

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Snack - aplikácia na textovú komunikáciu v štýle IRC (Slack)

Semestrálny projekt

Marek Dieška, 127133

Viktória Latičová, 127219

Cvičiaci: Ing. Eduard Kuric, PhD.

Predmet: Vývoj progresívnych webových aplikácií

2025/2026

Obsah

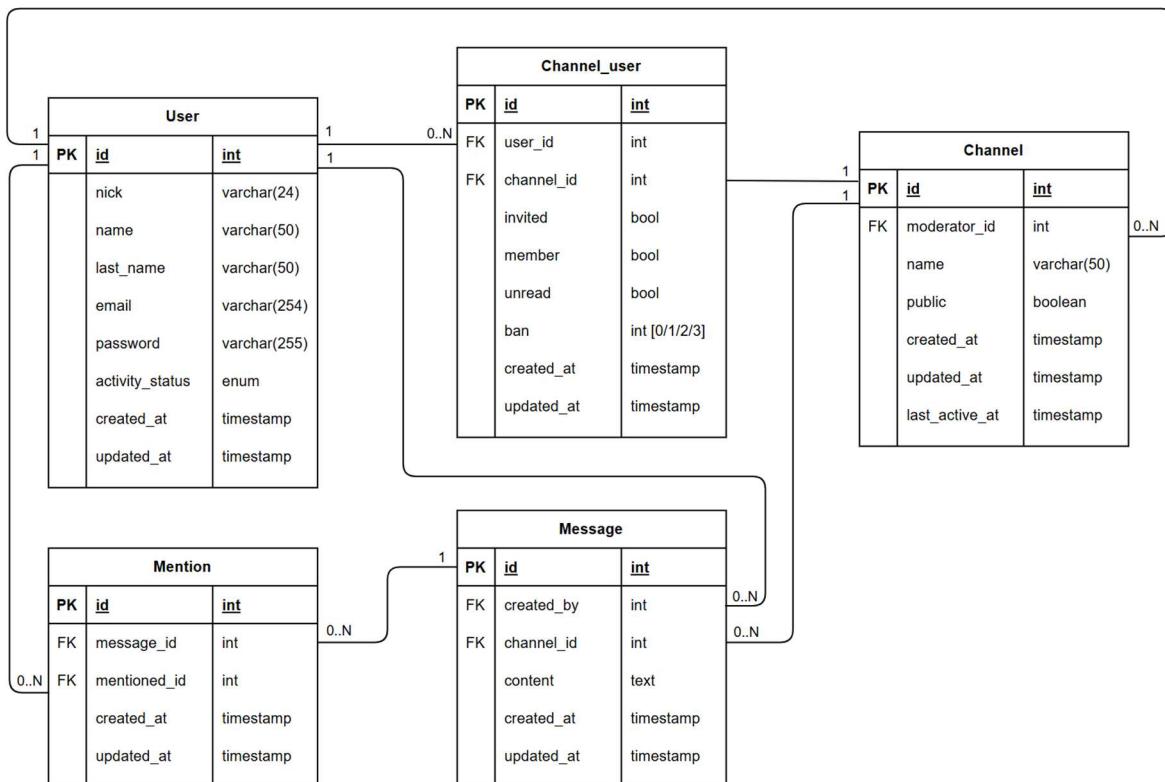
Zadanie	3
Diagram dátového modelu	3
Použité technológie	4
Diagramy architektúry aplikácie	4
Návrhové rozhodnutia	5
AdonisJS 6.....	5
Luxon	5
Node-Cron.....	5
Socket.io.....	5
Socket.io-client.....	5
Better-sqlite3	5
Pinia.....	5
Vue Router	5
Jednotlivé obrazovky.....	6
Login a registrácia	6
Komunikácia v kanáli.....	7
Živé sledovanie prichádzajúcich správ.....	7
Vytváranie nových kanálov.....	7
Nastavenie stavu používateľa.....	8
Vyhodenie používateľa	8
Aplikácia v mobilnej verzii	8
Používateľská príručka	9
Link na github	9
Záver.....	9

Zadanie

Cieľom zadania bolo vytvoriť progresívnu webovú aplikáciu na textovú komunikáciu v štýle IRC. Aplikácia mala obsahovať interaktívnu real-time komunikáciu v channeloch, kde sa okrem iného môžu používatelia navzájom pozývať, odstraňovať. písat' si, označovať sa, sledovať naživo kto čo píše ešte pred tým, ako správu odošle. Na tento projekt mali byť využité frameworky AdonisJS a Quasar. Živá komunikácia mala byť dosiahnutá pomocou websocketov.

Diagram dátového modelu

Diagram od prvej fázy odovzdania prešiel viacerými zmenami. Tabuľku notifications sme z návrhu úplne vypustili. Prišli sme na to, že notifikácie posielané cez sockety nie je potrebné do databázy zapisovať. Čo je potrebné ale zapisovať sú označenia @nick, aby sa mohli tieto spr, a preto sme pridali tabuľku mentions. Do tabuľky channel_users sme pridali riadky invited – či má užívateľ aktuálne pozvánku do daného channelu, riadok member a riadok ban – počet príkazov /kick smerovaných na daného používateľa v channeli. Ban = 3 označuje trvalý ban do channelu. Poslednou zmenou bolo pridanie riadku last_active_at do tabuľky channels, ktorý je potrebný pri určení, či channel neboli používaný viac ako 30 dní.

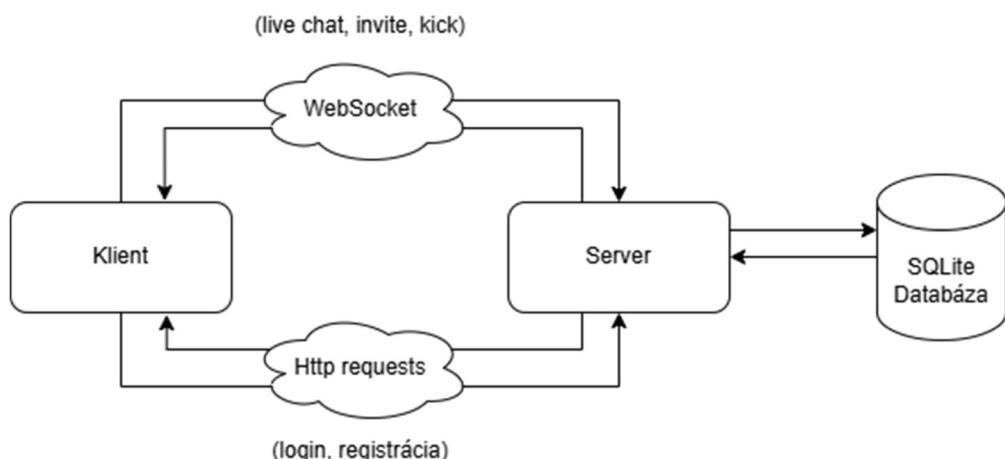


Použité technológie

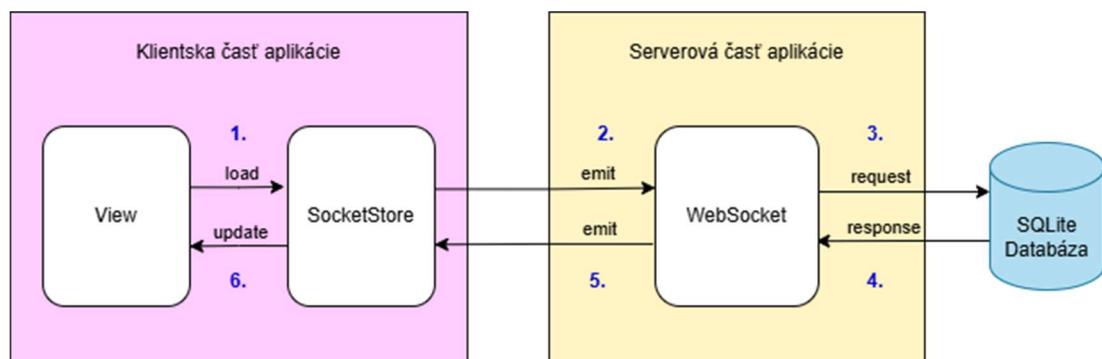
- **Dátový model** bol modelovaný cez online nástroj Visual Paradigm.
- Databáza bola implementovaná relačnou databázou **SQLite**
- **Backend** aplikácie je implementovaný vo frameworku **AdonisJS 6** postavenom na Node.js 22.14 → autentifikácia, REST API, práca s databázou (lucid ORM)
- Na komunikáciu v reálnom čase sú použité **WebSockety** (Socket.IO)
- **Frontend** je vytvorený pomocou frameworku **Quasar** postaveného na Vue 3
- Projekt je písaný v jazyku Typescript pomocou **Composition API**
- Celá aplikácia je dokopy vybudovaná pomocou nástroja **Vite (npm)**

Diagramy architektúry aplikácie

Na všetky úlohy, ktoré majú prebiehať v reálnom čase klient využíva komunikáciu cez web socket a následný emit pomocou serveru. Http požiadavky využíva klient pri registrácii, prihlásení a vytvorení nových kanálov, kde nie je potrebná aktualizácia u ostatných používateľov. Prístup k databáze klient nemá priamo, zabezpečuje to server a výsledky vráti späť klientom.



Klient nikdy neodosiela správy priamo ostatným klientom, ale všetka komunikácia prebieha cez server, ktorý zabezpečuje validáciu, rozposielanie správ do kanálov a ak je potrebné, aj ukladanie a čítanie dát z databázy.



Návrhové rozhodnutia

AdonisJS 6

Kvôli komplikáciám s inštalovaním AdonisJS 5 sme sa rozhodli použiť framework adonisJS 6. Kvôli tomuto použitie predpripraveného ws na githube nebolo možné a pridávali sme si ho sami.

Luxon

Použitý na prácu s časom a dátumami. V aplikácii slúži na formátovanie časových značiek správ a zobrazovanie správ relatívneho času (napr. pred 5 minútami)

Node-Cron

Knižnica na plánovanie úloh, použitá na vytvorenie časovej úlohy na odstraňovanie kanálov, ktoré sú neaktívne dlhšie ako 30 dní.

Socket.io

Knižnica použitá na real-time komunikáciu cez WebSockets. Použitá na okamžité doručovanie správ, notifikácie o pripojení/odpojení používateľov a indikátor písania.

Socket.io-client

Klientská časť Socket.IO. Zabezpečuje pripojenie frontendu k WebSocket serveru a prijímanie/odosielanie real-time udalostí.

Better-sqlite3

SQLite databáza použitá pre lokálny vývoj a testovanie bez potreby externého datbázového servera.

Pinia

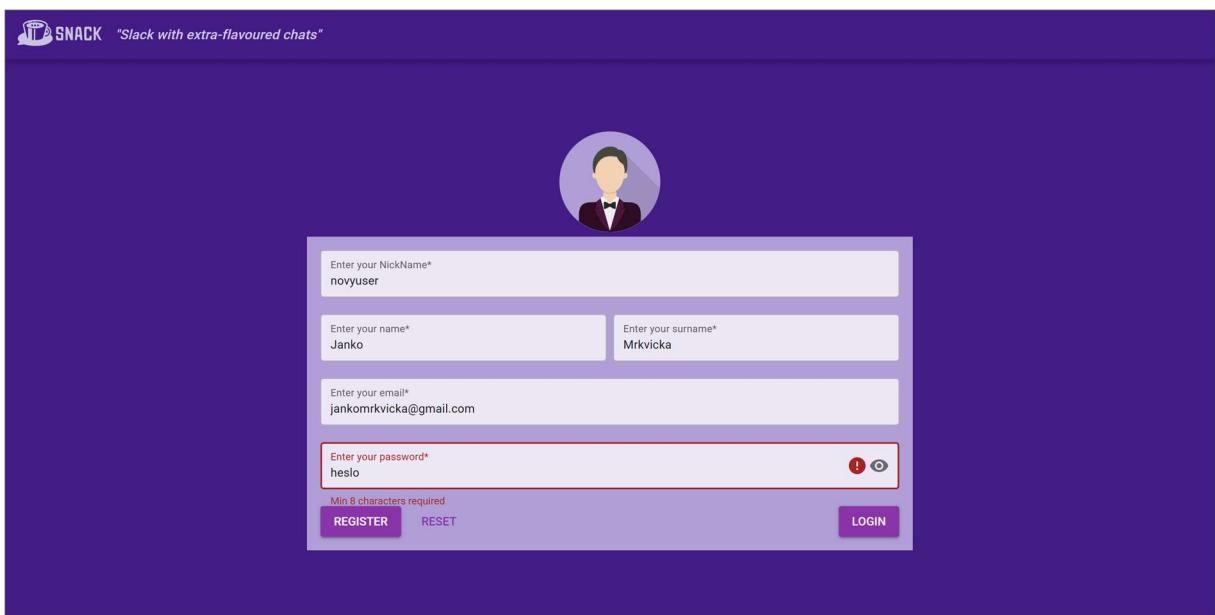
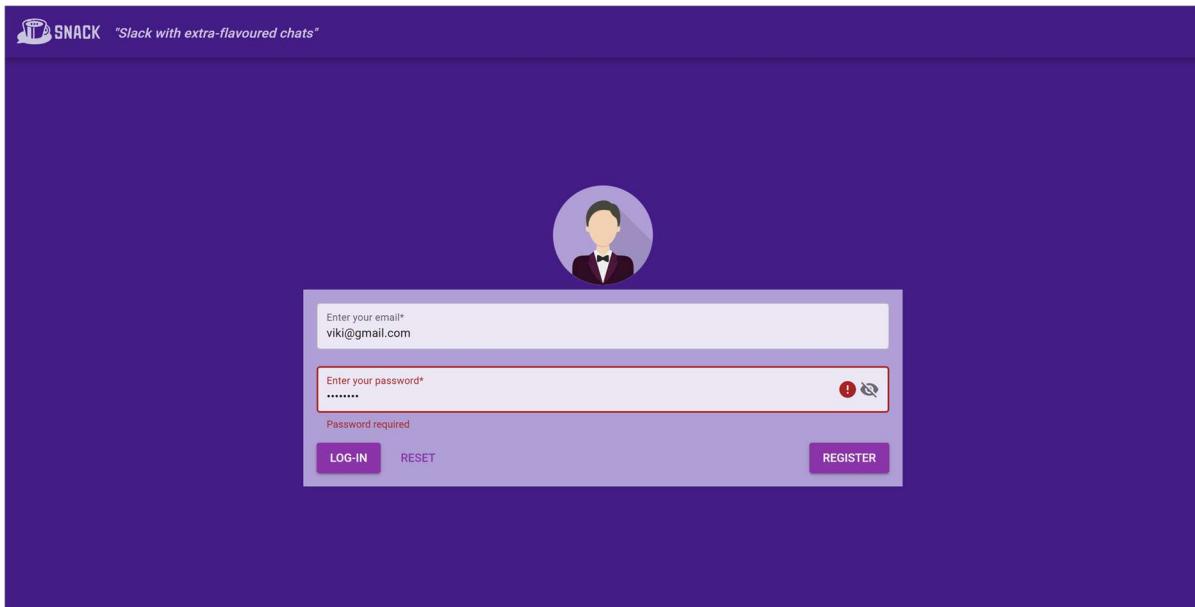
State manažment pre Vue 3. Uchováva globálny stav aplikácie ako informácie o prihlásenom používateľovi, zoznam kanálov a aktívne konverzácie.

Vue Router

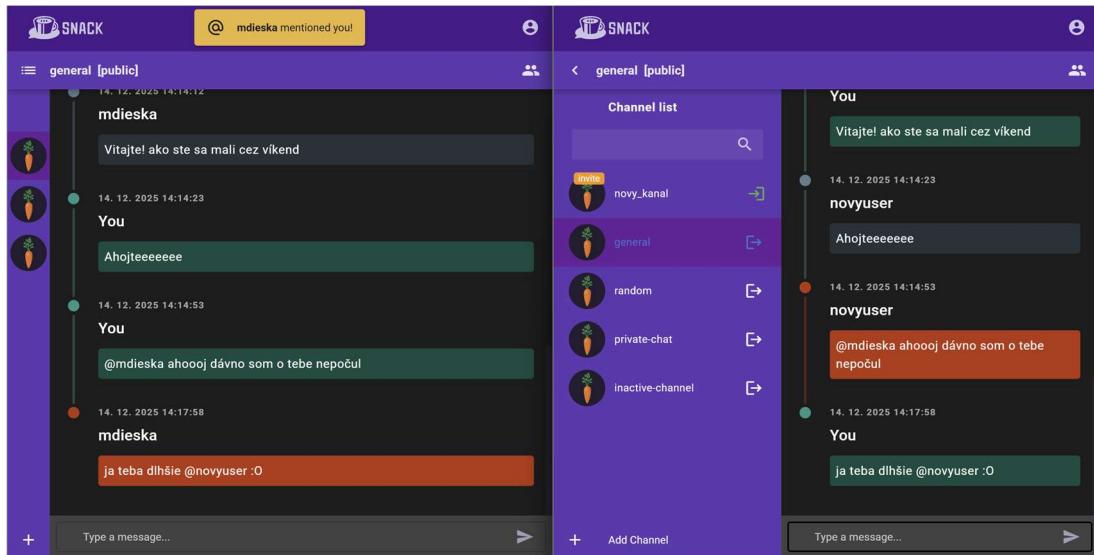
Zabezpečuje navigáciu medzi stránkami aplikácie

Jednotlivé obrazovky

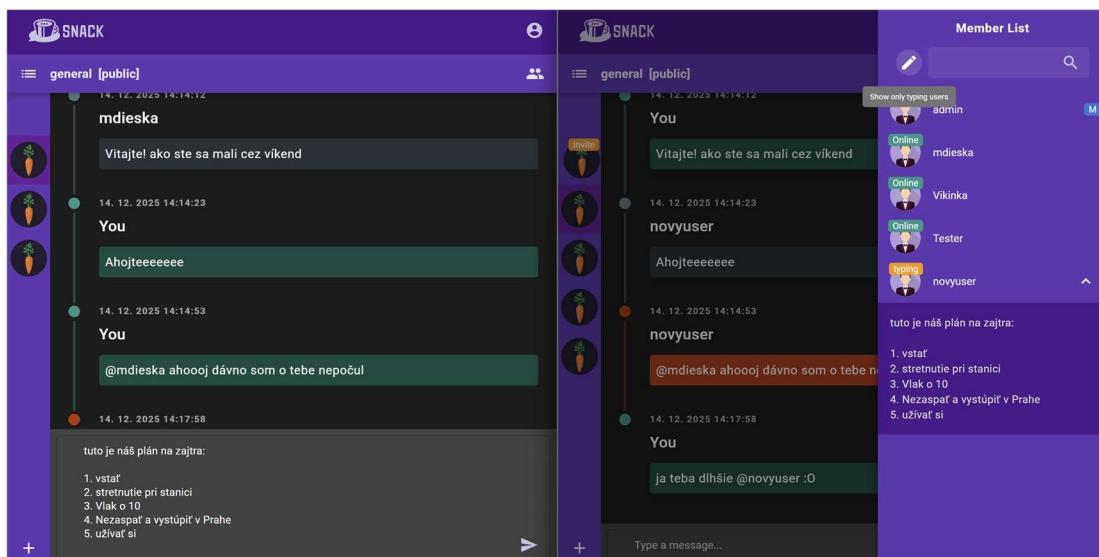
Login a registrácia



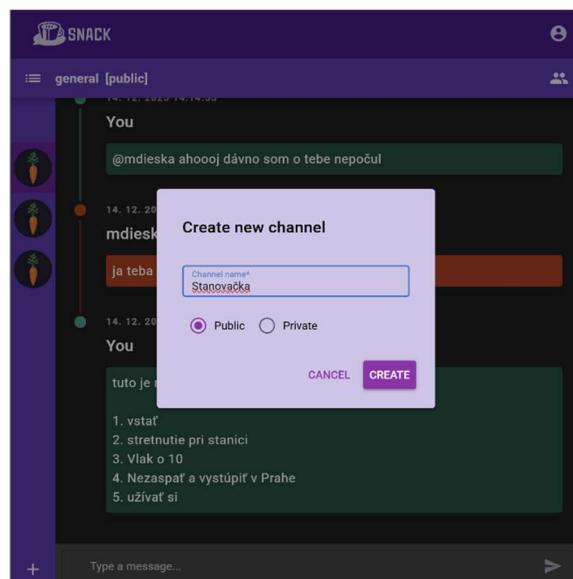
Komunikácia v kanáli



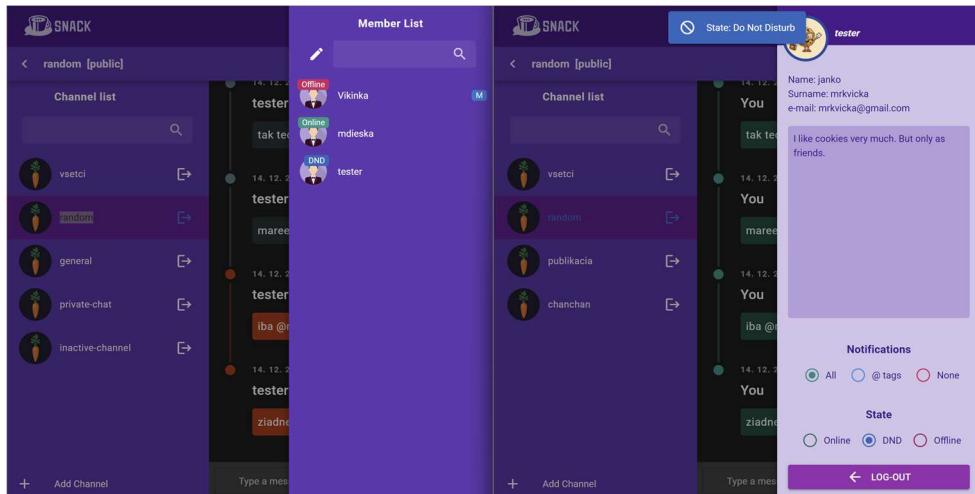
Živé sledovanie prichádzajúcich správ



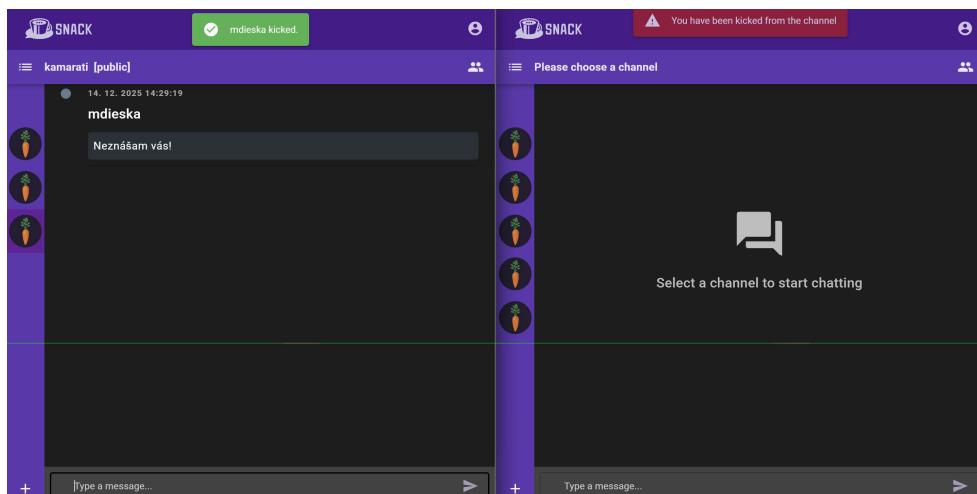
Vytváranie nových kanálov



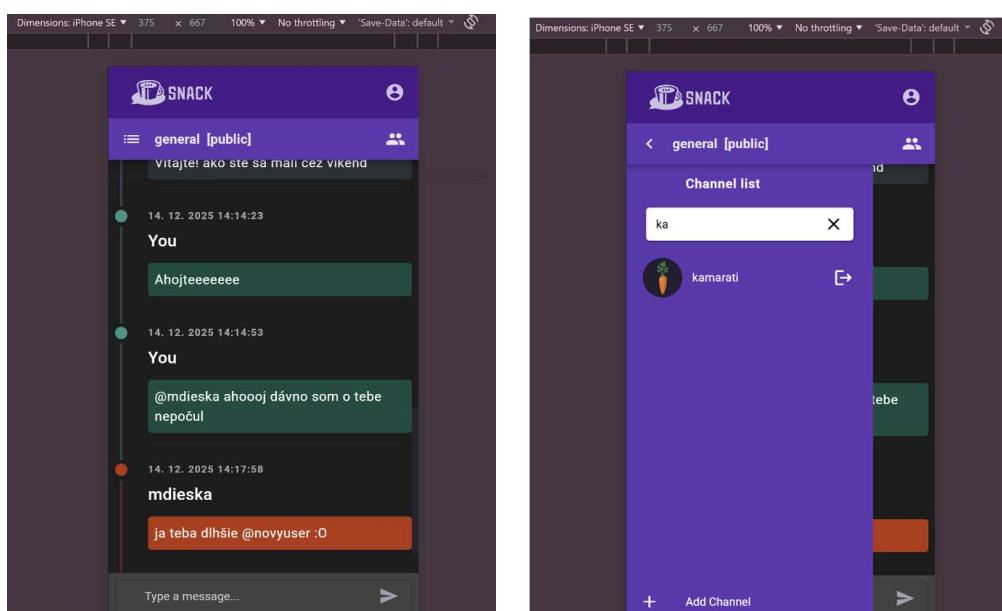
Nastavenie stavu používateľa



Vyhodenie používateľa



Aplikácia v mobilnej verzii



Používateľská príručka

1. Stiahnuť a nainštalovať Node.js (v18+)
2. Pullnúť z GitHubu projekt do svojho zariadenia
3. V príkazovom riadku: cd Snack-server, **npm install**
4. Skopírovať **.env.example** do **.env**
5. Vygenerovať APP_KEY príkazom **node ace generate:key** a skopírovať ho do **.env**
6. Spustiť migrácie a seedy: **node ace migration:fresh --seed**
7. Spustiť backend server: **npm run dev**
8. Otvoriť nový terminál, cd **Snack-client**, **npm install**
9. Spustiť frontend: **npm run dev**

Link na github

https://github.com/Vikii712/VPWA_Dieska_Laticova



Hash posledného commitu: 9ad6f3ff7b3c8bc0c9b4fc32cc3ec407346a7b4d

Záver

Cieľom tejto práce bolo navrhnutie dátového modelu a vytvorenie funkčnej, responzívnej progresívnej webovej aplikácie na textovú komunikáciu v štýle IRC s využitím progresívnych frameworkov Quasar na frontende a AdonisJS na backende. Aplikácia splňa všetky zadané kritériá, ktoré obsahujú komunikáciu v reálnom čase, možnosť sledovania písania neodoslanej správy a rôzne príkazy cez príkazový riadok slúžiace na vytváranie kanálov, pridávanie, odstraňovanie používateľov. Taktiež je implementovaná možnosť prepínať stavy používateľa a preferencie zobrazovaných upozornení a veľa ďalších funkcionalít.