**Laikas: 4 val.**

**Pateikimas vertinimui:   
- Zip formatu atsiųstas programinis kodas (kodo failai, be node\_modules aplanko)  
- Duomenų bazės URL nurodomas .env kodo viduje .env faile (nurodoma, kad DB pasiekiama iš visų IP adresų).**

Užduotis.

Sukurkite REST API, naudodami Node.js/Express.js/MongoDB, kuris grąžina informaciją apie vartotojus.

Įsivaizduokite, kad į jus kreipėsi įmonė, kuri turi savo REST API "endpoint" arba "route", kuris grąžina daug informacijos apie vartotojus (<https://prnt.sc/1v663x5> ). Jie pastebėjo, kad dažnu atveju ši informacija yra perteklinė ir ne visada visa naudojama. Todėl buvo nuspręsti turėti savo vartotojų duomenų bazę (users\_db, duomenų modelis: \_id, name, email, address) bei naudoti vartotojus, kurie grąžinami iš čia <https://jsonplaceholder.typicode.com/users>. Todėl atsirado poreikis sukurti naujus penkis naujus "endpoint" (arba "route"), kurie:

1. Kreipsis į <https://jsonplaceholder.typicode.com/users> ir į users\_db ir apdirbę informaciją grąžins:
   1. GET - /api/users/names - masyvą su vartotojais, kurie bus objekte ir jame bus matomas vartotojo id ir vardas
   2. GET - /api/users/emails - masyvą su vartotojais, kurie bus objekte ir jame bus matomas vartotojo id, vardas ir email
   3. GET - /api/users/address - masyvą su vartotojais, kurie bus objekte ir jame bus matomas vartotojo id, vardas ir pilnas adresas (gatvė ir miestas kaip vienas string)
2. Sudarytoje duomenų bazėje users\_db:
   1. POST - /api/users - talpins vartotojus (name, email, address)
   2. GET - /api/users - grąžins vartotojus (\_id, name, email, address)

*Svarbu:*

*Duomenis imti iš čia:* [*https://jsonplaceholder.typicode.com/users*](https://jsonplaceholder.typicode.com/users) *ir sukurtos MongoDB duomenų bazės users\_db.*