MIKROSTORITVE

Contents

[NAČRT MIKROSTORITEV 3](#_Toc206388178)

[UPRAVLJANJE UPORABNIKOV 3](#_Toc206388179)

[API KONČNE TOČKA 3](#_Toc206388180)

[KATALOG KNJIG 4](#_Toc206388181)

[API KONČNE TOČKA 4](#_Toc206388182)

[UPRAVLJANJE BRALNIH POLIC 4](#_Toc206388183)

[API KONČNE TOČKA 5](#_Toc206388184)

[RECENZIJA, OCENE, KOMENTARJI 6](#_Toc206388185)

[API KONČNE TOČKA 6](#_Toc206388186)

[PRIPOROČILNI SISTEM 6](#_Toc206388187)

[API KONČNE TOČKA 7](#_Toc206388188)

[NOTIFIKACIJE 7](#_Toc206388189)

[API KONČNE TOČKA 7](#_Toc206388190)

[STATISTIKA IN BRALNI IZZIVI 8](#_Toc206388191)

[API KONČNE TOČKA 8](#_Toc206388192)

[ARHITEKTURNI DIAGRAM 9](#_Toc206388193)

[RAČUNALNIŠTVO V OBLAKU 10](#_Toc206388194)

[STORITEV: feature-flags 10](#_Toc206388195)

[STORITEV: spam-check-review 10](#_Toc206388196)

[STORITEV: content-notice 10](#_Toc206388197)

[STORITEV: book-badges 11](#_Toc206388198)

[STORITEV: xx 11](#_Toc206388199)

[STORITEV: xx 11](#_Toc206388200)

[STORITEV: xx 11](#_Toc206388201)

[STORITEV: xx 11](#_Toc206388202)

# NAČRT MIKROSTORITEV

## UPRAVLJANJE UPORABNIKOV

Odgovorna: Barbara Ferlinc

Tehnologije: Node.js/Express, Firebase

### API KONČNE TOČKA

1. POST /users/register:

* dodajanje novega uporabnika
* klicana s strani: frontend

1. POST /users/login:

* prijava uporabnika
* klicano s strani: frontend

1. POST /users/id:

* najde uporabnike po ID-ju
* klicano s strani: frontend

1. GET /users/allUsers:

* pridobi vse uporabnike
* klicano s strani: frontend

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

* pridobi email uporabnika
* klicano s strani: xx

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

* pridobi profil uporabnika
* klicano s strani: xx

1. PUT /users/{id}:

* posodobi podatke o uporabniku
* klicano s strani: xx

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

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

1. DELETE /users/{id}:

* izbriše uporabnika po ID-ju
* klicano s strani: frontend

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

* izbriše vse uporabnike po email domeni
* klicano s strani: frontend

## KATALOG KNJIG

Odgovorna: Barbara Ferlinc

Tehnologije: Node.js/Express, Firebase

### API KONČNE TOČKA

1. POST /books/addBook:

* doda novo knjigo
* klicana s strani: frontend

1. POST /books/addMultipleBooks:

* doda več novih knjig
* klicano s strani: xx

1. GET /books/allBooks:

* pridobi vse knjige
* klicano s strani: frontend

1. GET /books/{id}:

* pridobivanje knjige po njenem ID-ju
* klicano s strani: recenzije storitev, frontend

1. PUT /books/{id}:

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

1. DELETE /books/{id}:

* brisanje knjige po ID-ju
* klicano s strani: frontend

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

* pridobivanje knjig glede na žanr
* 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
* klicano s strani: frontend

## UPRAVLJANJE BRALNIH POLIC

Odgovorna: Vivien Štampfer

Tehnologije: Node.js/Express, MongoDB

### API KONČNE TOČKA

1. GET /shelves:

* pridobivanje seznama polic uporabnika
* klicana s strani: frontend

1. POST /shelves:

* ustvarjanje police za uporabnika
* klicano s strani: frontend

1. GET /shelves/{id}:

* pridobivanje police po ID-ju

1. PUT /shelves/{id}:

* posodabljanje police po ID-ju

1. DELETE /shelves/{id}:

* brisanje police po ID-ju
* klicano s strani: frontend

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

* dodajanje knjige na polico WantToRead
* klicano s strani: frontend

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

* dodajanje knjige na polico Reading

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

* dodajanje knjige na polico Read

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

* premikanje knjige iz ene police na drugo
* klicano s strani: frontend

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

* brisanje polic za uporabnika

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

* brisanje knjige iz police

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

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

1. GET / shelves/user/userId:

* Klicano s strani: frontend

## 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
* klicana s strani: frontend

1. POST /obvestila:

* ustvarjanje novega obvestila
* povezano s storitvijo: Quote api na vercelu
* klicano s strani: frontend

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

* brisanje vseh obvestil za uporabnika
* klicano s strani: frontend, user-service

1. GET /obvestila/{id}:

* pridobivanje enega obvestila
* klicano s strani: frontend

1. PUT /obvestila/{id}:

* posodabljanje obstoječega obvestila
* klicano s strani: frontend

1. DELETE /obvestila/{id}:

* brisanje obvestila
* klicano s strani: frontend

## 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 company

AI-generated content may be incorrect.

# RAČUNALNIŠTVO V OBLAKU

## STORITEV: feature-flags

odgovorna: Lena Bojc

ponudnik: Render (container)

končna točka:

* POST /goal-hints
* opis: Izračuna tempo branja za tekoče leto in vrne namig coacha (status napredka ter priporočeni tempo/teden).
* telo: { "targetBooks": 10, "completedBooks": 3, "year": 2025 }
* odziv (200 OK): { "status": "behind", "pacePerWeek": 1.5, "daysLeft": 138, "behindBy": 2, "note": "Behind by ~2 book(s). ~1.5/week hits the target."}

povezava do dashboarda: <https://dashboard.render.com/web/srv-d2h1teruibrs73ertum0/deploys/dep-d2h8tg3uibrs73f272i0>

klicano s strani: statistics-service

## STORITEV: spam-check-review

odgovorna: Lena Bojc

ponudnik: Vercel (stateless)

končna točka:

* POST /api/spam-check-review
* opis: Pregleda besedilo za nedovoljene besede. Če najde nedovoljeno besedo, vrne to besedo in njen tip.
* telo: { "text": "Your review text here" }
* odziv (200 OK):
  + { "allowed": false, "type": "harassment", "word": "idiot" }
  + { "allowed": true, "type": null, "word": null }

povezava do dashboarda: <https://vercel.com/lenabojcs-projects/spam-check-review>

klicano s strani: review-service

## STORITEV: content-notice

odgovorna: Barbara Ferlinc

ponudnik: Render (container)

končna točka:

* POST /v1/content-notice
* opis: Iz naslova/opisa izlušči opozorila (npr. violence, strong\_language, substance, romance, mature\_themes) in predlaga starostni razpon (All ages, 12+, 16+, 18+).
* telo: { "title": "Book title", "description": "Short synopsis..." }
* odziv (200 OK): { "notices": ["violence","mature\_themes"], "recommendedAge": "16+" }

povezava do dashboarda: <https://dashboard.render.com/web/srv-d2hfu6vdiees73eg1u90>

klicano s strani: book-service (pri GET /books/:id – odziv razširiš z contentInfo)

## STORITEV: book-badges

odgovorna: Barbara Ferlinc

ponudnik: Vercel (stateless)

končna točka:

* POST /api/book-badges
* opis: Vrne badge oznake in osnovno UI temo za kartico knjige glede na meta-podatke (npr. Short read, New release, Top rated) ter icon in color za prikaz na frontendu.
* telo: {"title": "Book title", "genres": ["Fantasy","Adventure"], "pages": 420, "averageRating": 4.7, "ratingsCount": 120, "publishedYear": 2025}
* odziv (200 OK): { "badges": ["New release","Top rated"], "icon": "📘", "color": "#1e90ff" }

povezava do dashboarda: <https://vercel.com/barbaraferlincs-projects/book-badges>

klicano s strani: book-service (pri GET /books in GET /books/:id – odziv razširiš z badgesInfo)

## STORITEV: Quotes

odgovorna: Vivien Štamfper

ponudnik: Vercel

končna točka:

* POST/GET /api/quotes : {"quote":"Znanje je moč, beri naprej!"}
* opis: vrne generiran citat in ga da k knjigi, ki jo dodaš v read.

povezava do dashboarda: <https://quotes-olive-seven.vercel.app/api/quotes>

klicano s strani: Obvestilo service

## STORITEV: Badges

odgovorna: Vivien Štampfer

ponudnik: Render

končna točka:

* GET- { "userId": "123",   "readCount": 35,  "badge": "Silver"} - <https://badges-rvmz.onrender.com/badges/123>
* POST - https://badges-rvmz.onrender.com
* opis: pogleda koliko je knjig v read in dodeli badge glede na to koliko jih je prebranih

povezava do dashboarda: https://badges-rvmz.onrender.com

klicano s strani: shelves service

## STORITEV:Nasveti (Tips Service)

odgovorna: [ime odgovorne osebe]

ponudnik: Vercel (stateless)

končna točka: POST /api/tips

opis: Vrne seznam nasvetov, ustvari nov nasvet ali izbriše obstoječega preko GraphQL. Vsak nasvet vsebuje id, besedilo in kategorija.

telo (primer za pridobitev vseh nasvetov): query { tips {id besedilo kategorija}}

odziv (200 OK):{"data": {"tips": [{ "id": "1", "besedilo": " Vzemi knjigo s sabo na avtobus.", "kategorija": " Praktični nasveti" }}

povezava do dashboarda: <https://nasvet-serverless.vercel.app/>

klicano s strani: frontend aplikacija (MotivationPage.jsx – prikaz nasvetov uporabniku)

Opis: Pridobi seznam nasvetov ali posamezen nasvet, ustvari nov nasvet ali izbriše obstoječega preko GraphQL query in mutation.

## STORITEV: xx

odgovorna: xx

ponudnik: xx

končna točka:

* POST/GET
* opis: xx

povezava do dashboarda: xx

klicano s strani: xx