

Naslov projekta: MicroHub

Ime skupine: Hub Devs (Skupina 2)

Člani:

- Jan Gračner
- Marko Horvat

Namen in cilji projekta:

Opis projekta:

Spletna stran MicroHub je preprosta družbena platforma, zasnovana kot alternativa sodobnim družbenim omrežjem, ki so postala preobremenjena z nepotrebnimi funkcijami in oglaševanjem. Namen projekta je razviti "cloud native" aplikacijo, ki ponovno vzpostavlja bistvo interakcij na družabnih omrežjih: deljenje objav, všečkanje, komentiranje ter prejemanje obvestil o dejavnostih drugih uporabnikov.

Problem, ki ga naša rešitev naslavljajo:

Problem, ki ga projekt MicroHub rešuje, je prezasičenost sodobnih družbenih omrežij z vsebinami, oglaševanjem in nepotrebnimi funkcijami, ki zmanjšujejo pristnost interakcij med uporabniki. Zaradi tega se uporabniška izkušnja pogosto poslabša, družbena omrežja pa izgubljajo svoj prvotni namen – preprosto in neposredno povezovanje ljudi. MicroHub ponuja rešitev v obliki minimalistične spletne platforme, ki se osredotoča na osnovne elemente družbene interakcije in s tem uporabnikom omogoča čistejšo, hitrejšo in bolj pristno komunikacijo.

Ogrodje in razvojno okolje:

Frontend:

- **React (TypeScript, Vite):** za izdelavo uporabniškega vmesnika naše spletne aplikacije. Uporabniški vmesnik bo z mikrostoritvami komuniciral preko REST API klicev. Prav tako bo klical tudi potrebne zunanje API-je (npr. DiceBear za avatars in profilne slike).

Backend (mikrostoritve):

- **Node.js (TypeScript):** izvajalno okolje, v katerem bodo delovale vse mikrostoritve. Vsaka mikrostoritev (Users, Posts, Likes, ...) bo samostojen Node.js program, zapakiran v Docker kontejner.
- **Express.js:** knjižnica, ki bo uporabljena znotraj vsake mikrostoritve za definiranje REST endpointov, obdelavo HTTP zahtev, validacijo in obdelavo podatkov ter komunikacijo z drugimi mikrostoritvami in bazo.

Podatkovna baza:

- **PostgreSQL:** za shranjevanje podatkov. Centralna relacijska podatkovna baza, kjer bo vsaka mikrostoritev imela svojo lastno shemo oziroma instanco, da ohranimo neodvisnost mikrostoritev.

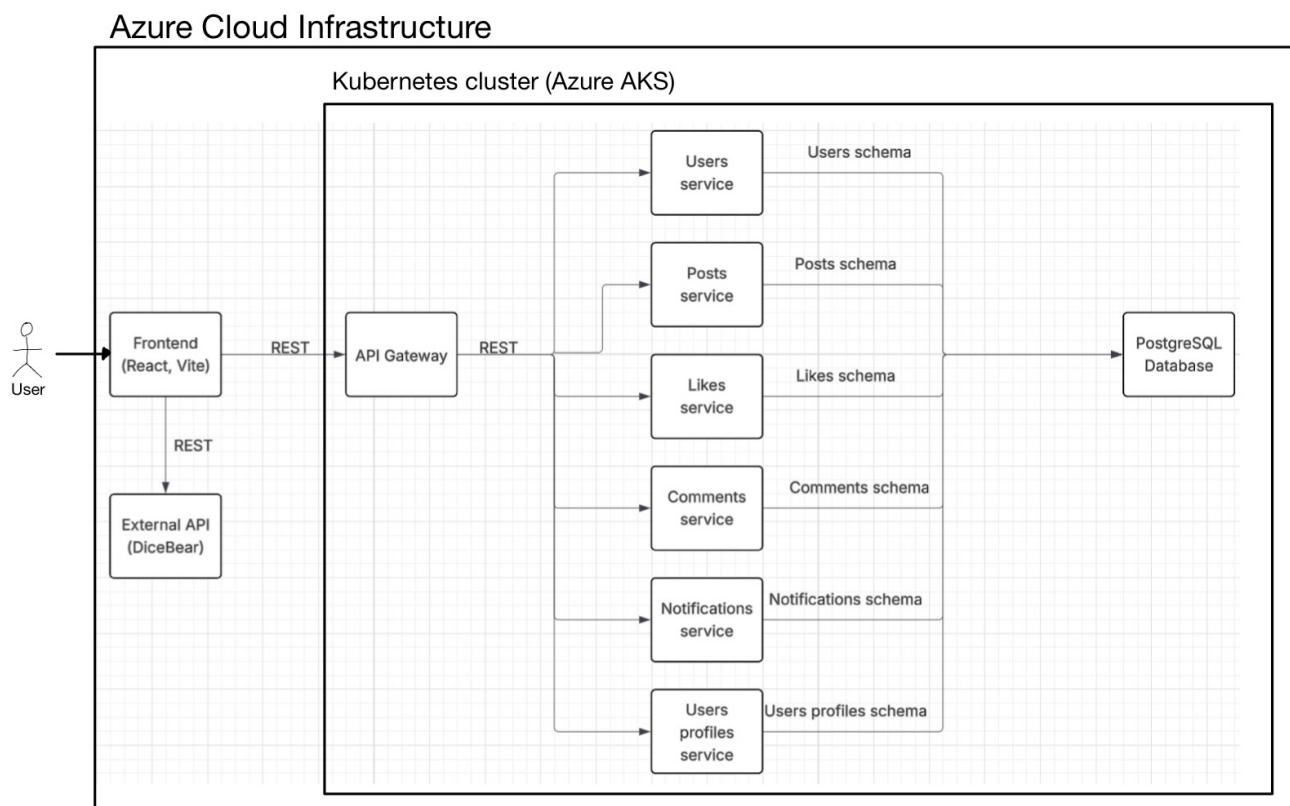
Infrastruktura:

- **Docker:** za pakiranje vsake mikrostoritve v svoj/neodvisen vsebnik
- **Kubernetes:** za izvajanje in upravljanje mikrostoritev, zapakiranih v Docker slike (images), v cloud-native okolu.
- **Azure Cloud Computing Services:** pri projektu bo storitev uporabljena kot oblačna platforma za gostovanje, izvajanje in upravljanje celotne aplikacije.

Razvojno okolje:

- **VS Code:** osrednje razvojno okolje za pisanje, testiranje in rashroščevanje kode (frontend in backend). Integrirano orodje za delo z Gitom, Dockerjem in Kubernetesom.
- **GitHub:** Služil bo kot repozitoriji za shranjevanje, deljenje in verzioniranje izvirne kode, sodelovanje med člani ekipe, CI/CD avtomatizacijo in dokumentacijo projekta.

Shema arhitekture:



Seznam funkcionalnosti mikrostoritev:

1. Users

Mikrostoritev za kreiranje, upravljanje in avtentikacijo uporabnikov. Skrbi za registracijo, prijavo, varno shranjevanje gesel ter razmerja med uporabniki (npr. prijateljstva, sledenje).

Glavne funkcionalnosti:

- Registracija novega uporabnika (e-pošta, uporabniško ime, geslo)
- Prijava in avtentikacija uporabnika (npr. preko izdaje JWT žetonov)
- Preverjanje in obnavljanje prijave (session management)
- Sledenje in prenehanje sledenja drugim uporabnikom: pošiljanje sporočila **user.followed** mikrostoritvi za obvestila
- Validacija dostopa prek JWT pri drugih mikrostoritvah

2. Posts

Mikrostoritev, ki upravlja objave uporabnikov: ustvarjanje, branje, urejanje in brisanje objav. Poleg tega omogoča označevanje objav s temami (tagi) in integracijo z obvestili.

Glavne funkcionalnosti:

- Ustvarjanje nove objave
- Brisanje in urejanje lastnih objav
- Pridobivanje objav posameznega uporabnika
- Prikaz objav vseh uporabnikov ali po temah (feed)
- Iskanje objav po ključnih besedah ali tagih
- Pošiljanje sporočila **post.created** mikrostoritvi za obvestila
- Shranjevanje metapodatkov za vsako objavo (npr. število všečkov, komentarjev)

3. Likes

Mikrostoritev za všečkanje objav.

Glavne funkcionalnosti:

- Dodajanje in odstranjevanje všečka na objavi
- Preverjanje, ali je uporabnik že všečkal objavo
- Štetje skupnega števila všečkov za posamezno objavo
- Pošiljanje sporočila **post.liked** mikrostoritvi za obvestila

4. Comments

Mikrostoritev, ki omogoča komentiranje objav.

Glavne funkcionalnosti:

- Dodajanje komentarjev pod objavo
- Brisanje lastnih komentarjev
- Pridobivanje vseh komentarjev za izbrano objavo
- Pošiljanje sporočila **postcommented** mikrostoritvi za obvestila

5. Notifications

Mikrostoritev za pošiljanje obvestil uporabnikom, kadar nekdo všečka ali komentira njegovo objavo ter ko nekdo začne slediti njegovemu profilu. Sprejema dogodke iz drugih storitev preko message brokerja (npr. RabbitMQ).

Glavne funkcionalnosti:

- Sprejem in obdelava sporočil **post.liked**, **post/commented**, **post.created** in **user.followed**
- Pošiljanje obvestil uporabnikom v aplikaciji in (po želji) tudi preko e-pošte ter potisnih obvestil
- Prikaz seznama neprebranih obvestil
- Označevanje obvestil kot prebranih
- Brisanje starih obvestil po določenem času

6. User Profiles

Mikrostoritev za prikazovanje svojega osebnega profila oziroma profila prijateljev, ki vsebuje statistiko in analitiko, torej prikaz števila objav, všečkov, komentarjev ter rasti dejavnosti.

Glavne funkcionalnosti:

- Prikaz podatkov o uporabniku (javna stran profila)
- Urejanje uporabniškega profila (opis, slika, ime, lokacija ipd.)
- Pridobivanje osnovne statistike profila - število objav, všečkov, komentarjev
- Pridobivanje seznama sledilcev in sledenih uporabnikov
- Generiranje tedenskih ali mesečnih poročil o aktivnosti
- Agregacija podatkov iz *Posts*, *Likes* in *Comments* storitev
- Generiranje grafov aktivnosti uporabnika (npr. dnevni graf dejavnosti)
- Možnost kasnejše integracije z vizualizacijo (npr. Grafana)

Primeri uporabe:

Okvirni primeri uporabe:

1. Registracija uporabnika

Uporabnik izpolni obrazec za registracijo (uporabniško ime, e-pošta, geslo). Mikrostoritev *Users* preveri, ali uporabnik že obstaja, shrani podatke v bazo (PostgreSQL) in vrne potrditev o uspešni registraciji.

2. Prijava uporabnika

Uporabnik vnese prijavne podatke. Mikrostoritev *Users* preveri uporabniško ime in geslo, ustvari JWT žeton (avtentifikacija) ter ga pošlje aplikaciji, ki ga uporablja za izvajanje nadaljnjih zahtev.

3. Ustvarjanje objave

Uporabnik napiše besedilo objave in po želji doda sliko. Mikrostoritev *Posts* shrani objavo v bazo in sproži asinhroni dogodek **post.created**, ki ga kasneje uporabijo *Notifications* in *User Profiles* mikrostoritve.

4. Všečkanje objave

Uporabnik klikne gumb za "všečkanje" objave. Mikrostoritev *Likes* zabeleži všeček v bazi in sproži dogodek **post.liked**, ki ga *Notifications* uporabi za obvestilo avtorju objave.

5. Komentiranje objave

Uporabnik napiše komentar pod objavo. Mikrostoritev *Comments* shrani komentar in sproži dogodek **post/commented**, ki ga *Notifications* uporabi za obveščanje avtorja objave.

6. Sledenje drugemu uporabniku

Uporabnik klikne gumb "Sledi" pod profilom nekega uporabnika. Mikrostoritev *Users* zabeleži odnos med uporabnikoma in sproži dogodek **user.followed**, ki ga *Notifications* uporabi za pošiljanje obvestila slednjemu uporabniku.

7. Ogled profila

Uporabnik odpre profil drugega uporabnika. Mikrostoritev *User Profiles* vrne osnovne podatke o profilu in podatke o aktivnosti (število objav, všečkov, komentarjev).

8. Prejem obvestil

Ko nek drug uporabnik všečka, komentira ali začne slediti profilu uporabnika, mikrostoritev *Notifications* pošlje obvestilo uporabniku v aplikacijo ali preko API klicev na e-pošto ozirna preko potisnih sporočil.

Kompleksnejši primer uporabe:

"Uporabnik všečka objavo drugega uporabnika"

Akterji:

- Uporabnik A (avtor objave)
- Uporabnik B (všečka objavo)

Opis scenarija:

1. Uporabnik B na uporabniškem vmesniku svoje aplikacije klikne "Všečkaj" pod objavo uporabnika A.
2. Uporabniški vmesnik preko REST API klicev mikrostoritve obvesti o dogodku (npr. **POST /api/v1/posts/:id/like**).
3. Mikrostoritev *Users* ustrezno identificira prejemnika dogodka - všečkanje objave.
4. Mikrostoritev *Likes* zabeleži všeček v bazo in pošlje sporočilo **post.liked**.
5. Sporočilo **post.liked** se pošlje prek message brokerja (npr. RabbitMQ).
6. Mikrostoritev *Notifications* prejme dogodek, ustvari obvestilo za uporabnika A in ga shrani v svojo bazo.
7. V aplikaciji uporabnika A *Notifications* prek uporabniškega vmesnika pošlje obvestilo v realnem času ("Uporabnik B je všečkal tvojo objavo"). Opcijsko se obvestilo lahko pošlje tudi preko e-pošte in potisnih obvestil.
8. Mikrostoritev *User Profiles* posodobi statistiko uporabnika A (npr. +1 všeček pod relevantno objavo).

Vključene mikrostoritve:

- **Likes:** logika za všečkanje objave
- **Notifications:** prejem dogodka in ustvarjanje obvestila
- **Users:** identifikacija prejemnika obvestila
- **User Profiles:** posodobitev statistike