

# NODE, EXPRESS, JSON API, CRUD

## Užduotis - TYPE19

1. Sukurti serverį naudojant expressjs

2. Sukurti duomenų bazę naudojant mysql

Kuriant visą projektą naudoti git su logiškais commitais.

3. Sukurti lenteles duomenų bazėje:

3.1. 'users' su laukais: id, name, email, password, role\_id

3.2. 'shop\_items' su laukais: id, name, price, description, image, item\_type\_id

3.3. 'item\_types' su laukais: id, name

3.4. 'orders' su laukais: id, user\_id, shop\_item\_id, quantity, total\_price, status

3.5. 'user\_roles' su laukais: id, name

4. Sukurti CRUD maršrutus/operacijas vartotojams

4.1. POST /api/auth/register - registruoti vartotoją su name, email, password, role\_id

4.2. POST /api/auth/login - prisijungti vartotoją naudojant email ir password

5. Sukurti CRUD maršrutus/operacijas parduotuvės prekėms

5.1. POST /api/shop\_items - sukurti parduotuvės prekę su name, price, description, image, item\_type\_id

5.2. GET /api/shop\_items - gauti visas parduotuvės prekes

5.3. GET /api/shop\_items/:id - gauti parduotuvės prekę pagal id

5.4. DELETE /api/shop\_items/:id - ištrinti parduotuvės prekę pagal id

6. Rankiniu būdu (phpmyadmin) įterpti 4 ar daugiau item\_types: food, drink, electronic, clothes

## 7. Sukurti CRUD maršrutus/operacijas užsakymams

7.1. POST /api/orders - sukurti užsakymą su user\_id, shop\_item\_id, quantity, total\_price, status

7.2. GET /api/orders - gauti visus užsakymus. gauta informacija turi būti su vartotojo vardu ir prekės pavadinimu bei vieneto kaina

7.3. GET /api/orders/:id - gauti užsakymą pagal id

7.4. GET /api/orders/user/:user\_id - gauti visus užsakymus, priklausančius tam tikram vartotojui. gauta informacija turi būti su vartotojo vardu ir prekės pavadinimu bei vieneto kaina

## 8. Sukurti GET maršruta /api/user\_roles - gauti visas roles

8.1 Rankiniu būdu (phpmyadmin) įterpti 3 user\_roles lentelės: admin, user, guest

## 9. Sukurti middleware POST, PUT ir PATCH maršrutams patikrinti. Tikrinti ar užklausoje yra visi reikalingi laukai ir validuoti juos pagal tipą ir logiką.

9.1. Jei užklausoje nėra visų reikalingų laukų, siųsti klaidos pranešimą iš serverio pusės, kuriame būtų klaidos informacija kas negerai. Jei tai validacija formai tai turetu būti visi laukai ir klaidos žinutės, kas negerai.

# FRONTEND

1. Sukurti HTML puslapius: login.html, register.html, shop.html, add\_item.html, orders.html

2. Pridėti stiliaus ir JS failus prie kiekvieno puslapio

3. Sukurti prisijungimo puslapį

3.1. Sukurti formą su email ir password laukais

3.2. Naudoti JavaScript, kad būtų galima siųsti email ir password į serverį

3.3. Jei email ir password yra teisingi, išsaugoti vartotojo email ir rolę localStorage, nukreipti į shop.html

3.4. Jei email ar password yra neteisingi, rodyti klaidos pranešimą iš serverio pusės

4. Sukurti registracijos puslapį

4.1. Sukurti formą su name, email, password, repeatPassword ir role laukais

4.1.1. role turėtų būti select įvestis su pasirinkimais iš user\_roles lentelės (arba rankiniu būdu sukurtos, bet tai bus laikoma maziau pliusu)

4.2. Naudoti JavaScript, kad būtų galima siųsti name, email, password ir role į serverį

4.3. Jei email nėra užimtas, sukurti vartotoją su name, email, password ir role

4.4. Jei email yra užimtas, rodyti klaidos pranešimą iš serverio pusės

5. Sukurti shop.html puslapį

5.1. Sukurti prisitaikančio dizaino gridą su parduotuvės prekėmis

5.2. Naudoti JavaScript, kad būtų galima gauti visas parduotuvės prekes iš serverio/backendo

5.3. Naudoti JavaScript, kad parodyti visas parduotuvės prekes kaip prisitaikančio dizaino korteles

5.4. Sukurti mygtuką, skirtą pridėti parduotuvės prekę į krepšelį ('orders' table)

5.5. Sukurti mygtuką, skirtą ištrinti parduotuvės prekę iš shop\_items lentelės

## 6. Sukurti add\_item.html puslapį

6.1. Sukurti formą su name, price, description, image, item\_type\_id (select input)

6.1.1. item\_type\_id turėtų būti select įvestis su pasirinkimais iš item\_types lentelės (arba rankiniu būdu sukurtos, bet tai bus laikoma mažiau pliusu)

6.2. Naudoti JavaScript, kad būtų galima siųsti naują parduotuvės prekę į serverį

6.3. Jei parduotuvės prekė yra sukurta, nukreipti į shop.html

6.4. Jei parduotuvės prekė nėra sukurta, rodyti klaidos pranešimą iš serverio pusės

## 7. Sukurti orders.html puslapį

7.1. Sukurti lentelę su užsakymais

7.2. Naudoti JavaScript, kad būtų galima gauti visus užsakymus iš serverio

7.3. Sukurti select įvestį, skirtą pasirinkti vartotojus. Jei vartotojas yra pasirinktas, rodyti tik tuos užsakymus, kurie priklauso tam vartotojui

## 8. Kiekvienas puslapis turėtų turėti naršymo juostą su nuorodomis į kitus puslapius

8.1. Jei vartotojas prisijungęs, rodyti atsijungimo mygtuką

8.2. Jei vartotojas neprisijungęs, rodyti prisijungimo ir registracijos mygtukus

## 9. Sukurti atsijungimo mygtuką

9.1. Naudoti JavaScript, kad būtų galima pašalinti vartotojo email ir rolę iš localStorage

9.2. Nukreipti į login.html

## Extra užduotys gauti daugiau nei 10 balu

1. Jei vartotojas prisijungęs ir turi rolę admin, tik tada rodyti nuorodą į add\_item.html

2. Tik jei vartotojas prisijungęs ir yra admin, tik tada rodyti ištrynimo mygtuką

3. orders.html. Jei vartotojas prisijungęs ir turi rolę admin, tik tada rodyti pasirinkimo įvestį, skirtą pasirinkti vartotojus.