

Naziv projekta:

Tražim cimera



Prijedlog upućen: **Selma Skopljaković**, selma.skopljakovic@gmail.com

Prijedlog upućuje: **trazimcimera.ba**

Tim:

Fuad Herac, fuad.herac@gmail.com,

Emir Bračković, brackovic97@gmail.com i

Enver Kovačević, enverkovacevic@hotmail.com

Sadržaj:

Uvod	3
Motiv i ideja	4
Planovi i cilj	4
Analiza sistema	5
1. Dizajn	5
2. Unos content-a	10
3. Edit content-a	12
4. Galerija	13
5. Baza podataka	13
6. User managment	14
Tehnologije	15
React	15
Apollo - GraphQL	20
Detaljan razvoj baze podataka	21
Korisnik	22
Objava i tip objave	22
Slike	22
Admin	23
Kontakt i utisci	23
Node – Apollo – GraphQL API	24

Uvod

Tim **trazimcibera.ba** osnovan je 25.02.2020. godine. Kroz nekoliko sastanaka i online diskusija razvijena je ideja o web sistemu za studente pomoću kojeg bi tražili cimera/cimere u određenom gradu.

Tim trenutno čini 4 studenata sa smjera Informacione Tehnologije na Prirodno-matematičkom fakultetu.

Članovi su sljedeći:

- **Fuad Herac** (Project manager i Front-end developer),
- **Emir Bračković** (Dizajn i Baza podataka),
- **Enver Kovačević** (Back-end developer).

Motiv i ideja

Opće poznata činjenica je da svake godine, stotine studenata/studentica traže cimer u različitim gradovima BiH (kao što su: Sarajevo, Banja Luka, Mostar itd.). Naime, mi kao tim (i kao studenti između ostalog) smo primijetili da se ti studenti inače „muče“ da nađu prikladnog cimera za iduću akademsku godinu. Stoga smo odlučili da razvijemo web sistem kako bi olakšali isto.

Srž ideje je da razvijemo sistem gdje bi svaki student mogao imati profil, te bi se mogao povezati sa drugim studentima. Iduća stvar koja je esencijalna za ovakav sistem je CMS koji bi omogućio korisnicima našeg sistema da objavljuju oglase o traženju cimera, da odgovaraju, prate oglase drugih, te uređuju ili brišu svoje već objavljene oglase. Pomoću integrisanih filtera (radi lakše kategorizacije i spajanja određenih korisnika sa sličnim preferencama) prethodno pomenuto bi bilo nekoliko klikova umjesto mukotrpne potražnje cimera po raznim portalima i stranicama koje nisu namijenjene za to.

Planovi i cilj

Plan je da cijeli sistem bude što jednostavniji za upotrebu, odnosno da sistem bude što intuitivniji. Planiramo da što prije podijelimo zadatke i da se svaki član našeg tima fokusira na dio posla u kojem je najbolji. Pokušat ćemo da se stavimo u poziciju budućih korisnika i u potpunosti razumijemo njihove potrebe. Također, planiramo da u roku od nekoliko sedmica napravimo kostur sistema, sa osnovnim funkcionalnostima, kako bi imali dobru osnovu na koju bi poslije dodavali funkcionalnosti i koji bi stilizovali. Ta prva verzija sistema bi nam bila dovoljna za kratku prezentaciju osnovne ideje našeg projekta.

Jedan od glavnih cijeva nam je da sistem bude završen do kraja semestra, jula 2020, te da položimo Web Programiranje II.

Radovi na projektu su počeli dana 28.02.2020.

Analiza sistema

1. Dizajn

Za izgled prve verzije našeg sistema koristili smo template sa wordpress-a sa temom pod nazivom: **Stratford**. Ovu temu smo poslije preuredili za naše potrebe. Koristili smo istu temu i na ostalim stranicama. U nastavku pogledajte logo našeg sistema i nekoliko snimki zaslona našeg web sajta.



Slika 1: Logo

Početna stranica:

Na početnoj stranici nalaze se linkovi koji vode na ostale stranice našeg sajta. Dugmići koji vode na stranicu za prijavu i registraciju. Na centralnom dijelu stranice se nalaze kratki tekstovi u kojima su raniji korisnici podijelili svoje mišljenje.

[Početna](#)[Oglasi](#)[Kontakt](#)[Prijavi se](#)[Registracija](#)

Ako imate stan koji iznajmljujete ili tražite stan i treba vam neko da bi s njim podijelili troškove stanovanja, na pravom ste mjestu !

[PRIJAVI SE](#)

Šta korisnici naših usluga kažu o nama?

Mnogo sam se mučio kako bih našao osobu sa kojom bih dijelio troškove stanovanja, ali mi je ova stranica taj postupak mnogo olakšala. Preko ove stranice našao sam 2 nova dobra prijatelja vjerovatno i za cijeli život i sa njima trenutno živim u blizini Kampusa

— Miralem Pjanić,
student
Elektrotehničkog
fakulteta u Sarajevu

Stvarno mi se sviđa ovaj način pronalaženja novih cimera! Prije sam stavljao oglase na svom Facebook profilu, ali to nije puno pomagalo. Ovo mjesto je spojilo puno ljudi koji imaju isti problem kao i ja i to mi mnogo olakšava proces pronalaska cimera!

— Emir Kovačević,
student
Prirodno-matematičkog
fakulteta u Sarajevu



Još se niste registrovali?

Pridružite se našoj zajednici i upoznajte nove ljude koji bi Vam mogli biti potencijalni cimeri. Objavljujte na našoj stranici oglase i nađite ono što Vam treba !

[REGISTRUJ SE](#)

Oglasi:
Na
stranici
oglasu su
prikazani
oglas koji
se nalaze
u bazi
podataka.
Prikazan
je tekst,
kontakt i
autor
oglasa a u
nekim
slučajevi
ma i
cijena.

Slika 2: Početna stranica



Oglasi

Soba sa svim pratecim u stanu na Kosevskom brdu 230 KM,uracunate režije .Pogodno za studente medicinskog, arhitektonskog, gradjevinskog i stomatoloskog fakulteta. Udaljenost od KCUS cca 5min.

TEL: +387 65 136 937

U Tuzli studentica traži cimericu u 2,5 sobnom potpuno namještenom stanu, internet. Stan je u mirnom i sigurnom kvartu, u neposrednoj blizini fakulteta. Ukupni trošak stanarine i režija 190KM po osobi.

SELMA MARIĆ; TEL:063/963/987

Tražim cimericu.Stan se nalazi u Merkur zgradi preko puta ekonomskog fakulteta u Sarajevu. 150KM plus režije.Stan je useljiv od 20.3.2020.

Zainteresovane djevojke neka se jave na 0603058178 za više informacija.

SELMA

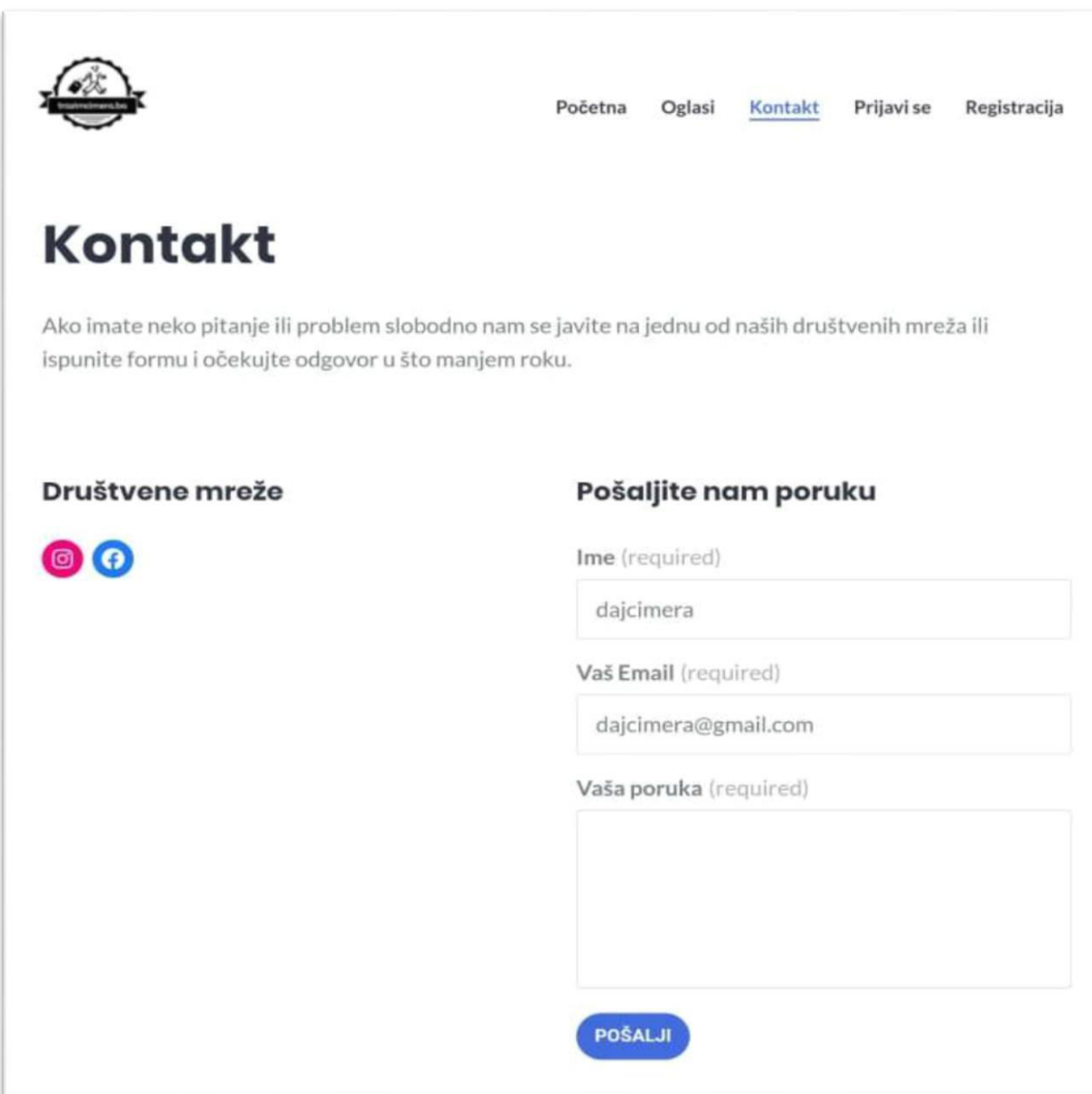
Trazim iskljucivo devojke i studentkinje za stanovanje u Zenici. Stan je opremljen i renoviran (dvorisni) . Udaljen 8km,od autobuske sanice .

SARA BEGIĆ TEL: +387-223-882

Slika 3: Stranica oglasi

Kontakt:

Za dodavanje ikonica koje vode na naš profil na društvenim mrežama, koristili smo widget: Social Icons. Pomoću njega smo na jednostavan način dodali ikonice i linkove.





Početna Oglasi Kontakt Prijavi se Registracija

Kontakt

Ako imate neko pitanje ili problem slobodno nam se javite na jednu od naših društvenih mreža ili ispunite formu i očekujte odgovor u što manjem roku.

Društvene mreže

Pošaljite nam poruku

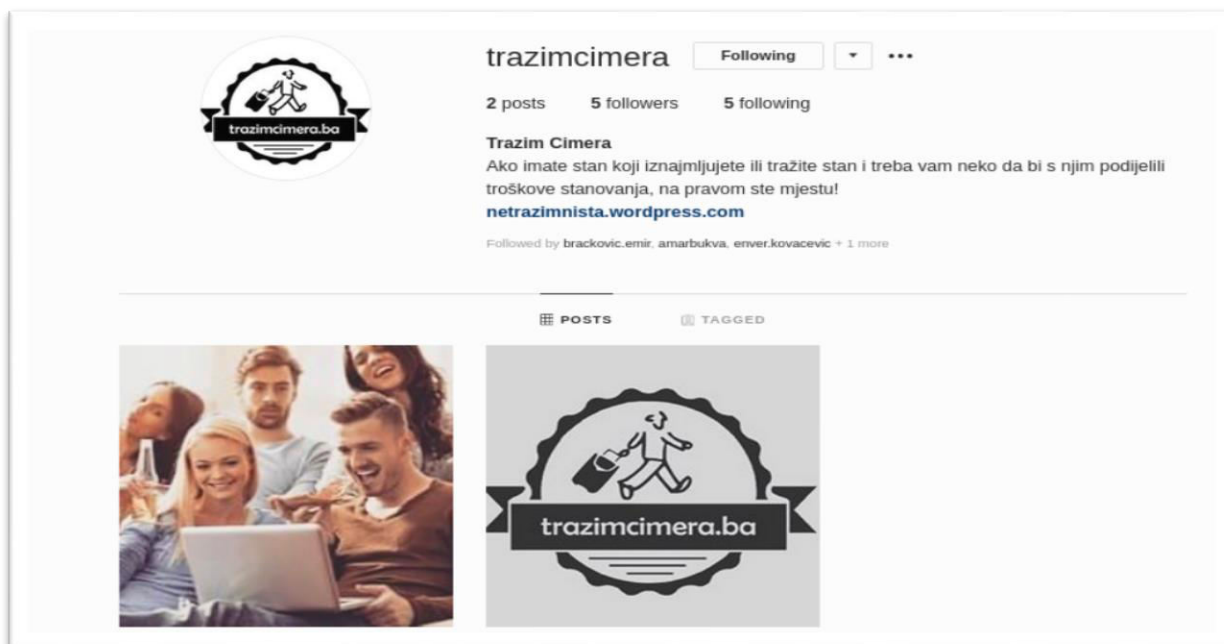
Ime (required)

Vaš Email (required)

Vaša poruka (required)

POŠALJI

Slika 4: Stranica kontakt



Slika 5: Instagram profil

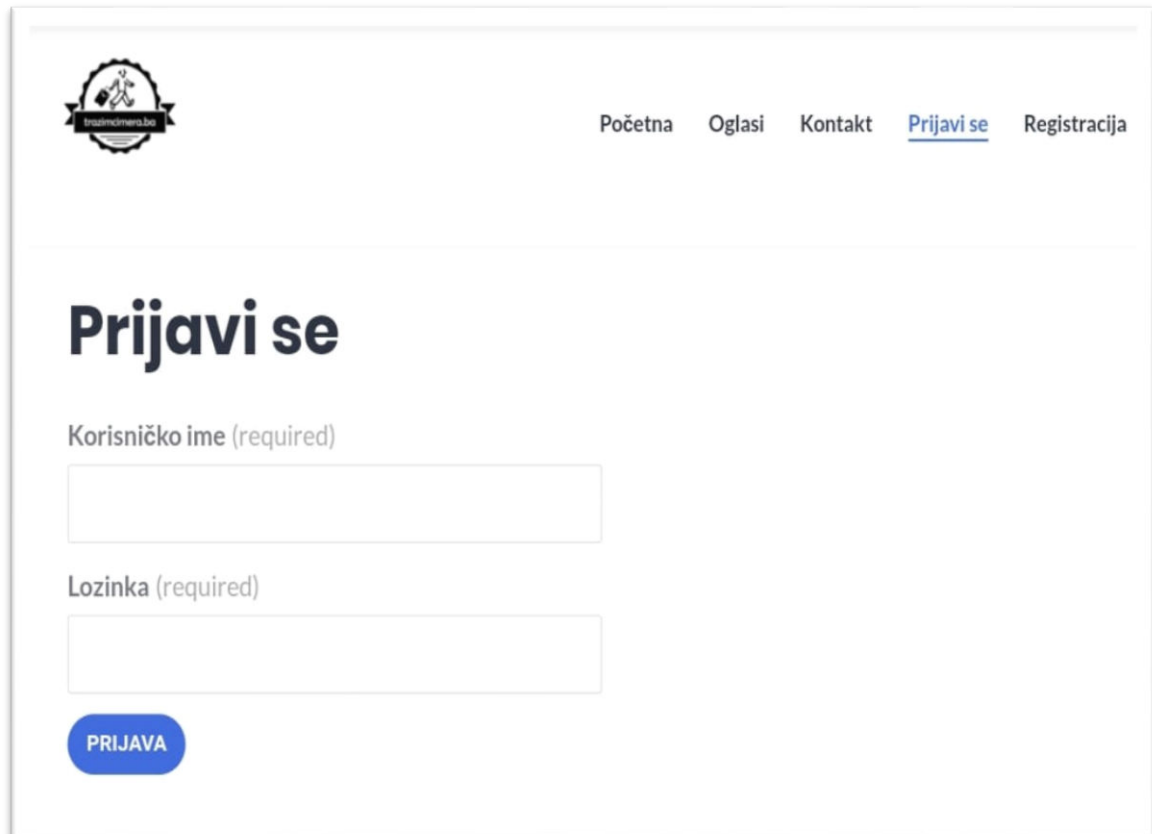


Slika 6: Facebook profil

Ikonice su zapravo linkovi koji vode na naše profile na društvenim mrežama.

2. Unos content-a

Na našem web sajtu podaci se unose u formama za prijavu i registraciju. Nakon unosa podataka kreiraju se profili i dodaju u bazu. Tip podataka mora odgovarati tipu podataka kolona u bazi podataka. Postoje i obavezna polja koja moraju biti popunjena.



The screenshot shows a web page for logging in. At the top left is a logo for 'Institucija.ba'. To the right of the logo is a navigation menu with links: 'Početna', 'Oglasi', 'Kontakt', 'Prijavi se' (which is underlined and highlighted in blue), and 'Registracija'. Below the navigation menu, the main heading 'Prijavi se' is displayed in a large, bold, dark font. Underneath the heading, there are two input fields. The first is labeled 'Korisničko ime (required)' and the second is labeled 'Lozinka (required)'. Below the second input field is a blue button with the text 'PRIJAVA' in white capital letters.

Slika 7: Stranica prijava

[Početna](#)[Oglasi](#)[Kontakt](#)[Prijavi se](#)[Registracija](#)

Registracija

Pridruživanjem našoj zajednici dobijate mogućnost da objavljujete oglase na našoj stranici!

Ime (required)

Prezime (required)

Datum rođenja (required)

Email (required)

Korisničko ime (required)

Lozinka (required)

Adresa i mjesto stanovanja

Sa koliko cimera bi željeli živjeti

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

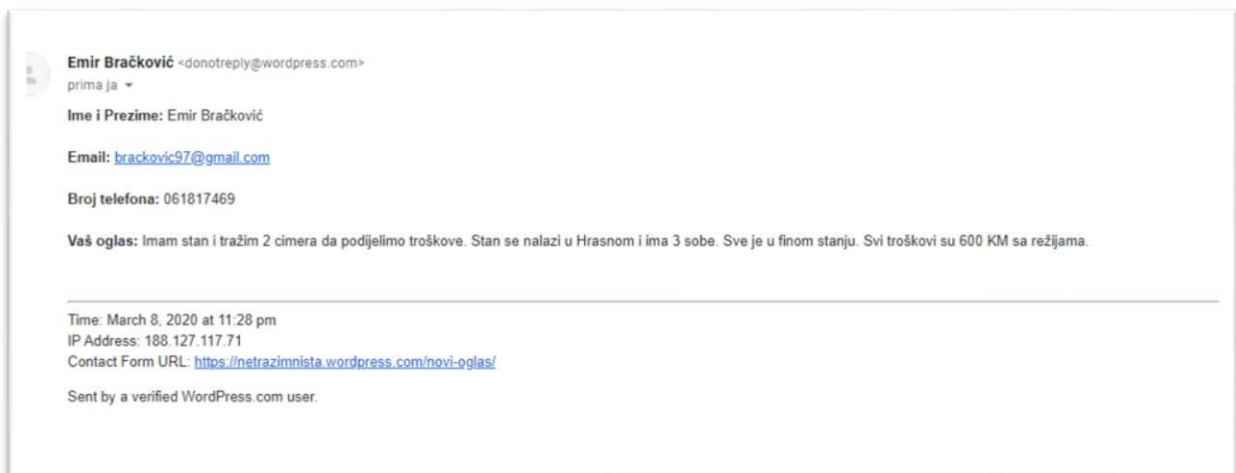
Nešto više o sebi

REGISTRUJ SE

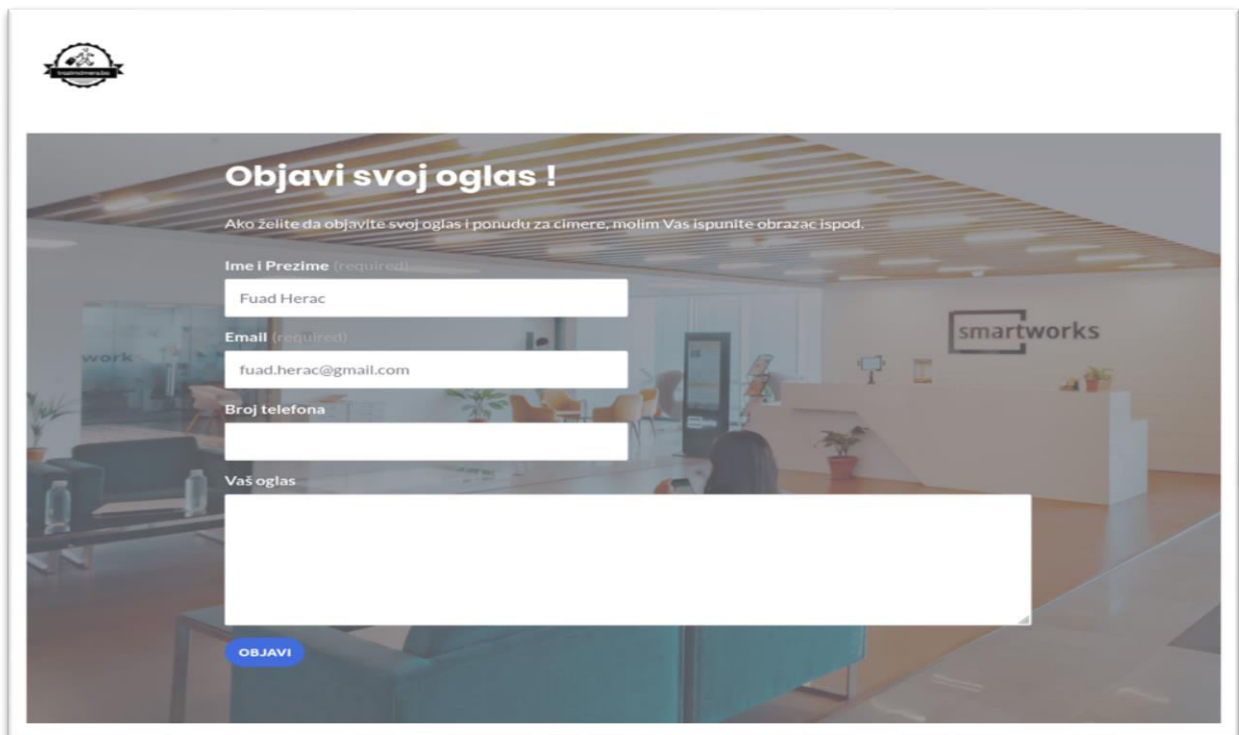
Slika 8: Stranica registracija

3. Edit content-a

Korisnik nakon uspješne registracije ima mogućnost da objavi oglas. Administrator dobija na mail sadržaj oglasa. Ukoliko je sve po propisima, odobrava oglas i on se objavljuje na stranici. Mail i stranicu za objavu oglasa pogledajte u nastavku.



Slika 9: Mail koji dolazi na naš server



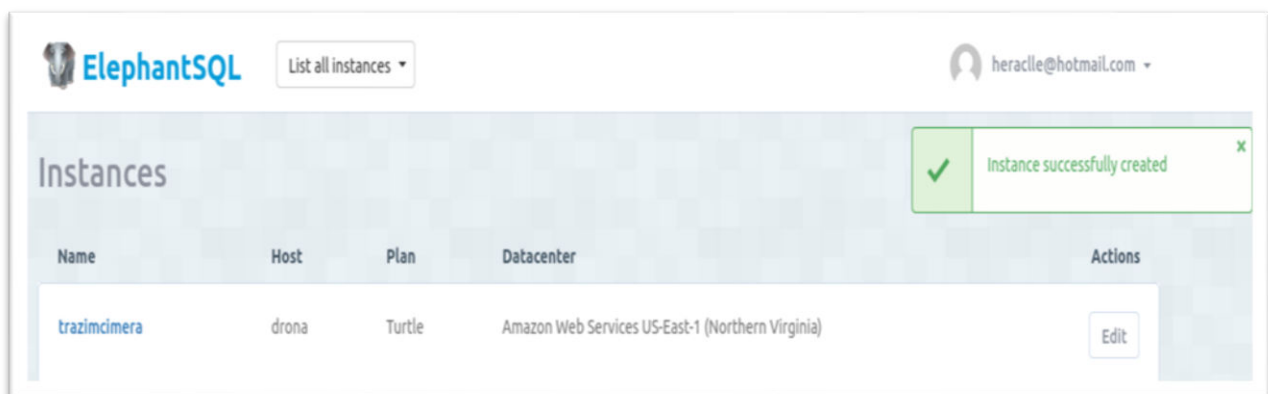
Slika 10: Stranica na kojoj kreira novi oglas

4. Galerija

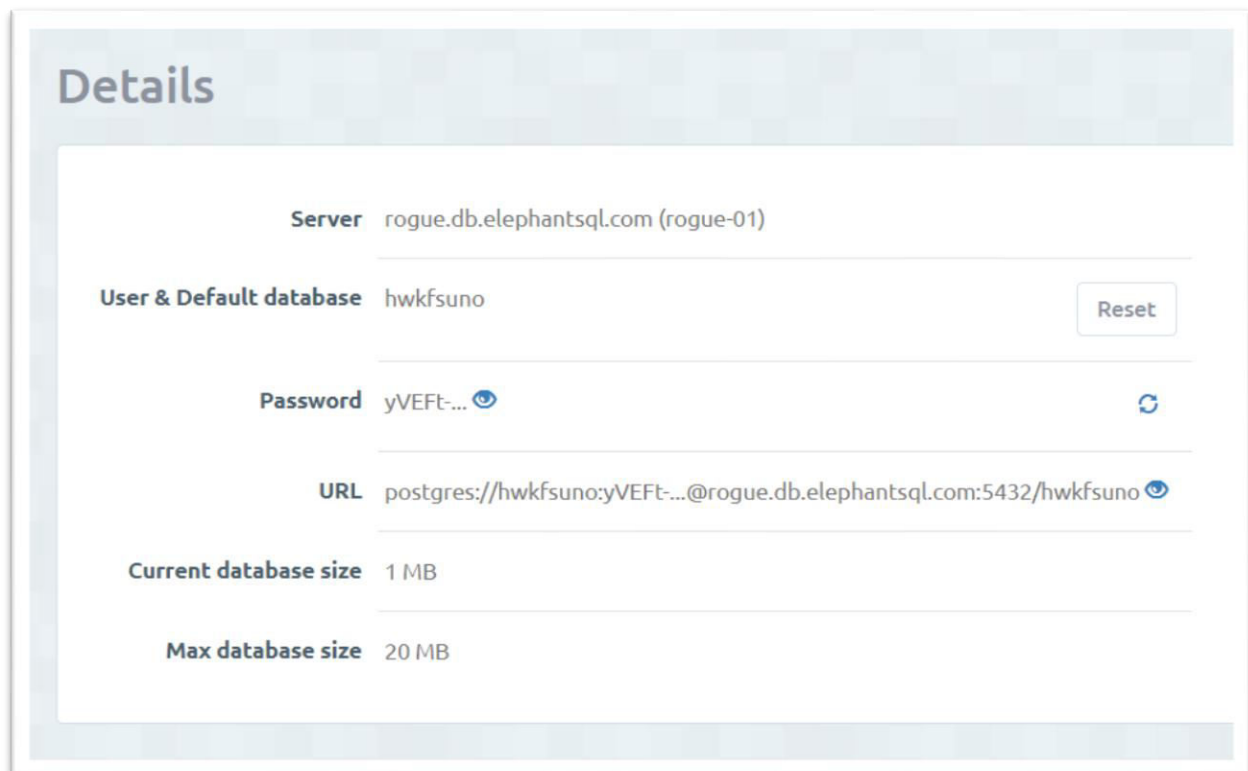
Slike se nalaze na cloud serveru, a u bazi podataka se čuvaju samo linkovi na te slike.

5. Baza podataka

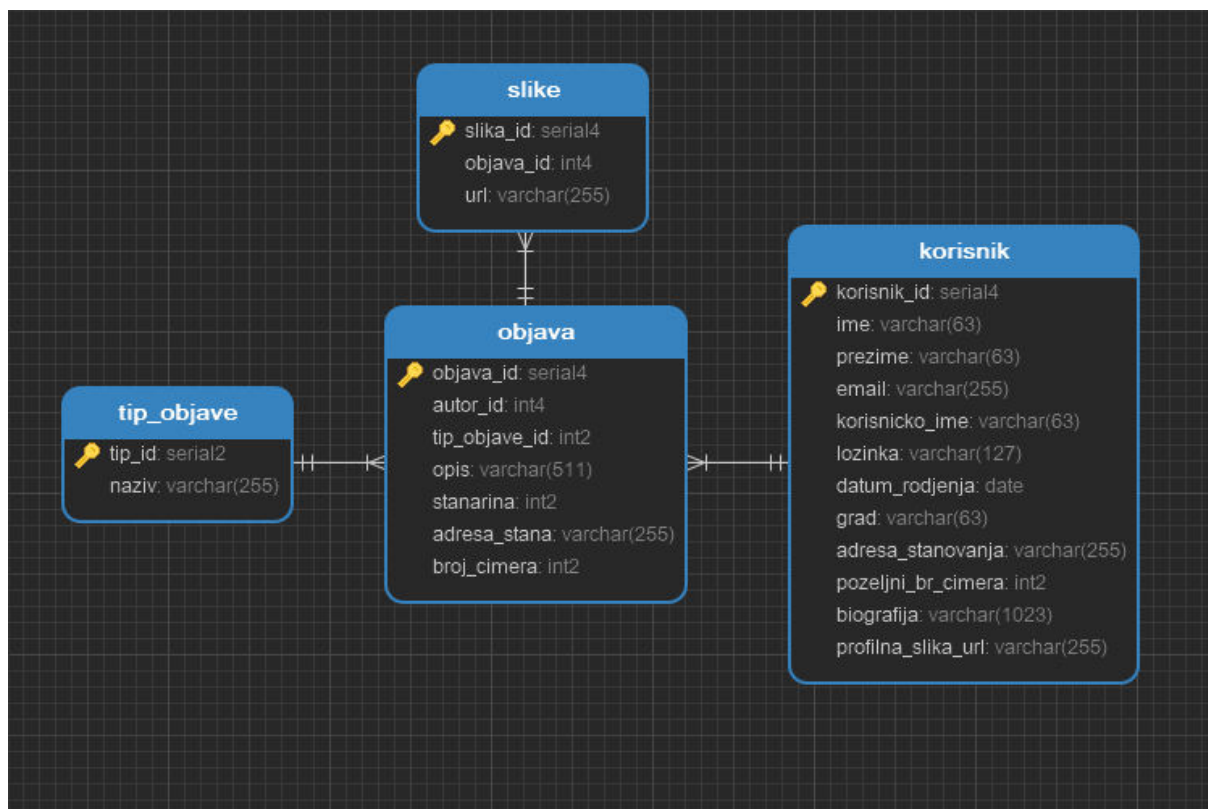
Za naš web sistem smo koristili PostgreSQL bazu podataka. Bazu podataka smo podigli na serveru ElephantSQL-a.



Slika 11: Kreiranje nove instance



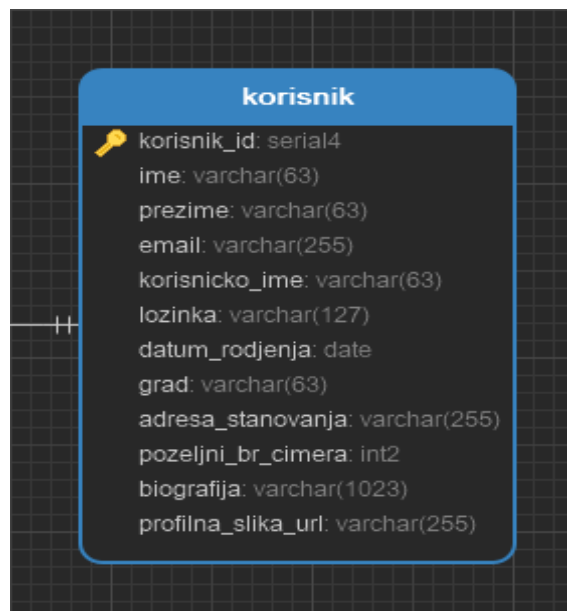
Slika 12: Podaci za pristup bazi



Slika 13: Trenutna šema naše baze podataka

6. User managment

Nakon uspješne registracije u bazu podataka se unose podaci o korisniku. Da bi se korisnici mogli razlikovati, koristi se primarni ključ `korisnik_id`, koji je unikatno za svakog korisnika. Ostali podaci predstavljaju lične podatke. Korisničko ime je unikatno. Dva korisnika ne mogu imati isto korisničko ime. Polje `grad` je polje po kojima će se vršiti provjera podudaranja lokacija i slanje notifikacija. Za profilnu sliku se čuva samo link koji vodi na cloud server na kojem je slika spašena.



Slika 14: Dio baze za korisničke podatke

Tehnologije

React

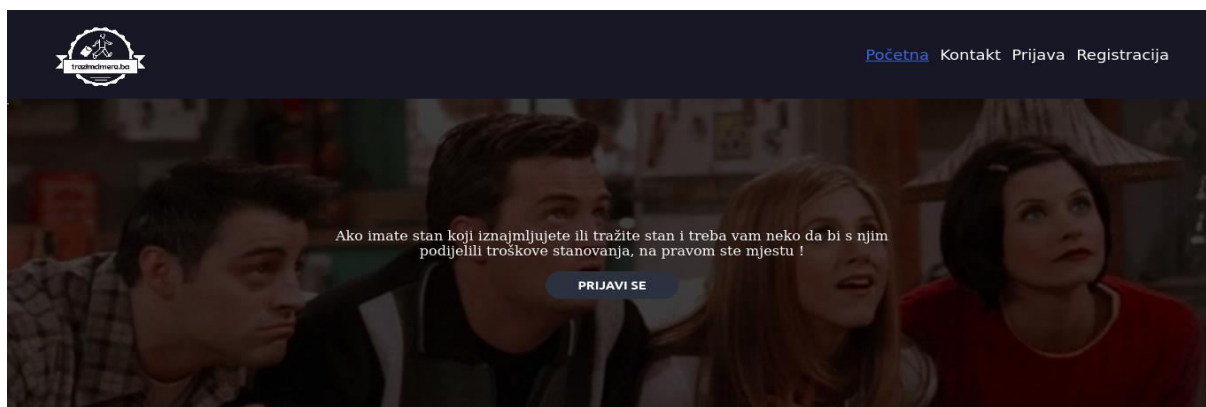
Za fronted smo koristili React. React su kreirali inženjeri Facebook-a kako bi kreirali aplikacije koje imaju bolju skalabilnost i održivost. React je JavaScript biblioteka za kreiranje korisničkog okruženja.

Srž svake React aplikacije čine komponente. Komponente su samostalne i ponovo upotrebljive cijeline koda. One se mogu kreirati na dva načina. Jedan je da se koriste klase a druga funkcije. U našem projektu smo kreirali komponente kao funkcije.

Za slanje zahtijeva prema serveru smo koristili popularni HTTP client axios. Pomoću axios-a veoma lahko šaljemo zahtjeve i dobijamo odgovor od servera, i na jednostavan način upravljamo greškama.

Napravili smo WEB site sa šest stranica. Napravili Layout komponentu koju smo koristili na svakoj stranici. Layout se sastoji od navbar-a i main dijela. U navbaru se nalaze linkovi do drugih stranica i Logo, koji je opet zasebna komponenta. Za redirekciju korisnika sa jedne stranice na drugu smo koristili react-router.

Na početnoj stranici se ispisuju mišljenja korisnika o našim uslugama koji se prvo povlače iz baze i dinamički ispisuju na stranici.



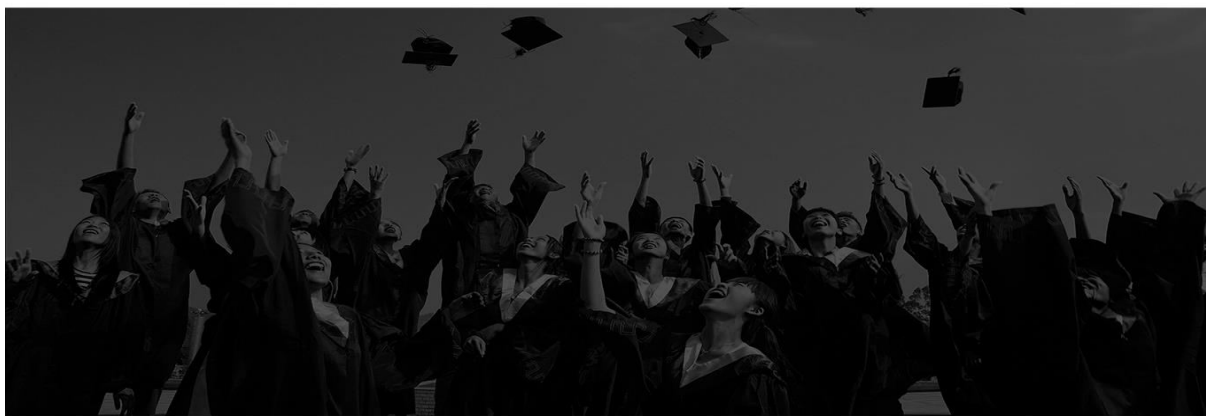
Šta korisnici naših usluga kažu o nama?

Mnogo sam se mučio kako bih našao osobu sa kojom bih dijelio troškove stanovanja, ali mi je ova stranica taj postupak mnogo olakšala. Preko ove stranice našao sam 2 nova dobra prijatelja vjerovatno i za cijeli život i sa njima trenutno živim u blizini Kampusa

-Miralem Pjanić,
student Elektrotehničkog fakulteta u Sarajevu

Stvarno mi se sviđa ovaj način pronalaženja novih cimera! Prije sam stavljao oglase na svom Facebook profilu, ali to nije puno pomagalo. Ovo mjesto je spojilo puno ljudi koji imaju isti problem kao i ja i to mi mnogo olakšava proces pronalaska cimera!

-Emir Kovačević,
student Prirodno-matematičkog fakulteta u Sarajevu



Još se niste registrovali ?

Pridružite se našoj zajednici i upoznajte nove ljude koji bi Vam mogli biti potencijalni cimeri. Objavljujte na našoj stranici oglase i nađite ono što Vam treba !

REGISTRUJ SE

Forma za prijavu je ostala ista, stim da se nakon što kliknemo dugme prijava, šalje post zahtjev na server, pomoću axios-a. Ukoliko je autentikacija uspješna korisnik se pomoću routera šalje na stranicu sa oglasima.



Prijavi se

Korisničko ime (required)

Lozinka (required)

PRIJAVA

Stranica kontakt sadrži i nove ikonice koje su zapravo linkovi do profila na društvenim mrežama.



Kontakt

Ako imate neko pitanje ili problem slobodno nam se javite na jednu od naših društvenih mreža ili ispunite formu i očekujte odgovor u što manjem roku.

Društvene mreže Pošaljite nam poruku



Ime (required)

Vaš Email (required)

Vaša poruka (required)

POŠALJI

Forma za registraciju funkcioniše isto kao i forma za prijavu. Ovaj put je validacija unesenih podataka nešto duža.



Početna Kontakt Prijava [Registracija](#)

Registracija

Pridruživanjem našoj zajednici dobijate mogućnost da objavljujete oglase na našoj stranici!

Ime (required)

Prezime (required)

Datum rođenja (required)

Email (required)

Korisničko ime (required)

Lozinka (required)

Grad (required)

Adresa stanovanja (required)

Sa koliko cimera bi živjeli?

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

Nešto više o vama

Browse...

No file selected.

REGISTRUJ SE

Nakon prijave na WEB sistem, korisnik se preusmjerava na stranicu oglasi. Navbar komponentu zamjenjuje komponenta Toolbar i sada korisnik ima opciju da pregleda sve oglase, doda novi oglas i odjavi se sa sistema.

[Oglasi](#)[Dodaj oglas](#)[Odjavi se](#)

Oglasi

Oglas1

TEL: +387 65 136 937
CIJENA: 450KM plus režije

Oglas2

TEL: +387 65 136 937
CIJENA: 450KM plus režije


Oglas3

TEL: +387 65 136 937
CIJENA: 450KM plus režije

Oglas4

TEL: +387 65 136 937
CIJENA: 450KM plus režije

Kada korisnik klikne dugme za odjavu tada se preusmjerava na početnu stranicu.

[Oglasi](#)[Dodaj oglas](#)[Odjavi se](#)

Objavi svoj oglas!

Ako želite da objavite svoj oglas i ponudu za cimore, molim Vas ispunite obrazac ispod.

Pošaljite nam poruku

Ime i Prezime (required)



Email (required)

Broj telefona (required)

Vaša oglas (required)

OBJAVI

ZAPRATITE NAS NA DRUŠTVENIM MREŽAMA



Apollo - GraphQL

Što se tiče back-end dijela našeg sistema, izabrali smo kombinaciju Node – Apollo – GraphQL.

GraphQL je query-jezik veoma fleksibilne prirode. Njegova struktura je u suštini stablo, gdje čvorovi grananja predstavljaju složenije tipove (koje definiše back-end programer) za koje se *resolveri* definišu eksplicitno, a listovi predstavljaju primitivne tipove (skalare) za koje postoje implicitni *resolveri*.

Fleksibilnost ovog jezika je u slobodi programera na front-end-u da bira čvorove i listove koje želi da dobije kao povratnu informaciju od servera. Sama struktura podataka koju front-end programer želi da dobije zauzvrat piše se u JSON stilu pisanja objekata.

GraphQL omogućava fleksibilno modeliranje podataka iz bilo kakvog izvora, bilo da je to direktno iz baze podataka (SQL ili noSQL), preko udaljenog REST API-ja ili nekog drugog izvora.

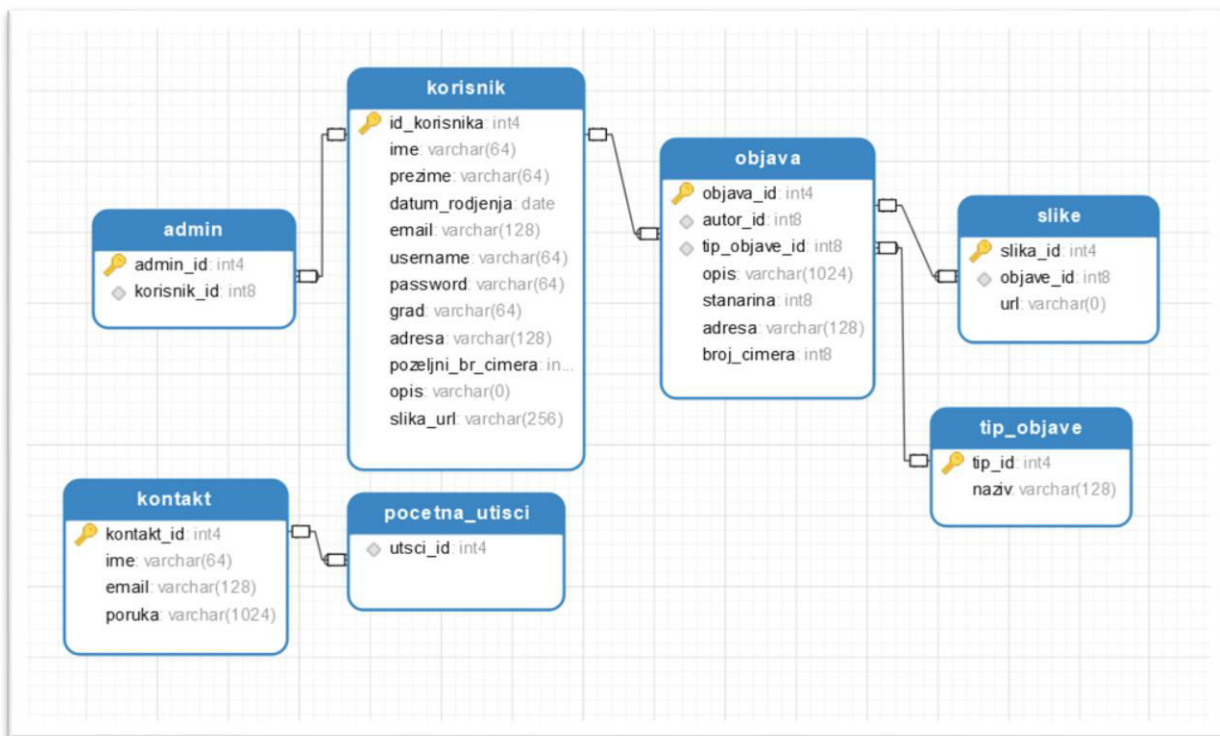
Apollo server je ono što pokreće i veže sve komponente jednog GraphQL API-ja. Instancira se te pokreće na Node.js runtime.

Za **ORM** (Object-relational-mapping) alat smo koristili **TypeORM**.

Korisničke šifre smo enkriptovali paketom **bcrypt**.

Detaljan razvoj baze podataka

U kreiranju detaljnije baze podataka koja će pokriti sve potrebe našeg web sistema ostali smo pri PostgreSQL. Kompletom analizom našeg web sistema upotpunili smo našu privremenu bazu podataka novim tabelama i finalizirali njen izgled. Šema nove baze podataka izgleda ovako :




Slika 15: Finalna šema baze podataka

U svrhu boljeg razumijevanja proći ćemo kroz svaki dio baze podataka i detaljnije ga opisati. Naša baza podataka ima sedam tabela u kojima stoje podaci sa naše web stranice koji su međusobno povezani.


Korisnik

U ovoj tabeli kao što smo već spomenuli, nakon registracije na našoj stranici, korisnik se dodaje u našu bazu podataka. Korisnici se razlikuju i pretražuju po primarnom ključu („Primary key“). U ovu tabelu dodajemo korisničko ime, prezime, datum rođenja, email, grad, adresu stanovanja i njegov opis. Pored toga korisnik ispunjava username i password koji služi za pristup korisničkom nalogu. Korisnik pri registraciji bira sa koliko bi otprilike cimera želio živjeti. Pored ovoga korisnik ima mogućnost da doda sliku profila koju ćemo mi skladištiti putem url-a.

korisnik	
	id_korisnika: int4
	ime: varchar(64)
	prezime: varchar(...)
	datum_rođenja: ...
	email: varchar(12...)
	username: varchar(...)
	password: varchar(...)
	grad: varchar(64)
	adresa: varchar(1...)
	pozelnji_br_cimera...
	opis: varchar(0)
	slika_url: varchar(...)

Objava i tip objave

Ovi dijelovi baze podataka zaduženi su za skladištenje informacija o oglasima na našoj web stranici. Prije svega i ovdje se objave razlikuju po primarnom ključku. Tabela objava sadrži podatke koji je autor postavio oglas, opis njegovog oglasa, kolika bi bila stanarina, adresa i broj cimera. Korisnik pri kreiranju objave mora izabrati koju vrstu oglasa želi da postavi. Za to je zadužena tabela tip objave koja skladišti podatke o tome da li je korisnik odabrao opciju da ima stan i traži samo cimere koji će mu se useliti i olakšati troškove življenja ili opciju u kojoj nema stan i traži cimera/e koji posjeduju stan ili cimera/e da sa njima zajedno neki novi stan iznajme. Ove 2 tabele su povezane na taj način da tabela Objave sadrži primarni ključ tabele Tip objave i povezuje informacije o oglasu. Od tipa objave korisnika, zavisit će i izgled oglasa na našoj web stranici.

objava	
	objava_id: int4
	autor_id: int8
	tip_objave_id: int8
	opis: varchar(102...)
	stanarina: int8
	adresa: varchar(1...)
	broj_cimera: int8

tip_objave	
	tip_id: int4
	naziv: varchar(128)

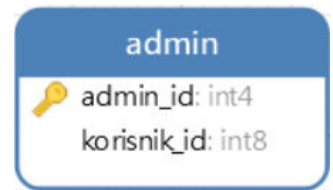
Slike

Ovaj dio baze služi za skladištenje slika sa naše galerije. Zbog lakšeg korištenja i manjeg zauzimanja prostora u našoj bazi i ovdje smo odlučili da skladištimo samo linkove za sve slike koje se koriste na našoj stranici. Ova tabela po primarnom ključu povezana je sa tabelom objave i na taj način povezujemo da u oglasima stoje slike koje korisnik želi objaviti.

slike	
	slika_id: int4
	objave_id: int8
	url: varchar(0)

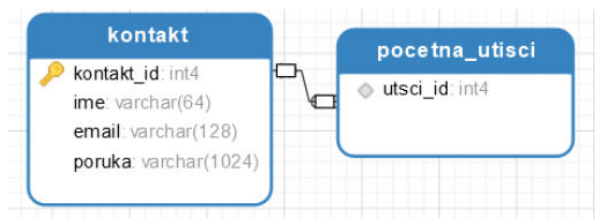
Admin

U ovom dijelu baze stoje informacije ko je od naših korisnika admin na ovoj stranici, tačnije ko ima privilegije da upravlja i uređuje oglase, ko ima pristup svim podacima na stranici i slično. Ova tabela je povezana sa tabelom korisnik i sadrži id korisnika koji imaju privilegije.



Kontakt i utisci

Ove dvije tabele su posljednji fragment naše stranice. Kao što smo već naveli u dijelu stranice „kontakt“ naši korisnici mogu administraciji poslati mail, ako ih nešto zanima ili imaju neko mišljenje/utisak odnosno kritiku za našu stranicu. Na tom dijelu ispunjava se forma koja se šalje na mail. Tabela kontakt sadrži id poruke, ime korisnika, email na koji mu možemo odgovoriti i tekst svoje poruke koju nam želi dostaviti. Ove tabele smo kreirali da bi imali trajni pristup tim podacima, kao rezervu za mail servis. Pored toga tabela kontakt nam olakšava pregled tih upita odnosno utisaka i samim tim imamo veće mogućnosti da naša stranica napreduje. Kada smo već odlučili da skladištimo sve upite odnosno utiske naših korisnika, kreirali smo i tabelu Početna utisci u kojoj stoje utisci korisnika koje smo željeli staviti na početnu stranicu.



Node – Apollo – GraphQL API

Kao što je spomenuto, API smo implementirali sa kombinacijom tehnologija Node – Apollo – GraphQL.

Pored ovih tehnologija za **ORM** smo koristili **TypeORM** koji je jedan od popularnijih izbora za **ORM** u JavaScript/TypeScript okruženju (pretežno korišten uz TypeScript, za koji smo se i mi odlučili).

Izgled/struktura našeg GraphQL API-ja je definisan u GraphQL schema definicijskom jeziku.

Query i **Mutation** tipovi:

```
# Query definitions

type Query {
  users: [User]!
  me: User
  impressions: [Impression]
  posts: [Post]
  postById(id: ID!): Post
  impressionById(id: ID!): Impression
}

# Mutation definitions

type Mutation {
  login(username: String, password: String): UserReturn
  register(user: UserRegisterDetails): String # Message
  addPost(postInfo: PostInfo): Post
}
```

Tipovi podataka:

```
# Type definitions

type User {
  user_id: Int
  username: String
  name: String
  surname: String
  city: String
  address: String
  email: String
  bio: String
  gender: String
  date_of_birth: String
  pr_picture_url: String
  pref_roommate_num: Int
}

type Contact {
  contact_id: ID
  name: String
  email: String
  message: String
}

type Impression {
  impression_id: ID
  contact: Contact
}

type Post {
  post_id: ID
  description: String
  address: String
  rent: Int
  num_of_roommates: Int
  author: User
  post_type: PostType
  images: [Image]
}

type PostType {
  post_type_id: ID
  name: String
  posts: [Post]
}
```

```
type Image {
  image_id: ID
  url: String
  post: Post
}

type UserReturn {
  token: String
  id: ID
  username: String
}
```

Input tipovi:

```
# input definitions

input UserRegisterDetails {
  username: String!
  password: String!
  name: String
  surname: String
  city: String
  address: String
  email: String
  bio: String
  gender: String
  date_of_birth: String
  pr_picture_url: String
  pref_roommate_num: Int
}

input PostInfo {
  author: ID!
  post_type: ID!
  description: String
  address: String
  rent: Int
  num_of_roommates: Int
}
```

Primjer modela urađenog u **TypeORM**:

```
import { Entity, PrimaryGeneratedColumn, Column, ManyToOne, OneToMany } from "typeorm";
import { User } from "../User";
import { PostType } from "../PostType";
import { Image } from "../Image";

@Entity()
export class Post {

    @PrimaryGeneratedColumn()
    post_id: number;

    @Column({ length: 1024, nullable: true })
    description: string;

    @Column({ length: 128, nullable: true })
    address: string;

    @Column()
    rent: number;

    @Column()
    num_of_roommates: number;

    @ManyToOne((type) => User, (author) => author.posts)
    author: User;

    @ManyToOne((type) => PostType, (postType) => postType.posts)
    post_type: PostType;

    @OneToMany((type) => Image, (image) => image.post)
    images: Image[];
}
```

Za definiciju određenih svojstava kolona u tabeli, u ovom slučaju to su *property* klase **Post** koriste se tzv. dekoratori koji se pišu sa @ te nazivom dekoratora. Sa njima možemo da definišemo svojstva tipa:

- Da li je kolona primarni ključ tabele,
- Da li je kolona strani ključ tabele (tj. definišemo relacije tipa **ManyToOne**, **OneToMany**, itd...),

- Osnovna svojstva kao dužinu varchar tipa ili da li ta kolona dozvoljava null vrijednosti i dr.

Vezu back-end servera sa bazom ostvarujemo putem **createConnection()** funkcije kojoj se proslijeđuje konfiguracijski objekat koji definiše osnovne informacije o bazi, te druge stvari kao put do entiteta, migracijskih fajlova, itd.

```
import { createConnection } from "typeorm";

export const connector = () => createConnection({
  type: "postgres",
  host: process.env.DB_HOST,
  port: 5432,
  username: process.env.DB_USERNAME,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_USERNAME,
  synchronize: true,
  logging: false,

  entities: [
    "src/models/*.ts"
  ],

  migrations: [
    "src/typeorm/migration/*.ts"
  ],

  subscribers: [
    "src/typeorm/subscriber/*.ts"
  ],

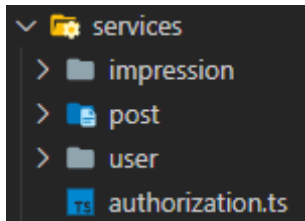
  cli: {
    entitiesDir: "src/typeorm/entity",
    migrationsDir: "src/typeorm/migration",
    subscribersDir: "src/typeorm/subscriber"
  }
});
```

Resolveri su funkcije koje GraphQL API mapira na schemu, te poziva kada schema parsira odgovarajuće nazive. Resolveri treba da vraćaju podatke tačno one strukture koja odgovara GraphQL schemi.

Naši resolveri su sljedeći:

```
module.exports = {
  Query: {
    me: (_, __, { dataSources, user }) =>
      dataSources.middleware.authorize(() => dataSources.userAPI.getUserById(user), {
        user,
      }),
    users: (_, __, { dataSources, user }) =>
      dataSources.middleware.authorize(dataSources.userAPI.getAllUsers, {
        user,
      }),
    impressions: (_, __, { dataSources }) =>
      dataSources.impressionsAPI.getAllImpressions(),
    impressionById: (_, id, { dataSources }) =>
      dataSources.impressionsAPI.getImpressionById(id),
    posts: (_, __, { dataSources, user }) =>
      dataSources.middleware.authorize(
        () => dataSources.postAPI.getAllPosts(),
        {
          user,
        }
      ),
    postById: (_, id, { dataSources, user }) =>
      dataSources.middleware.authorize(
        () => dataSources.postAPI.getPostById(id),
        {
          user,
        }
      ),
  },
  Mutation: {
    login: (_, input, { dataSources }) => dataSources.userAPI.loginUser(input),
    register: (_, input, { dataSources }) =>
      dataSources.userAPI.registerUser(input),
    addPost: (_, input, { dataSources, user }) =>
      dataSources.middleware.authorize(
        () => dataSources.postAPI.createPost(input),
        {
          user,
        }
      ),
  },
};
```

Procesiranje podataka se dešava u servisima, čija struktura je prikazana na sljedećoj slici:



Primjer jednog servisa (za dobavljanje svih utisaka):

```
import "reflect-metadata";
import { connector as createConnection } from "../../connector";
import { Impression } from "../../models/Impression";

// RETRIEVE ALL USERS FROM DB

export const getAllImpressions = async () => {
  let connection;
  try {
    // establish connection
    connection = await createConnection();

    // retrieve users
    const impressions =
      await connection
        .manager
        .getRepository(Impression)
        .createQueryBuilder("impression")
        .leftJoinAndSelect("impression.contact", "contact")
        .getMany();

    // close connection
    await connection.close();

    return impressions;
  } catch (err) {

    // close connection if not yet closed
    if (connection && !connection.closed) {
      await connection.close();
    }

    return err;
  }
};
```

Primjer kverija napisanog u GraphQL Playground-u:

```
1 query {
2   impressions {
3     impression_id
4     contact {
5       name
6       message
7     }
8   }
9 }
```

```
{
  "data": {
    "impressions": [
      {
        "impression_id": "1",
        "contact": {
          "name": "Miralem Pjanić",
          "message": "Mnogo sam se mučio kako bih našao osobu sa kojom bih dijelio troškove stanovanja, ali mi je ova stranica taj postupak mnogo olakšala. Preko ove stranice našao sam 2 nova dobra prijatelja vjerovatno i za cijeli život i sa njima trenutno živim u blizini Kampusa."
        }
      },
      {
        "impression_id": "2",
        "contact": {
          "name": "Emir Kovačević",
          "message": "Stvarno mi se sviđa ovaj način pronalaženja novih cimera! Prije sam stavljao oglase na svom Facebook profilu, ali to nije puno pomagalo. Ovo mjesto je spojilo puno ljudi koji imaju isti problem kao i ja i to mi mnogo olakšava proces pronalaska cimera!"
        }
      }
    ]
  }
}
```

Primjer mutacije napisane u GraphQL Playground-u:

```
1 mutation AddPost($info: PostInfo!) {
2   addPost(postInfo: $info) {
3     post_id
4     description
5     address
6     rent
7     num_of_roommates
8     author {
9       user_id
10      username
11    }
12    post_type {
13      post_type_id
14      name
15    }
16    images {
17      image_id
18      url
19    }
20  }
}
```

```
{
  "data": {
    "addPost": {
      "post_id": "22",
      "description": "U potrazi sam za dva cimera, potpuno namješten trosoban stan sa pogledom na PNE!",
      "address": "Zmaja od bosne, 15",
      "rent": 480,
      "num_of_roommates": 2,
      "author": {
        "user_id": 27,
        "username": "enver"
      },
      "post_type": {
        "post_type_id": "1",
        "name": "Imam stan"
      },
      "images": []
    }
  }
}
```

QUERY VARIABLES HTTP HEADERS (1)

```
1 {
2   "info": {
3     "author": 27,
4     "post_type": 1,
5     "description": "U potrazi sam za dva cimera, potpuno namješten trosoban stan sa pogledom na PNE!",
6     "address": "Zmaja od bosne, 15",
7     "rent": 480,
8     "num_of_roommates": 2
9   }
10 }
```