MIKROSTORITVE

Contents

[NAČRT MIKROSTORITEV 2](#_Toc206297155)

[UPRAVLJANJE UPORABNIKOV 2](#_Toc206297156)

[API KONČNE TOČKA 2](#_Toc206297157)

[KATALOG KNJIG 3](#_Toc206297158)

[API KONČNE TOČKA 3](#_Toc206297159)

[UPRAVLJANJE BRALNIH POLIC 3](#_Toc206297160)

[API KONČNE TOČKA 4](#_Toc206297161)

[RECENZIJA, OCENE, KOMENTARJI 5](#_Toc206297162)

[API KONČNE TOČKA 5](#_Toc206297163)

[PRIPOROČILNI SISTEM 5](#_Toc206297164)

[API KONČNE TOČKA 6](#_Toc206297165)

[NOTIFIKACIJE 6](#_Toc206297166)

[API KONČNE TOČKA 6](#_Toc206297167)

[STATISTIKA IN BRALNI IZZIVI 7](#_Toc206297168)

[API KONČNE TOČKA 7](#_Toc206297169)

[ARHITEKTURNI DIAGRAM 8](#_Toc206297170)

# NAČRT MIKROSTORITEV

## UPRAVLJANJE UPORABNIKOV

Odgovorna: Barbara Ferlinc

Tehnologije: Node.js/Express, Firebase

### API KONČNE TOČKA

1. POST /users/register:

* dodajanje novega uporabnika
* povezano s storitvijo: xx
* klicana s strani: frontend

1. POST /users/login:

* prijava uporabnika
* povezano s storitvijo: xx
* klicano s strani: frontend

1. POST /users/id:

* najde uporabnike po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /users/allUsers:

* pridobi vse uporabnike
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /users/email/{id}:

* pridobi email uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /users/profile/{id}:

* pridobi profil uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /users/{id}:

* pridobi podatke o uporabniku
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /users/preferences/{id}:

* posodobi preference uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /users/{id}:

* izbriše uporabnika po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /users/emailDomain/{domain}:

* izbriše vse uporabnike po email domeni
* povezano s storitvijo: xx
* klicano s strani: xx

## KATALOG KNJIG

Odgovorna: Barbara Ferlinc

Tehnologije: Node.js/Express, Firebase

### API KONČNE TOČKA

1. POST /books/addBook:

* doda novo knjigo
* povezano s storitvijo: xx
* klicana s strani: xx

1. POST /books/addMultipleBooks:

* doda več novih knjig
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /books/allBooks:

* pridobi vse knjige
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /books/{id}:

* pridobivanje knjige po njenem ID-ju
* povezano s storitvijo: xx
* klicano s strani: recenzije storitev

1. PUT /books/{id}:

* posodabljanje podatkov o knjigi po njenem ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /books/{id}:

* brisanje knjige po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /books/genre/{genre}:

* pridobivanje knjig glede na žanr
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /books/rating/{id}:

* posodabljanje ocene knjige glede na recenzije
* povezano s storitvijo: recenzije storitev
* klicano s strani: frontend

1. DELETE /books/author/{author}:

* brisanje knjig avtorja
* povezano s storitvijo: xx
* klicano s strani: xx

## UPRAVLJANJE BRALNIH POLIC

Odgovorna: Vivien Štampfer

Tehnologije: Node.js/Express, MongoDB

### API KONČNE TOČKA

1. GET /shelves:

* pridobivanje seznama polic uporabnika
* povezano s storitvijo: xx
* klicana s strani: xx

1. POST /shelves:

* ustvarjanje police za uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /shelves/{id}:

* pridobivanje police po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /shelves/{id}:

* posodabljanje police po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /shelves/{id}:

* brisanje police po ID-ju
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /shelves/{userId}/wantToRead:

* dodajanje knjige na polico WantToRead
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /shelves/{userId}/reading:

* dodajanje knjige na polico Reading
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /shelves/{userId}/read:

* dodajanje knjige na polico Read
* povezano s storitvijo: xx
* klicano s strani: xx

1. POST /shelves/{userId}/move:

* premikanje knjige iz ene police na drugo
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /shelves/by-user/{userId}:

* brisanje polic za uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /shelves/{userId}/{shelf}/{bookId}:

* brisanje knjige iz police
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /shelves/{userId}/read:

* pridobi knjige iz READ police po ID-ju uporabnika
* povezano s storitvijo: xx
* klicano s strani: statistika storitev

## RECENZIJA, OCENE, KOMENTARJI

Odgovorna: Lena Bojc

Tehnologije: FastApi, Python, MongoDB

### API KONČNE TOČKA

1. POST /reviews:

* ustvarjanje nove recenzije uporabnika za knjigo
* povezano s storitvijo: knjige storitev
* klicana s strani: frontend

1. POST /comments:

* ustvarjanje komentarja uporabnika za knjigo
* klicano s strani: frontend

1. PUT /reviews/{id}/text:

* posodabljanje teksta v recenziji
* klicano s strani: frontend

1. PUT /reviews/{id}/rating:

* posodabljanje ocene knjige v recenziji
* klicano s strani: frontend

1. GET /reviews/user/{userId}/book/{bookId}:

* pridobivanje recenzije po uporabnikovem ID-ju in ID-ju knjige
* klicano s strani: frontend

1. GET /books/{bookId}/reviews:

* pridobivanje recenzij za knjigo
* povezano s storitvijo: knjige storitev
* klicano s strani: frontend

1. GET /books/{bookId}/comments:

* pridobivanje vseh komentarjev za knjigo
* klicano s strani: frontend

1. GET /reviews/{bookId}/average:

* pridobivanj povprečne ocene za knjigo
* klicano s strani: knjige storitev

1. DELETE /comments/{id}:

* odstranjevanje komentarja
* klicano s strani: frontend

1. DELETE /reviews/{id}:

* odstranjevanje recenzije
* klicano s strani: frontend

## PRIPOROČILNI SISTEM

Odgovorna: Klara Kirbiš

Tehnologije: Node.js/Express, MongoDB

### API KONČNE TOČKA

1. GET /recommendations/{userId}:

* pridobivanje priporočila za uporabnika
* povezano s storitvijo: xx
* klicana s strani: xx

1. POST /recommendations/{userId}:

* ustvarjanje novega priporočila za uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /recommendations/{userId}:

* posodabljanje priporočila za uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /recommendations:

* pridobivanje vseh priporočil
* povezano s storitvijo: xx
* klicano s strani: xx

1. POST /recommendations/{userId}/add:

* dodajanje knjige v priporočila uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /recommendations/{userId}/book/{bookId}:

* brisanje knjige iz priporočila uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

## NOTIFIKACIJE

Odgovorna: Vivien Štampfer

Tehnologije: Node.js/Express, MongoDB

### API KONČNE TOČKA

1. GET /obvestila:

* pridobivanje seznama obvestil
* povezano s storitvijo: xx
* klicana s strani: xx

1. POST /obvestila:

* ustvarjanje novega obvestila
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /obvestila/po-uporabniku/{uporabnikId}:

* brisanje vseh obvestil za uporabnika
* povezano s storitvijo: xx
* klicano s strani: xx

1. GET /obvestila/{id}:

* pridobivanje enega obvestila
* povezano s storitvijo: xx
* klicano s strani: xx

1. PUT /obvestila/{id}:

* posodabljanje obstoječega obvestila
* povezano s storitvijo: xx
* klicano s strani: xx

1. DELETE /obvestila/{id}:

* brisanje obvestila
* povezano s storitvijo: xx
* klicano s strani: xx

## STATISTIKA IN BRALNI IZZIVI

Odgovorna: Lena Bojc

Tehnologije: FastApi, Python, MongoDB

### API KONČNE TOČKA

1. POST /goals:

* ustvarjanje novega cilja za uporabnika
* povezano s storitvijo: bralne police storitev
* klicana s strani: frontend

1. POST /readBooks:

* ustvarjanje prebrane knjige, ki je bila odstranjena iz letnega cilja
* klicano s strani: frontend

1. PUT /goals/{id}/targetBooks:

* posodabljanje števila knjig za bralni izziv
* klicano s strani: frontend

1. PUT /goals/{id}/books:

* dodajanje knjige na seznam prebranih knjig v bralnem izzivu
* klicano s strani: frontend

1. DELETE /goals/{id}/books:

* brisanje knjige iz bralnega izziva uporabnika
* klicano s strani: frontend

1. GET /goals/user/{userId}:

* pridobivanje bralnega izziva za uporabnika
* klicano s strani: frontend

1. GET /goals/{id}/pages:

* pridobivanje števila prebranih strani iz uporabnikovega izziva
* klicano s strani: frontend

1. GET /goals/{userId}/genres:

* pridobivanje distribucije žanrov prebranih knjig
* klicano s strani: uporabniki storitev

1. DELETE /goals/{id}:

* brisanje bralnega izziva za uporabnika
* klicano s strani: frontend

# ARHITEKTURNI DIAGRAM

<https://online.visual-paradigm.com/share.jsp?id=333231303633382d38>

A diagram of a network

AI-generated content may be incorrect.

# RAČUNALNIŠTVO V OBLAKU

## STORITEV 1

odgovorna: XX

ponudnik: xx

opis storitve: xx

povezava do dashboards: xx

dokazilo namestitve v oblaku: (screenshoti dashboards, URL-ji endpointov, logi po klicu)

povezano s storitvijo:

klicano s strani:

CONTAINER STORITVE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Tip** | **Ime storitve** | **Kaj dela (minimalno)** | **Kdo kliče** | **Kam v UI** | **Predlagan ponudnik** |
| C1 | Container | **stats-service** | Sprejme evente (*bookViewed, reviewAdded*) in vrača agregate (top knjige/žanri, dnevni ogledi). | Frontend/Backend ob obisku knjige in ob POST/PUT na **reviews**/**books** | “Statistika” widget (Top 5, danes/ta teden) | **Render** (free container) |
| C2 | Container | **logs-service** | Preprost HTTP “sink”: POST /logs shranjuje JSON loge; GET /logs?limit=50 za pregled. | Vse MS pri uspešnih/napak klicih (npr. **/users**, **/books**, **/reviews**) | Admin stran “Dnevniki” | **Koyeb** ali **Fly.io** (free app) |
| C3 | Container | **feature-flags** | Vrača JSON z zastavicami (npr. showNewStatsWidget: true). | Frontend pred renderjem strani | Nastavitve/feature preklopi (brez DB) | **Render** (drugi račun) ali **Railway** free dev plan\* |
| C4 | Container | **badges-service** | Generira simple “badges” po pravilih (npr. 5 prebranih → “Novinec”). POST /events/read + GET /badges/{userId}. | **shelves**/**goals** po uspešnih spremembah polic/ciljev | “Moji dosežki” (seznam značk) | **Fly.io** ali **Koyeb** |

SERVERLESS STORITVE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Tip** | **Ime funkcije** | **Kaj dela (minimalno)** | **Trigger** | **Kam v UI** | **Predlagan ponudnik** |
| S1 | Serverless | **notify-new-review** | Ob novi recenziji pošlje obvestilo (e-mail/push ali vsaj zapiše v logs-service). | Backend po POST /reviews | Toast “Recenzija poslana” | **Vercel Functions** |
| S2 | Serverless | **spam-check-review** | Preprost “keyword filter” za recenzije; vrne allowed:true/false. | Backend pred POST /reviews | UI show: “Recenzija zavrnjena (spam)” | **Vercel** ali **Netlify Functions** |
| S3 | Serverless | **recompute-weekly-trends** | Preračuna tedenske trende iz stats-service in zapiše nazaj (ali samo vrne). | Ročni HTTP klic (gumb “Osveži trende”) | Gumb “Osveži statistiko” | **AWS Lambda** (HTTP/API Gateway) |
| S4 | Serverless | **goal-reminder** | Po klicu vrne/pošlje opomnik za bralni cilj (npr. še N strani). | Backend/cron-simulacija (ročno HTTP) po **/goals** | Banner “Današnji cilj” | **Cloudflare Workers** (free) |

DELITEV NALOG

Vivi: C4 + S4

Klara: C1 + S3

Barbara: C2 + S1

Lena: C3 + S2

INTEGRACIJE

* Po prikazu strani knjige frontend pošlje POST /events v stats-service (C1).
* Po POST /reviews backend pokliče najprej spam-check-review (S2); če allowed, shrani recenzijo in nato pokliče notify-new-review (S1) ter POST /events (C1).
* Ob PUT na /shelves ali /goals backend pošlje event v badges-service (C4); frontend prikazuje značke iz GET /badges/{userId}.
* Frontend pred renderjem pokliče feature-flags (C3), da skrije/prikaže nove widg­ete.
* Admin/debug: vsak backend klic na koncu pošlje POST /logs v logs-service (C2).
* Gumb “Osveži trende” pokliče recompute-weekly-trends (S3), potem frontend z GET /stats/top osveži widget.

DEMO STORYJI

1) odpreš knjigo → števec zraste,

2) dodaš recenzijo → toast + log,

3) klikneš “Osveži trende” → top lista se spremeni,

4) dodaš knjigo na polico → pojavi se značka.

SKELETI MINIMALNIH API-jev

stats-service (C1):

* POST /events { type, userId, bookId, … } → 200
* GET /stats/top?window=7d → [ {bookId, views}, … ]

logs-service (C2):

* POST /logs { service, level, msg, meta }
* GET /logs?limit=50

feature-flags (C3):

* GET /flags → { showStats:true, showBadges:true }

badges-service (C4):

* POST /events/read { userId, bookId }
* GET /badges/{userId} → [ "Novinec", "Železna volja" ]

notify-new-review (S1):

* POST /api/notify { userId, bookId, reviewId } → { status:"sent" }

spam-check-review (S2):

* POST /api/spam-check { text } → { allowed:true, score:0.03 }

recompute-weekly-trends (S3):

* POST /api/recompute → { updated:true }

goal-reminder (S4):

* POST /api/remind { userId } → { reminder:"Še 12 strani do cilja!" }