



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

Laboratorinių darbų ataskaita

Informacinių sistemų kūrimas karkasais

Darbą atliko:
Dangis Kažukauskas IFF-9/8

Darbą priėmė:
Mantas Bataitis

KAUNAS, 2022

Turinys

1.	Svetainės aprašymas	3
2.	Duomenų bazės schema su laukų paaiškinimais.....	3
2.1.	Duomenų bazės schema	3
2.2.	Lentelių aprašymai	4
2.3.	Lentelių duomenų pavyzdžiai.....	4
3.	Serverio API paslaugos modeliai, valdikliai, bibliotekos.....	5
3.1.	Modeliai	5
3.2.	Valdikliai	6
4.	Naudotojo sąsajos realizacija.	8
4.1.	Realizacijos struktūra ir funkcijos.....	8
4.3.	Bibliotekos.....	10
5.	Išvados.....	10

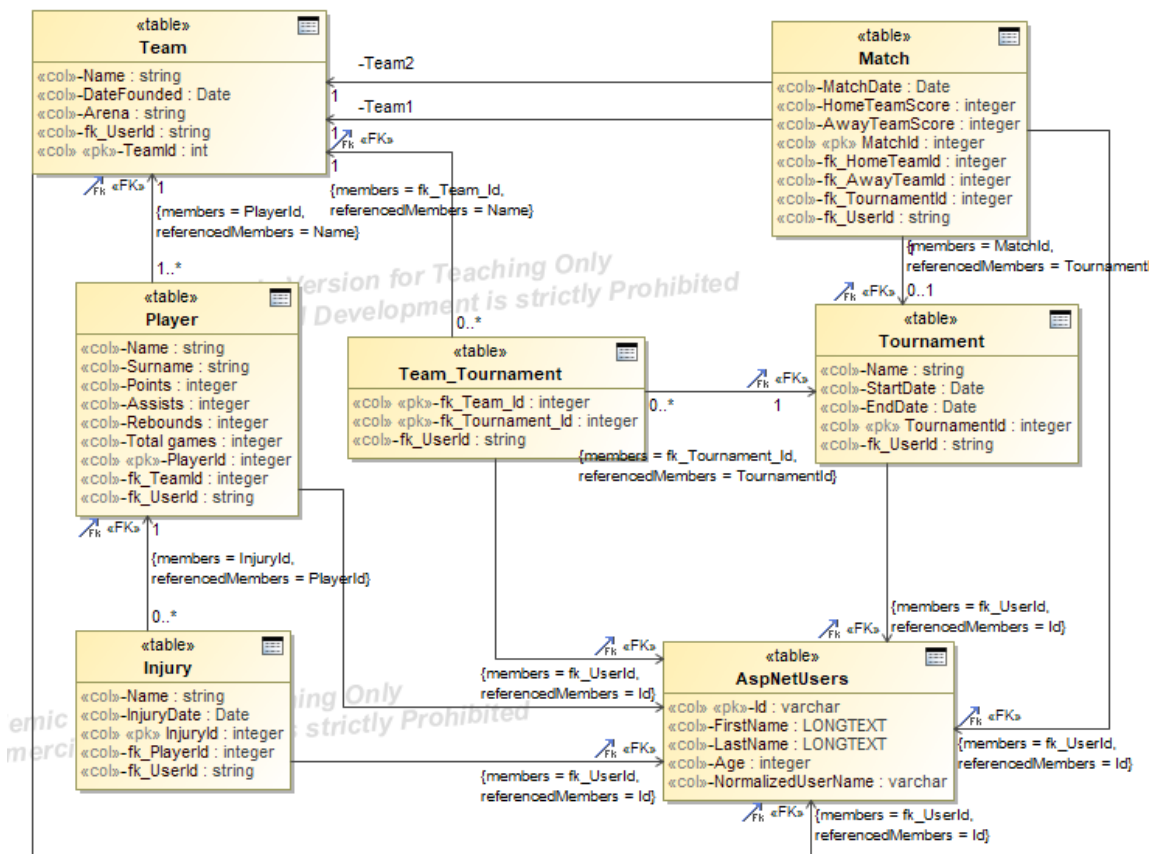
1. Svetainės aprašymas

Svetainė yra apie krepšinį. Norint naudotis svetaine, reikia užsiregistruoti. Yra du naudotojų tipai: vartotojas ir administratorius. Vartotojai, naudodamiesi svetaine, mato tik savo sukurtus duomenis, juos gali matyti, kurti, keisti bei šalinti. Administratorius gali naviguoti svetaine taip pat, kaip ir kiti naudotojai, tačiau jam informacija visa prieinama, taip pat gali viską kurti, keisti, šalinti bei visus duomenis matyti jam skirtoje „Secret!“ sekcijoje.

Funkcionalumas: susikūrus anketą, galima eiti į dvi pagrindines skiltis: komandas ir turnyrus. Kiekviena komanda susideda iš pavadinimo, arenos ir įkūrimo datos. Į ją įeina žaidėjai, turintys vardus, pavardes, surinktų taškų, rezultatyvių perdavimų, atkovotų kamuolių ir žaistų rungtynių kiekį. Taip pat sistemoje saugojama informacija apie jų traumų istoriją, kurią galima pridėti. Toliau – turnyrai. Jiems sukurti reikia pavadinimo, turnyro pradžios ir pabaigos laikų. Į turnyrus įeina rungtynės, kuriose dalyvauja dvi komandos, todėl jau reikia būti sukūrus komandas, norint sukurti rungtynes. Kiekvienos rungtynės turi rezultatą, areną, kurioje žaista (šeimininkų arena) bei datą.

2. Duomenų bazės schema su laukų paaiškinimais

2.1. Duomenų bazės schema



2.2. Lentelių aprašymai

Injury lentelė: Name laukas reiškia traumos pavadinimą, Date – kada įvyko trauma, privatus raktas Id bei išorinis raktas į Player lentelę.

Player lentelė: Name ir Surname laukai reiškia žaidėjo vardą ir pavardę atitinkamai, toliau – žaidėjo statistika: Points (taškai), Assists (rezultatyvūs perdavimai), Rebounds (atkovoti kamuoliai), Total games (kiek iš viso rungtynių sužaista), privatus raktas Id bei išorinis raktas į Team lentelę.

Team lentelė: komanda susideda iš: Name (pavadinimo), DateFounded (įkūrimo datos) bei Arena (arenos pavadinimo).

Match lentelė: rungtynės susideda iš Date (rungtynių datos), HomeTeamScore (kiek įmetė taškų namų komanda), AwayTeamScore (kiek įmetė taškų svečių komanda), turi privatų raktą Id, išorinį raktą į Tournament lentelę bei du išorinius raktus į Team lentelę, nes vienoje rungtynėje būtinai turi dalyvauti dvi komandos.

Tournament lentelė: norint užregistruoti turnyrą, reikia suteikti jam Name (pavadinimą), StartDate (turnyro pradžios datą), EndDate (turnyro pabaigos datą) bei privatų raktą Id.

Team_Tournament lentelė: skirta ryšiui daug-daug pavaizduoti. Ji susideda iš turnyrų ir komandų.

AspNetUsers lentelė: vartotojų lentelė, paveldinti IdentityUser klasę. Kiekviena lentelė turės savyje UserId, kuris padės atsekti, kam priklauso resursas.

Lentelėje esantys duomenų tipai yra nurodyti.

2.3. Lentelių duomenų pavyzdžiai

Injury lentelė:

	InjuryId	Name	InjuryDate	PlayerId	UserId
	4	Trauma is admins	12/23/2022 3:2...	2	22024ccc-1996-...
🗑️	NULL	NULL	NULL	NULL	NULL

Player lentelė:

	PlayerId	Name	Surname	Points	Assists	Rebounds	TotalGames	TeamId	UserId
▶	1	Vardas	Pavardė	243	22	54	7	1	d6fe0552-b31d...
	2	Vardas	Pavardas	66	25	20	8	1	d6fe0552-b31d...
	4	zaid	ejas	25	25	5	4	5	bee41713-3335...
	5	zaidejas	pavardis	25	51	22	6	5	bee41713-3335...
	7	adminas	pridejo	0	0	0	0	5	22024ccc-1996-...
⊕	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Team lentelė:

	TeamId	Name	Arena	DateFounded	UserId
▶	1	Useris1 Koman...	Useris1 Arenaa	12/27/2022 5:3...	d6fe0552-b31d...
	2	Useris1 2Koman...	Useris1 2Arenaa	12/9/2022 2:24...	d6fe0552-b31d...
	4	Useris2 Komanda	Useris2 Arena	12/21/2022 4:2...	bee41713-3335...
	5	Useris2 2Koman...	Useris2 2Arena	12/9/2022 3:23...	bee41713-3335...
⊗	NULL	NULL	NULL	NULL	NULL

Match lentelė:

	MatchId	HomeTeamSco...	AwayTeamScore	MatchDate	HomeTeamId	AwayTeamId	TournamentId	UserId
▶	1	50	40	12/29/2022 12:...	1	2	2	d6fe0552-b31d...
	3	221	144	12/29/2022 1:2...	2	1	2	d6fe0552-b31d...
	5	25	40	12/27/2022 10:...	4	5	3	bee41713-3335...
	6	102	90	12/30/2022 12:...	5	4	3	bee41713-3335...
⊗	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tournament lentelė:

	TournamentId	Name	StartDate	EndDate	UserId
▶	1	LKL	12/23/2022 3:2...	12/30/2022 3:2...	d6fe0552-b31d...
	2	KMT	12/2/2022 2:22...	12/14/2022 4:3...	d6fe0552-b31d...
	3	turnyras useris2	12/21/2022 3:2...	12/29/2022 3:2...	bee41713-3335...
⊗	NULL	NULL	NULL	NULL	NULL

Team_Tournament lentelė:

	TeamId	TournamentId	UserId
⊗	NULL	NULL	NULL

3. Serverio API paslaugos modeliai, valdikliai, bibliotekos.

3.1. Modeliai

Sukurti modeliai atitinka duomenų bazės struktūrą. Žemiau pateikiamas Player modelio paaiškinimas, kuris tinka apibūdinti kaip atrodo modeliai sistemoje.

Kiekvienam modeliui yra sukuriama DTO (Data Transfer Objects). Player atveju jie susideda iš modelio DTO (PlayerDTO), sukuriama modelio DTO (CreatePlayerDTO), redaguojamo modelio DTO (UpdateInjuryDTO).

Player modelis:

```
public class Player : IUserOwnedResource
```

```

{
    public int PlayerId { get; set; }
    public string? Name { get; set; }
    public string? Surname { get; set; }
    public int Points { get; set; }
    public int Assists { get; set; }
    public int Rebounds { get; set; }
    public int TotalGames { get; set; }
    public int? TeamId { get; set; }
    [ForeignKey("TeamId")]
    public Team? Team { get; set; }
    public string? UserId { get; set; }
    [ForeignKey("UserId")]
    public User? User { get; set; }
}

```

Matome, kad modelis atitinka struktūrą, atvaizduotą schemeje. Kad mainytis duomenimis ir neatskleisti per daug nereikalingos informacijos, kuriame DTOs:

PlayerDTO:

```

public record CreatePlayerDTO(string name, string surname, int points, int assists,
int rebounds, int totalGames);

```

CreatePlayerDTO:

```

public record UpdatePlayerDTO(string name, string surname, int points, int assists,
int rebounds, int totalGames);

```

UpdatePlayerDTO:

```

public record PlayerDTO(int id, string name, string surname, int points, int
assists, int rebounds, int totalGames, string teamName, int teamId, string username,
string userId);

```

Likę modeliai įgyvendinti panašiu stiliumi, todėl neverta visų aprašyti.

3.2. Valdikliai

Kiekvienam valdiklio metodui yra uždėta autorizacija pagal roles. Vartotojas gali atlikti veiksmus tik su savo resursais, administratorius veiksmus gali atlikti su bet kieno resursais.

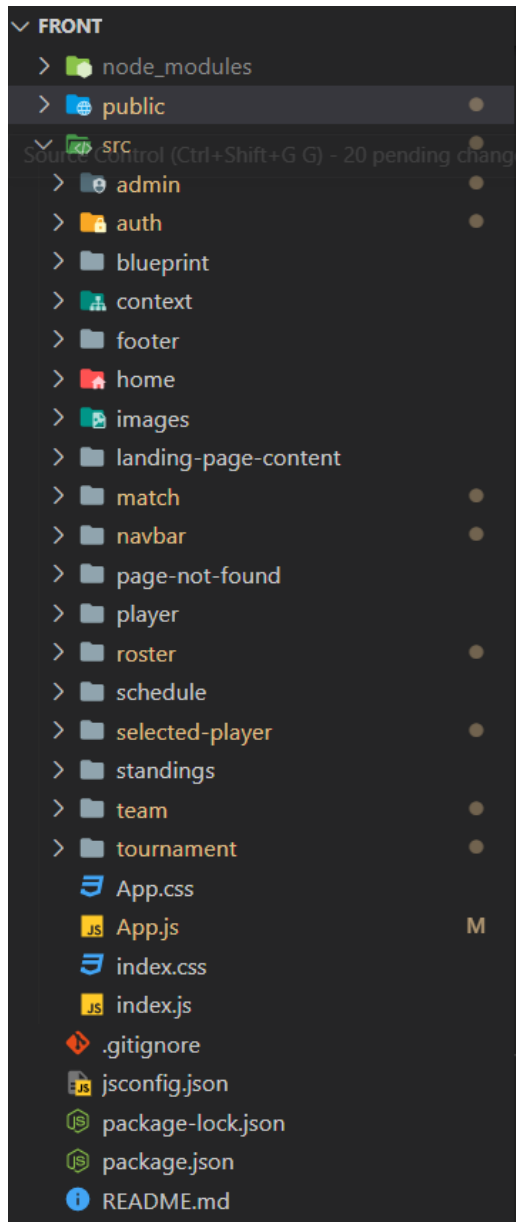
- AuthController:
 - Registracijos funkcija,
 - Prisijungimo funkcija, grąžinant vartotojui JWT.
- InjuriesController:
 - Įgyvendintas traumų sąrašo gavimas,
 - Vienos traumos gavimas,
 - Traumos kūrimas,
 - Traumos redagavimas,
 - Traumos ištrinimas. Toliau šios operacijos bus vadinamos CRUD.
- InjuriesDbListController:
 - Traumų sąrašo gavimas. Nereikia žaidėjo Id, kuris yra privalomas InjuriesController, norint atlikti veiksmus su traumomis.
- MatchesController:
 - Įgyvendintas CRUD.
- MatchesDbListController:

- Rungtynių sąrašo gavimas. Nereikia nurodyti turnyro Id, kuris yra privalomas MatchesController, norint atlikti veiksmus su rungtynėmis.
- PlayersController:
 - Įgyvendintas CRUD.
- TeamsController:
 - Įgyvendintas CRUD.
- TeamTournamentsController:
 - Įgyvendintas CRUD.
- TeamTournamentsDbListController:
 - Įgyvendintas komandų turnyrų sąrašo gavimas. Nereikia nurodyti turnyro Id, kuris yra privalomas TeamTournamentsController, norint atlikti veiksmus.
- TournamentsController:
 - Įgyvendintas CRUD.

4. Naudotojo sąsajos realizacija.

4.1. Realizacijos struktūra ir funkcijos

Failų struktūra:



Žvelgiant iš viršaus į apačią: admin aplankale yra tik administratoriui pasiekiami duomenys, atvaizduoti lentelėse. Toliau – auth aplankalas, kuriame aprašyta prisijungimo ir registracijos logika, context aplankalas skirtas suteikti aplikacijai kontekstą, šiuo atveju – vartotojo informaciją iš JWT.

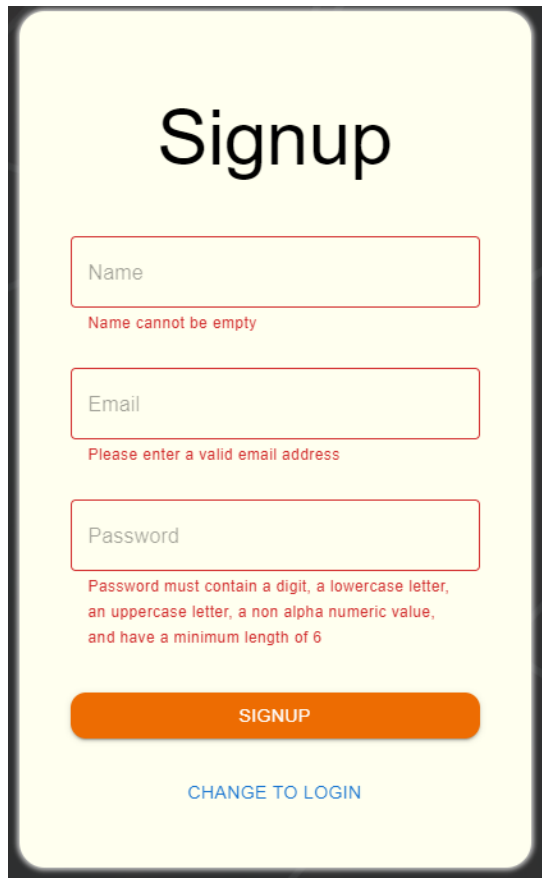
Likusiuose aplankaluose aprašyti komponentai, naudojami visoje aplikacijoje.

4.2. Duomenų įvedimo formos

Duomenys tikrinami kliento pusėje. Prisijungiant ir registruojantis naudojamos validacijos pagal IdentityUser klasės reikalavimus. Įvedant ar keičiant duomenis, *submit* mygtukas nėra aktyvus ir neišveina jo

paspausti, iki kol duomenys nebus tinkamai supildyti. Taip pat, norint sukurti turnyrą, reikia turnyro pradžios ir pabaigos datų, neleidžiame išsiųsti užklauso, jei pradžios data yra vėlesnė nei pabaigos data.

Registracijos validacijos, kai nei vienas laukas nėra teisingas:



A mobile app interface for a 'Signup' screen. The title 'Signup' is at the top. Below it are three input fields: 'Name', 'Email', and 'Password'. Each field has a red border and a red error message below it. The 'Name' field error is 'Name cannot be empty'. The 'Email' field error is 'Please enter a valid email address'. The 'Password' field error is 'Password must contain a digit, a lowercase letter, an uppercase letter, a non alpha numeric value, and have a minimum length of 6'. At the bottom is an orange 'SIGNUP' button and a blue 'CHANGE TO LOGIN' link.

Signup

Name

Name cannot be empty

Email

Please enter a valid email address

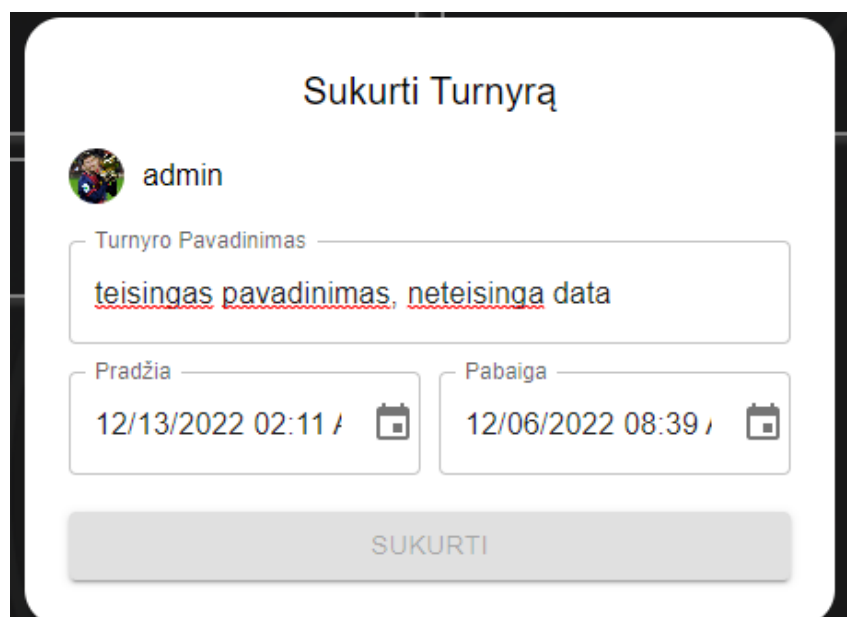
Password

Password must contain a digit, a lowercase letter, an uppercase letter, a non alpha numeric value, and have a minimum length of 6

SIGNUP

CHANGE TO LOGIN

Formos validacija, kai kažkuris iš laukų užpildytas neteisingai (stebėti *submit* mygtuką):



A mobile app interface for a 'Sukurti Turnyrą' (Create Tournament) screen. The title 'Sukurti Turnyrą' is at the top. Below it is a user profile section with a circular avatar and the text 'admin'. There are two input fields: 'Turnyro Pavadinimas' (Tournament Name) and 'Pradžia' (Start). The 'Turnyro Pavadinimas' field has a red border and a red error message below it: 'teisingas pavadinimas, neteisinga data'. The 'Pradžia' field has a red border and a red error message below it: '12/13/2022 02:11 /'. There is also a 'Pabaiga' (End) field with a red border and a red error message below it: '12/06/2022 08:39 /'. At the bottom is a grey 'SUKURTI' button.

Sukurti Turnyrą

admin

Turnyro Pavadinimas

teisingas pavadinimas, neteisinga data

Pradžia

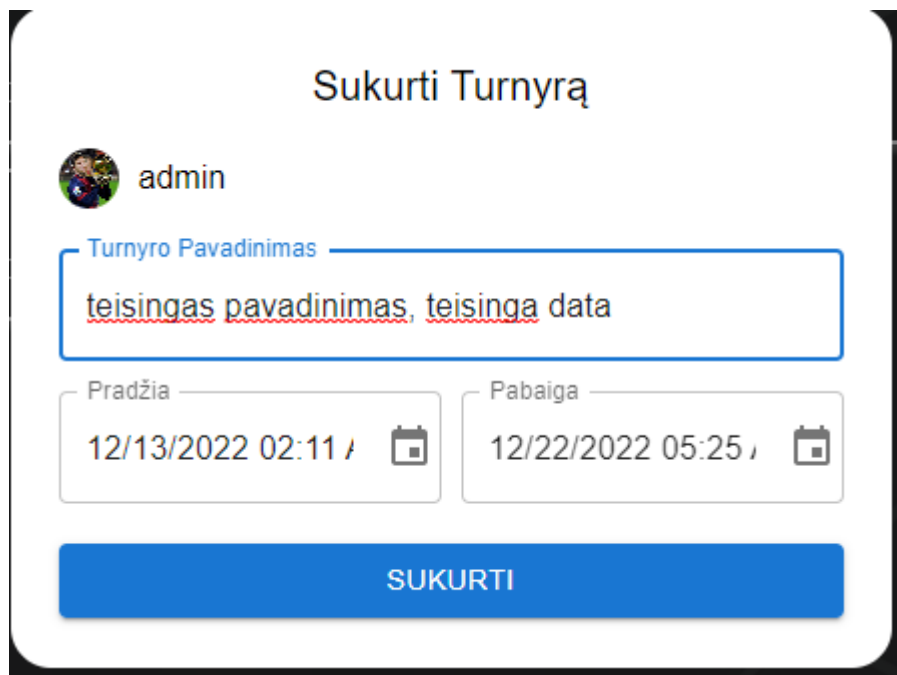
12/13/2022 02:11 /

Pabaiga

12/06/2022 08:39 /

SUKURTI

Formos validacija, kai laukai užpildyti teisingai (stebėti *submit* mygtuką):



The screenshot shows a web form titled "Sukurti Turnyrą" (Create Tournament). At the top, there is a user profile section with a circular avatar icon and the text "admin". Below this is a form with the following fields:

- A text input field labeled "Turnyro Pavadinimas" (Tournament Name) containing the text "teisingas pavadinimas, teisinga data". The text is underlined with a red dashed line, indicating successful validation.
- A date and time input field labeled "Pradžia" (Start) containing the value "12/13/2022 02:11" and a calendar icon.
- A date and time input field labeled "Pabaiga" (End) containing the value "12/22/2022 05:25" and a calendar icon.

At the bottom of the form is a large blue button labeled "SUKURTI" (CREATE).

4.3. Bibliotekos

Svetainė pagrįsta kurta naudojantis MUI React komponentų biblioteką. Navigacijai pasirinkau react-router-dom.

5. Išvados

Šio laboratorinio metu išmokau daug: ne tik kurti ir modifikuoti savo rašytą API naudojant .NET Core karkasą, bet ir parašyti kliento dalį, naudojantis React pagalba. Iki šio laboratorinio nebuvo tekę kurti API. Jį kurti buvo gan nesunku, ypač, kai kontrolieriai sugeneruojami, belieka juos tik pataisyti taip, kad veiktų pagal svetainės logiką.

Prie svetainės pagražinimo ir funkcionalumo prisidėjo MUI (anksčiau vadinta Material UI): React komponentų biblioteka, kurioje yra nemažai komponentų pasirinkimo: nuo paprasčiausių mygtukų, iki puslapio išdėstymo įrankių ar piktogramų. Ją naudoti buvo smagu, ypač, kai įgundi ieškodamas informacijos dokumentacijoje.