Simple Contact Manager

Miha Prah, Anže Golob, Matija Krsnik

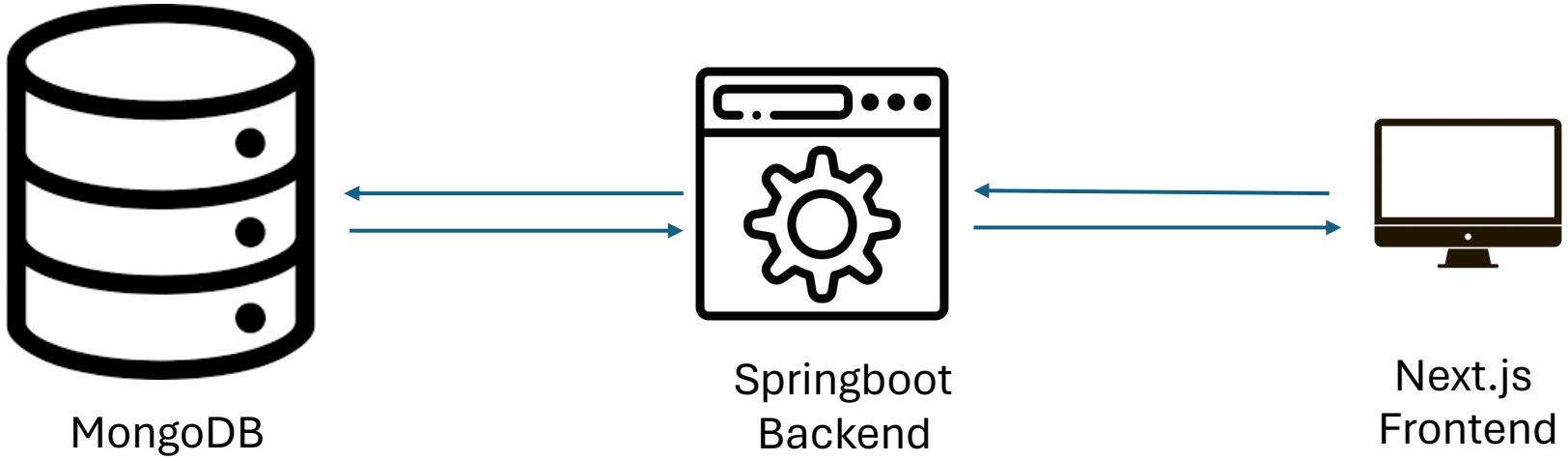
Opis

- Aplikacija za upravljanje s kontakti
- Omogoča filtriranje
- Ustvarjanje različnih skupin kontaktov
- Izvoz kontaktov
- Dodajanje poljubnih atributov in značk
- Uvoz kontaktov

Prednosti

- Preprostost uporabe pred ostalimi CRM sistemi
- Ostali CRM sistemi prekompleksni za uporabo za samo upravljanje s kontakti

Arhitektura rešitve



Nabor uporabljenih orodij

Backend

spring

Java

POSTMAN

mongoDB

Frontend

NEXT.JS

TS

tailwindcss

Ostalo

Vercel

Notion

GitHub

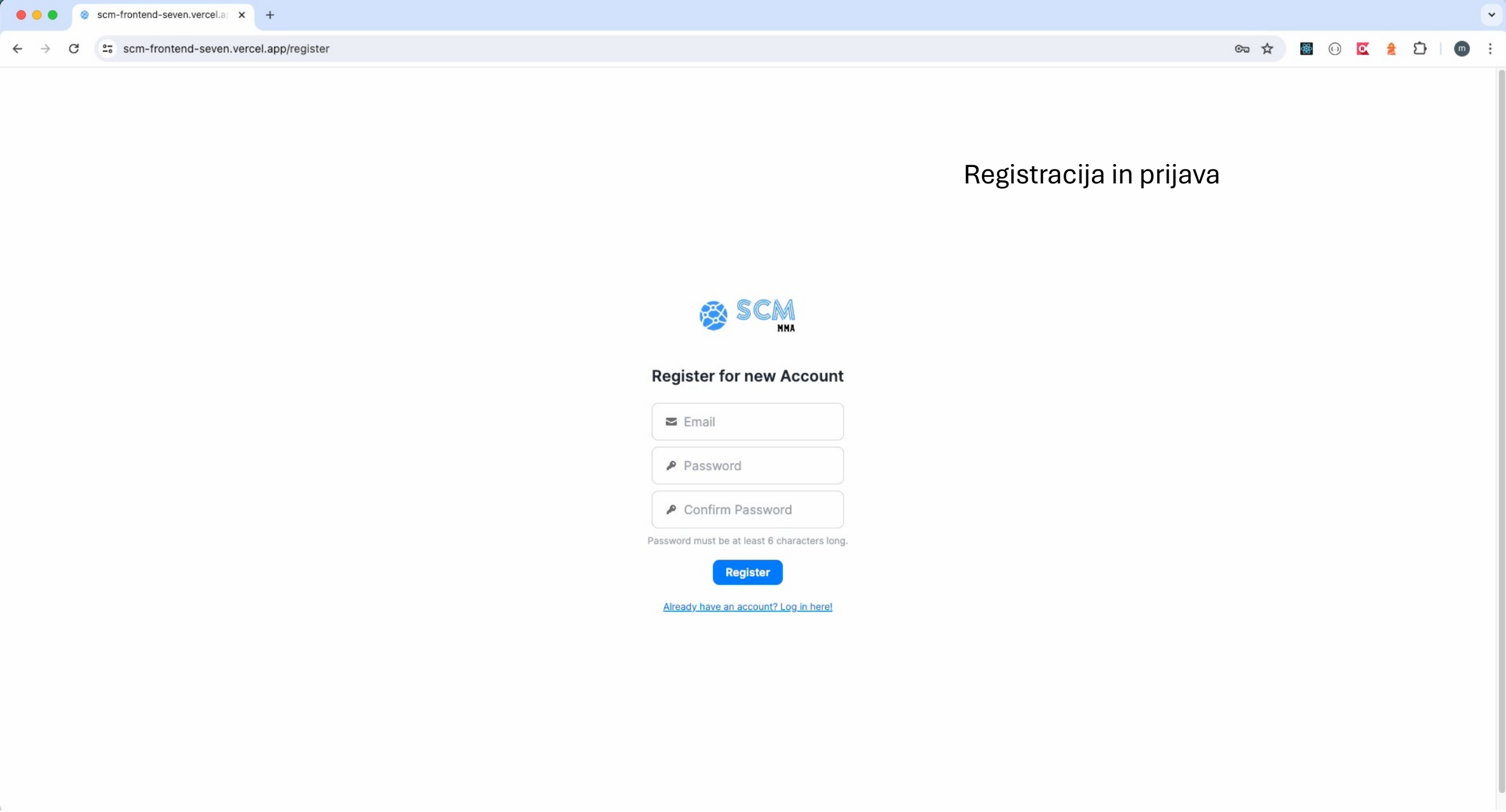
sonarcloud

IJ

WS

Firebase

Registracija in prijava



Sporočilo za potrditev novega računa

SCM <noreply@scm2024-31a54.firebaseapp.com>
to me

Thu, May 16, 9:34 AM

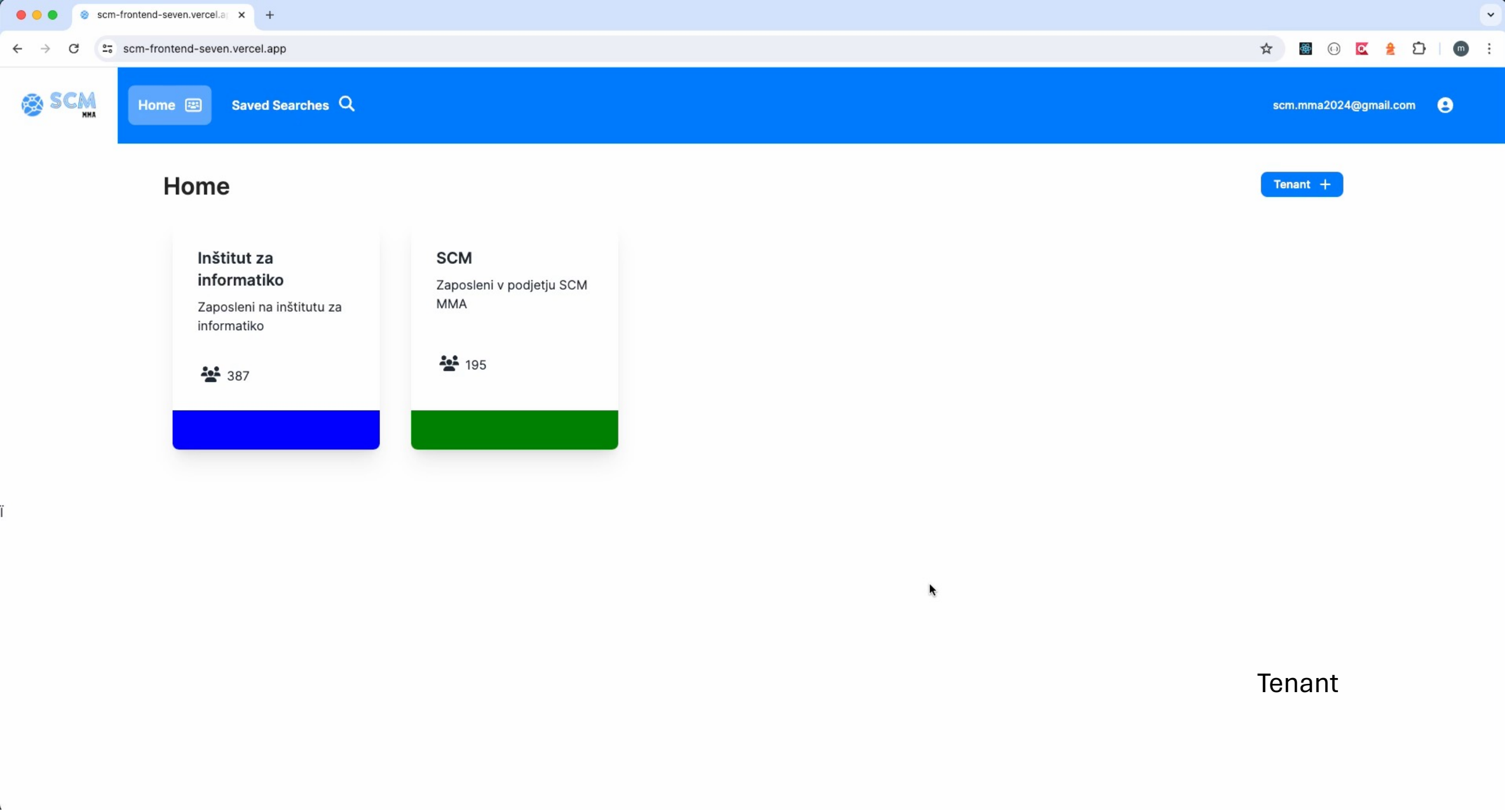
Hello,

Follow this link to verify your email address.

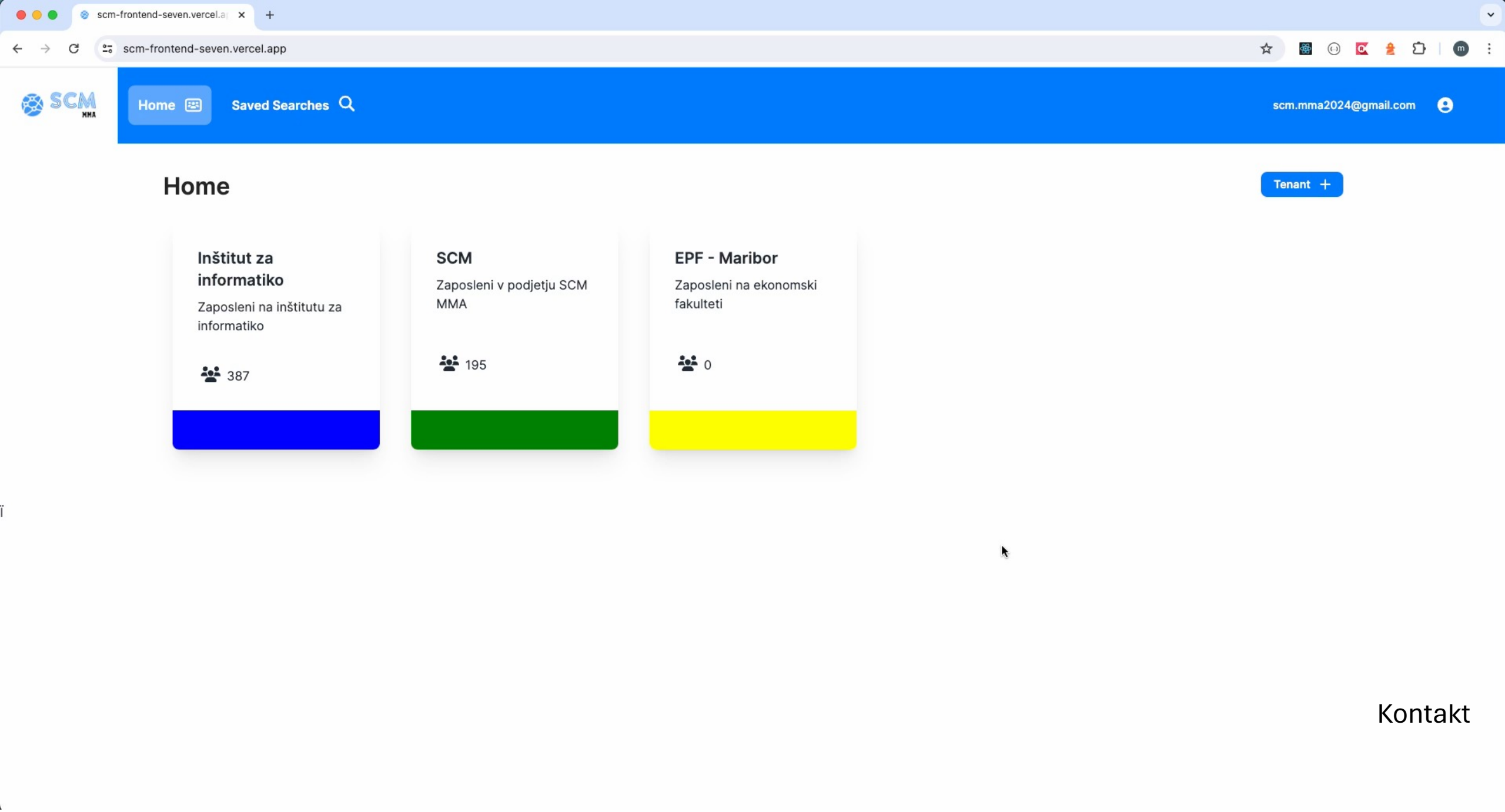
https://scm2024-31a54.firebaseapp.com/__/auth/action?mode=verifyEmail&oobCode=dO33OqcBzczWqJQOdilceQUfe7sORER9mSpb7I2O8sUAAAGPgFL7IA&apiKey=AIzaSyDS3qbHiEMU01pn4XdFn8G_JHWoZYVG8Ik&lang=en

•••

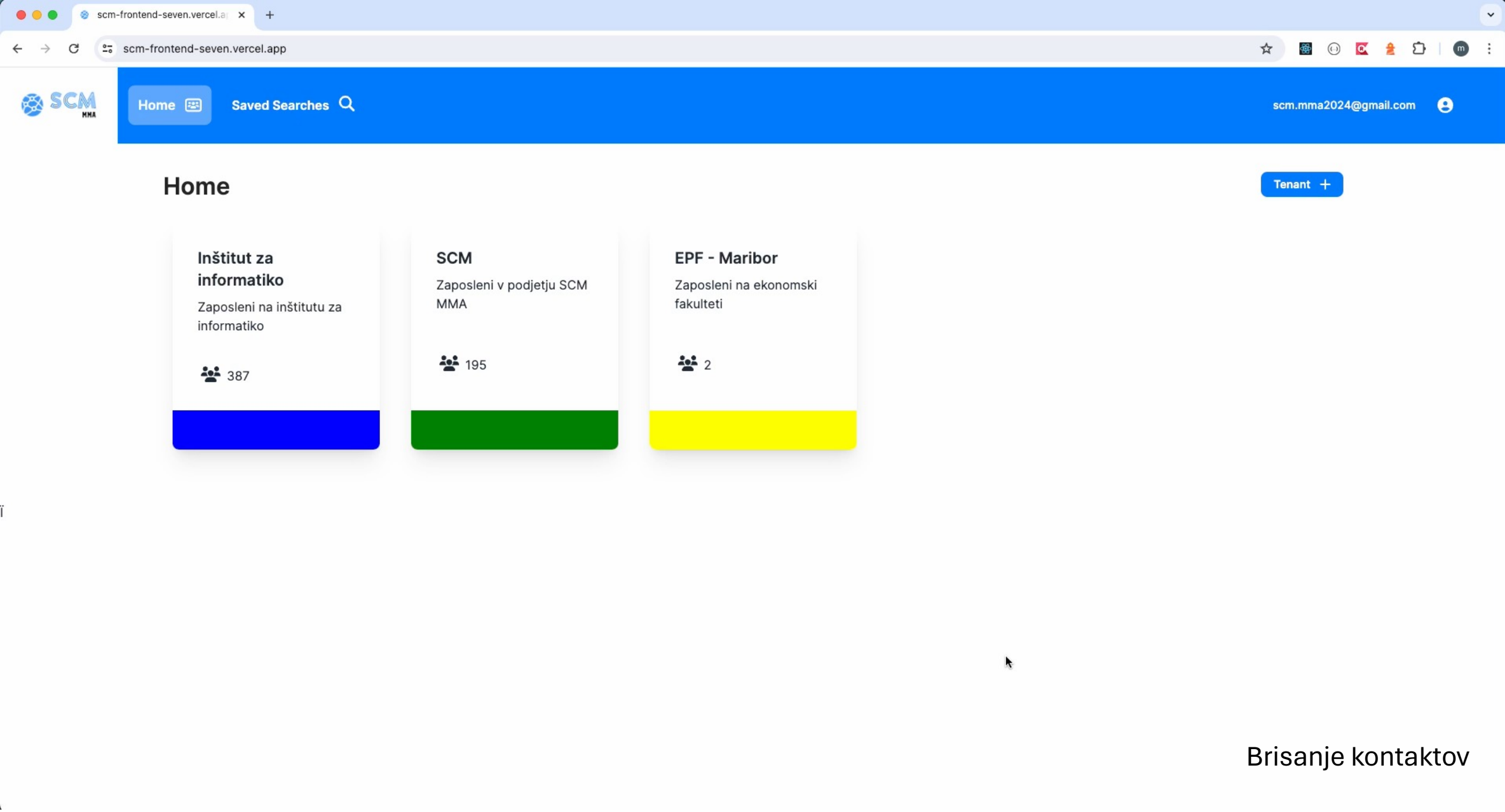
Reply Forward


scm-frontend-seven.vercel.app
SCM
MMA
Home
Saved Searches
scm.mma2024@gmail.com
Home
Tenant +
Inštitut za informatiko
Zaposleni na inštitutu za informatiko
387
SCM
Zaposleni v podjetju SCM MMA
195

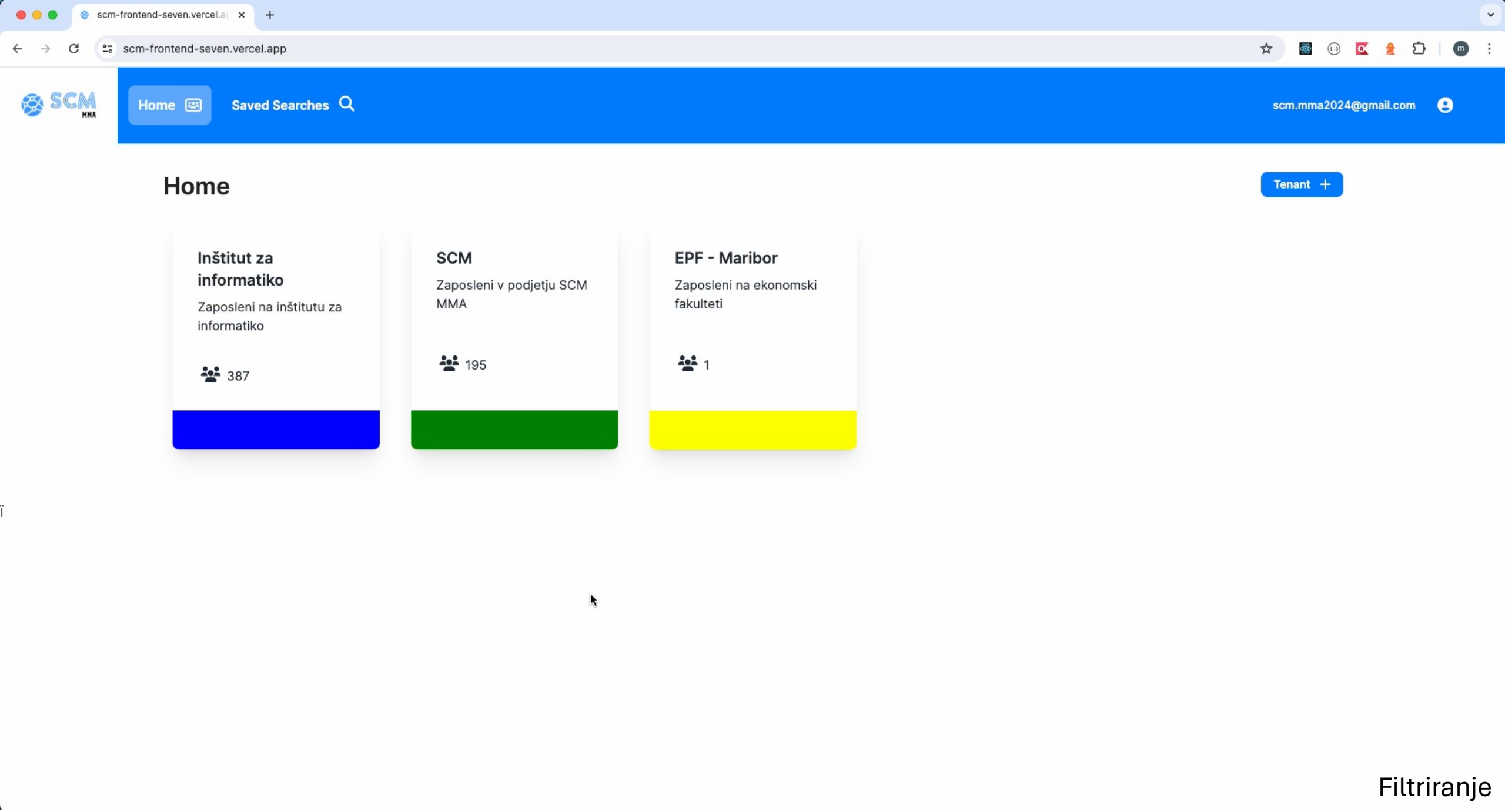
Tenant


scm-frontend-seven.vercel.app
SCM
MMA
Home
Saved Searches
scm.mma2024@gmail.com
Home
Tenant +
Inštitut za informatiko
Zaposleni na inštitutu za informatiko
387
SCM
Zaposleni v podjetju SCM MMA
195
EPF - Maribor
Zaposleni na ekonomski fakulteti
0

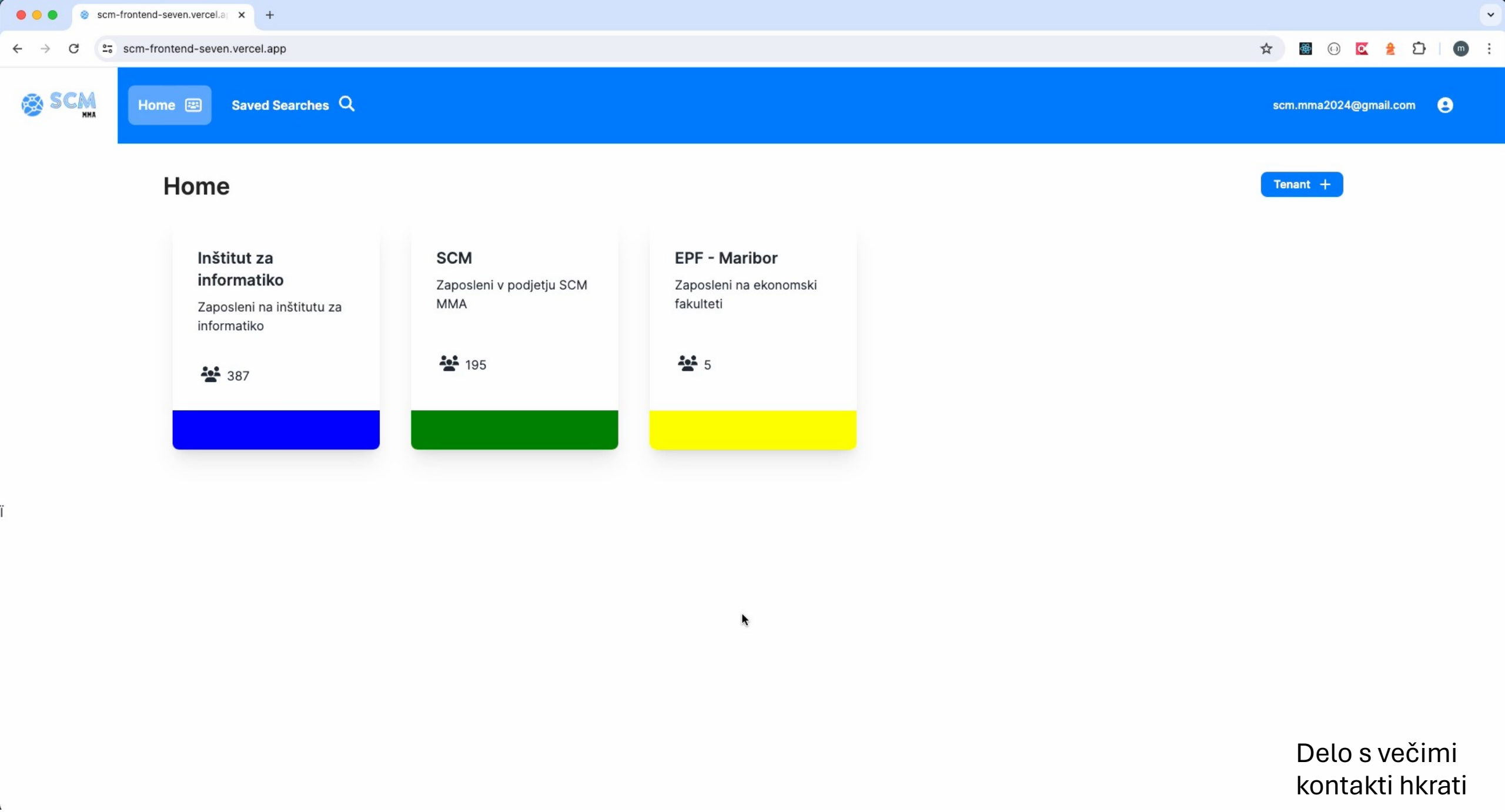
Kontakt



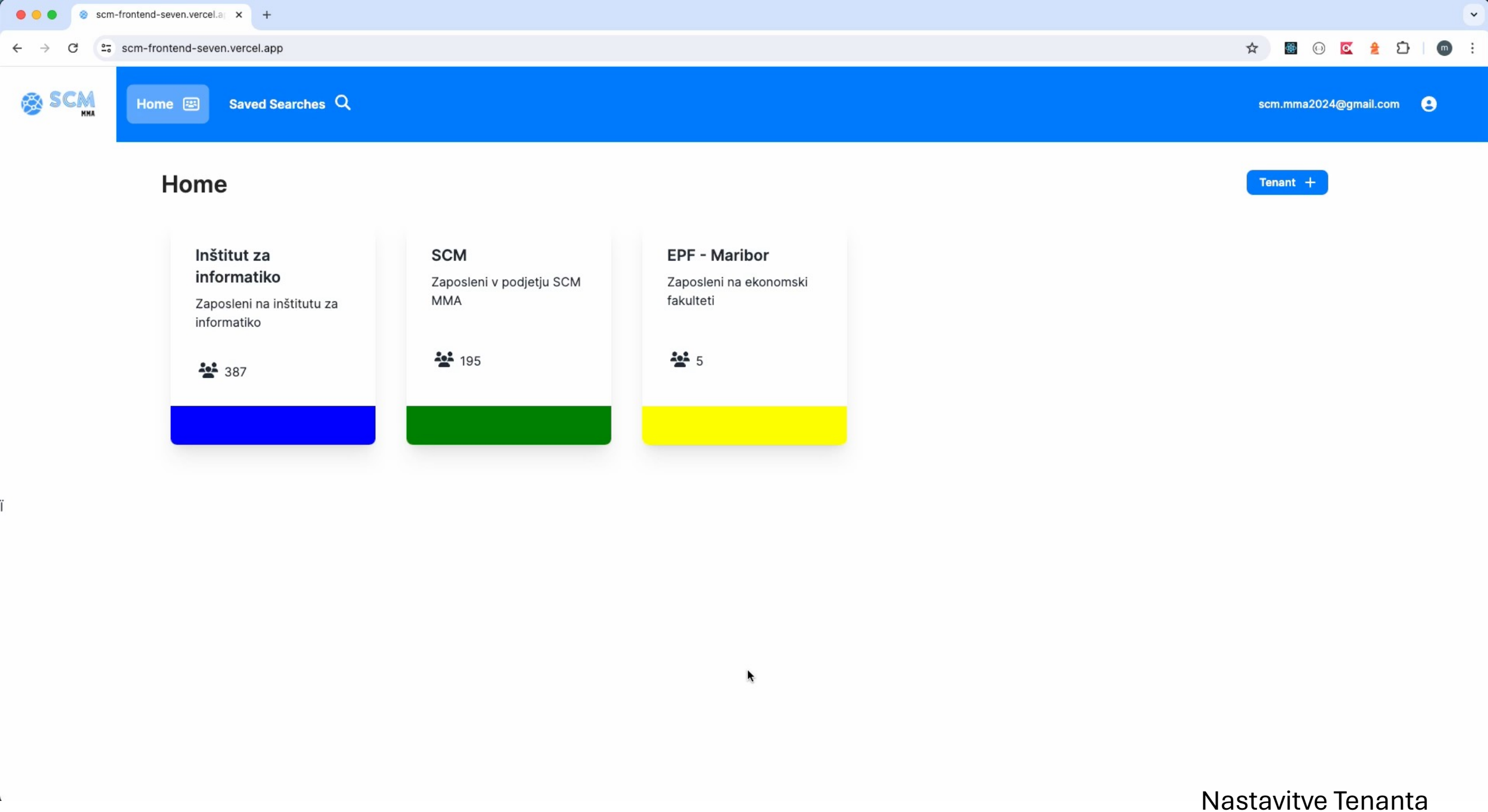
Brisanje kontaktov



Filtriranje



Delo s večimi kontakti hkrati



Nastavitve Tenanta

scm-frontend-seven.vercel.app

SCM MMA

Home | Saved Searches

scm.mma2024@gmail.com

Home

Tenant +

Inštitut za informatiko

Zaposleni na inštitutu za informatiko

387

SCM

Zaposleni v podjetju SCM MMA

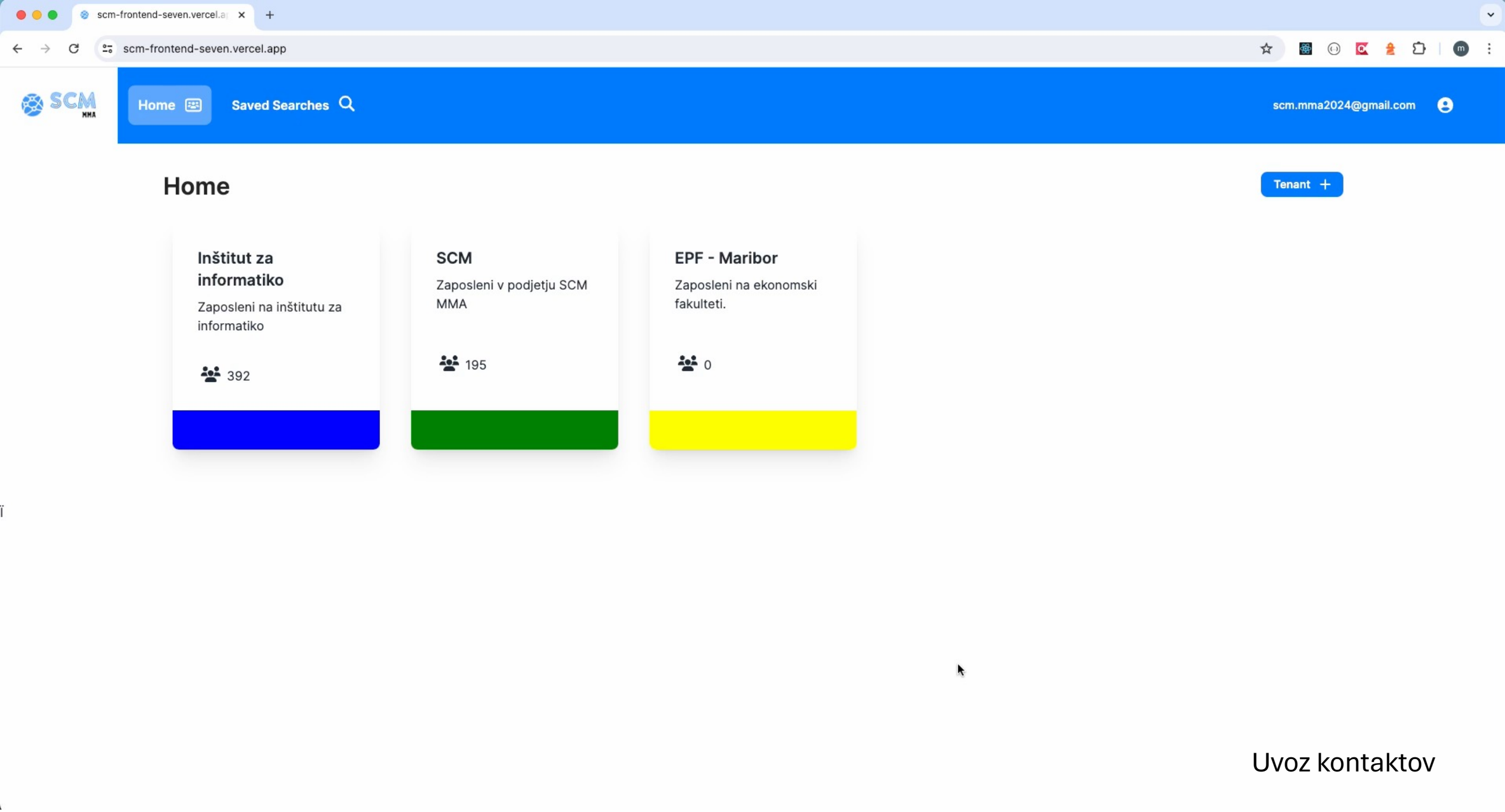
195

EPF - Maribor

Zaposleni na ekonomski fakulteti

5

Izvoz kontaktov



Uvoz kontaktov

Dokumentacija SCM

2. Dokumentacija

V dokumentaciji projekta je predstavljen celoten proces dela od začetka do konca. Prav tako so predstavljena vsa orodja in programska oprema, ki je bila tekom razvoja aplikacije uporabljena.

Funkcionalnosti

- Ustvarjanje skupin kontaktov
- Filtriranje in iskanje po kontaktih
- Shranjevanje filtrov
- Pregled izbrisanih kontaktov
- Dodajanje poljubnih značk kontaktom
- Ustvarjanje lastnih atributov za posamezen kontakt
- Revizijska sled kontaktov
- Nastavljanje poljubnih iemn atributov za skupine
- Izvoz kontaktov v format MS Excel (*.xsl/x*)
- Uzvoz kontaktov v formatu MS Excel (*.xsl/x*)

Tehnološki nabor

Backend (zaledje)

Zaledje aplikacije smo izdelali v programskem jeziku Java. Uporabili smo ogrodje SpringBoot, ki nam je pomagalo izgraditi REST vmesnik in komunicirati s podatkovno bazo. Za podatkovno bazo smo izbrali NoSQL dokumentno bazo MongoDB, ki nam je omogočila shranjevanje fleksibilnih, JSON podobnih dokumentov. Ta je bila potrebna, saj imajo lahko posamezni kontakti na sebi poljubno število različnih atributov in značk. Bazo smo poganli v dockerju in za prikaz uporabili Atlas Compass, ki nam je omogočal vizualizacijo podatkovne baze. V njem je bilo tudi možno kolekcije in podatke spreminjati in brisati. Za testiranje API-jev smo uporabili orodje Postman, ki nam je omogočilo kreacijo map v katerih smo imeli vse posamezne requeste za vse objekte. Te mape smo lahko med seboj tudi delili in tako zagotovili enako okolje za testiranje API-ja. Postman omogoča tudi pisanje skript, ki smo jih uporabili pri generiranju naključnih podatkov za namena polnjenja baze za testiranje. Za lažje dokumentiranje APIja smo tudi uporabili orodje Swagger, ki nam je lepo vizualno prikazalo, katere vse končne točke ima aplikacija, kaj sprejmejo, kje so dostopne in kaj vračajo. Omogoča tudi pregled vseh shem, ki se uporabljajo za pošiljanje podatkov preko http protokola.

Frontend (pročelje)

Spletno aplikacijo smo izdelali s pomočjo programske knjižnice React v programskem jeziku Typescript. Za ogrodje smo uporabili Next.js, ki ima številne prednosti:

- komponente smo lahko označili kot "server komponente", kaj izboljša samo delovanje aplikacije
- v osnovni obliki omogoča navigacijo s pomočjo app routerja, kaj je pomenilo, da ga ni potrebno lastno implementirati

Za vizulani izgled aplikacije smo uporabili nizko nivojski okvir za CSS Tailwind CSS, ki nam je omočil enostavno izgradnjo komponent s uporabo pripomočkov. Komponente so tako hitro postale odzivne in se držale enakega stila, saj smo glavne CSS atribute nastavili v **tailwind.config.ts** datoteki (border-radius, margin, padding, width, primary-color, background-color,...).

Za prijavo in registracijo uporabnikov smo uporabili Firebase, ki omogoča možnost prijave s emailom in geslom. Prav tako omogoča avtentikacijo uporabnikov tako na frontendu, kot na backendu.

Gostovanje

Za namene predstavitve smo projekt naložili na splet. Za gostovanje backenda in Mongo podatkovne baze smo uporabili ponudnika Railway, ki nam je omogočil enostavno nalaganje iz GitHub repozitorija in tudi enostavno dodajanje okoljskih spremenljivk, ki so potrebne za povezavo na bazo in avtentikacijo. Za gostovanje frontenda smo izbrali ponudnika Vercel, ki je oblačna platforma za gostovanje JavaScript storitev in podpira Next.js. Omogoča hitro gostovanje s pomočji GitHub repozitorija in okoljskih spremenljivk, ki so potrebne za avtentikacijo.

Organizacija in način dela

Komunikacija

Vsa komunikacija znotraj projekta je bila speljana preko Discord strežnika, razen komuniciranje s profesorjem, ki je bilo preko maila. Naredili smo ločene kanale za posamezne dele projekta. Tako smo uspeli ločit komunikacijo, da smo lahko reševali probleme ločeno in se zadeve niso izgubile oziroma se niso problemi spregledali. Prav tako smo preko Discord strežnika izvedli vse skupinske sestanke in vso delo na daljavo, kjer smo si pomagali s funkcijo deljenja zaslona.

Med izvedbo samega projekta smo izvedli tudi številne sestanke s profesorjem v živo. Namen teh je bil prikazati dosedanje delo, dobiti povratne informacije in morebitne napotke za nadaljne delo.

Simple Contact Manager - MMA

SCM oziroma Simple Contact Manager, je spletna aplikacija namenjena upravljanju s kontakti. Njena glavna prednost pred ostalimi CRM (Customer Relations Manager) programi, je njena preposta uporaba, saj je namenjena izključno delu s kontakti. Ta datoteka, bo služila kot predstavitev projekta, navodila za namestitev aplikacije lokalno, uporaba aplikacije na spletu in dokumentacija za aplikacijo SCM.

Člani ekipe MMA

- Miha Prah
- Matija Krsnik
- Anže Golob

Kazalo vsebine projekta

1. Predstavitev projekt
   - Podroben opis projekta
2. Dokumentacija
   - Funkcionalnosti
   - Tehnološki nabor
   - Organizacija in način dela
   - Zasnova podatkovne strukture
   - Diagram primerov uporabe
   - Graf registracije novega uporabnika in graf avtentikacije
   - Wireframe aplikacije (prototip izgleda)
   - Testiranje
   - GitHub Actions
   - Optimizacija kode
   - Uporabniške zgodbe
   - Testni scenariji
3. Navodila za namestitev lokalno
   - Testno lokalno okolje
4. Uporaba aplikacije
   - Uporabniški priročnik
   - Končni izgled aplikacije

1. Predstavitev projekta

Podroben opis projekta

Načrtovati, nadgraditi, implementirati in namestiti bo potrebno sistem, ki bo uporabnikom omogočal vodenje kontaktov na nekem projektu ali v nekem podjetju. Skupino kontaktov lahko uporabniki kreirajo sami, kontakti naj bodo vidni le znotraj te skupine; poskrbeti bo torej treba za izolacijo podatkov med različnimi uporabniki in njihovimi projekti. Podatki o uporabnikih naj bodo do določene mere predefinirani, omogočeno pa naj bo tudi poljubno definiranje kontaktov v smislu ključ-vrednost. Pričakuje se označevanje kontaktov s poljubnimi značkami, ki bodo služile kot kategorije kontaktov. Voditi bo potrebno tudi vse spremembe kontaktov (revizijska sled) in jih tudi primerno verzionirati. Dostop do zalednega sistema naj bo mogoč preko ustrezno zavarovanega REST vmesnika. Uporabniški vmesnik naj nudi prijazen pregled, iskanje, filtriranje ter izvoz označenih kontaktov v format MS Excel. Konkurenčna prednost izdelka prek konkurenčnimi CRM sistemi bo torej ravno enostavnost uporabe.

Zasnova podatkovne strukture

Za shranjevanje podatkov smo uporabili MongoDB, kjer smo uporabljali različne kolekcije. Struktura temelji na skupini kontaktov, ki smo jo imenovali "Tenant". Ko uporabnik ustvari novo skupino, se usvarijo 3 kolekcije:

- Tenant main -> kolekcija, ki beleži vse aktivne kontakte znotraj ene skupine
- Tenant deleted -> kolekcija, ki beleži vse izbrisane kontakte znotraj ene skupine
- Tenant activity -> kolekcija, ki beleži vse aktivnosti, ki jih izvede uporabnik znotraj eneke skupine, kot so recimo ustvarjanje, posodabljanje ali brisanje kontaktov. Ta kolekcija nam omogoča usvarjanje revizijske sledi.

Posamezen kontakt ima kot privzeto nastavljena title oziroma naziv, ostale atribute lahko poljubno dodaja uporabnik, kot pare ključ in vredost. Kontaktu se lahko dodajo tudi značke in komentarji. Vse spremembe na posamznem kontaktu se beležijo in hranijo v kolekcijo activity.

Obstaja tudi glavna kolekcija "Tenants", kjer so zabeleženi dodatni podatki o vsaki kolekciji, kot so njeno celo ime, opis, barva, ime uporabnikov, ki imajo do nje dostop, shranjeni atributi za prikaz in pa tako imenovai label, ki omogočajo mapiranje poljubno mapiranje ključ vrednosti za vsako skupino uporabnikov posebaj.

Kolekcija "Predefined searches", je namenjena shranjevanju iskalnih podatkov posameznih uporabnikov, če se ti uporabljajo recimo zelo pogosto. Uporabniik lahko ob filtriranju shrani nastavitve filtriranja za prihodnjo upšorabo.

Spodaj je skica, ki prikazuje strukturo naše podatkovne baze.

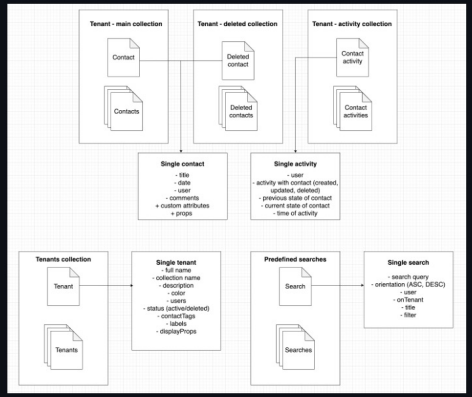
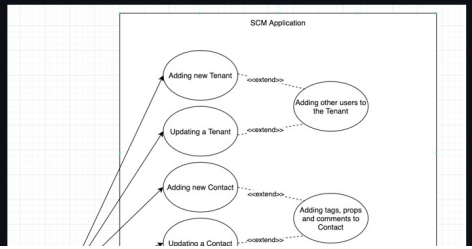
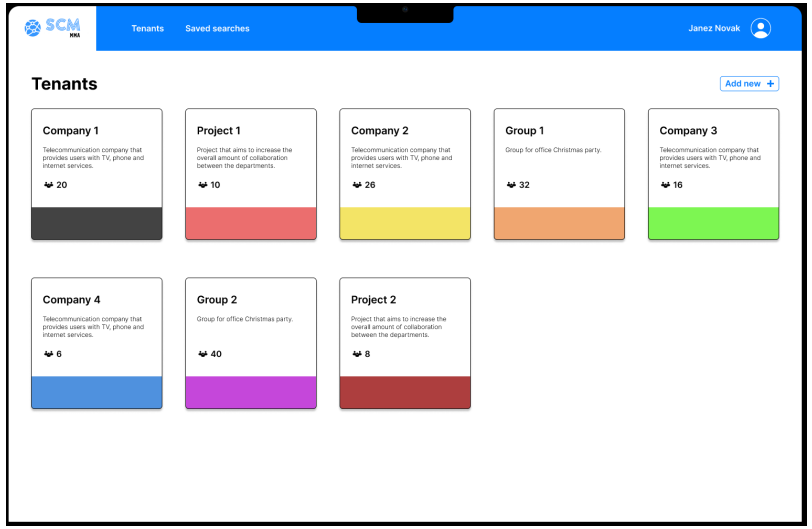


Diagram primerov uporabe

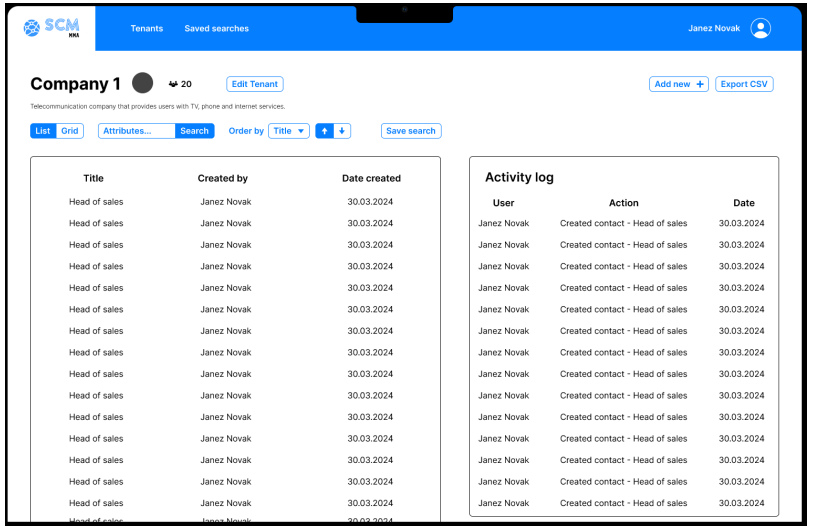


Wireframe

Pregled vseh skupin



Pregled vseh kontaktov seznam



Pregled vseh shranjenih filtrov

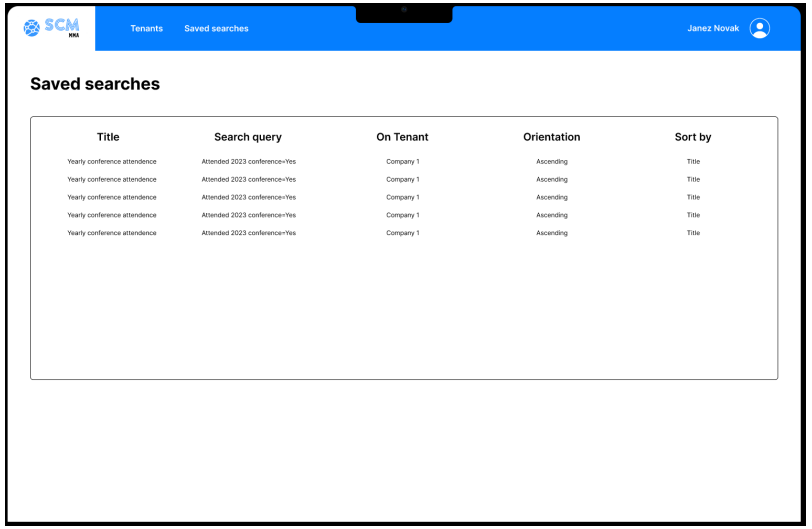
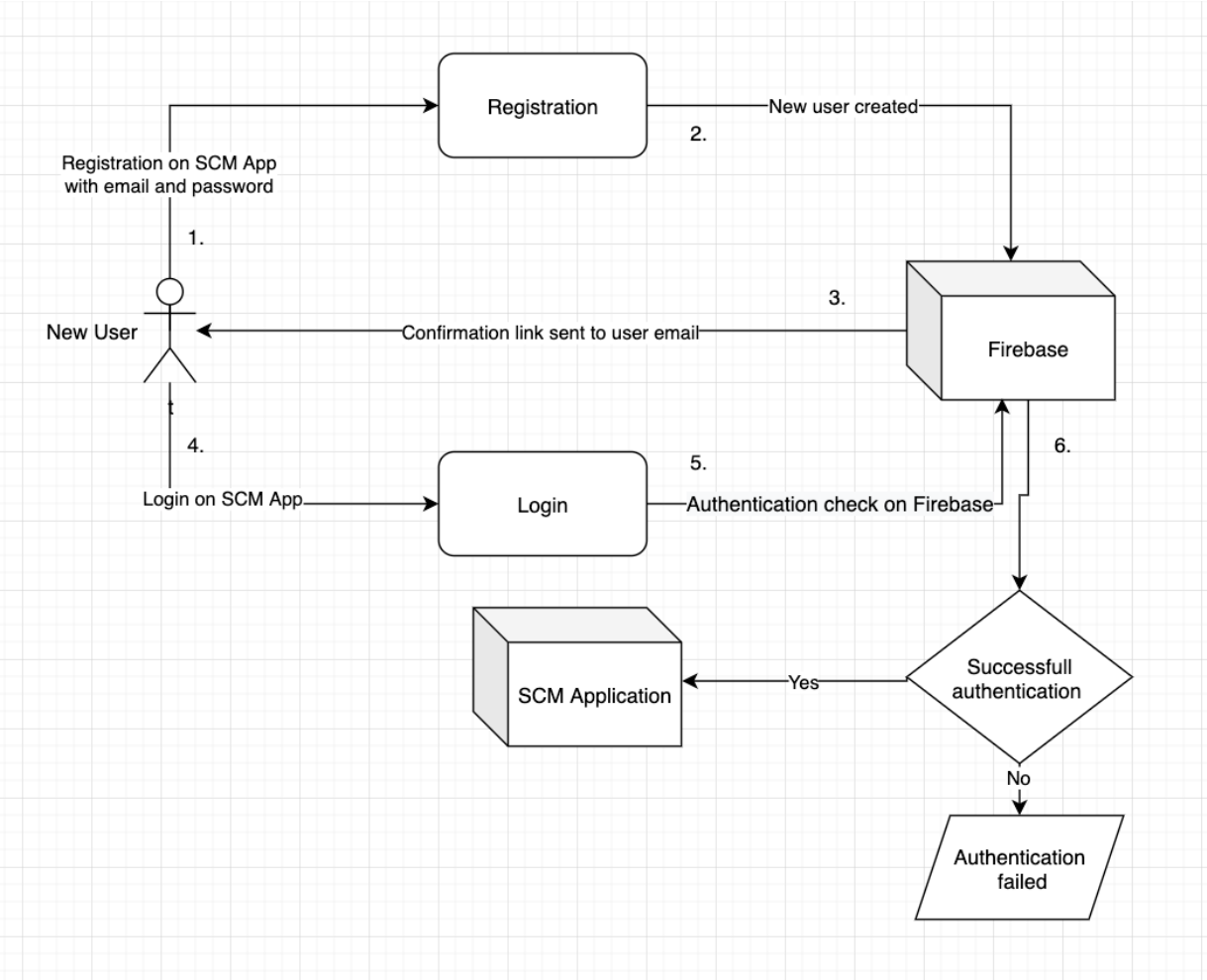
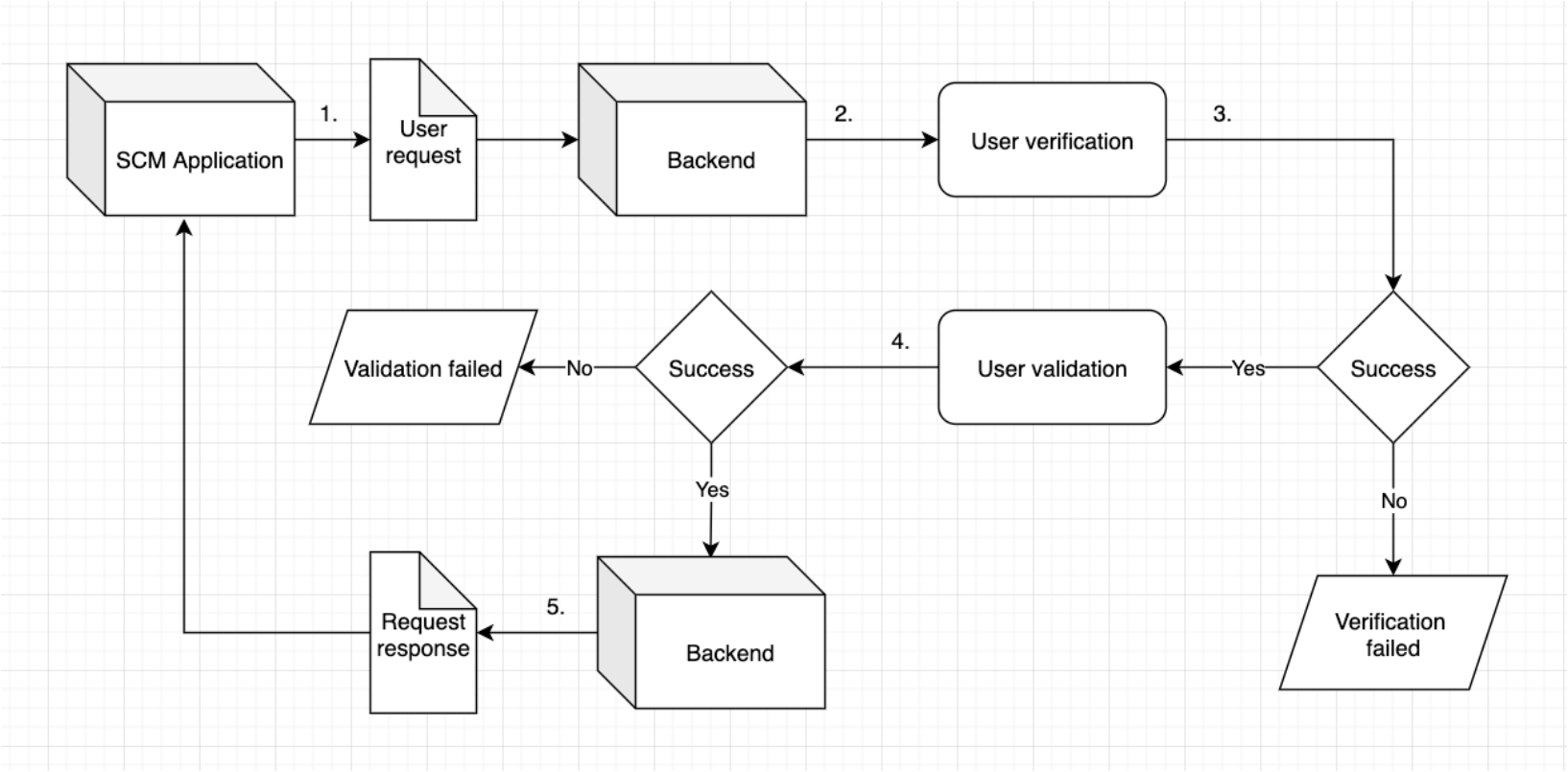


Diagram primerov uporabe

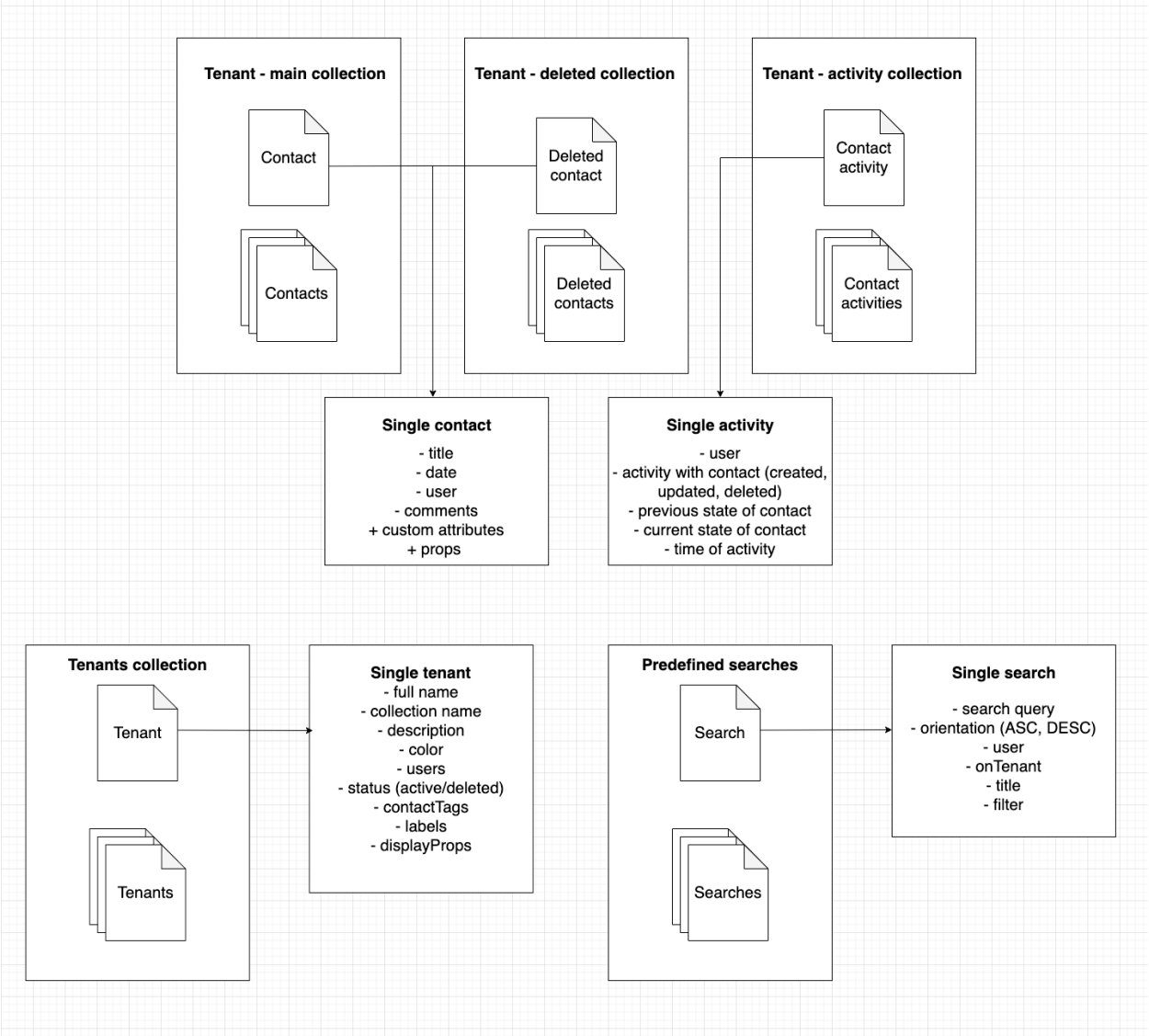
Registracija novega uporabnika



Avtentikacija uporabnika



Podatkovna struktura



SonarCloud

Main Branch Summary

2.6k Lines of Code ? • Version 0.0.1-SNAPSHOT

Quality Gate Status ?

✓ Passed

Measures

Last analysis 50 minutes ago • 076f8c1e

New Code | Overall Code

Security	Reliability	Maintainability
0 Open issues A	0 Open issues A	49 Open issues A
0 H · 0 M · 0 L	0 H · 0 M · 0 L	10 H · 21 M · 18 L

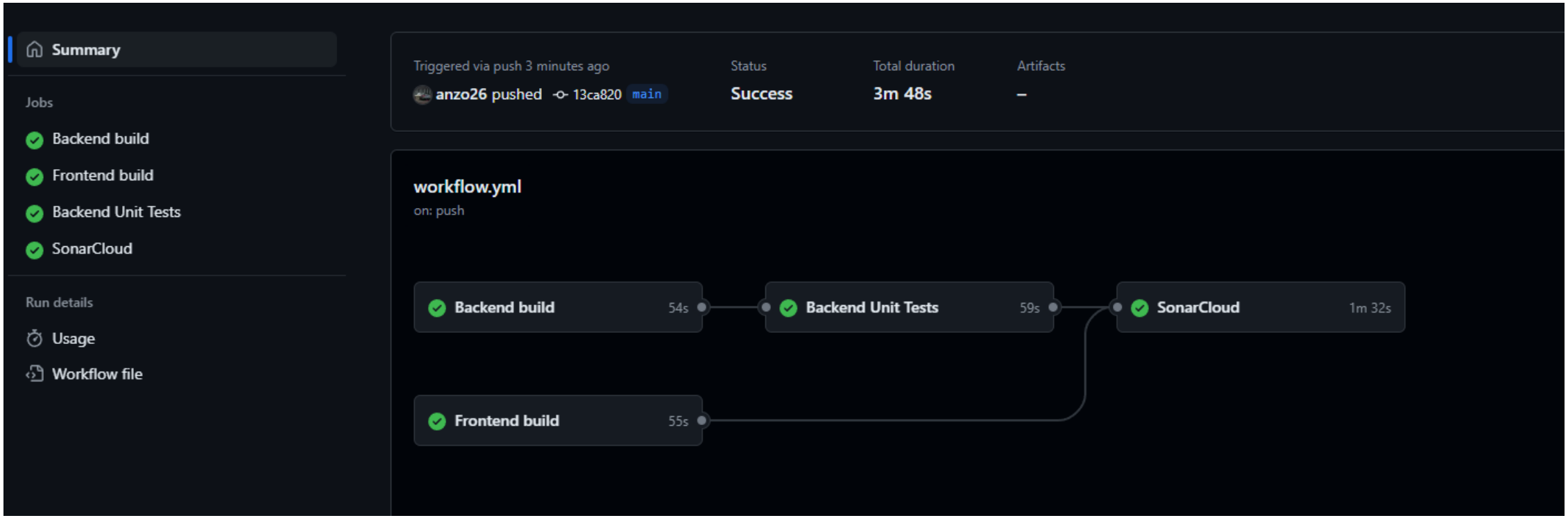
Accepted Issues
15

Coverage
54.9%
No conditions set on 1.5k Lines to cover

Duplications
1.5%
No conditions set on 3.2k Lines

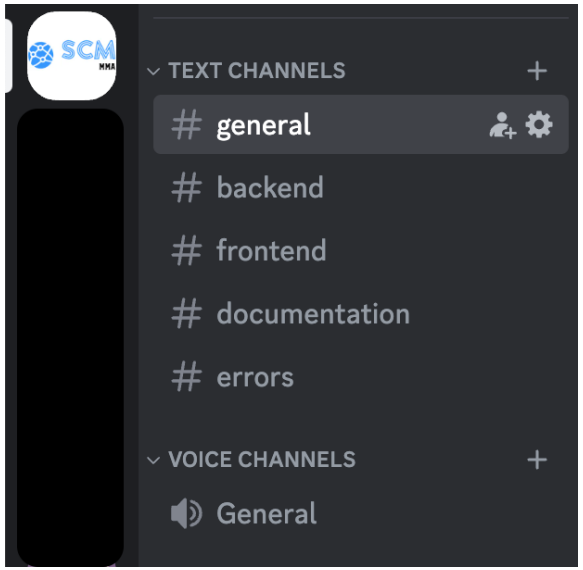
Security Hotspots
0

GitHub Actions

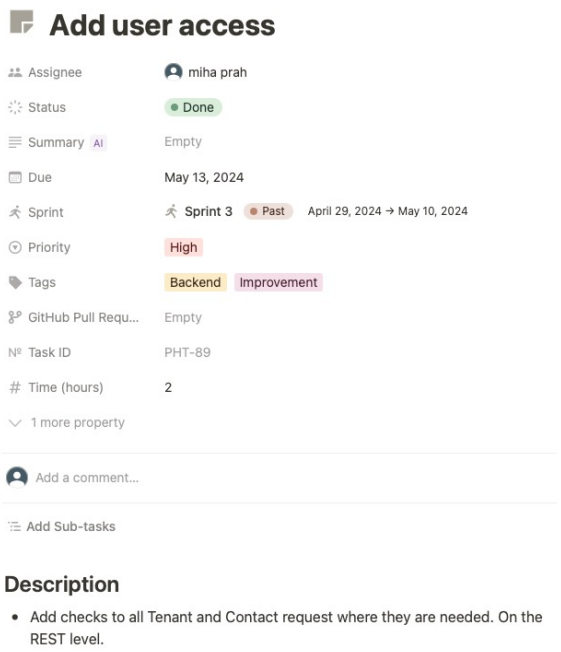


Vodenje projekta

Discord strežnik



Primer task



Add user access

Assignee	miha prah
Status	Done
Summary AI	Empty
Due	May 13, 2024
Sprint	Sprint 3 · Past · April 29, 2024 → May 10, 2024
Priority	High
Tags	Backend · Improvement
GitHub Pull Requ...	Empty
Nº Task ID	PHT-89
Time (hours)	2

1 more property

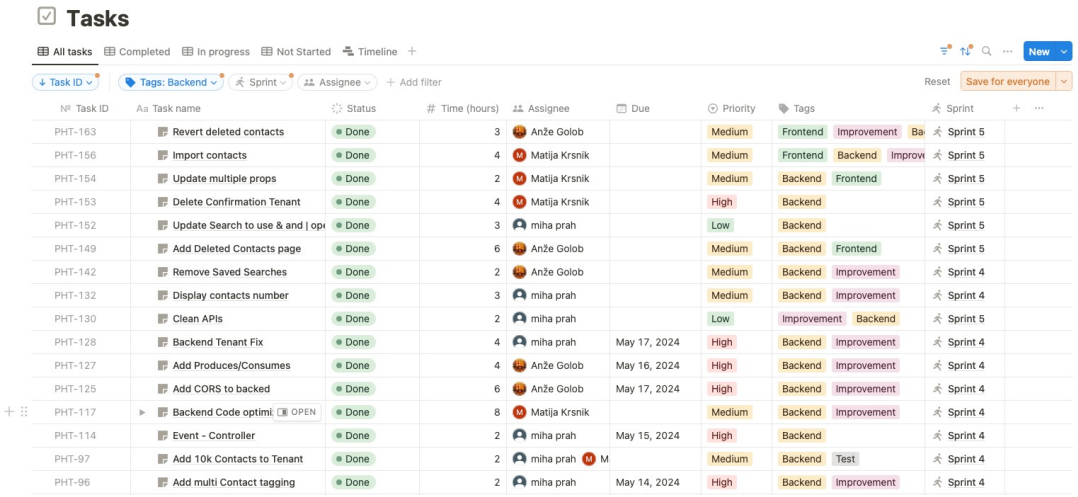
Add a comment...

Add Sub-tasks

Description

- Add checks to all Tenant and Contact request where they are needed. On the REST level.

Pregled vseh taskov



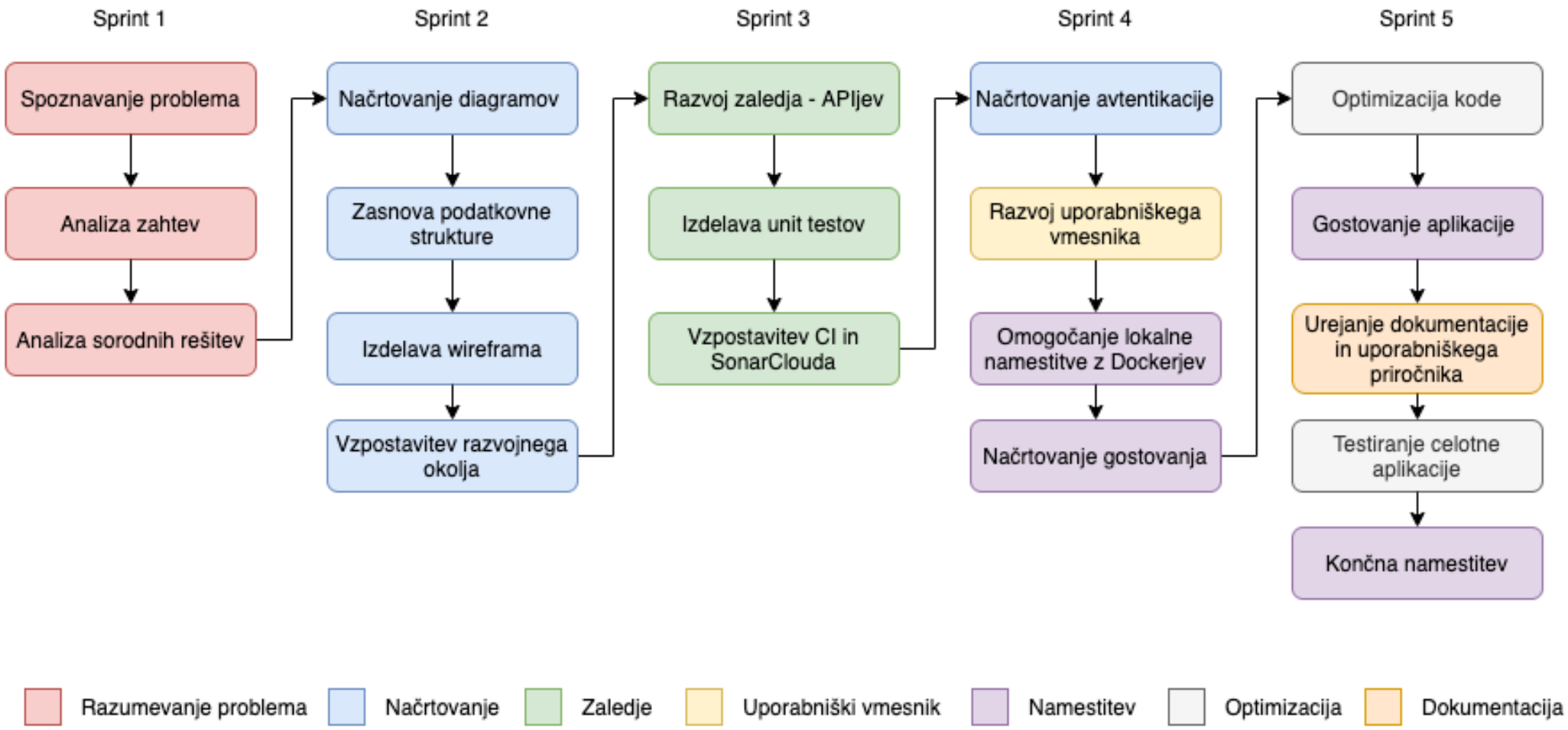
Tasks

All tasks · Completed · In progress · Not Started · Timeline

Task ID · Tags: Backend · Sprint · Assignee · Add filter · Reset · Save for everyone · New

Nº Task ID	Aa Task name	Status	# Time (hours)	Assignee	Due	Priority	Tags	Sprint
PHT-163	Revert deleted contacts	Done	3	Anže Golob		Medium	Frontend · Improvement · Ba	Sprint 5
PHT-156	Import contacts	Done	4	Matija Krsnik		Medium	Frontend · Backend · Improve	Sprint 5
PHT-154	Update multiple props	Done	2	Matija Krsnik		Medium	Backend · Frontend	Sprint 5
PHT-153	Delete Confirmation Tenant	Done	4	Matija Krsnik		High	Backend	Sprint 5
PHT-152	Update Search to use & and | ope	Done	3	miha prah		Low	Backend	Sprint 5
PHT-149	Add Deleted Contacts page	Done	6	Anže Golob		Medium	Backend · Frontend	Sprint 5
PHT-142	Remove Saved Searches	Done	2	Anže Golob		Medium	Backend · Improvement	Sprint 4
PHT-132	Display contacts number	Done	3	miha prah		Medium	Backend · Improvement	Sprint 4
PHT-130	Clean APIs	Done	2	miha prah		Low	Improvement · Backend	Sprint 5
PHT-128	Backend Tenant Fix	Done	4	miha prah	May 17, 2024	High	Backend · Improvement	Sprint 4
PHT-127	Add Produces/Consumes	Done	4	Anže Golob	May 16, 2024	High	Backend · Improvement	Sprint 4
PHT-125	Add CORS to backed	Done	6	Anže Golob	May 17, 2024	High	Backend · Improvement	Sprint 4
PHT-117	Backend Code optimi OPEN	Done	8	Matija Krsnik		Medium	Backend · Improvement	Sprint 4
PHT-114	Event - Controller	Done	2	miha prah	May 15, 2024	High	Backend	Sprint 4
PHT-97	Add 10k Contacts to Tenant	Done	2	miha prah M		Medium	Backend · Test	Sprint 4
PHT-96	Add multi Contact tagging	Done	2	miha prah	May 14, 2024	High	Backend · Improvement	Sprint 4

Iterativno proces razvoja aplikacije



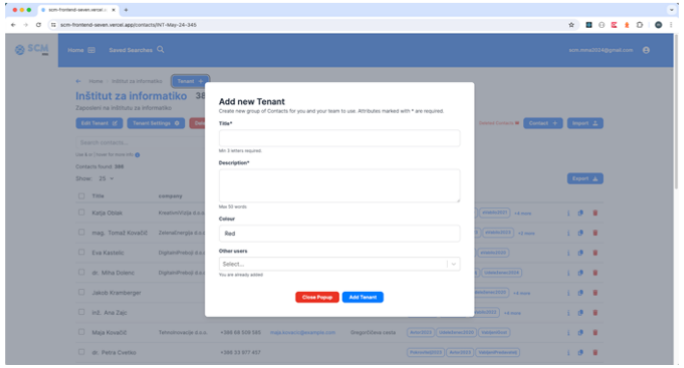
Uporabniški priročnik

SCM MMA

Uporabniški priročnik

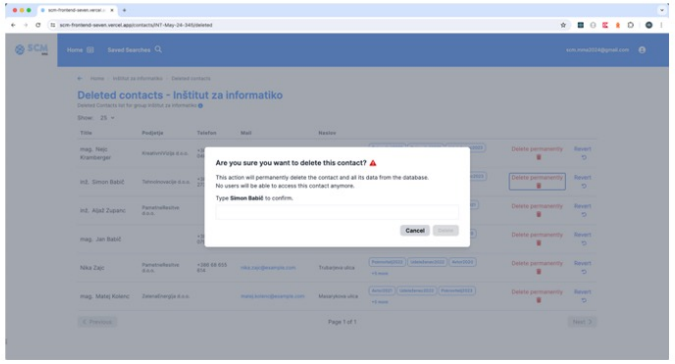
Dodajanje skupine (Tenant)

Uporabnik lahko doda novo skupino na domačem pogledu s pritiskom na gumb »Tenant +« ali pa na pregledu vseh kontaktov. Ob kreaciji nove skupine mora podati njen naslov, ki mora biti dolg vsaj 3 znake, opis, ki mora biti dolg vsaj 10 znakov. Opcijsko lahko izbere tudi barvo skupine, ki se bo prikazala na domačem zaslonu (privzeto je nastavljena rdeča barva). Uporabnik lahko tudi doda druge uporabnike, ki želi, da imajo dostop do skupine. Sebe uporabnik ne rabi naštevati med uporabnike (Users).



Trajno brisanje kontakta

Uporabnik lahko s pritiskom na gumb »Delete permanently« na pogledu vseh izbrisanih kontakov posamezne skupine izbriše kontakt za vedno iz vseh baz. Odpre se modalno okno, ki uporabnika opozori o resnosti njegovega dejanja. Za večjo varnost mora uporabnik vnesti naslov (Title) kontakta, preden ga lahko za vedno izbriše.



Miha Prah, Anže Golob, Matija Krsnik

Testni scenariji

SCM
MMA

Testni scenariji

Miha Prah, Anže Golob, Matija Krsnik

Uporabniška zgodba 1:

- Kot nov uporabnik želim uporabljati aplikacijo SCM.

Testni scenarij 1: Registracija novega uporabnika

Namen:

- Preveriti, ali je registracija novega uporabnika na spletni strani uspešna.

Predpogoji:

- Uporabnik še ni registriran v sistem.
- Nahaja se na spletni strani SCM.
- Na drugem zavihku ima odprto svojo e-pošto.

Vhodni podatki:

- Podatki novega uporabnika (e-pošta, geslo).

Scenarij testiranj:

	AKCIJA	PRIČAKOVAN REZULTAT
1.	Na login strani pod vnosnimi polji je link, ki nas vodi do registracijske strani	Prikaže se nam podstran /register.
2.	Vnesemo zahtevane podatke za registracijo ter potrdimo z klikom na gumb »Register«.	Na podan e-poštni naslov se pošlje sporočilo s povezavo za verifikacijo.
3.	Odpremo zavihek s svojo e-pošto.	Prikaže se nam novo sporočilo v naši e-pošti.
4.	Kliknemo na najnovejše sporočilo prejeto s strani noreply@scm2024-31a54.firebaseapp.com,	Prikaže se nam vsebina sporočila pošiljatelja noreply@scm2024-31a54.firebaseapp.com,
5.	Kliknemo na povezavo, ki nam je podana v sporočilu.	Povezava nas preusmeri na spletno stran, kjer se nam preveri vnesen e-poštni naslov.
6.	Če je verifikacija bila uspešna, smo se uspešno registrirali.	Uporabnik se lahko prijavi v spletno aplikacijo.

Pogoji uspešnega zaključka:

- Uporabnik je registriran, dodan v podatkovno bazo in se lahko uspešno prijavi v spletno aplikacijo.

Uporabniška zgodba 2:

- Kot registrirani uporabnik se želim prijaviti v aplikacijo SCM, da lahko dostopam do njenih funkcionalnosti in upravljam svoje naloge.

Testni scenarij 2: Prijava uporabnika

Namen:

- Preveriti, ali je prijava registriranega uporabnika na spletni strani uspešna.

Predpogoji:

- Uporabnik je že registriran v sistemu.
- Uporabnik se ob začetku nahaja na prijavni strani spletne aplikacije SCM.

Vhodni podatki:

- Podatki registriranega uporabnika (e-pošta, geslo).

Scenarij testiranj:

	AKCIJA	PRIČAKOVAN REZULTAT
1.	Uporabnik odpre prijavno (/login) stran aplikacije SCM.	Prikaže se prijavna stran z vnosnimi polji za e-pošto in geslo.
2.	Uporabnik v polje za e-pošto vnese svoj registrirani e-poštni naslov.	E-poštni naslov je vnesen v ustrezno vnosno polje.
3.	Uporabnik v polje za geslo vnese svoje pravilno geslo.	Geslo je vneseno v ustrezno vnosno polje.
4.	Uporabnik klikne na gumb »Log in«.	Sistem preveri vnesene podatke.
5.	Če so e-pošta in geslo pravilni, je uporabnik uspešno prijavljen.	Uporabnik je preusmerjen na nadzorno ploščo aplikacije SCM.
6.	Če so e-pošta in geslo nepravilni, sistem prikaže ustrezno sporočilo o napaki.	Uporabniku se prikaže sporočilo v desnem zgornjem kotu "Invalid email or password!"

Pogoji uspešnega zaključka:

- Uporabnik je uspešno prijavljen in preusmerjen na nadzorno ploščo aplikacije SCM.
- V primeru napačnih podatkov se prikaže ustrezno sporočilo o napaki in uporabnik ostane na prijavni strani.

Priložnosti za nadaljni razvoj

- Optimizacija algoritma iskanja
- Možnost izbire atributov za izvoz
- Avtomatska izdelava varnostnih kopij
- Naprednejše filtriranje

Hvala za pozornost !

- Ekipa MMA