



Tehnička i Korisnička dokumentacija



Autori: Karlo Kajba Šimanić, Karlo Smirčić i Toni Kukec

Mentor: prof. Goran Boneta

Gimnazija Andrije Mohorovičića Rijeka

Sadržaj

Uvod.....	4
O autorima.....	4
Karlo Smirčić.....	4
Toni Kukec.....	4
Karlo Kajba Šimanić	5
Što je to Jobscape?.....	6
Planovi za daljnji razvoj.....	6
Korisnička dokumentacija.....	7
Kreiranje računa / registracija	7
Početna stranica	8
Pretraga objava.....	9
Kreiranje nove objave	10
Razgovori / chat stranica	11
Profil	13
Uređivanje profila / pregledavanje vlastitih objava.....	14
Postavke	15
Tehnička dokumentacija	16
Pokretanje	16
Frontend	16
Dijagrami	17
Naslovna stranica.....	17
Pretraga objava	17
Objava oglasa.....	17
Uređivanje oglasa	17
Stranica s razgovorima	18
Stranica s blokiranim korisnicima	18
Profilna stranica.....	19
Stranica za uređivanje profila	19
Postavke	19
Odjava.....	19
Prijava	20

Registracija.....	20
Backend.....	21
Docker containeri	21
FastAPI server.....	21
MySQL	24
Komunikacija među korisnicima.....	25
Testiranje	26

Uvod

O autorima

Karlo Smirčić

Programirati sam počeo u 5. razredu osnovne škole. Tada sam znao programirati samo u FMS Logu te sam išao na natjecanja. Dvije godine kasnije, FMS Logo mi je dosadio te sam htio učiti nešto novo, stoga sam se upoznao s Arduinom koji me odmah zainteresirao. U 8. razredu, išao sam na natjecanje iz Automatike (Arduino) te ostvario 2. mjesto na državnom natjecanju. Nakon završetka osnovne škole upisao sam Gimnaziju Andrije Mohorovičića. Tamo sam se upoznao s kolegama i Pythonom. Naime, Python me jako zainteresirao te mi omogućio izradu različitih projekata. Moje se poznavanje Pythona poboljšavalo kroz godine te sam nedavno imao priliku okušati se u nekim težim projektima (Karlo Kajba Šimanić i ja napravili smo AI koji prepoznaće tip bolesti pluća samo po rendgenskoj slici). Prošle godine smo Karlo Kajba Šimanić i ja prisustvovali na natjecanju iz razvoja softvera gdje smo predstavili našu ideju Logicam. Tada sam prvi put počeo učiti o Dart-u. S obzirom na to da je Karlo pisao većinu korisničkog sučelja, imao sam jako malo iskustva s tim programskim jezikom. Par mjeseci kasnije, ponovno smo krenuli raditi na Logicam-u te sam tako imao prilike više naučiti o Dart-u.

Moja uloga u ovom projektu bila je spajanje frontenda s backendom. Osim toga, moj zadatak je bio postaviti sustav za razgovor među korisnicima.

Toni Kukec

Rođen sam 02.05.2004. godine u Rijeci. Pohađao sam Osnovnu školu Milana Brozovića u Kastvu te sam tamo pokazao velik interes za matematiku i druge prirodne znanosti. Nisam znao koju će srednju školu upisati sve do zadnjeg trenutka, kada sam na državnom natjecanju iz matematike upoznao učenika Gimnazije Andrije Mohorovičića, koji me nagovorio da upišem GAM.

U prvom razredu sam počeo programirati u Pythonu i od tada ga koristim skoro svaki dan. Koristim Python kako bih si olakšao svakodnevne stvari, poput izrade stranice koja omogućava praćenje i prikazivanje školskog rasporeda ili programa za učenje latinskih fraza. Od drugog razreda proučavam etičko hakiranje i zaštitu programa od napada.

U ovom projektu imam glavnu ulogu backend developera. Brinem se o networkingu i sigurnom spremanju podataka na serveru.

Karlo Kajba Šimanić

Prvi put sam u kontakt s programiranjem došao u 3. razredu osnovne škole kada sam slučajno otkrio Scratch. Zavolio sam programirati igrice u tom programu i to me „uvuklo“ u programiranje. Počeo sam sve više obraćati pažnju na detalje u igricama kako bih mogao sam probati rekreirati ono što sam video dok sam igrao. Naravno, tada sam igrao dvodimenzionalne igrice pa je to bilo lagano.

U 5. razredu osnovne škole, upoznao sam se s FMS Logom, a u 6. razredu odlučio sam preći na nešto ozbiljnije – naučiti Unity i C# za izradu igara. Naučio sam osnove, kreirao sam nekoliko igrica, no to mi je dosadilo pa sam odlučio preći na Python. U Pythonu sam počeo raditi na kraju 8. razreda. Preko ljeta napravio sam nekoliko aplikacija poput pretvarača valuta ili vlastitog malog tekst urednika.

Na početku 1. razreda srednje škole, ušao sam dublje u svijet elektronike i robotike, no nisam prestao programirati. Ubrzo sam počeo učiti programske jezike C++ i Arduino. 2020. godine natjecao sam se u razvoju softvera gdje sam s kolegom, Karлом Smirčićem, predstavio našu aplikaciju EEPROM Programmer. Osim toga, razvili smo igricu koju smo kasnije objavili na platformi „Google play“ pod imenom „Jump it“. Početkom 2021. godine, također sam sudjelovao na natjecanju u razvoju softvera na kojem smo Karlo Smirčić i ja prezentirali projekt „Logicam“. U razvoju projekta „Jobscape“ glavnu ulogu imam u razvijanju mobilne aplikacije i dizajniranju korisničkog sučelja.

Što je to Jobscape?

Jobscape je mobilna aplikacija namijenjena za primanje i pružanje freelance usluga te jednostavan razgovor između poslodavca i radnika. Unutar aplikacije korisnici mogu objavljivati objave koje su podijeljene u „Zaposli se“ i „Zapošljavamo“. Objave se također mogu filtrirati po kategorijama i potkategorijama. Korisnici jednim klikom na gumb mogu kontaktirati druge korisnike i time brzo započeti komunikaciju. Svaki korisnik je provjeren pomoću e-adrese te je odgovoran za svoj profil. Na profilu korisnik može postaviti svoje korisničko ime, puno ime, opis svojeg profila te profilnu sliku. Za vrijeme razvoja aplikacije, cilj je bio pomoći zajednici programera i freelancera, stoga je aplikacija otvorenog izvora (eng. Open Source).

Planovi za daljnji razvoj

Naravno, aplikacija nije u svom konačnom stanju te će biti ažurirana. Među prioritetima je razvoj web i desktop aplikacije, čime bi se korisnicima koji imaju pristup računalu olakšalo korištenje aplikacije. Cilj je i par manjih nadogradnji na trenutnu aplikaciju, npr. ocjenjivanje profila kojim korisnici dobivaju povratnu informaciju o njihovim uslugama te bi tako drugi korisnici mogli vidjeti koliko dobre usluge pruža korisnik. Osim toga, izbor teme – kako bi korisnici mogli prilagoditi izgled aplikacije prema njihovom izboru te pružiti podršku za engleski i druge jezike. Prilikom razvoja backenda odlučeno je kako bi trebalo olakšati hosting servera te omogućiti laku komunikaciju među serverima.

Korisnička dokumentacija

Upute za korištenje aplikacije

Kreiranje računa / registracija

Prvim klikom na aplikaciju, može se vidjeti stranica za prijavu u aplikaciju (*Slika 1*). Ova funkcija omogućena je korisnicima koji već imaju račun u Jobscape aplikaciji. U slučaju da korisnik nema račun, može ga kreirati pritiskom na gumb „Registracija“ (*Slika 2*).

The screenshot shows a light gray rectangular form. At the top left is a blue rounded rectangle containing a user icon and the placeholder text "Korisničko ime". Below it is another blue rounded rectangle containing three asterisks and the placeholder text "Lozinka". At the bottom are two buttons: a dark gray button on the left labeled "Registracija" and a blue button on the right labeled "Prijava".

Slika 1 – stranica za prijavu

Pritiskom na gumb „Registracija“ otvara se nova stranica (*Slika 3*) na kojoj korisnik mora ispuniti određene informacije da bi izradio korisnički račun. Prilikom registracije, korisnik prvo mora navesti korisničko ime koje želi koristiti. Korisničko ime može sadržavati isključivo mala slova a-z, velika slova A-Z, brojeve i znak „_“, odnosno donju crtu. Zatim korisnik mora upisati e-adresu. Važno je da korisnik unese valjanu e-adresu, formata user@domain.com, da bi mogao potvrditi registraciju svojeg računa pošto aplikacija pošalje poruku za verifikaciju računa. Korisnik mora proći proces verifikacije računa samo jednom, isključivo prilikom izrade računa. Treći korak pri izradi računa jest unos punog imena i prezimena. Prilikom popunjavanja navedenog polja, korisnik ne smije koristiti posebne znakove (\$, @, %, &, brojevi...). Treba naglasiti da ovo polje omogućuje lakšu identifikaciju među korisnicima tijekom traženja poslova i obraćanja poslodavcima.

The screenshot shows a light gray rectangular form with five input fields. From top to bottom: 1. A blue rounded rectangle with a user icon and placeholder "Korisničko ime" followed by an info icon. 2. A blue rounded rectangle with an envelope icon and placeholder "E-mail" followed by an info icon. 3. A blue rounded rectangle with a person icon and placeholder "Puno ime" followed by an info icon. 4. A blue rounded rectangle with three asterisks and placeholder "Lozinka" followed by an info icon. 5. A large blue rounded rectangle with a smiley face icon and placeholder "Opis". At the bottom are two buttons: a dark gray button on the left labeled "Prijava" and a blue button on the right labeled "Registracija".

Slika 3 – stranica za registraciju

Nakon toga korisnik mora odabrati svoju lozinku računa. Prilikom biranja lozinke, korisnik mora paziti da lozinka sadrži najmanje 8 znakova, među kojima su barem jedno veliko i jedno malo slovo, barem jedan broj i barem jedan poseban znak (\$, @, %, #, !, &...). Posljednje polje koje korisnik mora ispuniti jest opis profila. U tom polju korisnik može napisati bilo kakav tekst koji opisuje njegovo znanje, vještine i iskustva. Nakon što je korisnik gotov s unošenjem podataka, pritiskom na gumb „Registracija“ (*Slika 4*), server šalje mail korisniku čiji je sadržaj link kojim korisnik potvrđuje svoju registraciju. Pritiskom na link, korisnik je uspješno potvrdio svoju registraciju te tako završio izradu svojeg računa.



Slika 4 – gumb za registraciju

Početna stranica

Nakon prijave u aplikaciju, otvara se naslovna stranica (*Slika 5*), a nakon prijave na naslovnoj stranici, korisnik ulazi u aplikaciju. Na samom vrhu stranice nalazi se tražilica (*Slika 6*) pomoću koje korisnik pretražuje oglase po kategoriji, autoru ili imenu, odnosno ključnim riječima.



Slika 6 – tražilica

U nastavku se mogu primijetiti dva gumba, „Zaposli se“ i „Zapošljavamo“, kojima se mogu odabrati dvije opcije:

Zaposli se

Zapošljavamo

Slika 7 – „Zaposli se“ i „Zapošljavamo“

prikazivanje oglasa pojedinaca koji traže posao, odnosno nude freelance uslugu ili prikazivanje oglasa poslodavaca, odnosno onih koji traže freelance uslugu (*Slika 7*).

The main page of the Jobscape application. It features a search bar at the top with the placeholder "Pretraži poslove". Below the search bar are two tabs: "Zaposli se" (underlined) and "Zapošljavamo". Under "Zaposli se", there is a "Novo" button. On the right side, there is a "Sve" button with a refresh icon. A job listing for "Python developer" is displayed, showing the category "Python". The listing text reads: "Zovem se Karlo i bavim se programiranjem u programskom jeziku Python. Vrlo sam iskusan, sa svojih 5 godina rada u ovom jeziku stekao sam znanje i vještine u izradi umjetnih inteligencija korištenjem Tensorflow i OpenCV alata. Također sam iskusan u izradi server-side programa. Navedena cijena je po 1 satu rada na projektu. Za dodatne informacije molim da se javite u privatne poruke." Below the listing, it says "Price: 100 kn" and "test12345". There is a "Kontaktiraj" button. Another listing below it says: "Dobar dan, ja sam Toni i nudim instrukcije iz matematike, informatike i fizike" with a "Poduke" button. At the bottom, there are navigation icons for "Naslovница", "+", a list icon, and a user profile icon.

Slika 5 – naslovna stranica

Potom se mogu primijetiti dva gumba. Klikom na njih otvara se izbornik s opcijama filtriranja (*Slika 8*).

Ispod njih se vide kartice s oglasima (*Slika 9*). Svaki oglas sadrži naslov, kategoriju i opis posla te autora oglasa. Također, svaki oglas ima gumb „Kontaktiraj“. Pritiskom na taj gumb aplikacija automatski kreira razgovor s autorom oglasa i tako šalje automatiziranu poruku.

Na dnu naslovne stranice nalazi se navigacijska traka (*Slika 10*) koja sadrži četiri različita gumba. Svaki od njih vodi na novu stranicu, osim prvog koji služi za vraćanje na naslovnu stranicu.



Slika 8 – gumbi za filtriranje



Slika 9 – kartica sa oglasom



Slika 10 – navigacijska traka

Pretraga objava

Jobscape svojim korisnicima omogućuje pretraživanje tražilicom pomoću koje korisnik pretražuje oglase po kategoriji, autoru ili imenu, odnosno ključnim riječima. Za pretraživanje oglasa po kategoriji (*Slika 11*), korisnik u tražilicu unosi znak „#“ nakon čega slijedi ime kategorije. Ako želi pretraživati objave po autoru (*Slika 11*), korisnik unosi znak „@“ nakon čega slijedi ime autora čije oglase korisnik pretražuje. Ako korisnik želi pretražiti oglase po ključnoj riječi tj. imenu oglasa (*Slika 12*), upisuje tekst i pritom ne mora unositi posebne znakove kao za prijašnje dvije opcije.

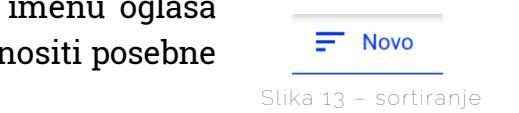
Osim što se korisnik može koristiti tražilicom, može se koristiti i opcijom sortiranja. Klikom na prvu opciju sortiranja (*Slika 13*) otvara se izbornik u kojem korisnik bira kako želi sortirati oglase (*Slika 14*).



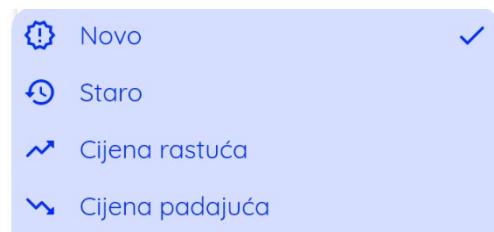
Slika 11 – pretraživanje autora i kategorija



Slika 12 – pretraživanje ključnim riječima



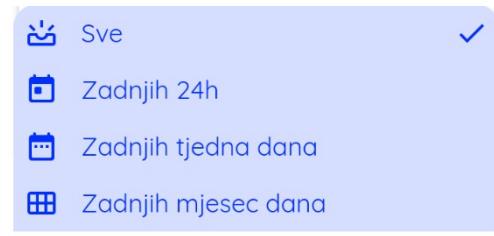
Slika 13 – sortiranje



Slika 14 – izbornik sortiranja

Klikom na drugu opciju filtriranja (*Slika 15*), otvara se izbornik u kojem korisnik bira kako želi filtrirati oglase u odnosu na vrijeme objavljivanja (*Slika 16*).

Slika 15 – filtriranje po kvaliteti



Slika 16 – izbornik

Kreiranje nove objave

Klikom na drugu ikonu „+“ u navigacijskoj traci, otvara se stranica za izrađivanje i objavu oglasa (*Slika 17*).

Pri kreiranju nove objave, korisnik je dužan navesti ime objave, kategoriju posla, kategoriju traži li uslugu ili nudi uslugu („Zaposli me“ ili „Zapošljavamo“), cijenu usluge te opis navedene usluge. Naslov usluge (*Slika 18*) ne smije biti duži od 128 znakova.

Naslov

0/128

Slika 18 – izrada naslova

Cijena ne smije biti duža od 9 znakova (*Slika 19*), a opis ne smije biti duži od 1024 znakova da bi sadržavao samo ključne informacije vezane za oglas (*Slika 20*).

Opis

0/1024

Slika 20 – polje za opis

Cijena

0/9

Slika 19 – polje za cijenu

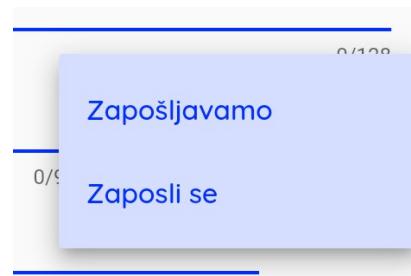
Slika 17 – stranica za izradu oglasa



Korisnik također određuje u koju kategoriju poslova pripada oglas koji izradi. Klikom na „Kategorije“ otvara se izbornik s kategorijama (*Slika 21*) te korisnik bira kategoriju koja najbolje odgovara njegovom oglasu. Osim toga, korisnik određuje zapošljava li nekoga ili traži posao. To čini odabirom jedne od opcija „Zaposli se“ ili „Zapošljavamo“ (*Slika 22*).



Slika 21 – izbornik kategorija



Slika 22 – izbornik vrsta usluga

Kada je korisnik zadovoljan sa svojim oglasom, klikom na gumb „Objavi“ potvrđuje objavu oglasa te je tada oglas dostupan svim korisnicima aplikacije.

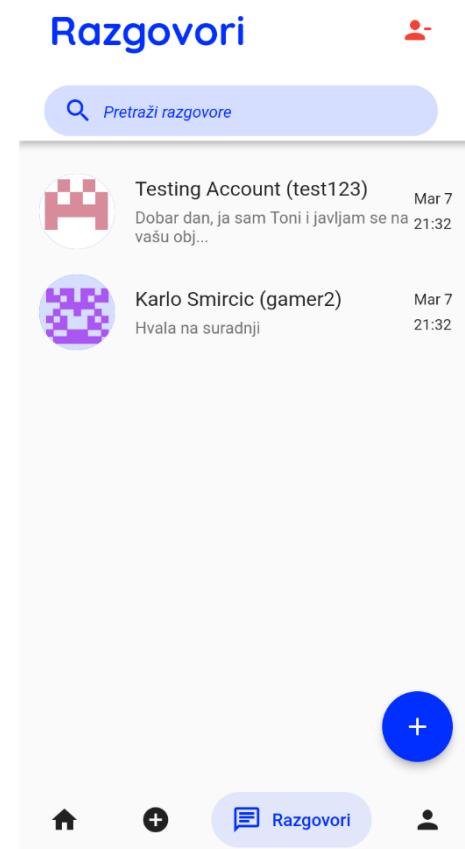
Razgovori / chat stranica

Pritiskom na gumb „Kontaktiraj“ na karticama, (na naslovnoj stranici) kreira se razgovor. Pristup takvim razgovorima omogućuje stranica s razgovorima (*Slika 23*).

Na samom vrhu stranice nalazi se tražilica pomoću koje korisnik može pretražiti određeni razgovor. Osim tražilice, može se uočiti i crvena ikona koja upućuje na stranicu s blokiranim korisnicima (*Slika 24*).



Slika 24 – stranica s razgovorima



Slika 23 – stranica s razgovorima

Na istoj stranici može ih se odblokirati jednostavnim klikom na ikonu pored korisnika (*Slika 25*). Na stranici s razgovorima mogu se vidjeti svi započeti razgovori, a klikom na određenog korisnika može se otvoriti stranicu koja nam omogućuje slanje i primanje novih poruka (*Slika 26*).



Slika 26 – blokirani korisnici

← Blokirani korisnici

	Testing Account username: test123	
	Karlo Smircic username: gamer2	

Slika 25 – blokirani korisnici

U slučaju da korisnik želi izbrisati neki od otvorenih razgovora, to može učiniti jednostavnim kliznim pokretom prsta po ekranu – s desna prema lijevo – pri čemu će korisnik biti pitan želi li završiti ili otkazati brisanje razgovora.

Na dnu ekrana nalazi se gumb za dodavanje razgovora. Klikom na taj gumb korisnik je pitan da upiše ime korisnika s kojim želi započeti razговор. Tada se pojavljuje gumb za nastavak s kojim korisnik potvrđuje kreiranje razgovora (*Slika 27*).

← Novi razgovor

Korisničko ime	0/128

Slika 27 –kreiranje razgovora

Profil

Klikom na ikonu identiteta (*Slika 28*), na navigacijskoj traci, otvara se stranica korisničkog profila. Na njoj korisnik može upravljati svojim objavama, uređivati osobne podatke, pristupiti postavkama aplikacije te odjaviti se iz aplikacije.



Slika 28 – ikona identiteta

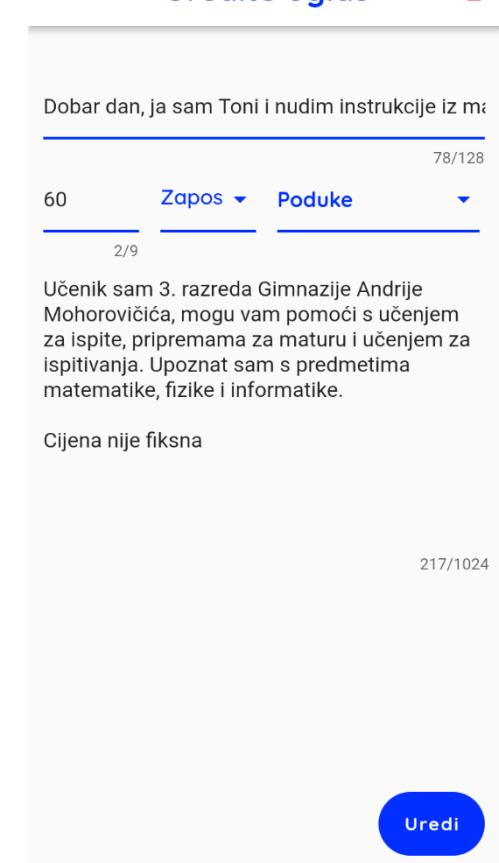
Na samom vrhu stranice vidimo dvije ikone. Jedna označava postavke aplikacije, a druga odjavu iz aplikacije. Klikom na ikonu postavki otvara se stranica s postavkama koje se mogu mijenjati.

Potom se može uočiti profilna kartica (*Slika 29*). Na prvi pogled korisnik može vidjeti samo profilnu sliku, korisničko ime te svoje puno ime. Klikom na ikonu pri dnu kartice, korisnik može proširiti profilnu karticu i time si omogućiti pregled svih korisničkih informacija računa. U slučaju da korisnik nije zadovoljan svojim profilom, može promijeniti podatke klikom na ikonu pri vrhu kartice koja otvara stranicu za mijenjanje informacija o profilu. U nastavku, korisnik ima uvid u sve svoje objave te može promijeniti njihov sadržaj.



Slika 29 – kartica profila

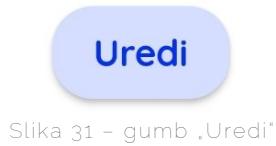
Ako se korisniku ne sviđa sadržaj ranije objavljen oglas, sadržaj navedenog oglasa može se promijeniti klikom na gumb „Uredi“, čime se otvara stranica za mijenjanje sadržaja oglasa (*Slika 30*). Osim što korisnik može mijenjati sadržaj oglasa, na navedenoj stranici može izbrisati oglas klikom na crvenu ikonu pri vrhu stranice. Time oglas prestaje biti vidljiv drugim korisnicima. Ta je funkcija nepovratna.



Slika 30 – stranica za uređivanje oglasa

Uređivanje profila / pregledavanje vlastitih objava

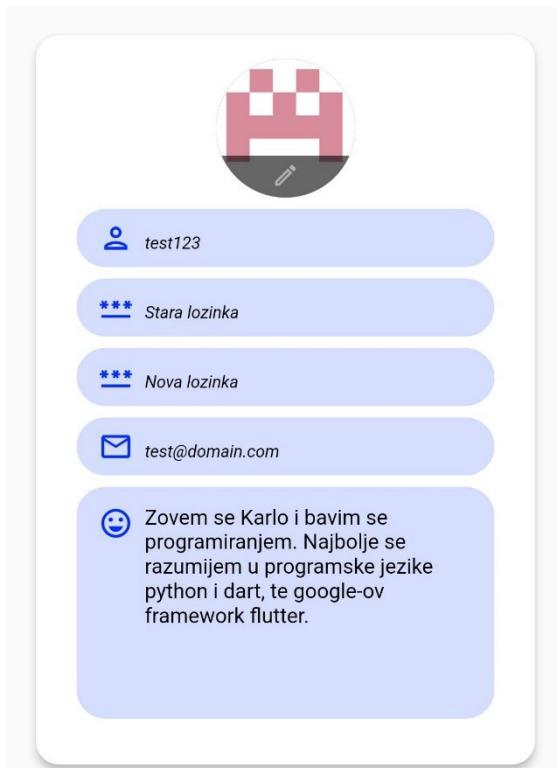
Ako se korisniku ne sviđa sadržaj ranije objavljenog oglasa, sadržaj navedenog oglasa može promijeniti klikom na gumb „Uredi“ (*Slika 31*), čime se otvara stranica za mijenjanje sadržaja oglasa.



Slika 31 – gumb „Uredi“

Osim što korisnik može mijenjati sadržaj oglasa, na navedenoj stranici može izbrisati oglas klikom na crvenu ikonu pri vrhu stranice. Time oglas prestaje biti vidljiv drugim korisnicima. Ta je funkcija nepovratna.

U slučaju da korisnik želi promijeniti podatke o svojem profilu, to može učiniti klikom na ikonu za uređivanje na kartici profila (*Slika 32*). Na taj način se otvara stranica za uređivanje informacija o profilu. Na toj stranici korisnik ima mogućnost promijeniti profilnu sliku, korisničko ime, e-adresu, lozinku te opis profila. Korisnik svoje izmjene može spremiti klikom na gumb „Spremi“ (*Slika 33*). Važno je napomenuti kako korisnik ne mora promijeniti sve informacije o profilu. Moguće je popuniti samo određena polja, pri čemu će se isključivo te informacije biti promijenjene. Također je važno napomenuti da je korisnik, pri mijenjaju lozinke, dužan unesti i trenutnu lozinku prije nego je promijeni. Pomoću toga se potvrđuje da je korisnik istinski vlasnik profila.



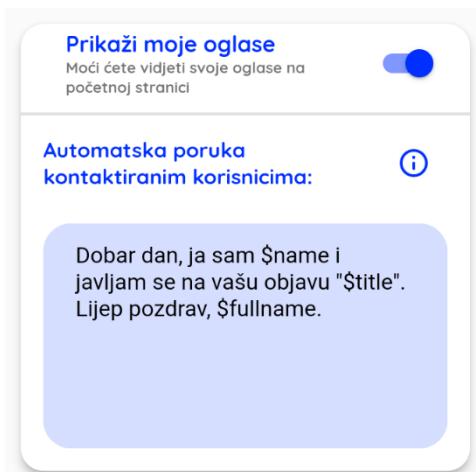
Slika 32 – stranica za uređivanje profila



Slika 33 – gumb za spremanje

Postavke

Na stranici s postavkama, korisnik može promijeniti dvije opcije (*Slika 34*). Prva opcija je prikaz svojih oglasa na naslovnoj stranici. Korisnik tu opciju može isključiti u slučaju da mu smeta vidljivost njegovih oglasa. Druga opcija je uređivanje automatske poruke. Automatska poruka je poruka koju aplikacija automatski pošalje autoru oglasa na čiji korisnik klikne „Kontaktiraj“. Ta je opcija napravljena kako bi se ubrzao i olakšao proces javljanja za korisnike Jobscape aplikacije. Korištenjem posebnih naredbi (\$name, \$title...), korisnik može pisati automatizirane poruke specifične za određeni oglas.



Slika 34 – postavke

Tehnička dokumentacija

Kako je aplikacija izrađena

Pokretanje

Aplikaciju je moguće pokrenuti preuzimanjem izvornog koda i upisivanjem naredbi flutter pub get i zatim flutter run za izgradnju i lansiranje aplikacije na uređaju spojenom preko ADB mosta.

Frontend

Frontend označava aplikaciju, web stranicu ili bilo koju drugu metodu za spajanje na Jobscape backend servere.

Aplikacija je napisana u [Dart](#) programskom jeziku, koristeći [Flutter](#) framework. Flutter je Googleov framework otvorenog izvora te omogućava izradu lijepih aplikacija koristeći Googleov Material dizajn te izvorno kompiliranih aplikacija. To omogućuje izradu jednog koda koji će imati odlične performanse na različitim uređajima. Prilikom izrade aplikacije korisničko sučelje napisano je u Flutteru, a povezano je s backendom pomoću Dart-a.

Aplikacija se na FastAPI server, koji koristimo za pohranu podataka, spaja preko modula http_requests.dart. Svaki endpoint ima funkciju sa zadanim endpointom i podatcima koje treba poslati backendu. Za funkcije chata i razgovora među korisnicima koristimo Openfire server i Flutter modul xmpp_stone koji koriste otvoreni standard XMPP za komunikaciju.

Za spremanje podataka u aplikaciji, koristi se [Hive](#). Hive je jednostavna, brza i sigurna baza podataka napisana u Dart-u. Radi po principu *ključ – vrijednost* te je jednostavna za korištenje. Dostupna je za mobilne i desktop uređaje te unutar aplikacije služi za spremanje tokena (koji se koristi za autentifikaciju nakon što je korisnik ulogiran). Također koristimo Hive za spremanje postavki, korisničkih podataka (korisničko ime, biografija, ...) te poslanih i primljenih poruka.

Dijagrami

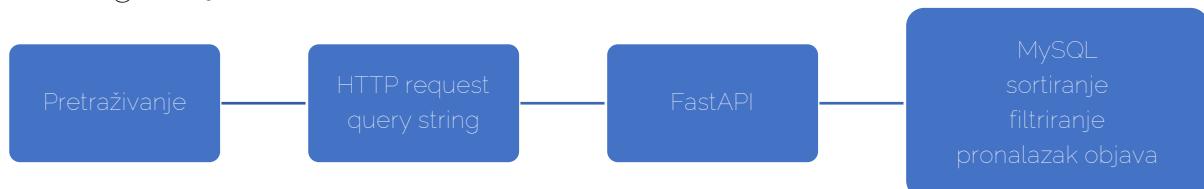
U nastavku se nalaze tokovi dijagrama sučelja aplikacije.

Naslovna stranica



U ovom dijagramu može se vidjeti kako naslovna stranica šalje HTTP request FastAPI-ju, koji će se s primljenim podatcima javiti MySQL bazi podataka kako bi zatražio objave. Nakon toga će backend serijalizirati podatke u JSON formatu koji će Flutter i Dart moći pročitati.

Pretraga objava



Objave se za pretraživanje učitavaju na sličan način kao na naslovnoj stranici, ali je u zahtjevu potrebno poslati query string koji backend može prevesti u ispravan SQL sintaksu.

Objava oglasa



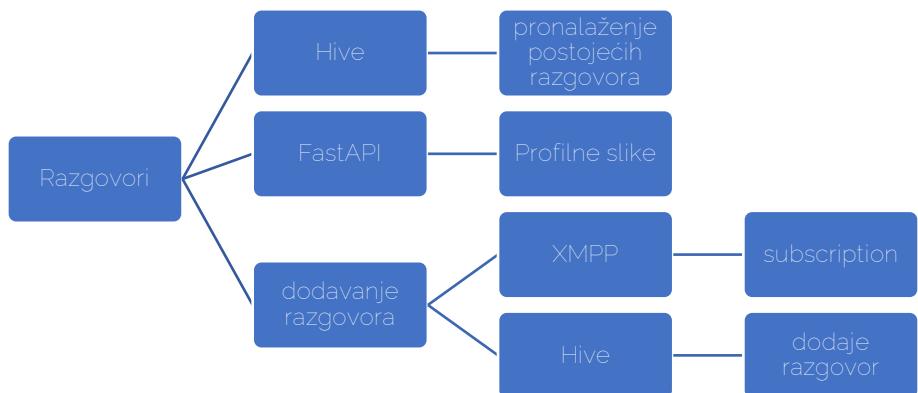
Objavljivanje oglasa također radi preko HTTP requesta u kojem se šalju sve informacije vezane uz oglas koje će backend spremiti u bazu podataka pomoću **INSERT** zahtjeva.

Uređivanje oglasa

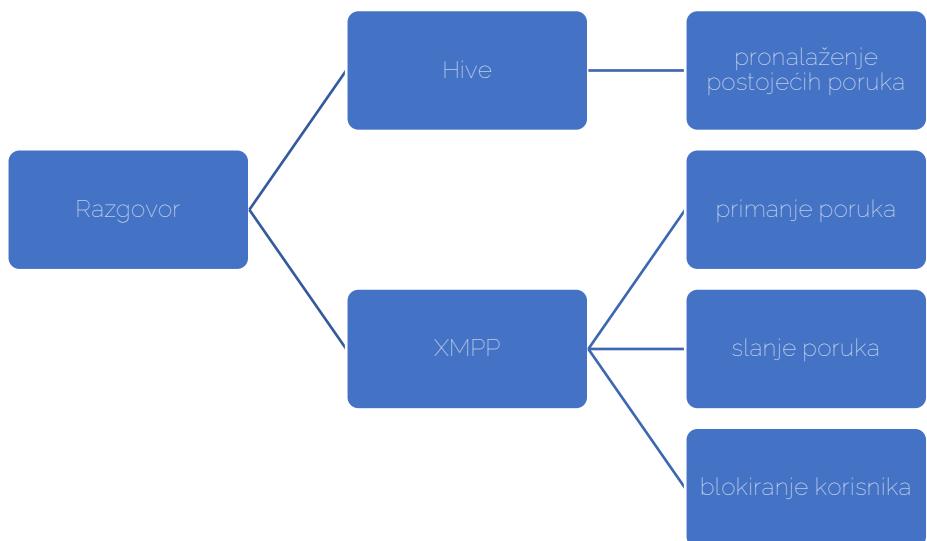


Oglasi se uređuju tako da se šalju samo oni podatci koji su promijenjeni kako bi se smanjilo opterećenje servera.

Stranica s razgovorima



Razgovori su preuzeti iz Hive baze podataka, jer Openfire sprema poruke samo u slučaju da korisnik nije u mogućnosti primiti poruke, tj. kada se nalazi izvan aplikacije. U nastavku se nalazi dijagram za stranicu koja se otvara kada korisnik klikne na neki aktivni razgovor. Prvo će se poslati request za učitavanje novih poruka koje će biti spremljene s ostalim starijim porukama u Hive bazi podataka.

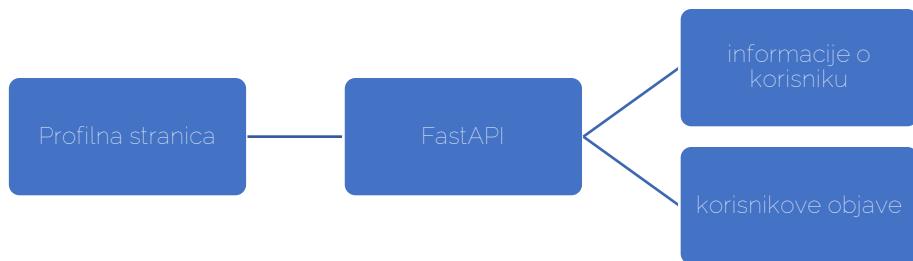


Stranica s blokiranim korisnicima



Na ovoj se stranici nalazi popis svih blokiranih korisnika, koji se preuzimaju XMPP upitom na Openfire serveru.

Profilna stranica



Na profilnoj stranici aplikacija šalje dva zahtjeva kako bi se dobili podatci o korisniku i kako bi se dobjale korisnikove objave.

Stranica za uređivanje profila



Na ovoj je stranici moguće promijeniti podatke o vlastitom profilu, preko zahtjeva koji se šalje FastAPI serveru za promjenu podataka. Ako neko od polja nije promijenjeno, ne šalje se u zahtjevu.

Postavke



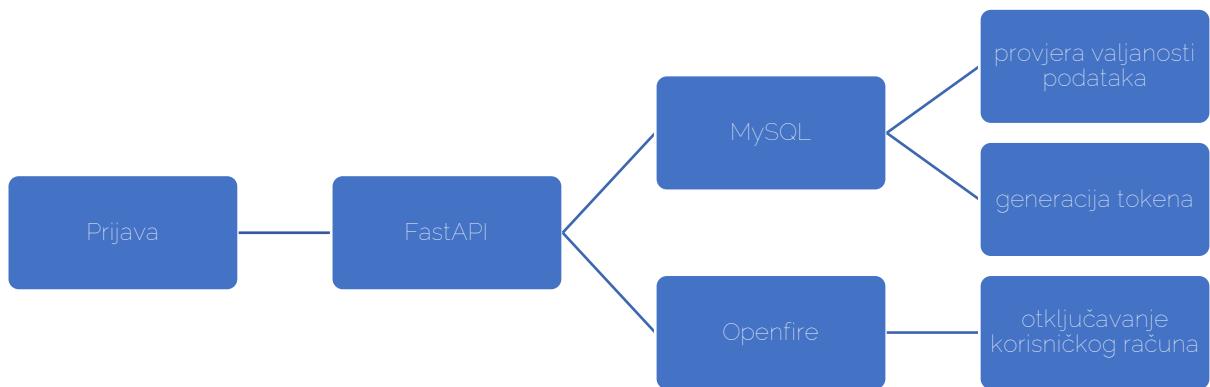
Postavke imaju lokalni utjecaj i nisu sinkronizirane na serveru.

Odjava



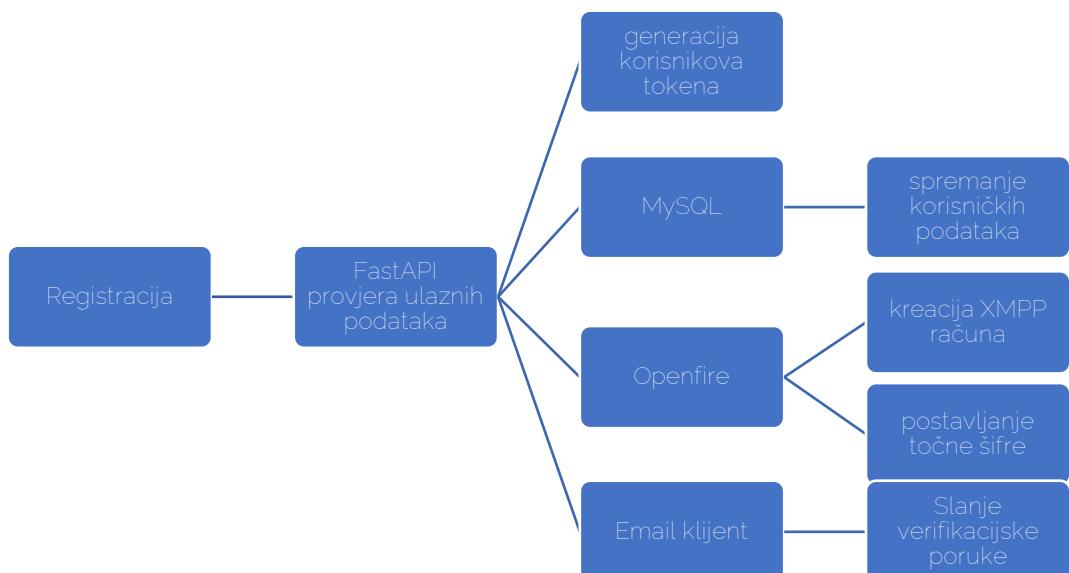
Odjava će promijeniti MySQL token u ništičnu vrijednost kako bi se onemogućila prijava pomoću tokena. FastAPI backend također će poslati zahtjev Openfire serveru za zaključavanje korisnikovog računa za razgovor koji će se otključati pri ponovnoj prijavi.

Prijava



Prijava zahtijeva slanje lozinke i korisničkog imena. Backend će pomoći imena potražiti hash lozinke u bazi podataka te će usporediti hash i lozinku da provjeri ispravnost lozinke. U slučaju da je lozinka ispravna generirat će se novi token, otključati račun za razgovor i odgovoriti aplikaciji s novim tokenom.

Registracija



Pri registraciji korisnik će unijeti svoje podatke. Lozinka će se koristiti za generiranje kriptografski sigurnog hasha koji će uz ostale korisničke podatke biti spremljen u bazi podataka, u tablici Users. Također će se stvoriti novi Openfire račun za razgovore pomoću rest API-ja te će se koristiti integracija sa serverom za e-adrese za slanje verifikacijske poruke korisniku. Odgovor servera sadržavat će token koji se nakon toga koristi za verifikaciju.

Backend

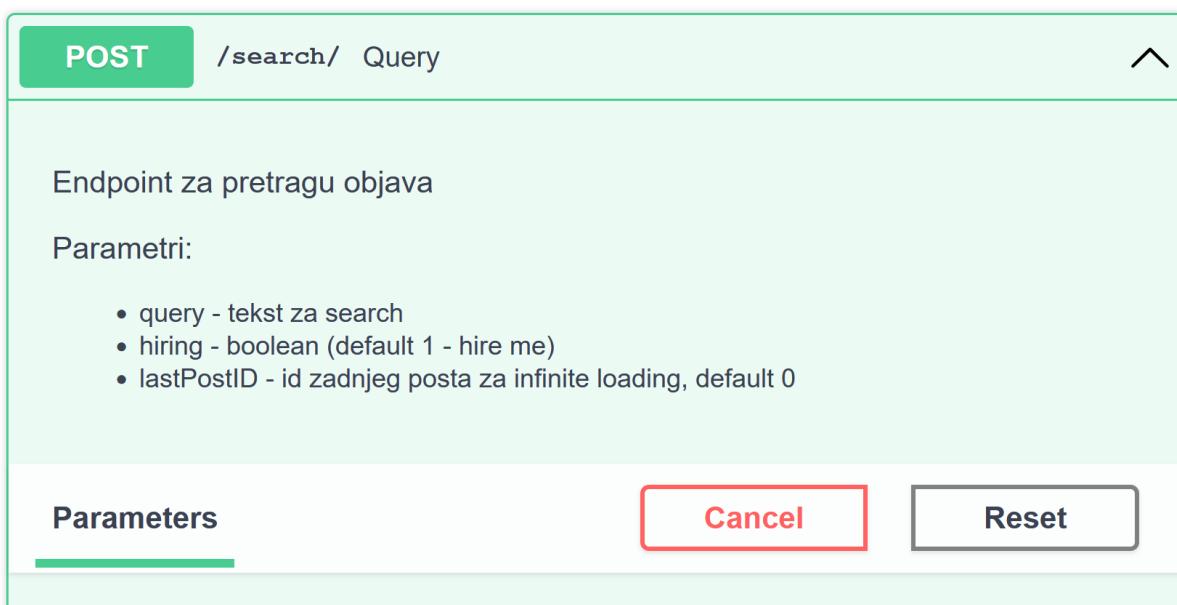
Docker containeri

Backend aplikacije je građen u razdvojenim cjelinama koje smo podijelili u Docker containere da bi se olakšalo održavanje. BackendRunner container je glavni container koji poslužuje backend aplikaciji te će komunicirati s MySQL Docker containerom i Openfire serverom za razgovore da bi omogućio jednostavno i centralizirano korisničko iskustvo.

FastAPI server

Glavni server na koji se aplikacija spaja izgrađen je pomoću programskog jezika Python i FastAPI frameworka. Taj framework odabran je jer je, kao što njegovo ime kaže, *brz*, i iako nije brži od Starlette frameworka na kojem je građen, ima puno više mogućnosti poput automatske validacije i serijalizacije podataka.

Omogućava testiranje pomoću pytest-a i automatsku dokumentaciju iz koje je moguće slati i simulirati requestove.



Slika 35 – automatska dokumentacija
API endpointova

```
$ pytest
=====
platform linux -- Python 3.8.10, pytest-7.0.1, pluggy-1.0.0
rootdir: /home/tony/Projekti/app/backend
plugins: anyio-3.5.0, Faker-8.1.4
collected 5 items
=====

test_backend.py ..... [100%]

===== 5 passed in 3.24s =====
```

FastAPI je građen na ASGI frameworku Starlette koji je optimiziran za async aplikacije i paralelizam, što je savršeno za naš API koji bi trebao moći podnijeti puno veza odjednom.

ASGI označava Asynchronous Server Gateway Interface i nastao je kao zamjena za stariji Python WSGI, specifikaciju za komunikaciju između web servera i web aplikacija/frameworkova. To znači da FastAPI podržava razne ASGI web servere, poput Uvicorna, Daphne ili Hypercorna.

FastAPI i većina drugih API modela funkcioniра na principu endpointova. Endpoint je http link koji prima post request s ulaznim podatcima koje potvrđuje i obrađuje da bi dobio rezultat koji formatira, najčešće u JSON formatu, i vraća korisniku. Često se tu nalazi i dodatan korak preuzimanja podataka iz baze poput MySQL-a. U našoj su aplikaciji endpointovi povezani u routere, od kojih svaki ima odvojenu ulogu, poput upravljanja korisničkim profilima. Routeri su spremljeni u folderu `modules`.

Dodavanje novog routera je jednostavno koristeći FastAPI klasu `APIRouter`.

```
# /modules/posts.py
from fastapi import APIRouter
from modules.shared import limiter

postRouter = APIRouter(prefix="/posts")

# zbog APIRouter prefixa je pravi endpoint example.com/posts/create
@postRouter.post("/create")
@limiter.limit("5/hour")
async def createPost(...):
```

Potrebno je router uvesti u glavni kod za upravljanje cjelokupnom FastAPI aplikacijom (u sljedećem primjeru nazvanom app).

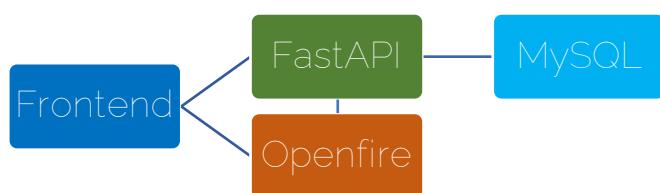
```
# main.py
from modules.posts import postRouter
app.include_router(postRouter)
```

Dodavanje novog endpointa je jednostavno. Koristimo pydantic BaseModel za oblikovanje tijela zahtjeva koji frontend šalje, funkciju checkToken za provjeru valjanosti tokena koji korisnik koristi za jedinstvenu identifikaciju i sigurnu komunikaciju. Osim toga, potrebno je uvesti limiter klasu koja koristi slowapi za upravljanje rate-limiterom, odnosno za vremensko ograničavanje broja zahtjeva da bi se spriječila zlouporaba endpointova i spriječili automatizirani napadi. SlowAPI koristi Redis, brzu bazu podataka koja funkcionira u radnoj memoriji računala da bi se pratio broj zahtjeva u određenom vremenskom roku i zabranio svaki zahtjev izvan dopuštenog broja. Response code s kojim backend u tom slučaju odgovara jest 429: Too Many Requests.

```
# Oblik podataka u zahtjevu
class postData(BaseModel):
    id: int
    ...

@postRouter.post("/getPost/")
@limiter.limit("20/minute")
async def getPost(post: postData, # podatci iz zahtjeva
                  request: Request,
                  x_token: str = Header(None, convert_underscores=True)):
    if not checkToken(x_token): # Provjera valjanosti tokena
        # ako je token neispravan, response code će biti 401
        raise InvalidTokenException

    postID = post.id
    ... # dohvatanje podataka iz zahtjeva
```



U ovom dijagramu može se uočiti kako su elementi naše aplikacije povezani. Bitno je vidjeti da MySQL nije dostupan javnosti, već korisnici dolaze do njegovih

podataka samo preko FastAPI backenda. Može se primijetiti i veza između FastAPI servera i Openfire servera za razgovore, preko kojih uspijevamo sinkronizirati korisničke podatke i ostale informacije.

MySQL

Za spremanje svih korisničkih podataka i objava koristimo MySQL server koji iz sigurnosnih razloga nije javno dostupan na internetu, već je dostupan samo backendu koji će prije slanja zahtjeva provjeriti sve podatke i potvrditi autentifikaciju korisnika. Upite u bazu podataka šalju se pomoću naredbe cursor.execute(), ali pritom se mora paziti da se ne koristi Pythonova ugrađena konkatenacija stringova ili da se ne koriste drugi načini za spajanje stringova poput f-stringova ili zbrajanja za dodavanje korisničkog inputa u MySQL upit, jer to dovodi do jednostavnog napada u obliku SQL injectiona. Zlonamjerni korisnik može unijeti posebno konstruiran tekst u polje za ime ili lozinku kako bi prevario bazu podataka u izvršavanje bilo kakvih naredbi. Zato je najbolje koristiti ugrađenu funkciju u MySQL connectoru za Python koja je nazvana „Prepared statements“ te će ta funkcija automatski sanitizirati svaki unos prije slanja bazi podataka.

Iako FastAPI ima automatsku provjeru tipa podataka, regexom se provjera svaki unos za validnost i koristi se prepared statemente kako bi bili sigurni da je baza podataka sigurna. Backend koji se često koristi treba biti brz, stoga je potrebno stvoriti poseban SQL upit da bi se većina računanja odvijala na strani MySQL servera koji je puno brži i optimiziraniji od Python koda. Na primjer za sortiranje stotinu tisuća objava Pythonu bi trebalo više sekundi, dok MySQL to može izvršiti u djeliću sekunde. Osim ubrzavanja upita, mora se pripaziti na smanjenje broja upita bazi podataka, jer uz više upita dolazi i do dodatnog vremenskog troška slanja upita bazi podataka i deserijalizacije odgovora u Python objekte.

Dobar primjer minimalizacije broja upita jest funkcija pretrage objava, koja gradi SQL upit korak po korak i koristi domišljato spajanje tablica kako bi broj potrebnih upita spustili s pet upita na samo dva. Nažalost, zbog rada MySQL-a nije moguće smanjiti broj upita na samo jedan, jer je cilj zadržati mogućnost upisa približnog korisničkog imena u polje za pretragu. Ako korisnik napiše ime poput „@test“ i ako taj korisnik ne postoji, backend bi trebao moći pretražiti slična imena i odabrati ono koje je najsličnije upitu. Za pronalazak sličnosti koristimo ugrađeni Python modul `difflib`, koji računa delta vrijednost između dviju riječi, i koristi delta vrijednost za pronalazak najbliže vrijednosti. Bitno je napomenuti da je `difflib` funkcija `get_close_matches()` case sensitive pa je potrebno zvati funkciju `lower()` kako bi se sve riječi prebacile u lowercase. Nakon pronalaska ispravnog imena koristimo ID tog korisnika za filtriranje svih objava. Nakon toga sortiraju se objave po broju ključnih riječi i odabiru se one koje imaju najveći broj podudaranja s ključnim riječima te koje su niže u tablici sortiranih objava od zadnje objave koju je korisnik učitao. Infinite scrolling je moguć na svakoj

stranici koja prikazuje objave, jer će tako korisnici vidjeti više objava pa će lakše pronaći freelance usluge.

MySQL baza podataka podijeljena je u tri tablice: Users, Posts i EmailConfirm. Tablica Users sadrži osnovne podatke o korisnicima, poput e-adrese, korisničkog imena, biografije i lozinke u hashed i salted obliku. Tablica Posts sadrži sve objave, a najbitnija su polja userID, koje sadrži ID korisnika koji je napravio objavu, polja title, content i price za informacije o objavi i polje hiring koje označava je li post u kategoriji „Zapošljavamo“ ili u kategoriji „Zaposli me“.

Komunikacija među korisnicima

Prilikom definiranja mogućnosti aplikacije, odlučeno je da je sustav za komuniciranje među korisnicima neophodan. Provedeno je istraživanje potencijalnih rješenja. Prilikom istraživanja pronađene su dvije interesantne opcije, Googleov Firebase i XMPP. Nakon rasprave, zaključilo se da je XMPP bolja opcija. To je odlučeno iz nekoliko razloga od kojih je najvažniji bio pokretanje s vlastitih servera te lakše dodavanje prilagođenih mogućnosti.

[XMPP](#) je „Extensible Messaging and Presence Protocol“, odnosno set otvorenih tehnologija namijenjenih brzom slanju poruka, prisutnosti (eng. „presence“), razgovora s više članova itd. Dakle, XMPP je kompletno besplatan i javnosti pristupačan protokol za sustave komunikacije. Poruke se šalju u XML formatu. Poruke su šifrirane, a tako i sigurne za prenošenje korisniku važnih podataka. XMPP pruža fleksibilan način rada i na taj način dopušta različite upotrebe. Definiran je listom [ekstenzija](#) (eng. „extension“) koje pružaju različite mogućnosti te s obzirom na to da je XMPP, kao što je već navedeno, otvorena tehnologija, uvijek je moguće predložiti nove ekstenzije. Naravno, većina potrebnih mogućnosti dodana je od početka razvoja samog XMPP protokola. Razvoj je počeo 1999., a kraj se još ne nazire. S obzirom na to da razvoj traje već preko 20 godina, ekstenzije su ažurirane mnogo puta te se stabilnost XMPP-a samo poboljšavala. Jedan od velikih prednosti ovog protokola jest decentraliziranost. Naime, protokol dopušta razgovor među više servera, a time i balansiranje opterećenja (eng. load balancing). Neke od najvećih platformi koje koriste XMPP za komunikaciju među korisnicima jesu: WhatsApp, Origin, Fortune, Zoom itd.

Prilikom postavljanja XMPP servera odlučeno je kako će se koristiti [Openfire](#). Openfire je poslužitelj za suradnju u stvarnom vremenu (eng. RTC). Openfire je vrlo jednostavan za postavljanje i administraciju te nudi odličnu sigurnost i performanse. Za još pojednostavljeni postavljanje koristi se [Docker](#) kontejner. Docker kontejner postavljen je s unaprijed izrađenom [slikom](#) (eng. Docker image) pronađenom na [DockerHub](#) serverima. Osim toga, Openfire nudi svoj sustav [pluginu](#). Jedan od važnih pluginu je „REST API“ koji pruža API za velik broj

mogućnosti unutar Openfirea. Osim navedenog plugin-a, za upravljanje porukama poslanih „offline“ korisnicima korišten je „Monitoring Service“ plugin.

Testiranje

FastAPI sadrži modul `TestClient`, pomoću kojeg je testiranje FastAPI backenda jednostavno. Testni primjeri su napisani u datoteci `test_backend.py` te je potpuno testiranje moguće sa samo dvije naredbe:

```
$ python3 -m pip install pytest faker
Successfully installed faker-13.3.1 pytest-7.0.1

$ pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.0.1, pluggy-1.0.0
rootdir: /home/tony/Projekti/app/backend
plugins: anyio-3.5.0, Faker-8.1.4
collected 5 items
=====

test_backend.py ..... [100%]

===== 5 passed in 3.24s =====
```

Ovo će testirati endpointove za kreiranje korisnika, prijavu, uređivanje podataka o korisniku, kreiranje objava i na kraju brisanja korisnika. Pomoću modula Faker kreiraju se lažni podaci radi testiranja te na kraju, zbog brisanja testnog korisnika, rezultati testiranja ne ostaju u bazi podataka.