

Table of contents

Intro	2
Opis	3
Raspoređivanje Zadataka	4
Baza Podataka	5
Supabase Postavke	6
Integracija Supabase-a s TypeScript Tipovima	13
Napomene	17

Intro

Detalji Projekta

Ime Teme: **Music Streaming Service**

Univerzitet	Fakultet	Predmet	Godina	Ciklus
Univerzitet u Zenici	Politehnički fakultet	Web Dizajn	Prva/Druga	Prvi

Članovi Tima
Safet Imamović
Eman Palavra
Dawud Žigo

Opis

Start typing here...

Raspoređivanje Zadataka

Safet Imamović	Eman Palavra	Dawud Žigo
<p>Baza Podataka:</p> <ul style="list-style-type: none"> • PostgreSQL • SupaBase 	<p>Početna stranica sa menijem</p>	<p>Implementacija Formi:</p> <ul style="list-style-type: none"> • Prijava • Registracija • Kontakt
<p>Payment Sistem:</p> <ul style="list-style-type: none"> • Stripe 	<p>Meni uključuje opcije</p> <ul style="list-style-type: none"> • o nama • prijava • kontakt 	<p>Integracija Google Maps na Kontakt stranicu</p>
<p>Implementacija uloga:</p> <ul style="list-style-type: none"> • Admin • Guest 	<p>Main layout dizajn</p>	<p>Query pjesama i prikaz liste</p>
<p>Implementacija Supabase Typescript tipova</p>		

Baza Podataka

Pregled Supabase-a

1. Šta je Supabase?

- Supabase je open-source alternativa za Firebase.
- Koristi PostgreSQL za svoju bazu podataka.
- Uključuje funkcije poput skladištenja, autentifikacije i real-time mogućnosti.
- Podržava slanje emailova za resetovanje lozinke sa prilagodljivim email šablonima.

Supabase Postavke

Kreiranje Računa

1. Kreiranje Supabase računa putem GitHub-a

- Registracija na Supabase koristeći GitHub račun.

Postavke Projekta

1. Postavke projekta

- Naziv projekta: `MusicStreamingService`
- Region: Frankfurt, EU (lokacija servera)

2. Lozinka projekta

- Generisana je lozinka za projekat i kopirana je u `password.txt` u korijenu projekta.
- Dodano je `password.txt` u `.gitignore` kako bi se osiguralo da se ne pošalje u repozitorij.

API Ključevi

1. Postavke API ključeva

- U postavkama Supabase -> API.
- Prikupljeno je sljedeće:
 - `NEXT_PUBLIC_SUPABASE_URL`
 - `NEXT_PUBLIC_SUPABASE_ANON_KEY`
 - `SUPABASE_SERVICE_ROLE_KEY`
- Dodano je ovo u `.env.local` datoteku.

- Osigurajte da je `.env.local` dodan u `.gitignore`.



Napomena **Objašnjenje API ključeva:**

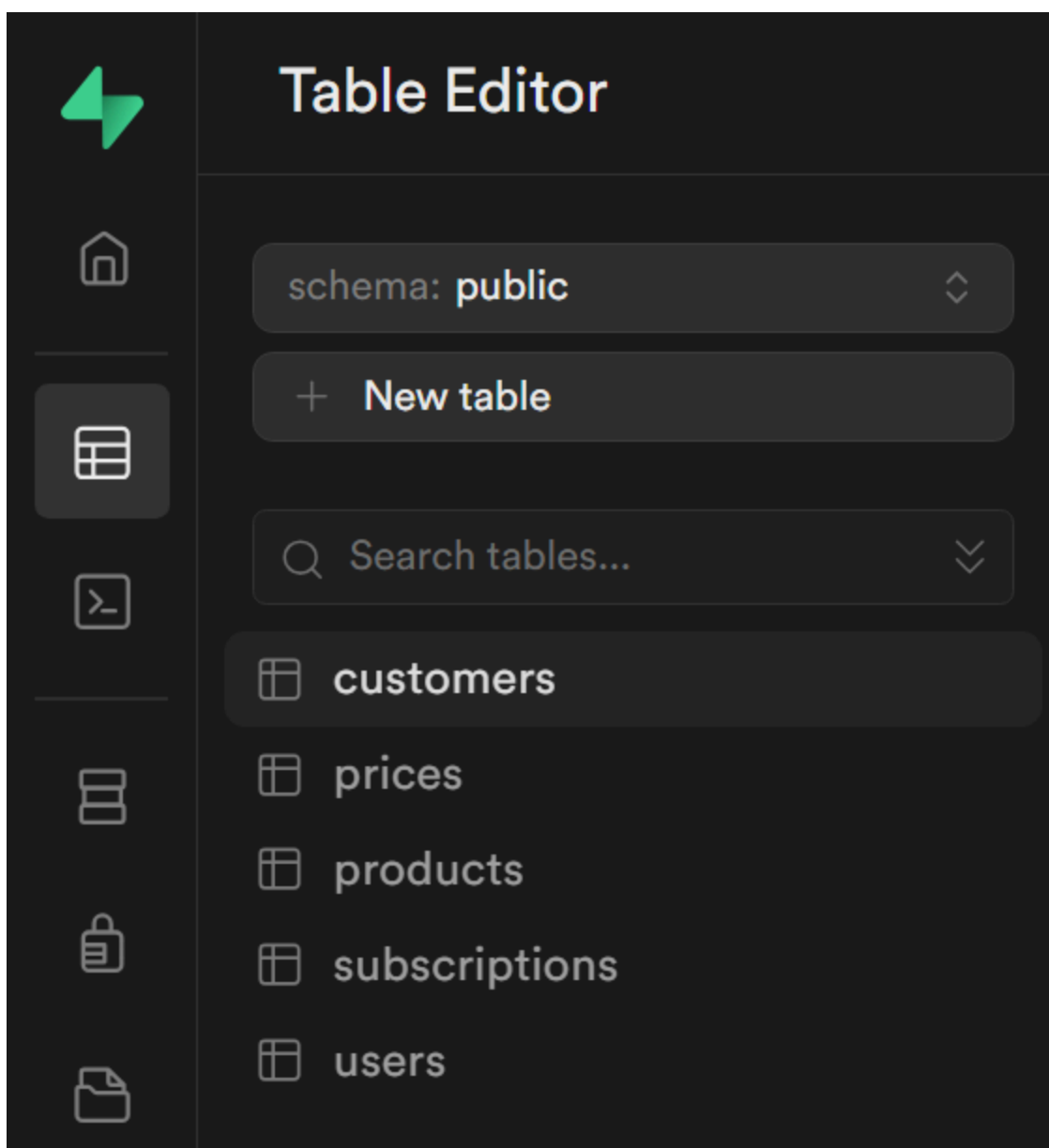
- **NEXT_PUBLIC_SUPABASE_URL:** URL vaše Supabase instance.
- **NEXT_PUBLIC_SUPABASE_ANON_KEY:** Anonimni ključ za javni pristup.
- **SUPABASE_SERVICE_ROLE_KEY:** Ključ sa povišenim ovlaštenjima za server-side operacije.

Šema Baze Podataka

Brzi Start Šabloni

1. Stripe Pretplate Šablon

// dodati šablon link



1 tables created

⚠ Napomena Objašnjenje SQL Koda:

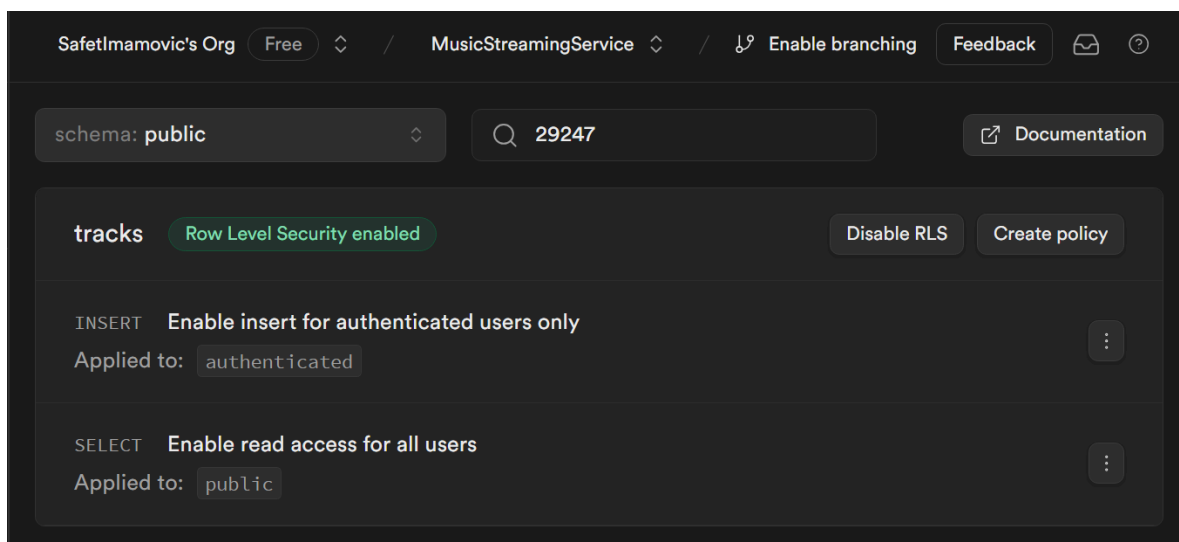
- **Users Tabela:** Čuva podatke o korisnicima sa sigurnošću na nivou reda kako bi se osiguralo da korisnici mogu pregledati i ažurirati samo svoje podatke.
- **Trigger za nove korisnike:** Automatski kreira unos u `users` tabeli kada se novi korisnik registruje.
- **Customers Tabela:** Povezuje ID korisnika sa Stripe ID-om korisnika, nije

dostupno korisnicima.

- **Products Tabela:** Čuva detalje o proizvodima, dostupno svima u read-only načinu.
- **Prices Tabela:** Čuva detalje o cijenama, sinhronizovano sa Stripe-om, dostupno svima u read-only načinu.
- **Subscriptions Tabela:** Čuva detalje o pretplatama, osiguravajući da korisnici mogu pregledati samo svoje pretplate.
- **Realtime Pretplate:** Omogućava real-time praćenje na javnim tabelama `products` i `prices`.

Dodatne Tabele i Politike

1. Tracks Tabela



2 table policies

```
create table public.tracks (  
  id bigint generated by default as identity,  
  created_at timestamp with time zone not null default now(),  
  title text null,  
  track_path text null,  
  image_path text null,
```

```

gif_path text null,
author
text null,
user_id uuid null,
constraint tracks_pkey primary key (id),
constraint tracks_user_id_fkey foreign key (user_id) references users
(id) on delete cascade
) tablespace pg_default;

```

2. Politike za Tracks Tabelu

```

alter policy "Omogući čitanje svim korisnicima" on "public"."tracks" to
public using (true);
create policy "Omogući unos samo autentificiranim korisnicima" on
"public"."tracks" as PERMISSIVE for INSERT to authenticated with check
(true);

```

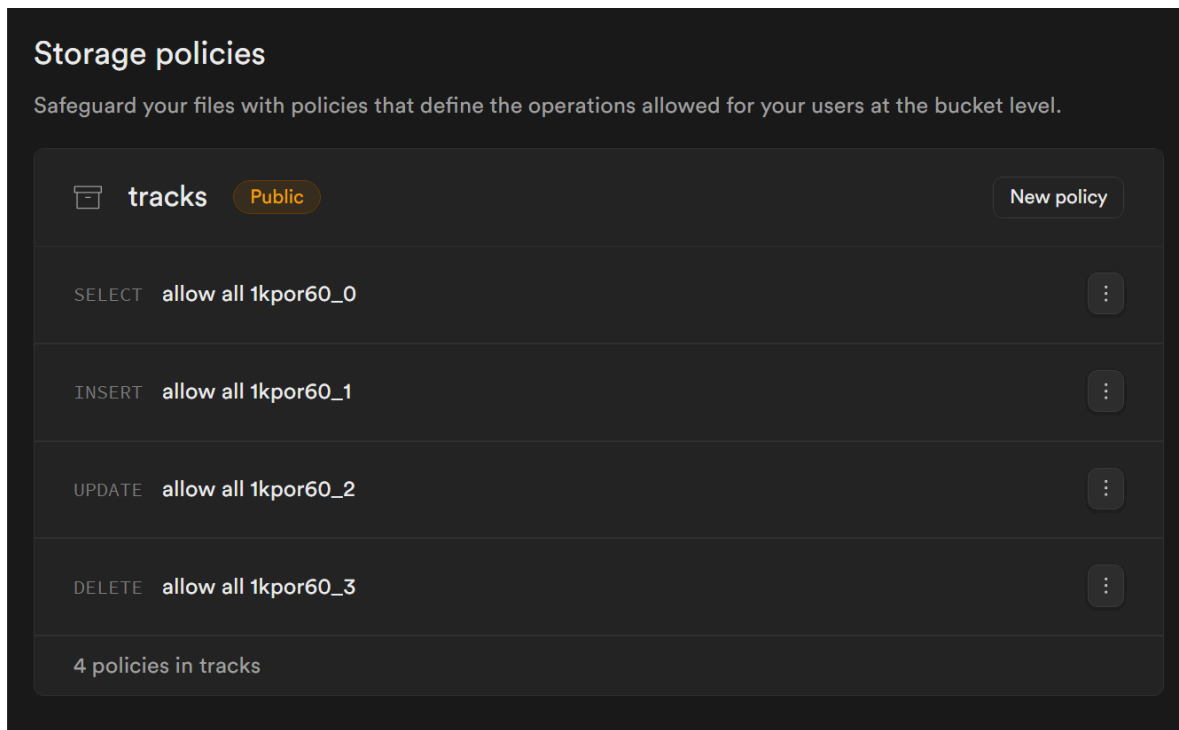
3. Tabela Liked Tracks

```

create table public.liked_tracks (
  user_id uuid not null,
  created_at timestamp with time zone not null default now(),
  track_id bigint not null,
  constraint liked_tracks_pkey primary key (user_id, track_id),
  constraint liked_tracks_track_id_fkey foreign key (track_id)
references tracks (id) on delete cascade,
  constraint liked_tracks_user_id_fkey foreign key (user_id) references
users (id) on delete cascade
) tablespace pg_default;

```

4. Politike Tabele Omiljenih Pjesama



3 bucket policies

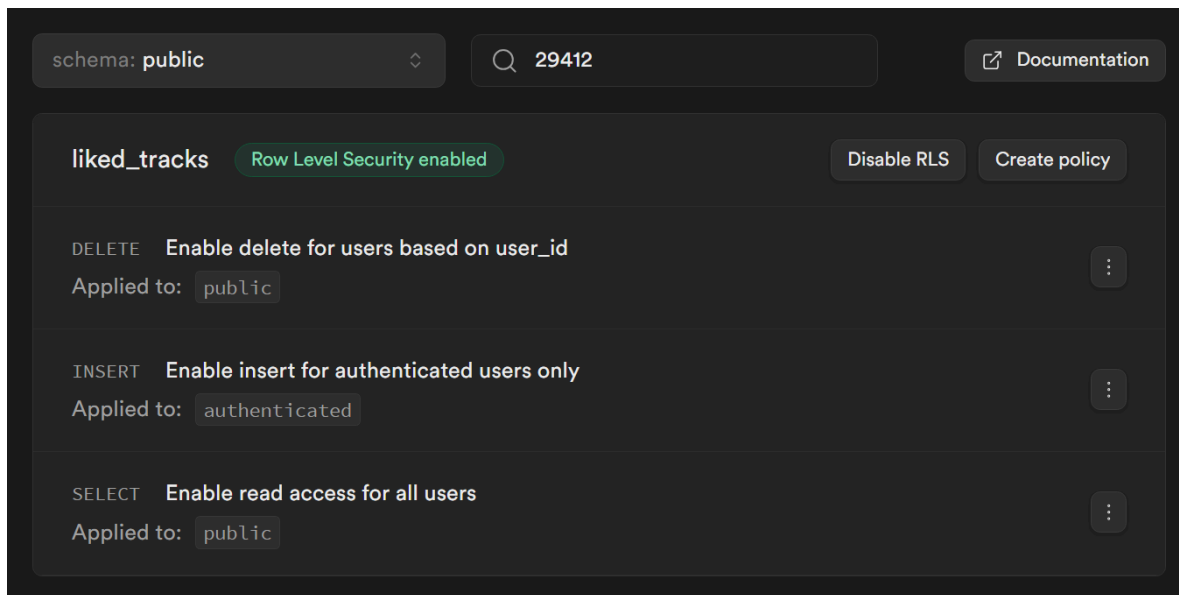
```
alter policy "Dozvoli brisanje korisnicima na osnovu user_id" na
"public"."liked_tracks" za javno korišćenje (((SELECT auth.uid() AS uid)
= user_id));
```

5. Postavljanje Skladišta

6. Kreiranje Kanti

- **Kanta za Pjesme:** Ograničena na audio/mpeg.
- **Kante za Slike i GIF-ove:** Nema ograničenja tipova.

7. Politike Skladišta za Kantu za Pjesme



Liked tracks rls policies

```
CREATE POLICY "Dozvoli sve 1kpor60 0" ON storage.objects FOR SELECT TO public KORIŠTENJEM (bucket_id = 'pjesme');
CREATE POLICY "Dozvoli sve 1kpor60 1" ON storage.objects FOR INSERT TO public SA PROVJEROM (bucket_id = 'pjesme');
CREATE POLICY "Dozvoli sve 1kpor60 2" ON storage.objects FOR UPDATE TO public KORIŠTENJEM (bucket_id = 'pjesme');
CREATE POLICY "Dozvoli sve 1kpor60 3" ON storage.objects FOR DELETE TO public KORIŠTENJEM (bucket_id = 'pjesme');
```

Integracija Supabase-a s TypeScript Tipovima

Korak 1: Instalacija Supabase CLI-a

Za početak, instaliran je Supabase CLI alat kao razvojna zavisnost. Ovaj alat pomaže u generisanju TypeScript tipova iz PostgreSQL šeme.

```
npm i supabase@">=1.8.1" --save-dev
```

Objašnjenje

- `npm i supabase@">=1.8.1" --save-dev`: Instalira se Supabase CLI paket s verzijom većom ili jednakom 1.8.1 kao razvojna zavisnost. `--save-dev` flag osigurava da se koristi samo u razvoju.

Korak 2: Prijava u Supabase

Potrebno je prijaviti se u Supabase kako bi se generisali tipovi. Ovo zahtijeva token za pristup, koji se može generisati sa stranice postavki Supabase projekta.

```
npx supabase login
```

Objašnjenje

- `npx supabase login`: Poziva se unos Supabase tokena za pristup, autentifikujući sesiju i omogućavajući pristup projektu.

Korak 3: Generisanje TypeScript tipova

Sa prijavljenim CLI-jem, generisani su TypeScript tipovi na osnovu PostgreSQL šeme.

```
npx supabase gen types typescript --project-id "$PROJECT_REF" --schema
```

```
public > types_db.ts
```

Objašnjenje

- `npx supabase gen types typescript`: Pokreće se Supabase CLI za generisanje TypeScript tipova.
- `--project-id "$PROJECT_REF"`: Specifikuje ID Supabase projekta.
- `--schema public`: Ukazuje na šemu za koju se generišu tipovi, u ovom slučaju `public` šema.
- `> types_db.ts`: Preusmjerava generisane tipove u fajl pod imenom `types_db.ts`.

Nakon ovog koraka, `types_db.ts` fajl će sadržavati TypeScript tipove za tabele i kolone iz PostgreSQL šeme.

Link do `types_db.ts`

(https://github.com/SafetImamovic/MusicStreamingService/blob/168a6be7bd7ed5989d6ef8bbbee168ac2756eb42/types_db.ts)

Korak 4: Postavljanje Supabase i User Context Providera

4.1: Kreiranje direktorija i fajla

Kreiran je direktorij pod nazivom `providers` i fajl pod nazivom `SupabaseProvider.tsx` u Next.js projektu.

```
mkdir providers
touch providers/SupabaseProvider.tsx
```

4.2: Instalacija Supabase Auth Helpers

Instalirani su potrebni paketi za upravljanje Supabase autentifikacijom u Next.js i React.

```
npm install @supabase/auth-helpers-nextjs @supabase/auth-helpers-react
```

4.3: Implementacija SupabaseProvider

Sljedeći kod je dodan u `SupabaseProvider.tsx` kako bi se kreirao Supabase klijent i obezbijedio kontekst sesije.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/providers/SupabaseProvider.tsx>)

4.4: Omotavanje glavnog layouta sa SupabaseProvider

`layout.tsx` je izmijenjen kako bi uključio `SupabaseProvider`.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/app/layout.tsx>)

Korak 5: Postavljanje User Context-a

5.1: Kreiranje direktorija i fajlova za hookove

Kreiran je direktorij `hooks` i fajl `useUser.tsx`.

```
mkdir hooks
touch hooks/useUser.tsx
```

5.2: Definisanje tipova korisnika

Novi fajl `types.ts` je dodan u korijenu projekta kako bi se definisali TypeScript tipovi za detalje korisnika, proizvode, cijene i pretplate.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/types.ts>)

5.3: Implementacija useUser hook-a

Sljedeći kod je dodan u `useUser.tsx` kako bi se upravljalo korisničkim kontekstom i detaljima pretplate.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/hooks/useUser.tsx>)

5.4: Kreiranje UserProvider-a

Kreiran je fajl `UserProvider.tsx` u `providers` direktoriju i dodan sljedeći kod.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/providers/UserProvider.tsx>)

5.5: Omotavanje layouta sa UserProvider-om

layout.tsx je izmijenjen kako bi uključio UserProvider.

(<https://github.com/SafetImamovic/MusicStreamingService/blob/4ac3ec20ea1d576d49a7e1b7a674a972ba0ab0f8/app/layout.tsx>)

Napomene

Start typing here...