Worth mentioning

3rd party auth dashboard

https://supabase.com/dashboard/project/vnhvmfczkbshtcuyfkeu/auth/users

Login : vpwasupabase@gmail.com

Heslo: Vpwa123!

Tam sa budu zobrazovat zaregistrovany pouzivatelia, ja ich ukladam aj u seba lebo, pretoze je to tak zle ze sa neda vytiahnut querinou iny user ako current co je v sessione, cize ten auth vyuzivam len na prihlasovacie tokeny. Ak budete mazat pouzivatela tak premazte aj DB, nech sa ten user vymaze aj v nasej tabulke.

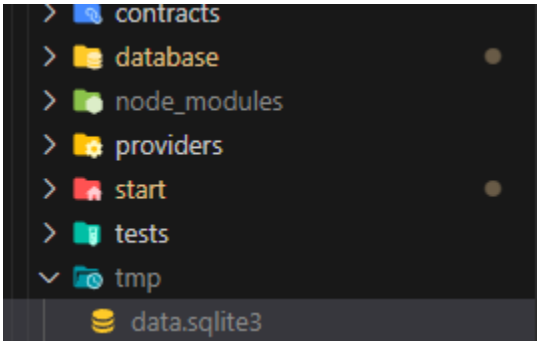
Po registracii je nutne potvrdit email confirmation, aby ste sa mohli lognut !

ak to niekde najdete v nastaveniach ze vypnut tak to vypnite, ja som to tam nenasiel…

Conn na DB

Netreba ziadne credentials menit len si zbehnite migracie a vsetko sa bude ukladat v tom subore, ked budete chciet premazat db tak nemazte subor len cely jeho content…

Lokacia: **\VPWA_project\backend\slack\tmp**



Websockety

Lokacia na backende - **\backend\slack\app\WebSocket\websocket.ts**

ked sa s frontendu posle funkcia **socket.emit**, tak to dorazi sem kde sa vyoknava logika

Userov udrzujem v mape vo formate… [{nickname: jeho_socket}]

Channeli vo formate [{channel_id: [sockety uzivatelov z channela]}]

```
const users = new Map();
const channels = new Map()
```

Pri prihlásení sa používateľov socket vloží do users, pri vytvoreni/joinuti channel sa vlozi do channels do daneho channel.

Na frontende je v mounte spraveny connection a listneery, cize ked emitnem zo subora spomenuteho vyssie co je na backende tak to dojde sem, napr. Aj s parametrom

```
mounted() {

  const user_id = supabase.auth.session().user.id
  const user_name = supabase.auth.session().user.user_metadata.nickname

  initializeSocket(user_id, user_name);

  this.socket = getSocket();

  this.socket.on('connect', () => {
    console.log('Connected to WebSocket server');
  });

  this.socket.on('create-channel', (channel) => {
    this.channels = [channel, ...this.channels];
  });

  this.socket.on('join-channel', (channel) => {
    this.channels = [channel, ...this.channels];
  });

  this.socket.on('leave-channel', (channel_id) => {
    this.channels = this.channels.filter((channel:any) => channel_id != channel.id);
    this.closeExitModal();
    this.$router.push({ path: '/channels' });
  });
```

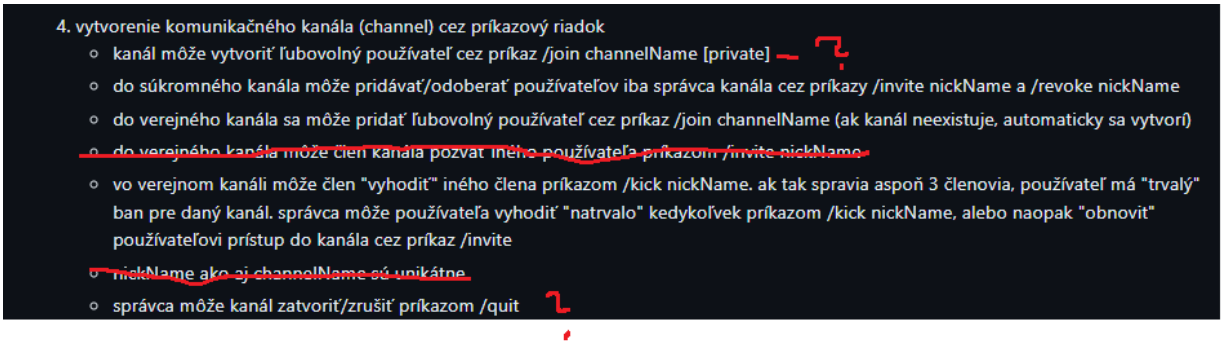
Hotové časti

Hotove časti

1. registrácia, prihlásenie a odhlásenie používateľa
   - používateľ má meno a priezvisko, nickName a email
2. používateľ vidí zoznam kanálov, v ktorých je členom
   - pri opustení kanála, alebo trvalom vyhodení z kanála je daný kanál odobratý zo zoznamu
   - pri pozvánke do kanála je daný kanál zvýraznený a topovaný
   - v zozname môže cez používateľské rozhranie kanál vytvoriť, opustiť, a ak je správcom aj zrušiť
   - dva typy kanálov - súkromný (private channel) a verejný kanál (public channel)
   - správcom kanála je používateľ, ktorý kanál vytvoril
   - ak nie je kanál aktívny (nie je pridaná nová správa) viac ako 30 dní, kanál prestáva existovať (následne je možné použiť channelName kanála pre "nový" kanál)

5. používateľ môže zrušiť svoje členstvo v kanáli príkazom /cancel, ak tak spraví správca kanála, kanál zaniká

1, 2, a 5. Sú hotové (v jednotke nefunguje este neaktivny kanal)



Zo 4.ky su spravene tie co su vyskrtnute, tam kde je otaznik je logika spravena len to nie je nalinkovane (vytvorenie kanala je len nie cez join channel… quit channel je len cez prikaz – cancel a nie /quit)

Ps: vacsina veci sa robi cez sockety, cize dobre zvazte ci pouzijete socket alebo vam postaci aj http request, ja som tak 75 % requestov prerabal na sockety….

V projekte by ste mali najst uz takmer vsetko co sa bude dat znovu pouzit ako vytvaranie a mazanie z many to many modelov, pracu s modelmi, pracu so socketmi aj requestami, je tam spraveny auth guard takze to netreba riesit tiez

Kod som este neupratoval a nestrukturoval do podsuborov, kvoli prehladnosti

TS tam vyhodi kopu errorov, lebo som bol lenivy tam robit typy a handlovat to, takze ked tak este to dorobim