Mobilna aplikacija za anketiranje

Knežević, Robert

Undergraduate thesis / Završni rad

2023

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:200:485014

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-16**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Sveučilišni studij

MOBILNA APLIKACIJA ZA ANKETIRANJE

Završni rad

Robert Knežević

Osijek, 2023.

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

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Robert Knežević
Studij, smjer:	Računalno inženjerstvo
Mat. br. Pristupnika, godina upisa:	R4513, 27.07.2020.
OIB Pristupnika:	29693877889
Mentor:	doc. dr. sc. Tomislav Galba
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna aplikacija za anketiranje
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak završnog rad:	Potrebno je napraviti mobilnu aplikaciju (Android) kojom će korisnici moći pristupiti anketiranju. Potrebno je omogućiti prijavu/registraciju, pregled dostupnih/popunjenih anketa. Anketa bi trebala biti u mogućnosti sadržavati tekst (odabir odgovora, unos odgovora), sliku. Uz mobilnu aplikaciju potrebno je osmisliti bazu podataka i API za pristup. Anketu bi trebalo biti moguće
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	15.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 18.09.2023.

Ime i prezime studenta:	Robert Knežević
Studij:	Računalno inženjerstvo
Mat. br. studenta, godina upisa:	R4513, 27.07.2020.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Mobilna aplikacija za anketiranje**

izrađen pod vodstvom mentora doc. dr. sc. Tomislav Galba

i sumentora ,

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

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PREGLED POSTOJEĆIH RJEŠENJA 2

2.1. SurveyMonkey 2

2.2. Jotform 3

2.3. AttaPoll 4

2.4. Forms App 5

3. ZAHTJEVI ZA PROGRAMSKO RJEŠENJE 6

3.1. Registracija i prijava korisničkog računa 6

3.2. Pregled dostupnih anketa 6

3.3. Ispunjavanje anketa i pohranjivanje rezultata 7

3.4. Pregled ispunjenih anketa 7

4. ALATI I TEHNOLOGIJE 8

4.1. Visual Studio Code 8

4.1.1. Node.js 9

4.1.2. Express 9

4.2. PostgreSQL 10

4.2.1. Dbeaver 10

4.3. Android studio 11

4.3.1. Kotlin 11

4.4. Firebase 12

5. IZRADA PROGRAMSKOG RJEŠENJA 14

5.1. Baza podataka 15

5.2. API za dohvat podataka 15

5.2.1. Uspostavljanje veze s bazom podataka 16

5.2.2. Definiranje ruta, kontrolera i modela 16

5.3. Android studio Kotlin 19

5.3.1. Korisničko sučelje 19

6. ZAKLJUČAK 26

LITERATURA: 27

SAŽETAK 28

ABSTRACT 29

PRILOZI 30

1. UVOD

Cilj ovog rada je istražiti i implementirati mobilnu aplikaciju koja će omogućiti korisnicima prijavu i registraciju, pregled dostupnih anketa te sudjelovanje u njima. Ankete će biti interaktivne i moći će sadržavati različite vrste pitanja kao što su odabir odgovora, unos odgovora u tekstualnom obliku i prikaz slika. Kroz aplikaciju, korisnici će moći pregledati anketu, odgovoriti na postavljena pitanja te dostaviti svoje odgovore.

Uz razvoj mobilne aplikacije, također je potrebno osmisliti i implementirati bazu podataka koja će omogućiti pohranu anketa, korisničkih podataka i rezultata ispunjenih anketa. Za pristup podacima iz baze, razvijat ćemo API (eng. *Application Programming Interface*) koji će omogućiti komunikaciju između mobilne aplikacije i baze podataka, a za autentifikaciju korisnika koristit će se firebase autentifikacija. API će pružiti potrebne funkcionalnosti za dohvat dostupnih anketa, prijavu korisnika, pohranu odgovora na ankete te upravljanje korisničkim računima.

Dodatno, aplikacija će podržavati dodjeljivanje anketa određenim korisnicima, grupama korisnika ili svim korisnicima. To će omogućiti prilagodbu anketiranja prema specifičnim korisničkim skupinama ili ciljevima istraživanja.

1.1. Zadatak završnog rada

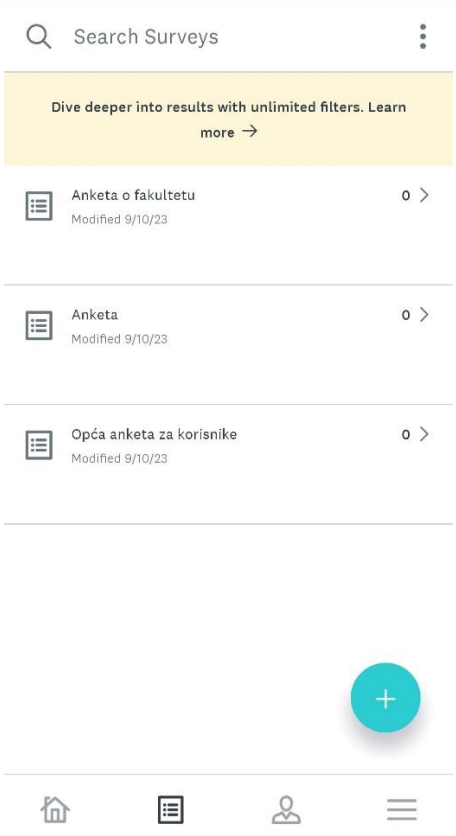
Potrebno je napraviti mobilnu aplikaciju (Android) kojom će korisnici moći pristupiti anketiranju. Potrebno je omogućiti prijavu/registraciju, pregled dostupnih/ispunjenih anketa. Anketa bi trebala biti u mogućnosti sadržavati tekst (odabir odgovora, unos odgovora), sliku. Uz mobilnu aplikaciju potrebno je osmisliti bazu podataka i API za pristup. Anketu bi trebalo biti moguće dodjeliti određenim korisnicima, grupama korisnika ili svim korisnicima.

2. PREGLED POSTOJEĆIH RJEŠENJA

U ovome dijelu završnog rada analizirati će se programska rješenja koja su služila kao uzor u izradi programskog rješenja, odnosno aplikacije sa sličnom funkcionalnošću. Svrha ove analize bila bi usporedba značajki aplikacija, njihove funkcionalnosti i performansi, kako bi se vidjele prednosti i nedostatci ove aplikacije. Aplikacije koje će biti spomenute su SurveyMonkey, Jotform, AttaPoll te Forms App.

2.1. SurveyMonkey

SurveyMonkey je platforma koja se primarno fokusira na stvaranje i dizajniranje anketa. Koristi se za izradu različitih vrsta anketa, i kasnije omogućuju prikupljanje povratnih informacija od različitih sudionika. Bitno je za reći da se unkcionalnosti aplikacije iz rada i funkcionalnosti aplikacije SurveyMonkey razlikuju u glavnoj stvari, a to je u pristupu prema anketama. Dok se u programskom rješenju aplikacije iz rada ankete samo ispunjavaju, u SurveyMonkey-u je omogućeno samo praviti ankete, a na slici 2.1. je prikazano kako se prikazuju napravljene ankete. Iako se funkcionalnosti razlikuju, korisničko sučelje aplikacija je slično. Glavne sličnosti bi bile mogućnost izlistavanja anketa i pregled ispunjenih anketa.



Sl. 2.1. Izgled sučelja SurveyMonkey

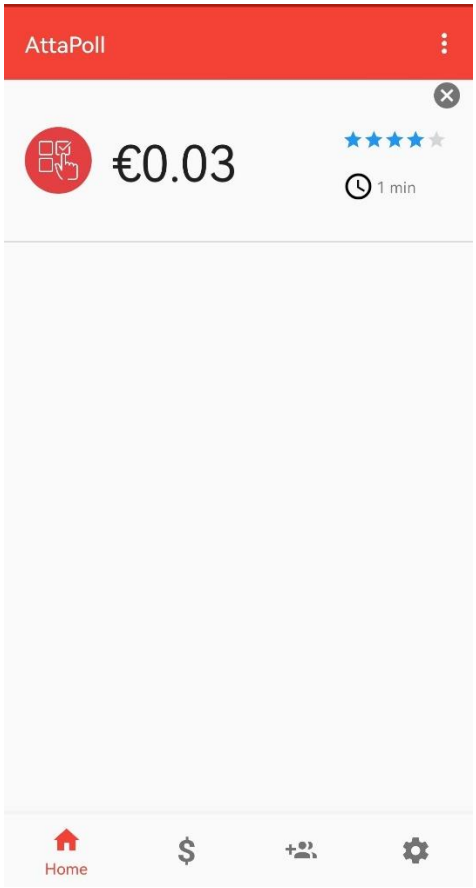
2.2. Jotform

Jotform je aplikacija koja omogućuje korisnicima da jednostavno izrađuju obrasce i ankete, prilagode ih prema vlastitim potrebama te prikupljaju odgovore. Jotform nudi različite alate za oblikovanje, te ima puno više funkcionalnosti od aplikacije u ovom radu. Jotform, kao i SurveyMonkey, nudi funkcionalnost kreiranja anketa. Također je moguće i ispuniti kreirane ankete. Ova aplikacija je po tome najsličnija aplikaciji u radu, te je aplikacija s najviše funkcionalnosti, u odnosu na ostale koje su spomenute, ili će biti spomenute. Jotform ima prikaz anketa kao i prethodno spomenuta aplikacija, pa je nepotrebno stavljati identično, ali je bitno spomenuti sučelje za ispunjavanje ankete koje je uvelike slično aplikaciji u radu (Sl. 2.2.).

Sl. 2.2. Sučelje za popunjavanje ankete u Jotform aplikaciji

2.3. AttaPoll

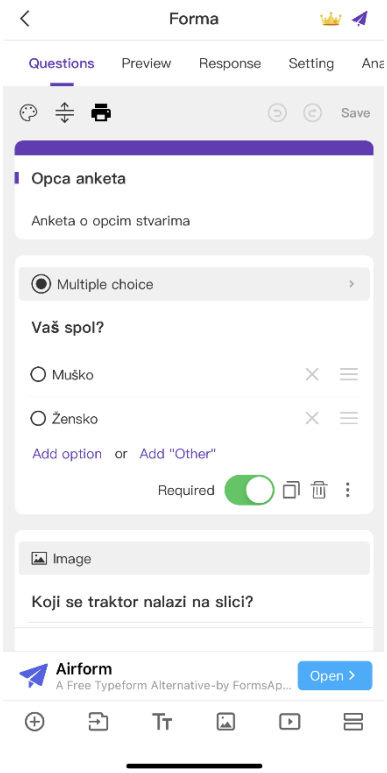
AttaPoll je aplikacija koja pruža jednaku funkcionalnost kao i aplikacija iz rada, omogućujući popunjavanje anketa. Ovo je aplikacija u kojoj velike firme dodjeljuju ankete korisnicima, te korisnici bivaju plaćeni nakon popunjavanja ankete. U postavkama aplikacije se može odrediti koliko se anketa dnevno želi primati i njihovu duljinu. AttaPoll, identično kao i aplikacija u radu, ima početnu stranicu na dodijeljenim anketama, što je još jedna sličnost (Sl. 2.3), dodijeljivanje anketa od strane drugih.



Sl. 2.3. Sučelje aplikacije AttaPoll

2.4. Forms App

Forms App je mobilna aplikacija koja omogućuje svojim korisnicima izradu i upravljanje obrascima i anketama putem mobilnog uređaja. Služi isključivo za kreiranje anketa, odnosno obrazaca, te prikupljanje odgovora istih. Glavna razlika, kao i u prethodnim aplikacijama, bila bi funkcionalnost ove aplikacija da kreira ankete, dok je funkcionalnost aplikacije opisane u radu isključivo za popunjavanje ankete, te pregled istih. Na slici 2.3. se mogu primijetiti sličnosti između ove aplikacije i aplikacije koja je opisana u radu. Obe aplikacije nude iste tipove pitanja i odgovora, uključujući unos teksta, mogućnost višestrukog odgovora, kao i odgovaranje na pitanja uz priloženu sliku.



Sl. 2.4. Sučelje aplikacije Forms App za kreiranje ankete

3. ZAHTJEVI ZA PROGRAMSKO RJEŠENJE

S obzirom na karakteristike aplikacije i funkcionalnosti koje su navedene ranije, u ovom poglavlju su prikazani zahtjevi programskog rješenja. Zahtjevi za programsko rješenje mogu se podijeliti na funkcionalne i nefunkcionalne zahtjeve. Funkcionalni zahtjevi proizlaze iz tražene funkcionalnosti sustava i eksplicitno su navedeni, dok nefunkcionalni zahtjevi obuhvaćaju aspekte koji su ključni za ispravno funkcioniranje sustava kao cjeline, pridonose poboljšanju performansi i korisničkog iskustva. U ovom slučaju funkcionalni zahtjevi su: registracija korisničkog računa, pregled dostupnih anketa, ispunjavanje anketa te pohrana rezultata istih.

3.1. Registracija i prijava korisničkog računa

Korisnička registracija predstavlja ključni korak u pružanju pristupa mobilnoj aplikaciji. Putem registracije, korisnici će moći stvoriti svoj vlastiti korisnički račun kako bi mogli u potpunosti iskoristiti sve funkcionalnosti aplikacije. Registracija korisničkog računa pruža korisnicima mogućnost personalizacije, pohranjivanja preferencija i pristupa relevantnim informacijama. Kroz jednostavan proces registracije, korisnici će biti u mogućnosti unijeti svoje osnovne podatke, kao što su ime, prezime, korisnično ime, e-mail adresa i lozinka. Ovaj korisnički račun omogućit će korisnicima prijavu i pristup aplikaciji s jednostavnim korisničkim vjerodajnicama koje su sigurno pohranjene u bazi podataka. Prijavom na svoj korisnički račun korisnici potvrđuju svoj identitet putem unesenih korisničkih vjerodajnica, te će isti biti u mogućnosti brzo pristupiti svojim anketama, ispunjavati ih, pregledati rezultate,i ostvariti sve interakcije unutar aplikacije.

3.2. Pregled dostupnih anketa

Pregled dostupnih anketa omogućava korisnicima da istraže različite ankete dostupne u aplikaciji. Ova funkcionalnost omogućuje korisnicima da pregledaju naslove anketa i kratki opis. Korisnici će biti u mogućnosti vidjeti informacije o otvorenim anketama, koje su dostupne za ispunjavanje, kao i informacije o već ispunjenim anketama na temelju njihovog korisničkog računa. Pregled dostupnih anketa omogućuje korisnicima da odaberu ankete koje ih zanimaju i sudjeluju u njima putem aplikacije. Ova funkcionalnost pruža korisnicima mogućnost aktivnog sudjelovanja u anketiranju i doprinosi njihovom iskustvu u aplikaciji.

3.3. Ispunjavanje anketa i pohranjivanje rezultata

Korisnici će moći pristupiti dostupnim anketama putem mobilne aplikacije i ispuniti odgovore na postavljena pitanja. Unos odgovora bit će prilagođen vrsti ankete, uključujući odabir višestrukih odgovora, unos teksta ili pridruživanje slika. Nakon što korisnik ispuni anketu, njegovi odgovori bit će pohranjeni u bazu podataka. Ova funkcionalnost omogućava analizu rezultata anketa i generiranje statističkih izvještaja radi dobivanja uvida u mišljenja i preferencije korisnika. Pohranjivanje rezultata provodi se sigurno i pouzdano, uz zaštitu privatnosti korisnika. Rezultati anketa bit će pridruženi odgovarajućim korisničkim računima radi osiguranja povjerljivosti informacija.

Ispunjavanje anketa i pohranjivanje rezultata doprinose interaktivnom iskustvu korisnika i omogućuju analizu prikupljenih podataka.

3.4. Pregled ispunjenih anketa

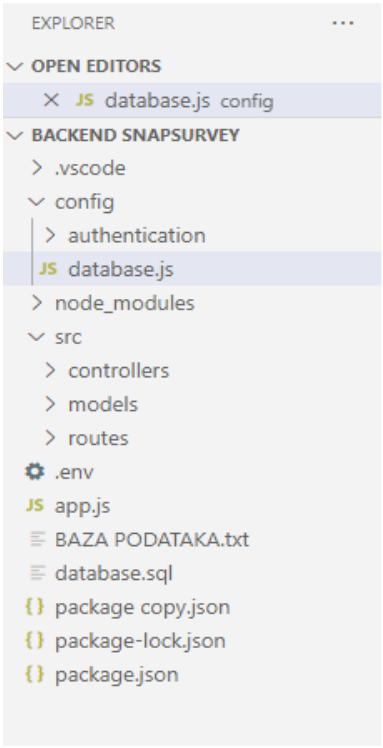
Pregled ispunjenih anketa, pokraj pregleda dostupnih anketa, omogućuje pregled korisnikovih odgovora na postavljena pitanja ankete. Ova funkcionalnost doprinosi boljem razumijevanju vlastitih odgovora, potiče povjerenje korisnika u istraživanje te unapređuje kvalitetu podataka. Važno je napomenuti da korisnik nije u mogućnosi mijenjati svoje odgovore.

4. ALATI I TEHNOLOGIJE

U razvoju mobilne aplikacije i *backend* sustava korišteni su različiti alati i tehnologije koje pružaju skalabilnost, sigurnost i efikasnost. *Backend* dio koda je izrađen u Node.js-u, koristeći Express okruženje. Express je poznati okvir koji omogućava razvoj brzih i skalabilnih web aplikacija. Podaci se pohranjuju i upravljaju pomoću PostgreSQL relacijske baze podataka koja pruža napredne mogućnosti upita. Za autentifikaciju korisnika korišten je Firebase, platformu koja pruža još mnoge razne usluge. Za *frontend* razvoj mobilne aplikacije korištena je Android platforma i programski jezik Kotlin. Android pruža robustno okruženje za razvoj mobilnih aplikacija, dok je Kotlin moderni programski jezik koji je interoperabilan s Java jezikom i pruža poboljšane performanse. Integracija svih ovih alata i tehnologija omogućuje izgradnju cjelovitog, pouzdanog i intuitivnog mobilnog iskustva za korisnike, uz brzu i sigurnu obradu podataka na *backendu*.

4.1. Visual Studio Code

Visual Studio Code (VS Code) je integrirano razvojno okruženje (eng. *IDE*) koje je razvila tvrtka Microsoft. Iako je besplatno, VS Code pruža bogate značajke i funkcionalnosti koje omogućavaju učinkovit i praktičan razvoj softvera. To je popularan izbor među programerima zbog svoje jednostavnosti korištenja, brzine izvršavanja i fleksibilnosti. VS Code podržava širok spektar programskih jezika, uključujući JavaScript, TypeScript, Python, Java, C#, C++, HTML, CSS i mnoge druge. Uz bogatu sintaksnu podršku, automatsko završavanje koda, ugrađeno nadziranje grešaka i pametno pretraživanje, VS Code olakšava pisanje i uređivanje koda kao što je spomenuto u [1]. U VS codeu pisan *backend* dio, te je na slici 4.1 prikazan primjer dokumenata projekta, i organizacije koda.



Sl. 4.1. Primjer organizacije datoteka u Visual Studio Code

4.1.1. Node.js

Node.js je platforma na strani poslužitelja omotana oko JavaScript jezika za izgradnju skalabilnih aplikacija usmjerenih na događaje. Node.js nije napisan na JavaScriptu, ali on koristi JavaScript kao interpretacijski jezik za obradu zahtjeva na strani poslužitelja [2]. Bitna stvar za napomenuti je da se Node.js ne može koristiti samostalno, već je za njegovo korištenje potrebno specijalno okruženje u vidu „JavaScript runtime environment“. Neki od mnogobrojnih frameworka za rad u Node.js bi bili: Express, Koa, Total.js, Sails...

4.1.2. Express

Express je minimalistički web okvir za Node.js koji pruža jednostavno i intuitivno sučelje za razvoj web aplikacija. On dodaje dodatnu apstrakciju i olakšava upravljanje rutama, obradu zahtjeva i odgovora, upravljanje srednjim slojevima i druge zajedničke funkcionalnosti web aplikacija. Express je fleksibilan i omogućuje prilagodbu prema specifičnim potrebama projekta. Prema [3], kombinacija Node.js-a i Expressa pruža brz i efikasan način za razvoj skalabilnih i pouzdanih *backend* aplikacija. Zahvaljujući velikom broju dostupnih paketa i modula putem NPM (eng. *Node Package Manager*), programerima je omogućen pristup raznim dodacima i bibliotekama koje olakšavaju razvoj i proširuju funkcionalnost aplikacija.

Korištenjem Node.js-a i Expressa, moguće je izgraditi API-je koji omogućuju komunikaciju između *frontend* i *backend* dijela aplikacije. Također je moguće obavljati različite operacije s bazom podataka, kao što su dohvaćanje, ažuriranje i pohrana podataka, provođenje poslovne logike, upravljanje autentifikacijom korisnika te druge *backend* funkcionalnosti.

4.2. PostgreSQL

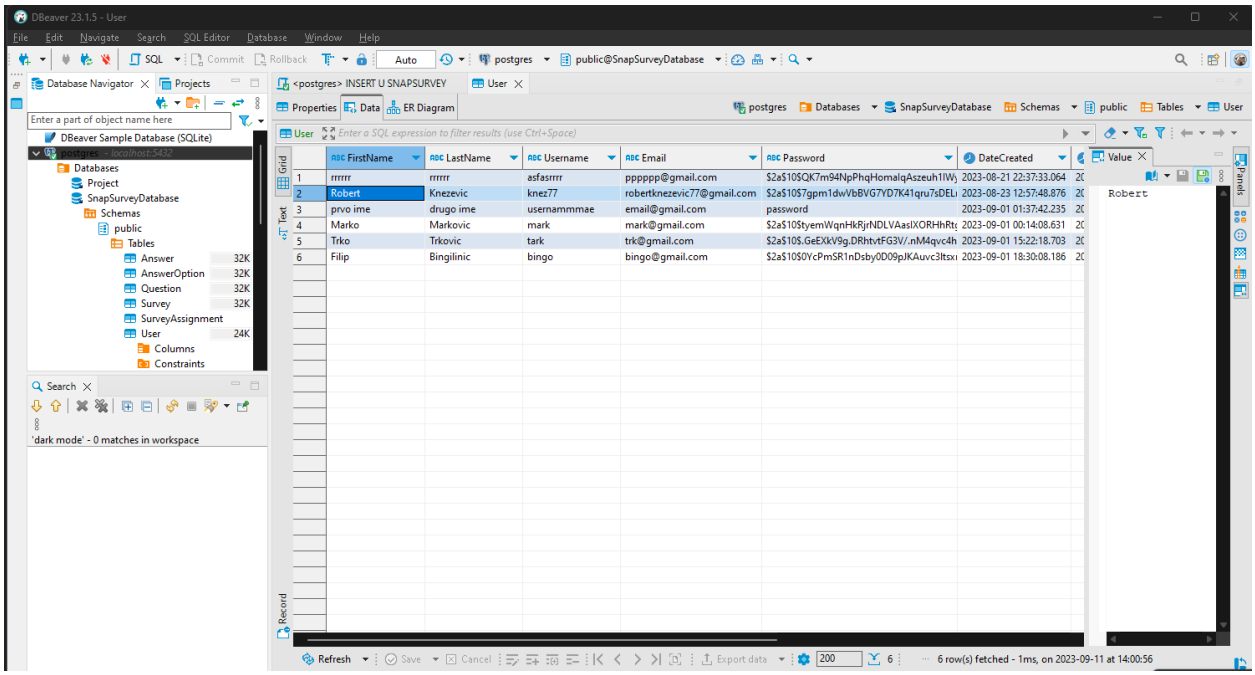
PostgreSQL je napredni objektno-relacijski sustav za upravljanje bazama podataka (eng. *DBMS*) koji se ističe visokom pouzdanošću, proširivošću i podrškom za kompleksne podatkovne operacije [6]. Ovaj open-source alat nudi širok spektar mogućnosti za rukovanje podacima, uključujući napredne tehnikama indeksiranja i podršku za geografske informacije. Snažna zajednica i bogata dokumentacija čine PostgreSQL popularnim izborom za aplikacije svih razmjera i složenosti

4.2.1. Dbeaver

DBeaver predstavlja SQL klijent i alat za administraciju baze podataka. To je desktop aplikacija napisana na Javi zasnovana na platformi Eclipse. *Community Edition* (*CE*) DBeavera besplatan je softver otvorenog koda koja se distribuira pod licencom Apache.

Ovaj univerzalni klijent omogućuje svojim korisnicima da učinkovito rade s različitim sustavima za upravljanje bazama podataka, uključujući PostgreSQL, MySQL, SQLite, Oracle i mnoge druge [7]. Njegova intuitivna grafička sučelja omogućavaju jednostavan pregled i upravljanje tablicama, indeksima, procedurama, korisnicima i drugim objektima u bazi podataka. DBeaver pruža snažan SQL alat koji olakšava izvođenje upita i formatiranja SQL koda. Dodatno, DBeaver podržava geografske informacije, što ga čini odličnim izborom za rad s prostornim podacima u PostgreSQL bazi. Ove značajke su glavni razlog zašto je odabran PostgreSQL u suradnji s Dbeaverom.

Na slici 4.2 nalazi se izgled sučelja Dbeavera, s lijeve strane se mogu vidjeti projekti, te tablice ovoga projekta. Također se može vidjeti podaci tablice korisnika (User).



Sl. 4.1 Korisničko sučelje DBeavera

4.3. Android studio

Android Studio je službeno integrirano razvojno okruženje za razvoj Android aplikacija. Pruža programerima alate i resurse za izradu, testiranje i distribuciju aplikacija za Android platformu. Sadrži grafički korisnički sučeljni uređivač, emulator za testiranje, debugger i integraciju s Android SDKom [4]. Koristi se za razvoj aplikacija za Android uređaje i pruža programerima sve potrebne alate za učinkovit i brz razvoj mobilnih aplikacija.

4.3.1. Kotlin

Kotlin je moderan programski jezik koji se koristi za razvoj Android aplikacija. Razvijen od strane JetBrainsa, Kotlin je postao službeni jezik za Android razvoj od 2017. godine. Kotlin je interoperabilan s postojećim Java kodom, što znači da se može koristiti za pisanje novih aplikacija ili za poboljšanje postojećeg Java koda.

Kotlin donosi mnoge prednosti programerima Android aplikacija. To uključuje smanjenje količine pisanog koda, poboljšanu sigurnost tipova, podršku za funkcionalno programiranje, jednostavnu integraciju s Android Studio IDEom i podršku za moderne Android značajke poput korutina [5]. Kotlin također donosi bolju čitljivost koda, smanjuje greške u kodiranju i olakšava održavanje aplikacija.

Kotlin se koristi za izradu raznih vrsta Android aplikacija, uključujući mobilne igre, aplikacije za produktivnost, e-trgovinu, društvene mreže i mnoge druge. Njegova popularnost među programerima Androida stalno raste zbog svoje jednostavnosti, izražajnosti i podrške zajednice. Kotlin je odličan izbor za programere koji žele efikasno razvijati Android aplikacije, poboljšati produktivnost i povećati kvalitetu svojeg koda.

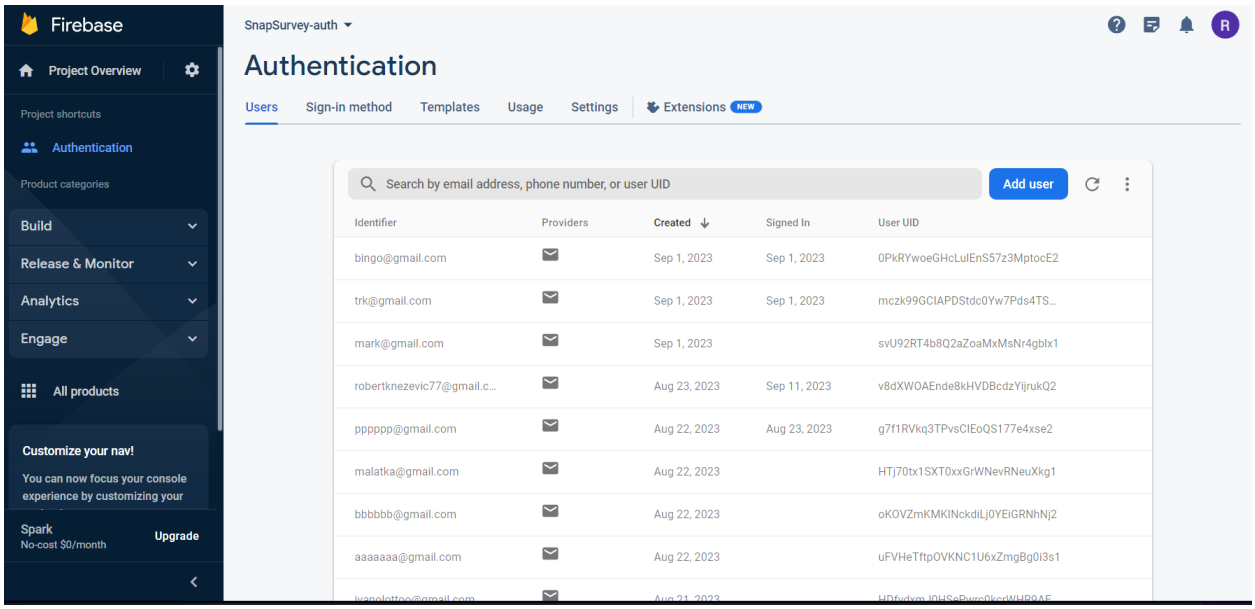
4.4. Firebase

Firebase je sveobuhvatan razvojni alat razvijen od strane Googlea koji pruža kompletnu infrastrukturu i alate za izradu raznovrsnih aplikacija, uključujući mobilne i web aplikacije. On funkcionira kao *BaaS* (eng. *Backend-as-a-Service*), što znači da pruža kompletnu pozadinsku infrastrukturu i jednostavno korisničko sučelje za korištenje različitih usluga [8]. To znači da korisnik ne mora brinuti o implementaciji i upravljanju vlastitom bazom podataka na poslužitelju.

Firebase obuhvaća niz značajki, uključujući autentikaciju i registraciju korisnika, strojno učenje, stvarno-vremensku bazu podataka, sustav za slanje poruka i mnoge druge. Ova funkcionalnost značajno olakšava razvojni proces.

Istaknuta karakteristika Firebase-a je mogućnost analitike koja omogućava praćenje aktivnosti korisnika u aplikaciji. Pomoću nje, možete pratiti različite podatke, uključujući broj korisnika, razmjenu podataka i druge metrike koje vam pomažu bolje razumjeti i poboljšati učinkovitost vaše aplikacije.

U ovom projektu firebase je korišten isključivo za autentifikaciju korisnika, jer ono pruža napredni sustav za autentifikaciju korisnika, te omogućuje sigurnu prijavu i registraciju korisnika, i još mnoge druge funkcionalnosti [9]. Na slici 4.3. se vidi kako izgleda sučelje firebasea koje pohranjuje korisnika.

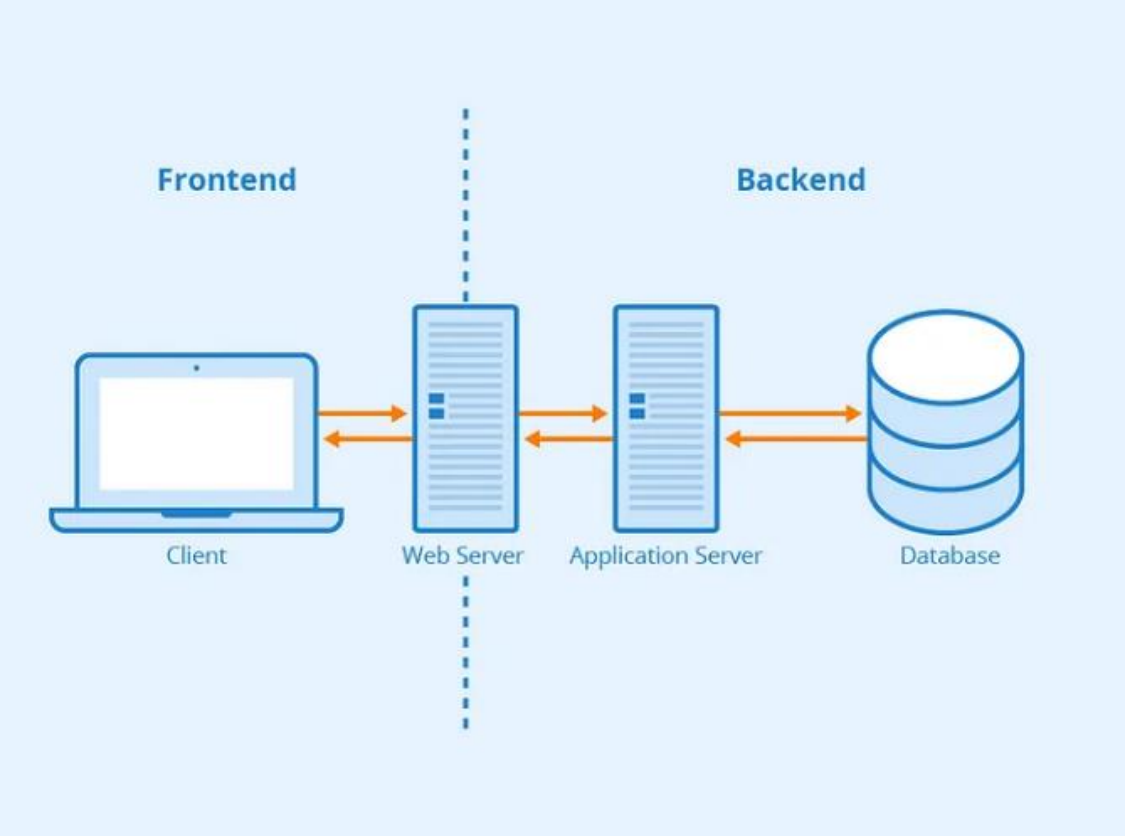


Sl. 4.2. Firebase sučelje korišteno za autentifikaciju

5. IZRADA PROGRAMSKOG RJEŠENJA

U ovom poglavlju predstavljena je arhitektura aplikacije te programsko rješenje aplikacije iz rada. Obuhvaćene su sve ključne komponente i tehnologije koje su korištene u izradi aplikacije. Za uspješno i kvalitetno kreiranje aplikacije korišteni su razni tutoriali s YouTubea, te mnoštvo drugih web stranica kako bi se uspješno implementirali sve značajke ovog završnog rada. Arhitektura aplikacije u radu sastoji se od korisničkog sučelja ili *frontenda* i poslužiteljske strane, odnosno *backenda*.

Razvoj aplikacije iz ovoga završnog rada podijeljen je na dva glavna dijela. Prvo je to razvoj poslužiteljske strane, gdje je prije svega izrađena PostgreSQL baza podataka te onda API pomoću kojeg će se dohvaćati i prerađivati podaci iz implementirane baze podataka. Također je implementirana autentifikacija korisnika, odnosno prijava i registracija korisnika, pomoću alata za prijavu iz Firebase programskog okvira. Drugi je dio razvoj korisničkog sučelja sa svim svojim funkcionalnostima, estetikom i detaljima. U dijelu razvoja korisničkog sučelja također spada i spajanje na API iz kojeg će se dobivati potrebne informacija o anketama i njihovim pitanjima. Na slici 5.1. je jasno prikazana komunikacija između *backenda* i *frontenda*.



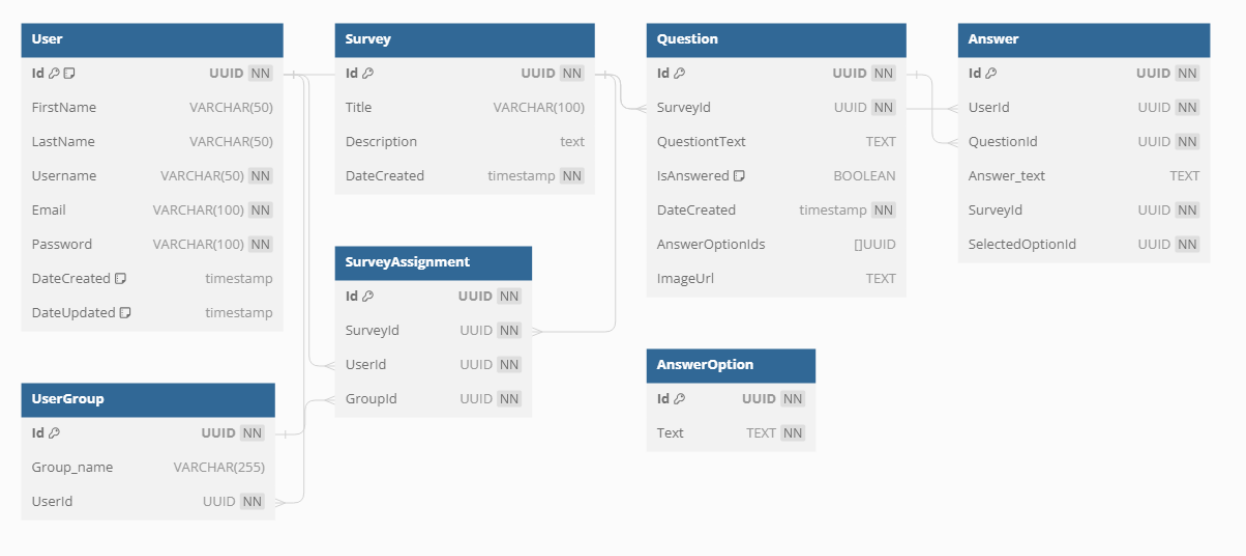
Sl. 5.1. Razlika između *frontenda* i *backenda*

5.1. Baza podataka

Izrađena shema PostgreSQL baze podataka može se vidjeti na slici 5.2. Baza podataka se sastoji od sedam tablica:

- korisnici aplikacije (User)
- detalji ankete (Survey)
- pitanja za ankete (Question)
- odgovori na pojedina pitanja (Answer)
- tablica koja služi za pridruživanje ankete korisnicima, ili grupi korisnika (SurveyAssignment)
- tablice za pridruživanje korisnika određenoj grupi (UserGroup)
- look-up tablice za slučaj kada pitanja imaju mogućnost višestrukog odgovora(AnswerOption).

Ova struktura podataka odražava specifičan zadatak aplikacije, koji se fokusira isključivo na ispunjavanje anketa i pregled istih. Iz prethodno navedenog razloga, podaci o anketama, pitanja i pridruživanje ankete korisnicima, kao i pridruživanje korisnika u određene grupe, su uneseni ručno u bazu podataka od strane kreatora. [10]



Sl. 5.2. Baza podataka dohvaćena sa stranice dbdiagram.io

5.2. API za dohvat podataka

API je skup definiranih metoda, protokola i alata koji olakšavaju komunikaciju između različitih sustava. Ovisno o potrebama, API se može implementirati korištenjem različitih tehnologija i protokola. Postoji raznovrsnost vrsta i svrha APIja. U okviru ovog projekta, izrađen je API koji omogućuje pozivanje funkcija s poslužiteljske strane aplikacije kako bi se pristupili podacima koji

se zatim prikazuju na *frontendu*. Korištenjem ovog APIja, u *frontendu* postiže se integracija između *frontend* i *backend* dijela aplikacije, omogućujući pozivanje funkcija koje se izvršavaju na poslužiteljskoj strani.

5.2.1. Uspostavljanje veze s bazom podataka

Prva stvar koju je bilo potrebno napraviti je uspostaviti vezu s bazom podataka. Za uspostavljanje veze s bazom podataka u Node.js obično se koristi odgovarajući paket, poput "pg" za PostgreSQL bazu podataka. Kroz konfiguraciju, kao što se vidi na prog. kodu 5.1., definiraju se postavke za povezivanje s bazom podataka, poput naziva hosta, korisničkog imena, lozinke i naziva baze podataka [11].

database.js

config > database.js > ...

```
1   const { Pool } = require('pg');
2
3   const pool = new Pool ({
4       host: 'localhost',
5       user: 'postgres',
6       password: "knezevic10",
7       database: 'SnapSurveyDatabase',
8       port: 5432
9   });
10
11    pool.connect((err, client, done) => {
12      if (err) {
13          console.error('Error connecting to database:', err);
14          process.exit(1);
15      }
16      console.log('Connected to database');
17      done();
18  });
19
20    module.exports = pool;
```

Prog. kod 5.1. Povezivanje s bazom podataka u Express.js

5.2.2. Definiranje ruta, kontrolera i modela

Nakon uspostavljanja veze s bazom podataka, kontroleri, modeli i rute se koriste za organizaciju logike aplikacije. Kontroleri sadrže logiku za obradu zahtjeva, modeli predstavljaju strukturu podataka i pružaju metode za interakciju s bazom podataka, dok rute definiraju *URLove* (eng. *Uniform Resource Locator*) i odgovarajuće metode za obradu zahtjeva. Uobičajena praksa je da

rute pozivaju odgovarajuće kontrolere, koji koriste modele za pristup i manipulaciju podacima u bazi podataka. Kontroleri zatim obrađuju zahtjeve, pripremaju podatke i šalju odgovor klijentu. Na ovaj način, struktura aplikacije u Node.js omogućuje organiziranu i modularnu izgradnju, gdje se logika aplikacije dijeli na više slojeva radi bolje čitljivosti i održivosti koda.

Za pobliže razumijevanje o čemu se tu radi, u narednom tekstu biti će prikazan primjer modela, kontrolera i rute za primjer dohvaćanja svih korisnika. Za početak, potreban je UserModel u kojem se definira konstruktor za kreiranje objekta User. U njemu su definirana svojstva, i dobivaju se vrijednosti koje su proslijeđene kao argument. Sljedeća stvar u UserModelu je statička funkcija za dohvaćanje svih korisnika, ona se poziva direktno na klasi, bez potrebe za stvaranjem novih instanci, funkcija je također i asinkrona. Sve ove funkcionalnosti su prikazane u prog. kodu 5.2.

JS UserModel.js

src > models > JS UserModel.js > ...

```
1   const e = require('express');
2   var pool = require('../../config/database');
3
4   var User = function (user) {
5       this.id = user.Id;
6       this.firstName = user.FirstName;
7       this.lastName = user.LastName;
8       this.username = user.Username;
9       this.email = user.Email;
10      this.password = user.Password;
11      this.loginTime = new Date();
12      this.logoutTime = new Date();
13  }
14
15  User.getAllUsers = async () => {
16      try {
17          const queryResult = await pool.query('SELECT * FROM \"User\"');
18          console.log('Users fetched successfully!!');
19          return queryResult.rows;
20      } catch (error) {
21          console.log("Error while getting users", error);
22          throw error;
23      }
24  }
```

Prog. kod 5.2. Datoteka UserModel.js

U UserControlleru obično se definiraju funkcije koje obrađuju zahtjeve (request) vezane uz korisnike (users) u aplikaciji. Te funkcije se nazivaju kontroleri i obavljaju određene radnje na

temelju primljenog zahtjeva te vraćaju odgovor. Potrebno je samo „importati“ UserModel kako bi se mogle koristiti funkcije istoga (Prog. kod 5.3.).

UserController.js

src > controllers > UserController.js > getUserById > getUserById

```
1   const UserModel = require('../models/UserModel');
2
3   exports.getAllUsers = async (req, res) => {
4       try {
5           const users = await UserModel.getAllUsers();
6           console.log("Users", users);
7           res.send(users);
8       } catch (error) {
9           res.status(500).send(error.message);
10      }
11  }
12
```

Prog. kod 5.3. Datoteka UserController

Da bi sve to funkcioniralo u praksi, potrebno je još stvoriti takozvane „*endpointe*“ (Prog. kod 5.4.), odnosno rute za dane funkcije. *Endpoint* predstavlja URL putem kojeg se može pozvati funkcija. U rutama se implementira UserController i preko njega se pozivaju funkcije.

UserRoute.js

src > routes > UserRoute.js > ...

```
1   const express = require('express');
2   const router = express.Router();
3   const userController = require('../controllers/UserController');
4
5   router.get('/', userController.getAllUsers);
6   router.get('/:id', userController.getUserById);
7   router.post('/', userController.createNewUser);
8   router.delete('/:id', userController.deleteUser);
9
10  module.exports = router;
```

Prog. kod 5.4. Datoteka UserRoute

Na kraju je implementiran glavni skriptni dokument koji se koristi za pokretanje *backend* dijela aplikacije. Kada se kaže da pokreće backend, misli se na to da server sluša zahtjeve korisnika i pruža odgovore.Ovaj dokument sadrži kod koji postavlja server, definira rute, obrađuje zahtjeve i

upravlja logikom iza aplikacije. U slučaju ove aplikacije, glavni skriptni dokument bio bi „app.js“, prikazan na slici 5.7.

app.js

```
const express = require("express");
const bodyParser = require("body-parser")
const app = express();

const authRoutes = require('./src/routes/AuthRoute');
const surveyRoutes = require('./src/routes/SurveyRoute');
const answerRoutes = require('./src/routes/AnswerRoute');
const surveyAssignmentRoutes = require('./src/routes/SurveyAssignmentRoute');
const questionRoutes = require('./src/routes/QuestionRoute');
const answerOptionRoutes = require('./src/routes/AnswerOptionRoute');

const userRoutes = require('./src/routes/UserRoute');

const port = process.env.APP_PORT || 3000;

app.use(express.json());
app.use(bodyParser.urlencoded({ extended: true }));

app.use('/api/users', userRoutes);
```

Prog. kod 5.5. Dokument app.js

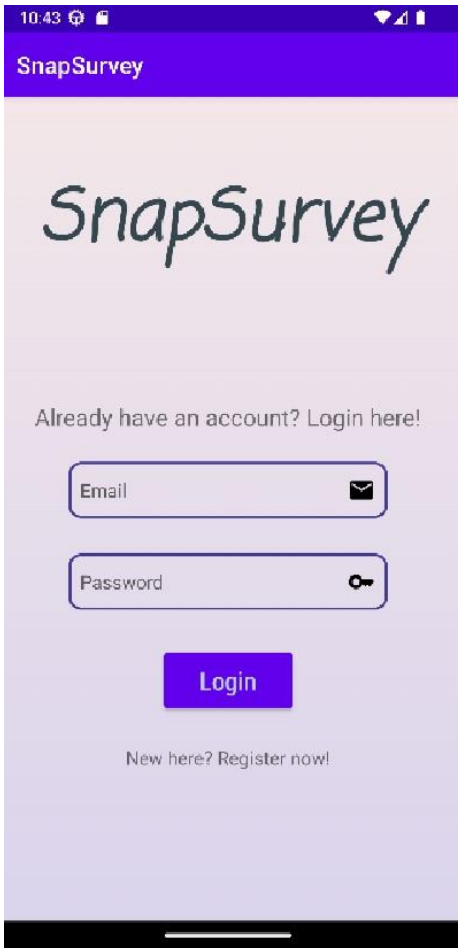
5.3. Android studio Kotlin

U ovome dijelu rada će biti prikazan *frontend* aplikacije. *Frontend* bi bio dio aplikacije koji sadrži korisničko sučelje te se bavi njegovim vizualnim izgledom, estetikom i interaktivnim elemntima. Cijeli razvoj *frontenda* izrađen je pomoću programa Android Studio u programskom jeziku Kotlin [12].

U sljedećim potpoglavljima detaljnije je opisano korisničko sučelje ove aplikacije i njene interaktivnosti.

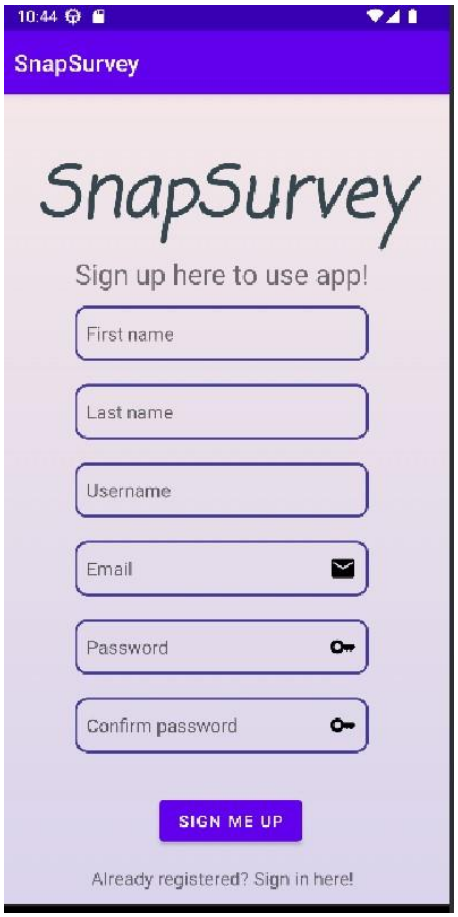
5.3.1. Korisničko sučelje

U ovome potpoglavlju pokazan je dizajn aplikacije i njene funkcionalnosti. Prilikom pokretanja aplikacije, aplikacija od korisnika traži prijavu. Izgled polaznog zaslona, koji je ujedno i zaslon za prijavu korisnika, prikazan je na slici 5.3.



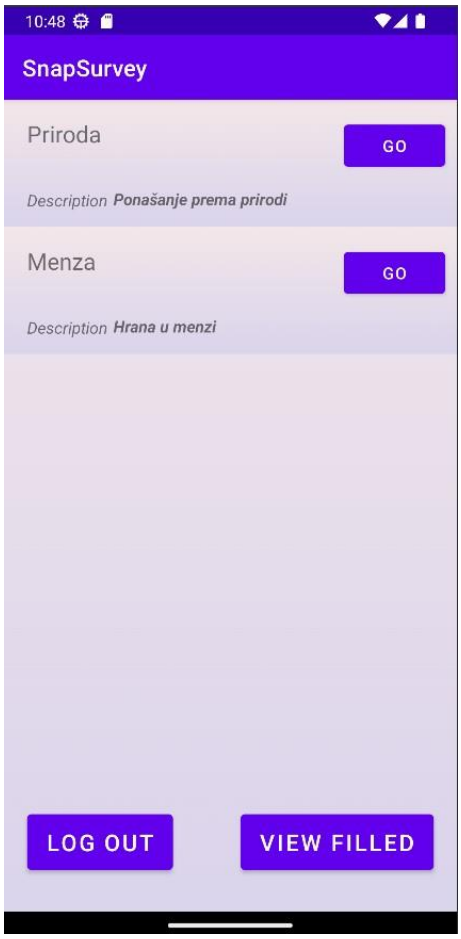
Sl. 5.3. Početni zaslon

Ako kojim slučajem korisnik prvi puta koristi aplikaciju, i primoran je registrirati se, to može napraviti na jednostavan način. Klikom na tekst „New here? Register now!“, korisnik biva odveden na zaslon za registraciju korisnika koji je prikazan na slici 5.4.



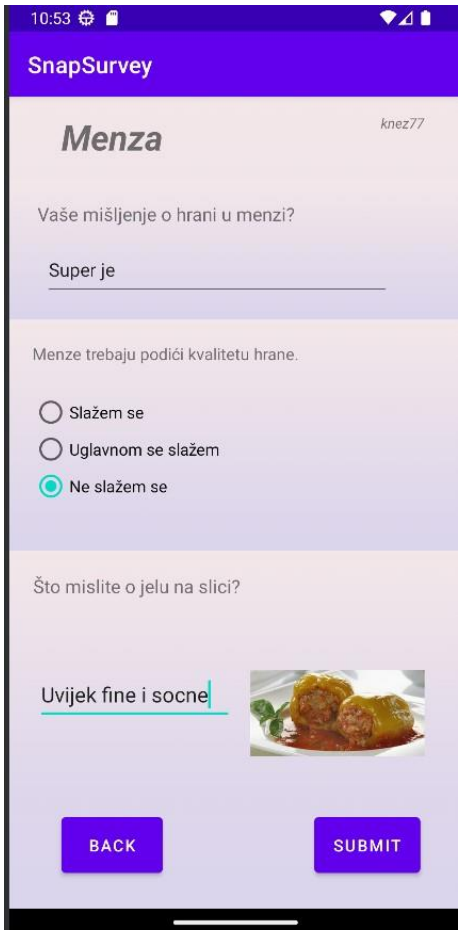
Sl. 5.4. Zaslon za registraciju

Sada kada korisnik ima račun, slijedi prijava. Nakon uspješne prijave, korisnik je odveden na početni zaslon, slika 5.5. Početni zaslon predstavljaju ankete koje su dodijeljene korisniku na ispunjavanje, te jednostavnim klikom gumba „GO“ korisnik biva preusmjeren na ispunjavanje ankete. Korisnik također na početnom zaslonu ima mogućnost odjave, te mogućnost pregleda ispunjenih anketa.



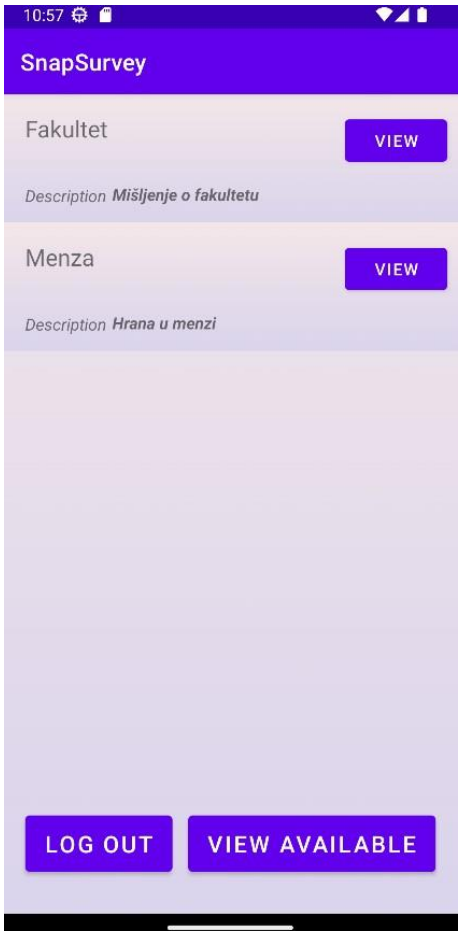
Sl. 5.5. Početni zaslon aplikacije SnapSurvey

Kada korisnik klikne gumb „GO“ na anketu imenom „Menza“, odvodi se na zaslon za popunjavanje te ankete, a na slici 5.6. se vidi izgled tog zaslona i način ispunjavanja ankete. Ovaj zaslon sadrži ime ankete u gornjem lijevom kutu, te korisničko ime korisnika koji je trenutno prijavljen i koji trenutno ispunjava anketu.



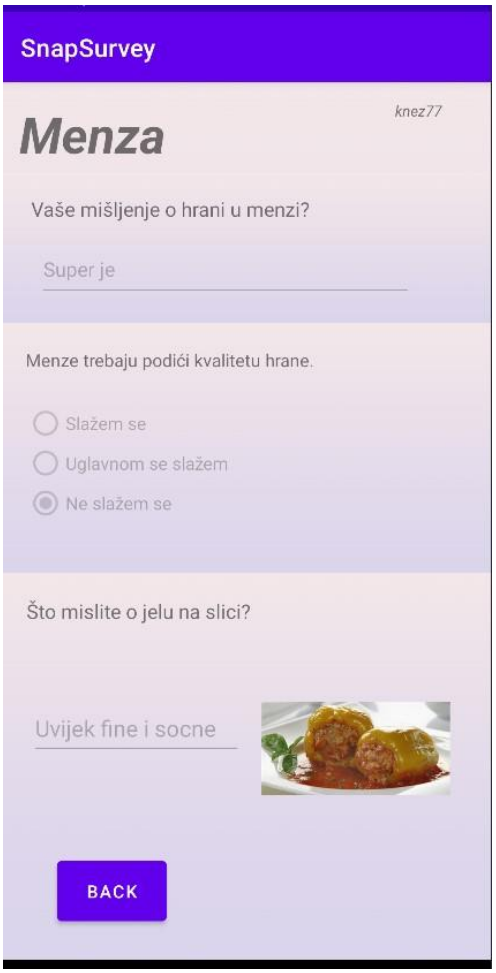
Sl. 5.6. Popunjavanje ankete

Nakon što korisnik ispuni anketu, klikom na gumb „SUBMIT“, odgovori su spremljeni u bazu podataka i korisnik je vraćen na zaslon za odabir anketa. Sada kada je ova anketa ispunjena, korisnik ima mogućnost pregledati svoje odgovore, bez mogućnosti uređivanja istih. Kada je korisnik na početnom zaslonu, klikom na "VIEW FILLED" preusmjerava se na ankete koje je sam ispunio. Tada se gumb koji je na početku bio "VIEW FILLED" mijenja u "VIEW AVAILABLE", koji sada služi za povratak na početni zaslon. Zaslon na kojem su prikazane ispunjene ankete može se vidjeti na slici 5.7. Također, gumb svake ankete je promijenjen. Na početnom zaslonu, gumb je bio „GO“. Sada je taj gumb postao „VIEW“, te služi za pogled na ispunjenu anketu i njene odgovore.



Sl. 5.7. Pregled ispunjenih anketa

Posljednji zaslon aplikacije pruža korisniku uvid u njegove odgovore. Također je i na ovom zaslonu prikazano korisničko ime prijavljenog korisnika u desnom gornjem kutu, što se vidi na slici 5.8.



Sl. 5.8. Prikaz ispunjene ankete "Menza"

6. ZAKLJUČAK

Cilj ovoga završnog rada je bio razviti aplikaciju koja podržava popunjavanje anketa od strane korisnika, te pregled istih. Sam koncept aplikacije se čini prilično jednostavan, ali njen razvoj zahtijevao je znatan napor. Pogotovo za nekoga tko prvi puta radi full-stack aplikaciju. „Samo“ popunjavanje anketa i mogućnost pregleda istih bio je izazovan, jer je bilo potrebno kreirati funkcionalni API, i razne funkcije za dohvaćanje podataka ove aplikacije koja se sastoji od 7 tablica. Bitno je reći da je uspješno riješena sva problematika ovoga rada, ali sigurno da ima prostora za poboljšati aplikaciju. Aplikacija bi se mogla nadograditi tako da se i korisniku dopusti kreiranje anketa, ali za potrebe završnoga rada, i ovo je bilo dovoljno. Također, ne bi bilo loše ni implementirati navigacijsku traku, koja bi olakšala „kretnju“ aplikacijom. Važno je istaknuti da su korištene raznolike tehnologije, obuhvativši *backend, frontend* i bazu podataka, što je predstavljalo izazov, ali i omogućilo razvoj sveobuhvatne aplikacije.

LITERATURA:

[1] "What is Visual Studio Code? Microsoft’s extensible code editor", dostupno na: https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html

[2] "Šta je node.js? ", dostupno na: https://www.webprogramiranje.org/uvod-u-node-js/

[3] "What is Express Js in Node Js? ", dostupno na: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js

[4]"How to use Kotlin in Android Studio", dostupno na: https://medium.com/@ajaysaini.official/how-to-configure-kotlin-in-android-studio-7fca851ec48e

[5] "Kotlin overview", dostupno na: https://developer.android.com/kotlin/overview

[6] "PostgreSQL: Up and Running" od Regina O. Obe i Leo S. Hsu, knjiga izdana 2017. godine, knjiga dostupna na: https://dokumen.pub/postgresql-up-and-running-a-practical-guide-to-the-advanced-open-source-database-3nbsped-9781491963418.html

[7] "DBeaver, upravljajte različitim vrstama baza podataka na jednostavan način", dostupno na: https://ubunlog.com/bs/dbeaver-upravlja-bazama-podataka/

[8] "Firebase, Google", dostupno na: https://firebase.google.com/

[9] "What is Firebase? All secrets unlocked", G. Batschinski, Back4App, dostupno na: https://blog.back4app.com/firebase/

[10] "Beginning Databases with PostgreSQL: From Novice to Professional, Second Edition", Neil Matthew and Richard Stones, 2005., knjiga dostupna na: https://theswissbay.ch/pdf/Gentoomen%20Library/Databases/postgresql/Beginning%20Databases%20With%20PostreSQL%20-%20From%20Novice%20To%20Professional%2C%202nd%20Ed%202005.pdf

[11] "Web Development with Node and Express: Leveraging the JavaScript Stack", Ethan Brown 2014., knjiga dostupna na: https://github.com/b055/books-1/blob/master/Web-Development-with-Node-and-Express_-2nd-Edition.pdf

[12] "Kotlin in Action", Dmitry Jemerov i Svetlana Isakova, 2017., knjiga dostupna na: https://github.com/cjnhust/ebook_collection/blob/master/Kotlin%20in%20Action.pdf

SAŽETAK

Završni rad fokusira se na razvoj aplikacije za popunjavanje anketa od strane korisnika i pregled istih. Glavni izazovi bili su implementacija sustava za upravljanje korisnicima, omogućiti autentifikaciju putem Firebase Firestorea, dizajn baze podataka koja je mijenjana par puta te razvoj *frontend* i *backend* komponenti. Korištenjem tehnologija poput maloprije spomenutog Firebasea, sveobuhvatnog Visual Studio Codea u razvnojnom okruženju Node.js gdje je korišten Express.js i PostgreSQLa u DBeaveru, *backend* je uspješno i učinkovito kreiran. Ovdje je fokus bio implementirati sve potrebne funkcije za dohvaćanje podataka. Veliku ulogu u implementiranju funkcionalnosti na *backendu* je imao alat Postman koji se koristi za testiranje i razvoj APIja. Nakon završetka *backenda*, prešlo se na *frontend*. Za *frontend* je korišten samo Android Studio i programski jezik Kotlin. Prilično je zgodan jer u sebi ima ugrađen emulator, koji služi kao android uređaj te se na njemu testira, odnosno pokreće aplikacija. Da se rezimira na kraju, aplikacije za anketiranje su poprilično prisutne u današnjom dobu, gdje ih se može instalirati na svoj uređaj u dva klika. Tako da ova aplikacija, na ovakvom tržištu, vjerojatno ne bi bila konkurentna. Bilo bi potrebno duže i kvalitetnije raditi na njoj da bi postojala nekakva nada.

Ključne riječi: anketa, aplikacija, *backend*, *frontend*, razvoj

ABSTRACT

Survey mobile application

The final work focuses on the development of an application for filling out surveys by users and reviewing them. The main challenges were the implementation of the user management system, enabling authentication through Firebase Firestore, the design of the database, which was changed a couple of times, and the development of *frontend* and *backend* components. By using technologies such as the aforementioned Firebase, comprehensive Visual Studio Code in the Node.js development environment where Express.js was used, and PostgreSQL in DBeaver, the *backend* was successfully and efficiently created. The focus here was to implement all the necessary data retrieval functions. The Postman tool, which is used for API testing and development, played a major role in implementing functionality on the *backend*. After finishing the *backend*, we moved on to the *frontend*. Only Android Studio and the programming language Kotlin were used for the *frontend*. It is quite convenient because it has a built-in emulator, which serves as an Android device and is tested on it, that is, the application is launched. To summarize at the end, survey apps are quite present in today's age where you can install them on your device in two clicks. So this application, in this kind of market, would probably not be competitive. It would be necessary to work on it longer and better for there to be any hope.

Key words: application, backend, development, frontend, survey

PRILOZI

Github link na kojem se nalazi implementacija *backenda* i *frontenda* aplikacije: https://github.com/rknezevic/SnapSurvey

ŽIVOTOPIS

Robert Kneževič rođen je 31.12.2000. u Slavonskom Brodu. Osnovnu školu je pohađao u Babinoj Gredi, nakon nje upisuje srednju tehničku školu, i stječe srednju stručnu spremu kao Računalni tehničar za strojarstvo. Poslije završetka srednje škole odlazi sa obitelji u Švedsku, gdje je radio godinu dana. Nakon vremena provedenog u Švedskoj, odluči se vratiti kako bi upisao sveučilišni preddiplomski studij računarstva, smjer računalno inženjerstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

_____

Potpis autora