada idući put opet navigira na formu za izradu anketa dobiti modalni prozor prikazan na slici 4.8.



Sl. 4.8. Primjer nastavljanja ankete koju je korisnik ranije započeo

Pritiskom na tipku „No“ ovaj modalni prozor se neće pojavljivati u budućnosti i korisnik može normalno nastaviti s izrađivanjem nove ankete. Pritiskom na tipku „Yes“ forma se automatski popunjava s podacima koje je korisnik unio onda kada je spremio tadašnju formu.

Kada korisnik smatra da je u potpunosti gotov s izradom ankete, klikom na tipku „Preview Survey“ se otvara modalni prozor kao na slici 4.9, u kojemu korisnik može pregledati kako izgleda njegova anketa. Ovo je korak u kojem korisnik odlučuje da li je zadovoljan izgledom svoje ankete, poretkom pitanja, ponuđenih odgovora i sl.



Sl. 4.9. Izgled ankete prije njenog spremanja

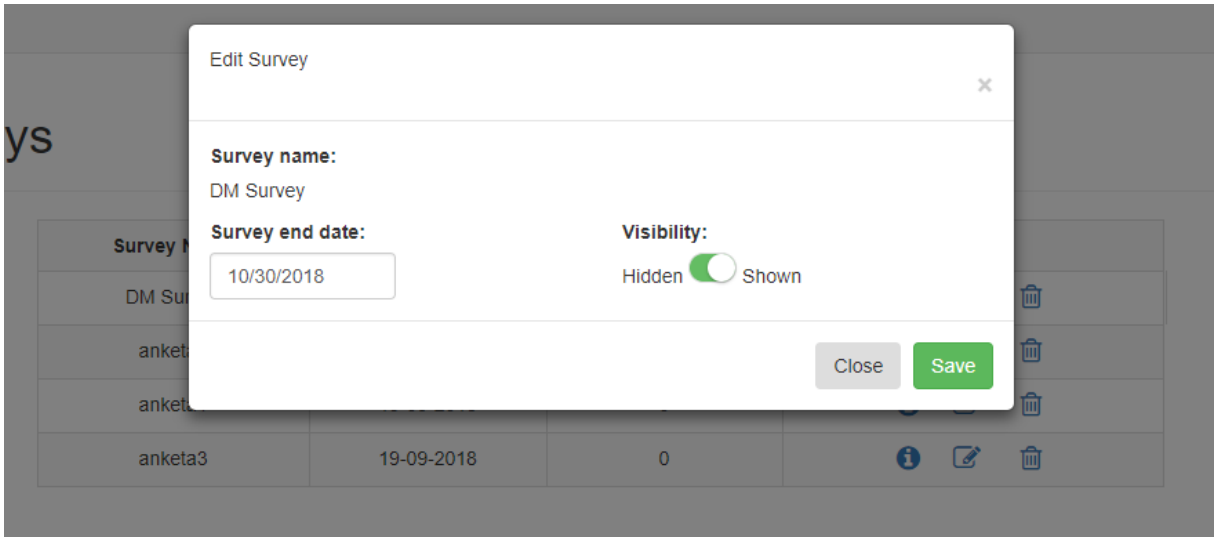
Ako je korisnik zadovoljan sa svime, nakon pritiska na tipku „Submit“ anketa će biti spremljena i dostupna za rješavanje.

Korisnik može vidjeti sve ankete koje je kreirao pritiskom na „My Profile“ u zaglavlju aplikacije, te odabiranjem „My Surveys“. Ovdje je moguće evidentirati koliko je ljudi sudjelovalo u anketi, izlistati detalje svake ankete, mijenjati razdoblje u kojemu je anketa

aktivna i njenu vidljivost, kao i brisanje anketa. Ove funkcionalnosti prikazane su na slikama 4.10. i 4.11.

Survey Name	Created On	Responses	
DM Survey	19-09-2018	3	
anketa2	19-09-2018	0	
proba	20-09-2018	0	
anketa1	19-09-2018	0	
anketa3	19-09-2018	0	
Anketa	19-09-2018	0	

Sl. 4.10. Prikaz anketa koje je korisnik izradio



Sl. 4.11. Modalni prozor za izmjenu već spremljene ankete

Klikom na detalje ankete, prikazuje se tablica s pitanjima te ankete, kao što je vidljivo na slici 4.12.

	Question Title	Question Type	
1.	Koji od ponudenih proizvoda vam je najdrazi?	Radiogroup	View results
2.	Izaberite proizvod koji kupujete	Checkbox	View results
3.	Kako ste zadovoljni s LVLY ulje za čišćenje lica?	Rating	View results
4.	Sto biste mijenjali kod dmBio Jackfruit	Text	View results

Sl. 4.12. Tablica s pitanjima ankete

Rezultati pitanja

Kao što je već spomenuto, postoje 4 tipa pitanja i rezultati svakog od tih tipova prikazani su na drugačiji način.

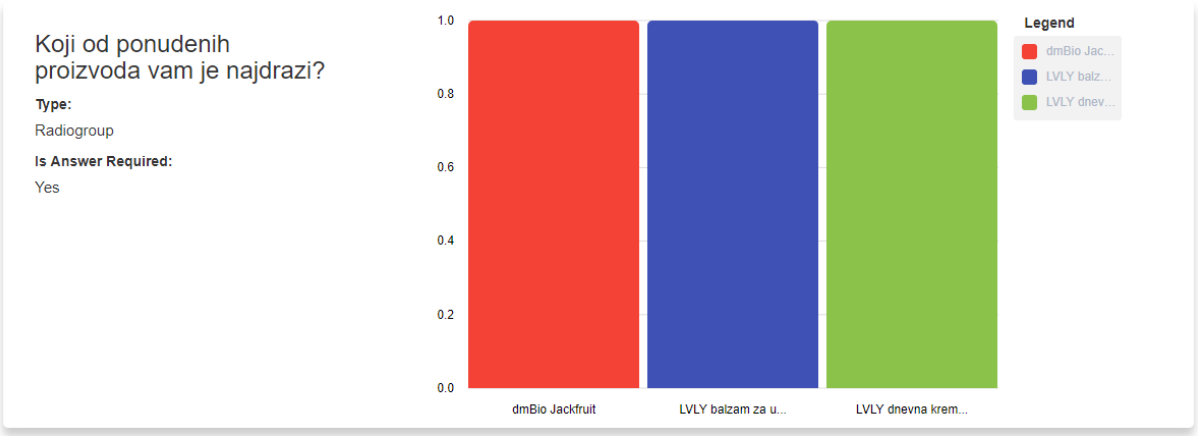
Sto biste mijenjali kod dmBio Jackfruit

Type:
Text

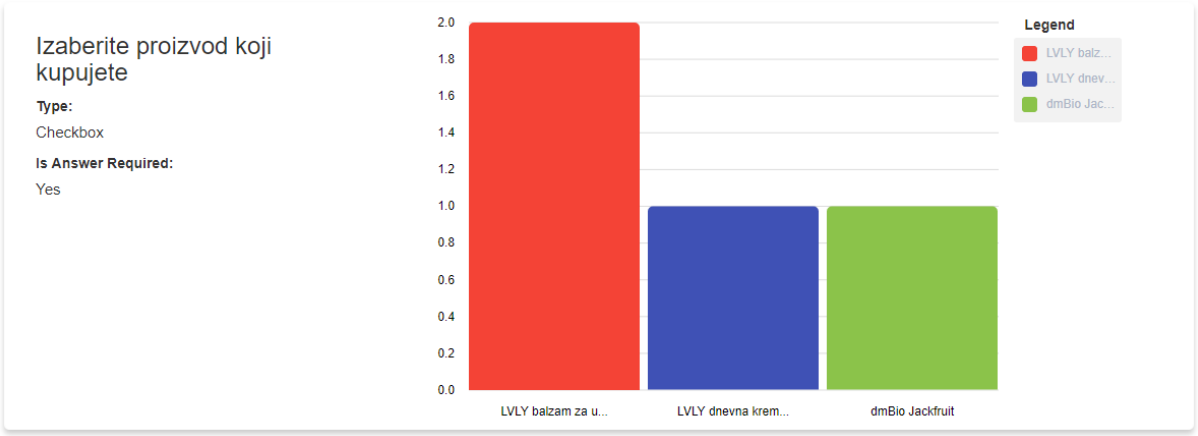
Is Answer Required:
No

	Answer
1	Cijenu
2	Pakiranje
3	Nista

Sl. 4.13. Rezultati „text“ pitanja



Sl. 4.14. Rezultati „radiogroup“ pitanja

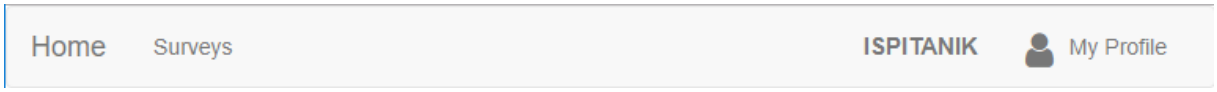


Sl. 4.15. Rezultati „checkbox“ pitanja

Sl. 4.16. Rezultati „rating“ pitanja

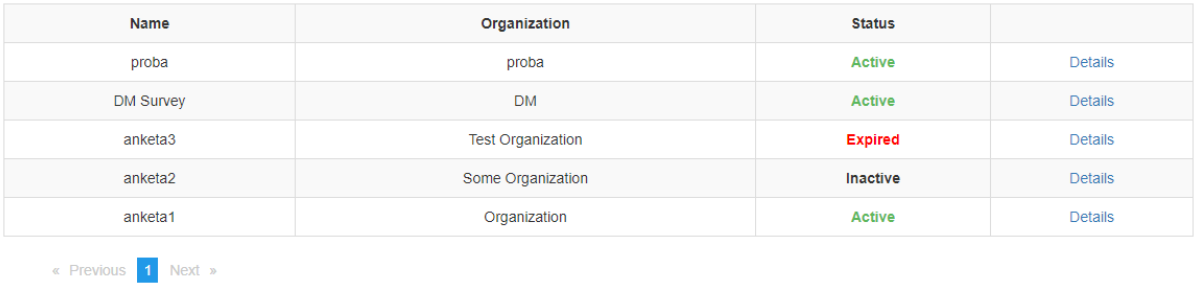
4.2. Korisnik s funkcijom *ispitanika*

Korisnik koji se prijavi kao *ispitanik* vidjeti će zaglavlje aplikacije prikazano na slici 4.17.



Sl. 4.17. Zaglavlje aplikacije kod ispitanika

Ovakav korisnik nije u mogućnosti izrađivati ankete, već ih samo popunjavati. Nakon što pritisne „Surveys“ u zaglavlju aplikacije, korisnik je preusmjeren na predložak gdje su prikazane sve ankete koje su postavljene da budu vidljive, a prikaz toga se vidi na slici 4.18.



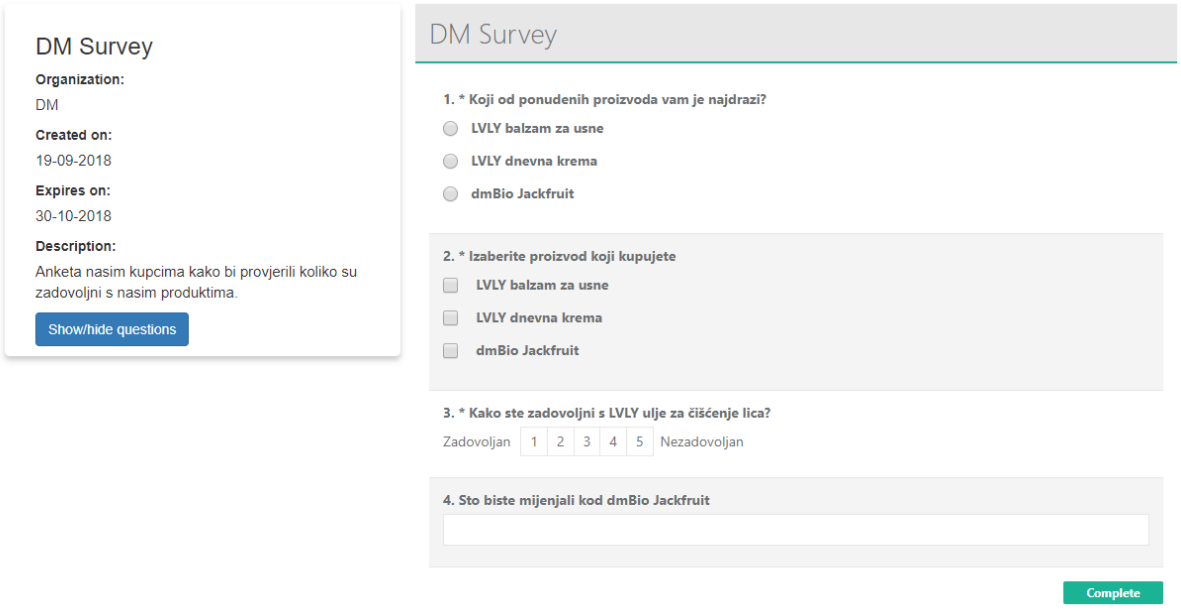
Name	Organization	Status	
proba	proba	Active	Details
DM Survey	DM	Active	Details
anketa3	Test Organization	Expired	Details
anketa2	Some Organization	Inactive	Details
anketa1	Organization	Active	Details

« Previous 1 Next »

Sl. 4.18. Prikaz svih dostupnih anketa

Kao što je vidljivo na slici 4.18., status ankete ovisi o njenom razdoblju u kojem je aktivna. Korisnik s ulogom *ispitivač* može popunjavati samo one ankete koje imaju status *active*. Ako bi korisnik htio detaljnije vidjeti anketu ili ju ispuniti, on to čini pritiskom na „Details“.

Nakon toga se otvara predložak gdje je detaljnije opisana anketa s poljima poput kada je anketa kreirana, kada završava, opis ankete itd. Pritiskom na tipku „Show/hide questions“ se mogu prikazati ili sakriti pitanja ankete, te je ovo ujedno i mjesto gdje korisnik može ispuniti anketu.



Sl. 4.19. Detalji ankete

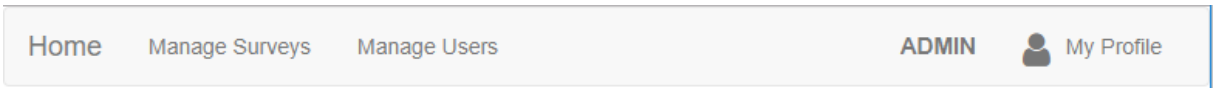
Ako je odabrana anketa kojoj status nije *active*, kao npr. „anketa3“ sa slike 4.18., gdje se vidi da joj je status *expired* (isto vrijedi i za ankete sa statusom *inactive*), tada korisnik neće imati pravo popunjavanja ankete, tj. neće biti tipke „Complete“ kojom se završava popunjavanje ankete i neće moći unijeti niti jedan odgovor. Ovaj slučaj vidljiv je na slici 4.20.

Sl. 4.20. Korisnik ne može popunjavati anketu koja je istekla

Nakon što korisnik ispuni anketu i pritisne „Complete“, njegovi odgovori će biti spremljeni.

4.3. Korisnik s funkcijom *administratora*

Korisnik koji se prijavi kao *admin* vidjeti će zaglavlje aplikacije kao na slici 4.21.



Sl. 4.21. Zaglavlje aplikacije kod administratora

Administratorski korisnik je samo jedan i ne postoji mogućnost registracije kao *administrator*, već je on ručno kreiran na poslužiteljskoj strani. Administratorski korisnik nema mogućnosti

dodavanja ili ispunjavanja anketa, ali može mijenjati vidljivost svake ankete, te ima uvid o svim registriranim korisnicima i ima pravo njihovog uklanjanja.

Survey Name	Status	Created by	Created On	Visibility
DM Survey	Active	ispitivac	19-09-2018	Hidden Shown
anketa2	Active	ispitivac	19-09-2018	Hidden Shown
proba	Active	ispitivac	20-09-2018	Hidden Shown
anketa1	Active	ispitivac	19-09-2018	Hidden Shown
anketa3	Inactive	ispitivac	19-09-2018	Hidden Shown

« Previous 1 Next »

Sl. 4.22. Prikaz anketa administratorskom korisniku

5. ZAKLJUČAK

Ankete danas igraju veliku ulogu u prikupljanju podataka o raznim društvenim problemima, marketingu, stvaranju statistika itd. Ankete dostupne na Internetu predstavljaju brz i jednostavan način za iznošenje vlastitog mišljenja o određenom problemu, te je vrlo lako prikupiti veliki broj rezultata u kratkom vremenu. Ankete su veoma značajne u marketingu jer su one najpouzdaniji način za uvid o povratnim informacijama kupaca prije investiranja previše vremena ili novca u neki novi produkt. One također pružaju vezu s kupcima gdje korisnici mogu brzo reagirati na promjene, bile one pozitivne ili negativne. Neki od glavnih razloga provođenja poslovnih anketa su kako bi se dokazala određena pretpostavka o kupcima, praćenje razina zadovoljstva kupaca, praćenje konkurencije, povratne informacije o novim proizvodima i sl Pitanja ovakvih anketa trebaju biti strateški isplanirana i strukturirana na najbolji mogući način kako bi se mogli dobiti najprecizniji podaci. Stoga je jako bitno imati dobar sustav za izradu anketa s raznovrsnim tipovima pitanja i gdje će odgovori moći biti grafički prikazani radi uspoređivanja rezultata, jednostavnog i učinkovitog prezentiranja istih itd.

Ujedno je to i zadatak ovog diplomskog rada, a pruženo rješenje u realizirano u obliku web aplikacije. Aplikacija je realizirana kao „single page aplikacija“, što znači da će ovakva stranica dinamički prepisati trenutnu stranicu, bez da mora svaki put slati zahtjev poslužitelju za novom stranicom, te ju to čini brzom i efikasnom. Značajke poput razdoblja u kojemu je anketa aktivna i nastavljanje prethodne ankete definitivno mogu olakšati posao korisniku koji kreira ankete, a samim time i korisniku koji ih popunjava. Prostor za napredak postoji, te bi se u slučaju objavljivanja nekih novih verzija aplikacije moglo dodati još tipova pitanja, više vrsta grafova za vizualizaciju odgovora i slično.

LITERATURA

[1] https://bs.wikipedia.org/wiki/.NET_Framework posjećeno: 13.09.2018.

[2] https://en.wikipedia.org/wiki/Entity_Framework posjećeno: 13.09.2018.

[3] https://www.computerhope.com/jargon/h/html.htm posjećeno: 15.09.2018.

[4] http://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx

posjećeno: 25.01.2017.

[5] https://docs.microsoft.com/en-us/aspnet/core/fundamentals/repository-pattern?view=aspnetcore-2.1 posjećeno: 02.02.2017.

[6] http://joelabrahamsson.com/inversion-of-control-an-introduction-with-examples-in-net/ posjećeno: 05.02.2017.

[7] https://jwt.io/introduction/ posjećeno: 15.09.2018.

SAŽETAK

U radu je implementiran sustav gdje korisnici mogu izrađivati ankete za ispitivanje javnog mnijenja, kao i odgovoarati na njih. On je realiziran u smislu web aplikacije. Aplikacija razlikuje tri vrste korisnika. Korisnik prijavljen kao *ispitanik* ima samo mogućnosti popunjavanja anketa. Korisnik s ulogom *ispitanika* može izrađivati vlastite ankete, te vidjeti njihove rezultate prikazane vizualno. *Administrator* korisnik ima mogućnost sakrivanja dostupnih anketa, kao i pregled i uklanjanje registriranih korisnika. Pri kreiranju anketa je potrebno odrediti razdoblje u kojemu će ona biti aktivna, te će korisnici moći popunjavati samo one ankete koje su u tom razdoblju označene kao aktivne. Pri dodavanju pitanja je moguće odabrati između 4 vrste pitanja, od kojih se odgovori od svake vrste vizualiziraju na njima pogodan način. Riješenje je realizirano korištenjem *ASP.NET* razvojnog okruženja i *MSSQL* baze podataka na poslužiteljskoj strani, te korištenjem *Angular 4* razvojnog okružennja, *HTML* prezentacijskog jezika i *CSS* stilskog jezika na korisničkoj strani.

Ključne riječi: *web aplikacija, ankete, ASP.NET, Angular, Entity framework, JWT, HTML, CSS, Bootstrap*

ABSTRACT

Web application for public opinion research

A system where users can create their own surveys to explore public opinion and respond to them is implemented in this thesis. It is implemented in a way of web application. Application has three different types of users. User logged as *examinee* can only respond to surveys. User with a role of *examineer* can create his own surveys, as well as check their responses shown graphically. *Administration* user has the privilege of hiding available surveys, as well as inspecting all the registered users and deleting them if needed. When creating surveys it is needed to set the period of time when survey will be active, which is needed so users cannot respond to them if they're not active. When adding questions to the survey, it is possible to choose between four different types of questions, whose answers of every of these types are visualised in a different and adequate way. Solution is implemented using the *ASP.NET* framework and *MSSQL* database on a server side, and using the *Angular 4* framework. *HTML* markup language and *CSS* style sheet language on client side.

Key words: *web application, surveys, ASP.NET, Angular, Entity framework, JWT, HTML, CSS, Bootstrap*

ŽIVOTOPIS

Petar Bagarić rođen je u Frankfurtu na Majni 31. listopada 1993. Osnovnu školu pohađao je i završio u Bilju. 2008. godine upisuje 3. gimnaziju u Osijeku, koju završava 2012. godine. Iste godine upisuje preddiplomski studij na Elektrotehničkom fakultetu u Osijeku, smjer računarstvo. 2015. godine ga završava i upisuje diplomski studij na spomenutom fakultetu, trenutno imena Fakulteta elektrotehnike, računarstva i informacijskih tehnologija, smjer računarstvo, izborni blok informacijske tehnologije kojeg trenutno pohađa.

Vlastoručni potpis:

_____

Petar Bagarić

PRILOZI

Na CD-u:

1. Diplomski rad „Web aplikacija za istraživanje javnog mnijenja.docx“
2. Diplomski rad „Web aplikacija za istraživanje javnog mnijenja.pdf“
3. Kod aplikacije