

Reflectus alternatyve

Frontend UX ir refleksijų formų tobulinimas – pokalbio irasu PDF

Data: 2026-01-04

Kalba: LT/EN (kaip pokalbyje)

Turinys:

1. Santrauka
2. Pokalbio transkriptas
3. Pateikti ekrano vaizdai (priedas)

Santrauka

Pokalbyje suformuota kryptis, kaip sutvarkyti Reflectus frontend: 1) sutvarkyti navigacija (visi mygtukai turi vesti i realius route), 2) sutvarkyti globalu layout (turinys centre, nera prilipimo prie krasto), 3) pagerinti refleksiju formu UX (korteles, spacing, 1-5 ivertinimai kaip pasirinkimai), 4) patobulinti refleksiju klausimus (pamokos, savaites, kontrolinio, projekto, savijautos). Taip pat aptarta, kaip pushinti projekta i GitHub per VS Code terminal.

Pokalbio transkriptas

User

sukuriam dbar frontend pilnai. daug geriau. reikia user friendly padaryt. iškark sukurk pagrindines kokias refleksijas.

Assistant

Žemiau yra pilnas, veikiantis, user-friendly frontend (React + Vite + Tailwind) su:

- Login (role: student/teacher)
- Student UI: “Nauja refleksija”, formos su progresu, autosave, istorija
- Teacher UI: klasės peržiūra, filtrai, refleksijos peržiūra + komentarai
- Pagrindinės refleksijų šablonai (pamokos, savaitės, kontrolinio, projekto, savijautos)
- Veikia ir be backend: Mock režimas (localStorage). Kai backend bus – perjungama per VITE_API_URL.

1) Sukuriam frontend (Ubuntu Terminal)

Atidaryk Ubuntu Terminal ir vykdyk (projekto šaknyje):

```
cd ~/projektai/reflect # pakeisk į savo repo kelią
npm create vite@latest frontend -- --template react
cd frontend
npm i

npm i react-router-dom zustand lucide-react react-hook-form zod @hookform/resolvers
npm i -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

2) Tailwind konfigūracija

tailwind.config.js (pakeisk turinį)

```
/** @type {import('tailwindcss').Config} */
export default {
  content: ["../index.html", "./src/**/*.{js,jsx}"],
  theme: { extend: {} },
  plugins: [],
};
```

src/index.css (pakeisk turinį)

```
@tailwind base;
@tailwind components;
@tailwind utilities;

:root { color-scheme: light; }
html, body, #root { height: 100%; }
```

3) Failų struktūra (sukurk VS Code)

Sukurk šituos failus:

```
src/  
App.jsx  
main.jsx  
router.jsx  
data/templates.js  
lib/storage.js  
lib/mockApi.js  
lib/api.js  
stores/authStore.js  
components/  
ui.jsx  
Layout.jsx  
ProtectedRoute.jsx  
ReflectionForm.jsx  
pages/  
Login.jsx  
student/  
StudentHome.jsx  
NewReflection.jsx  
FillReflection.jsx  
History.jsx  
teacher/  
Teacher
```

User

darbu plana codexui

Assistant

Žemiau yra darbo planas Codex/Copilot'ui (kad galėtum kopijuoti promptus į VS Code Copilot Chat / Codex ir jis generuotų kodą tvarkingai). Planas daromas taip, kad veiktų be backend (mock + localStorage), o vėliau tik pakeisi api.js į realius endpoint'us.

0) Tikslas ir taisyklės (viena tiesa visur)

User flow:

- Login su mokyklos duotais prisijungimais.
- Rolė (student/teacher) nustatoma pagal prisijungimą (vėliau backend atsakymas).
- Student: kuria refleksiją pagal šabloną, pildo, pateikia, mato istoriją.
- Teacher: mato mokinių refleksijas, filtruoja, atsidaro, palieka komentarą.

Technologijos:

- React + Vite + Tailwind
- react-router-dom
- zustand (auth + app state)
- react-hook-form + zod (formoms)
- localStorage mock režimui
- 1 vieta API: src/lib/api.js (vėliau prijungiam backend)

1) Repo ir paleidimas (Ubuntu Terminal)

Ubuntu terminale (projekto šaknyje):

```
cd ~/projektai/reflect
cd frontend
npm i
npm run dev
```

2) Milestone'ai (ką turi matyti po kiekvieno)

M1: veikia routing + UI layout + login puslapis

M2: auth store + protected routes + role-based meniu

M3: reflection templates + student dashboard + new reflection

M4: reflection form (autosave, progress, validate) + submit

M5: student history + detail view

M6: teacher dashboard (list+filters) + detail + comment

M7: api wrapper + mockApi + perjungimas į real API per .env

M8: UI polishing (empty states, error states, mobile) + build

3) Codex prompt'ai (kopijuok po vieną, eilės tvarka)

... (turinys tęsiamas šiame pokalbyje)

User

how to push to github

Assistant

Push to GitHub (Ubuntu Terminal)

1) Go to your project folder

```
cd ~/projektai/reflect # change to your real folder
```

2) Verify it's a git repo + see changes

```
git status
```

If it says "not a git repository", you're in the wrong folder (go one level up/down and try again).

3) Make sure the GitHub remote exists

```
git remote -v
```

If you see origin ...github.com... → OK.

If empty, add it (replace with your repo URL):

```
git remote add origin https://github.com/USERNAME/REPO.git
```

4) Add files + commit

```
git add -A
git commit -m "Initial frontend"
```

5) Push

Check your branch name

User

vscode terminal

Assistant

Push to GitHub using VS Code Terminal (Ubuntu)

1) Open the correct folder in VS Code

- VS Code → File → Open Folder...
- Choose your project folder (the one that has frontend/, backend/, etc.)

2) Open VS Code Terminal

- Top menu: Terminal → New Terminal
- It opens at the project folder (usually). Verify:

```
pwd  
ls
```

3) Check git status

```
git status
```

If you see “not a git repository”, you opened the wrong folder. Open the folder that contains .git.

4) Add remote (only if needed)

Check:

```
git remote -v
```

If nothing is shown, add your GitHub repo URL:

```
git remote add origin https://github.com/lmcorpofficial/reflect.git
```

5) Add → commit → push

```
git add -A  
git commit -m "Update frontend"
```

Now push (try one of these, depending on branch name):

```
git push -u origin main
```

If that fails because your branch is master:

```
git push -u origin master
```

If you don't know the branch:

```
git branch
```

The one with * is the current branch.

6) If GitHub asks for login/password

GitHub no longer accepts account passwords for git. Use Personal Access Token (PAT) as the password.

Quick setup (recommended):

```
sudo apt update  
sudo apt install -y gh
```

```
gh auth login
```

Then push again:

```
git push -u origin main
```

If you paste the exact error output from the terminal, I'll tell you the exact fix for your case.

User

Uploaded screenshots. turim struktūrą. bet neatrodo profesionaliai. ir tie mygtukai nevisi veikia ypač pas mokytojus ir mokinius. refleksijų

Assistant

Problema pagal screenshot'us

1) UI "išsitempęs" per visą ekraną, daug tuščios erdvės, tipografija/spacing nesuderinti → neatrodo profesionaliai.

2) "Kortelės-mygtukai" dalis jų greičiausiai yra tik <div> be realaus Link/onClick, arba veda į neegzistuojantį route.

3) Refleksijų srautas nėra pilnai sujungtas: sukurti → išsaugoti → atsiranda istorijoje / mokytojai peržiūroje.

Žemiau – konkretus Codex darbo planas, kad sutvarkytų UI + navigaciją + refleksijų veikimą.

0) Greitas audit (VS Code terminal)

```
cat package.json
```

1) Navigacija: sutvarkyti VISUS mygtukus, kad jie realiai veiktų

... (planas tęsiamas)

User

log in puslapis. man nepatinka kad jis negražus ir kad jis taip sugrūstas... reikia userfriendly. 1 refleksijos kūrimas iš mokinio puses... toliau negaliu nieko daryti nes mygtukai paspausti nieko nedaro

Assistant

Žinau, ką matai: UI yra „susigrūdęs“ kairėje, o likusi ekrano dalis tuščia. Ir svarbiausia – kortelės/mygtukai yra „dead click“ (paspaudi ir nieko nevyksta). Reikia sutvarkyti dviem etapais: (A) kad viskas būtų paspaudžiama ir vestų į puslapius, tada (B) profesionalus layout.

Žemiau – Codex darbo planas su aiškiais rezultatais.

A. 1 prioritetas: sutvarkyti, kad VISI mygtukai veiktų (navigacija)

... (planas tęsiamas)

User

MATAI PADARKYK KAD JOS NORMALIAI BUTU... PABANDYK PAGERINTI KALSUIMUS. SUTVARKOM DIZAINA... IŠ MOKINIO PUSĖS. PAKEITIMAI

Assistant

Žiūriu į ekranus: problema ne funkcijose, o layout'e ir UI komponentuose. Viskas prilipę prie kairio krašto, formos "išsitiesia" neteisingai, kai kur tekstas "išlenda", o 1-5 vertinimai yra kaip paprasti input'ai (ne user-friendly).

Žemiau – konkretus pakeitimų planas "iš mokinio pusės" + pagerinti klausimai.

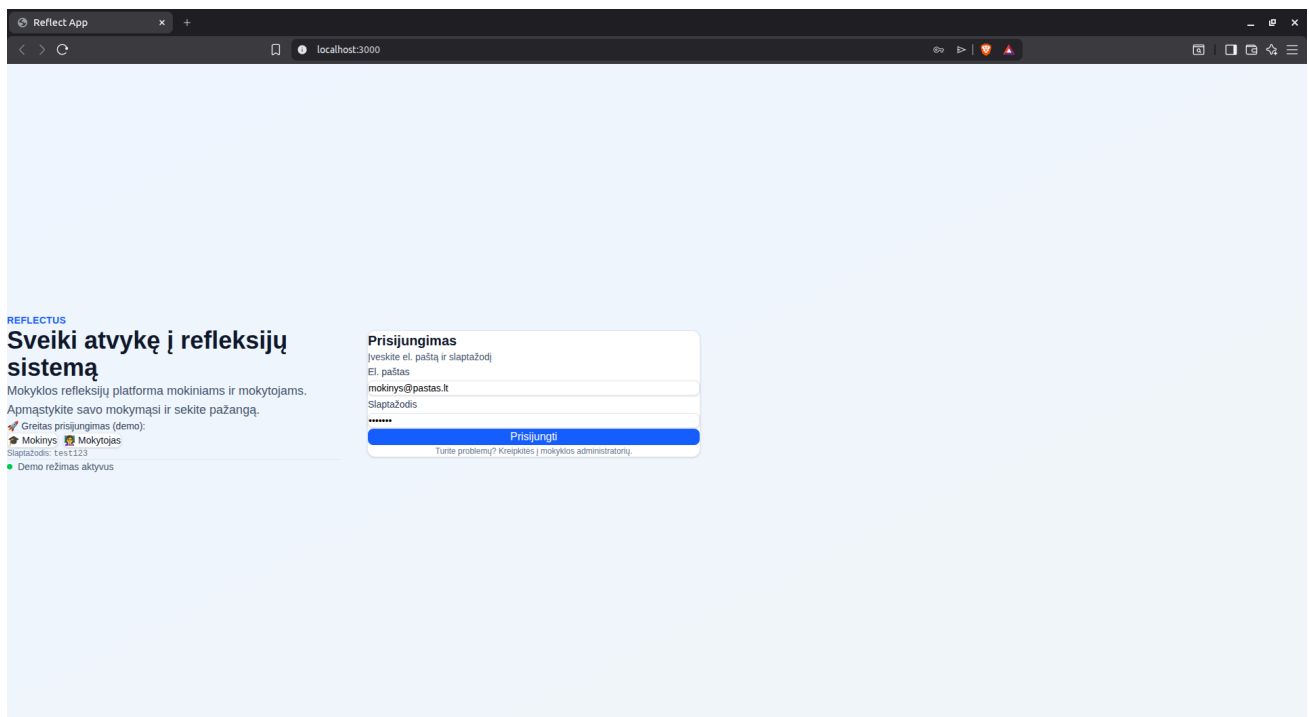
1) Globalus UI standartas: nieko prie krašto, viskas centre

... (toliau – dizainas ir klausimų patobulinimai)

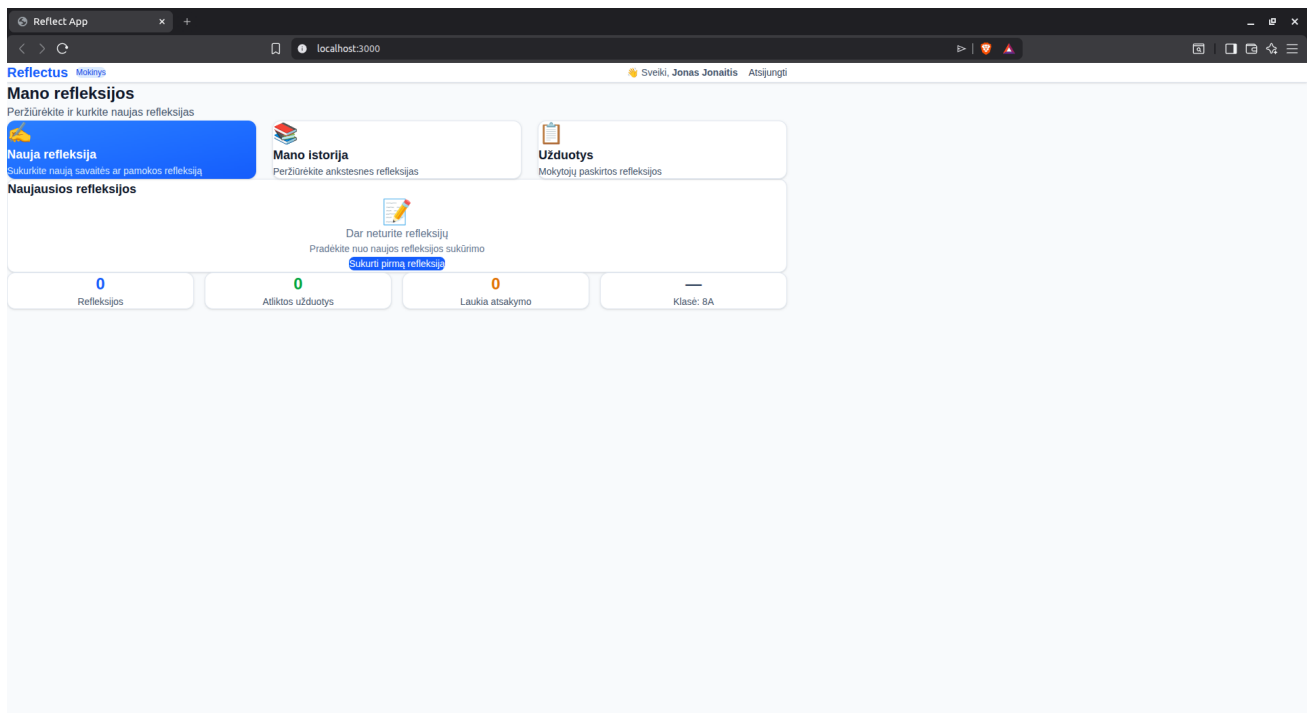
Priedas – pateikti ekrano vaizdai

Toliau pateikiami pokalbyje ikelti ekrano vaizdai, naudoti UI/UX problemoms iliustruoti.

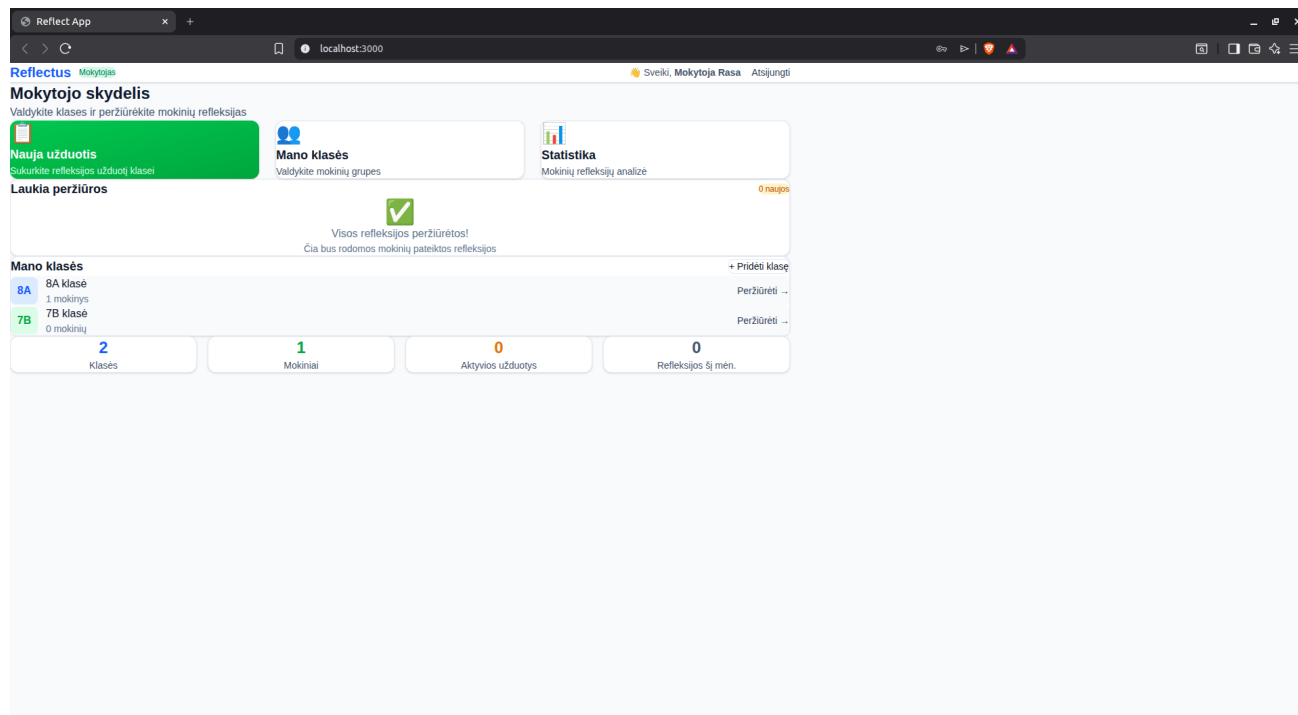
Screenshot 1: Login puslapis (demo režimas).



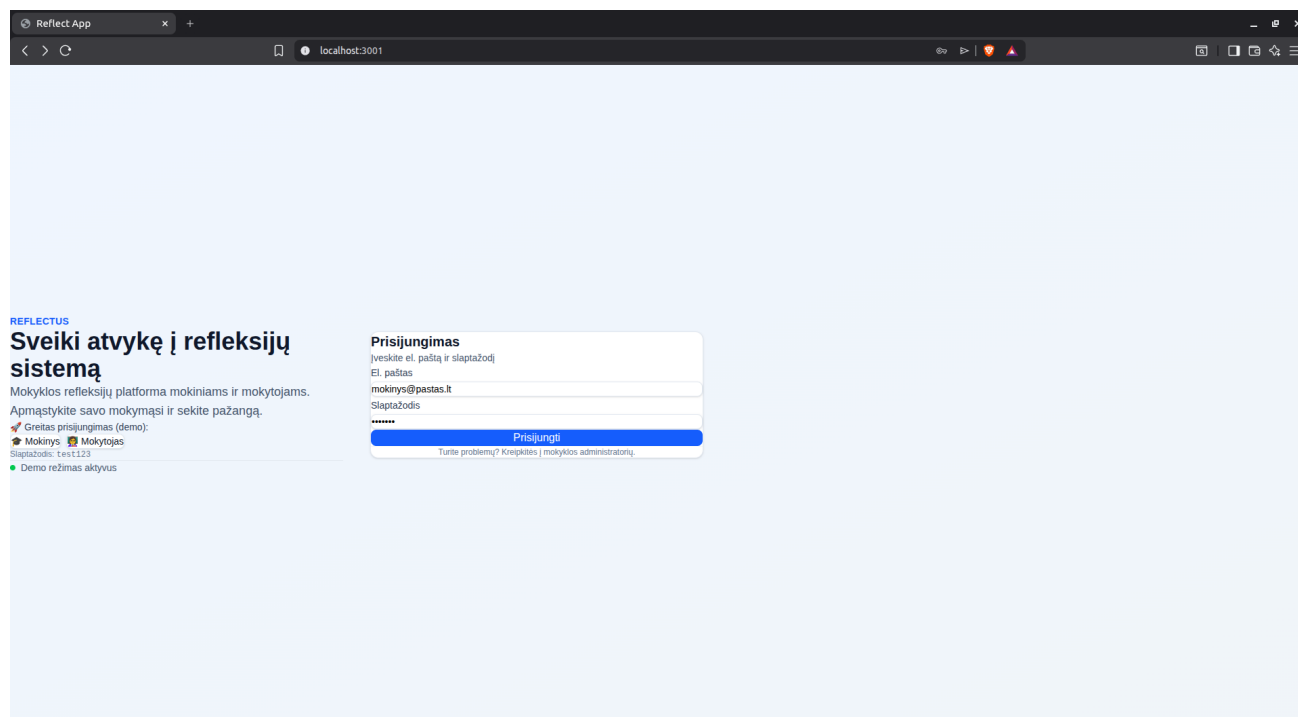
Screenshot 2: Mokinio skydelis (dashboard).



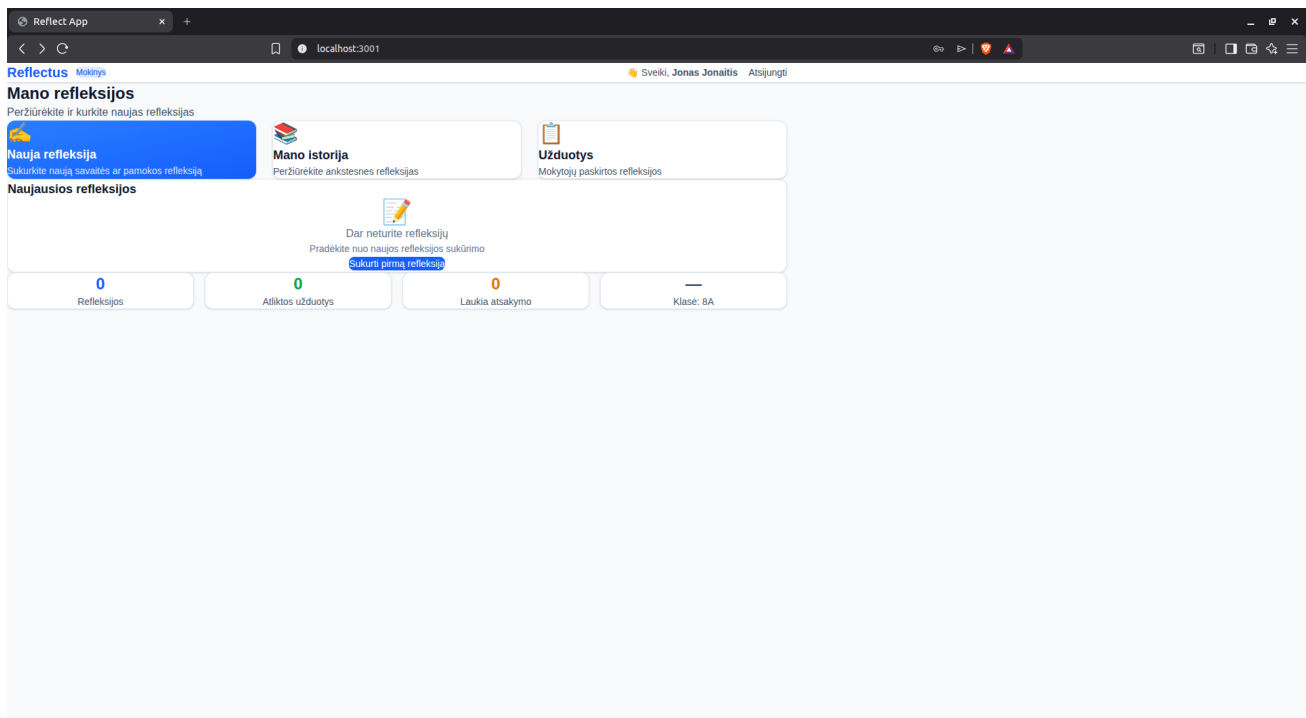
Screenshot 3: Mokytojo skydelis (dashboard).



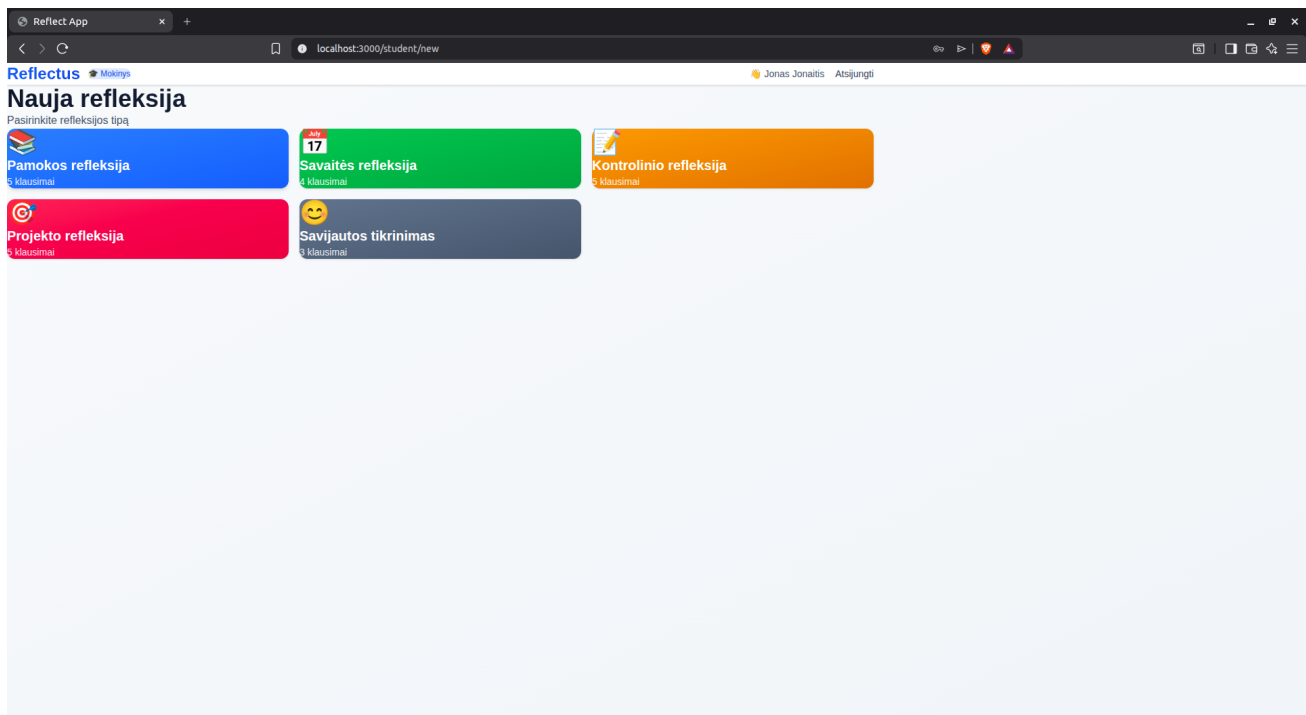
Screenshot 4: Login puslapio išdėstymas (problema: per daug tuščios vietos, elementai susigrūdę).



Screenshot 5: Mokinio skydelis – pagrindiniai mygtukai ir naujausios refleksijos (problema: dalis mygtukų neveikia).



Screenshot 6: Refleksijos tipo pasirinkimas (student/new).



Screenshot 7: Pamokos refleksija forma.

The screenshot shows a web browser window with the Reflectus application. The page title is 'Pamokos refleksija' (Lesson Reflection). The URL is 'localhost:3000/student/new/lesson'. The form contains the following fields:

- Dalykas*** (Subject): A text input field with a placeholder 'Įveskite dalykas...'. Below it is a label 'Tema*' (Topic).
- Tema*** (Topic): A text input field with a placeholder 'Įveskite temą...'. Below it is a label 'Ką išmokau?' (What did I learn?).
- Ką išmokau?** (What did I learn?): A text input field with a placeholder 'Įveskite ką išmokau?...'.
- Kas buvo sudėtinga?** (What was difficult?): A text input field with a placeholder 'Įveskite kas buvo sudėtinga?...'.
- Kokie klausimai liko?** (What questions remain?): A text input field with a placeholder 'Įveskite kokie klausimai liko?...'.

At the bottom of the form, there is a blue button labeled 'Pateikti refleksiją' (Submit reflection) and a link 'Atšaukti' (Cancel).

Screenshot 8: Savaitės refleksija forma.

The screenshot shows a web browser window with the Reflectus application. The page title is 'Savaitės refleksija' (Weekly Reflection). The URL is 'localhost:3000/student/new/week'. The form contains the following fields:

- Savaitės pasiekimai*** (Weekly achievements): A text input field with a placeholder 'Įveskite savaitės pasiekimai...'.
- Sunkumai** (Difficulties): A text input field with a placeholder 'Įveskite sunkumai...'.
- Tikslai kitai savaitei*** (Goals for next week): A text input field with a placeholder 'Įveskite tikslai kitai savaitei...'.
- Nuotaika (1-5)*** (Mood): A text input field with a placeholder '1 - 5'.

At the bottom of the form, there is a blue button labeled 'Pateikti refleksiją' (Submit reflection) and a link 'Atšaukti' (Cancel).

Screenshot 9: Kontrolinio refleksija forma.

The screenshot shows a web browser window with the URL `localhost:3000/student/new/test`. The page title is "Reflectus" with a "Mokymai" link. The main heading is "Kontrolinio refleksija". Below it, the instruction "Užpildykite visus laukus" is followed by four text input fields with labels: "Dalykas*", "Kaip nuošliauši?*", "Kas sekėsi?", and "Kas nesisekė?". Each field has a placeholder text starting with "Įveskite...". At the bottom, there is a blue button labeled "Pateikti refleksiją" and a link "Atšaukti".

Screenshot 10: Projekto refleksija forma.

The screenshot shows a web browser window with the URL `localhost:3000/student/new/project`. The page title is "Reflectus" with a "Mokymai" link. The main heading is "Projekto refleksija". Below it, the instruction "Užpildykite visus laukus" is followed by five text input fields with labels: "Projekto pavadinimas*", "Mano vaidmuo*", "Ko išmokau?*", "Kaip sekėsi komandinis darbas?", and "Ką daryt kitą kartą kitaip?". Each field has a placeholder text starting with "Įveskite...". At the bottom, there is a blue button labeled "Pateikti refleksiją" and a link "Atšaukti".

Screenshot 11: Savijautos tikrinimas forma.

The screenshot shows a web browser window with the title 'Reflectus App' and the address bar displaying 'localhost:3000/student/new/mood'. The page header includes the 'Reflectus' logo, a 'Mokymai' link, and a user profile for 'Jonas Jonaitis' with an 'Atsijungti' (Logout) button. The main heading is 'Savijautos tikrinimas' (Self-reflection), followed by the instruction 'Užpildykite visus laukus' (Fill in all fields). The form contains four input fields: 'Nuotaika (1-5)*' (Mood), 'Energijos lygis (1-5)*' (Energy level), 'Mintys' (Thoughts), and a larger text area for 'Įveskite mintys...' (Enter thoughts...). At the bottom of the form is a blue button labeled 'Pateikti refleksiją' (Submit reflection). Below the form, the text 'Atšaukti' (Cancel) is visible.