

Uzdevums: To-Do lista ar lietotāju autentifikāciju

Uzdevuma apraksts:

Paplašiniet iepriekš izveidoto To-Do list sistēmu, ieviešot lietotāju pieslēgšanos (login), reģistrāciju (signup) un iespēju katram lietotājam izveidot savus uzdevumus.

Prasības:

1. Lietotāju autentifikācija:

- Ieviesiet reģistrācijas (`POST /signup`) un pieslēgšanās (`POST /login`) maršrutus.
- Lietotāju dati jāglabā atsevišķā MongoDB kolekcijā (`User`), ar šādiem laukiem:
 - `username` (unikāls)
 - `password` (jābūt šifrētam, izmantojot `bcrypt`)

2. Autorizācija:

- Lietotājiem jāsaņem **JSON Web Token (JWT)** pēc pieslēgšanās.
- Visos uzdevumu (tasks) API pieprasījumos jābūt JWT tokenam (jāievieš middleware, kas pārbauda tokena derīgumu).
- Katrs uzdevums jāpiesaista konkrētam lietotājam, pievienojot lauku `userId`.

3. MongoDB struktūras izmaiņas:

- Uzdevumu modelī (`Task`) jāiekļauj papildu lauks:
 - `userId`: atsauce uz `User` kolekcijas objektu (`ObjectId`)

4. API izmaiņas:

- Visi uzdevumu API (GET, POST, PUT, DELETE) jāfiltrē un jāveic tikai konkrētā lietotāja kontekstā, balstoties uz `userId`, kas izvilks no JWT.

5. Front-end izmaiņas:

- Ievietojiet reģistrācijas un pieslēgšanās formas (vienkārši HTML vai kā popup).
- Saglabāji JWT tokenu pārlūkā (`localStorage`).
- Iekļauji šo tokenu visos `fetch` pieprasījumos pie API.

- Lietotājs var piekļūt tikai saviem uzdevumiem.

Tehnoloģijas un bibliotēkas:

- `express`, `mongoose`, `jsonwebtoken`, `bcrypt`, `cors`
- Front-end: HTML/CSS + JS (fetch)

Papildus uzdevumi (brīvprātīgi):

- Ieviesiet logout funkcionalitāti.
- Atjaunojiet front-end izkārtojumu, lai uzdevumi un lietotāja sesija tiktu pārvaldīti ērtāk.
- Validācija: gan front-endā, gan back-endā (piem., pārbaudīt tukšus laukus, paroles stiprumu).