

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Semestro projektas (PR00B121)

Dėstytojas: Eimutis Karčiauskas

Darbą parengė: Robertas Abromavičius

KAUNAS 2019

Turinys

Temos pasirinkimas.....	3
Darbo pavadinimas	3
Informacinė sistema.....	3
Darbo aprašymas	3
Darbo analizė	3
Darbo įrankiai.....	3
Pavyzdinės sistemos.....	3
Pavyzdinės sistemos privalumai.....	3
Pavyzdinės sistemos trūkumai	4
Individualus projektas.....	4
Sistemos veikimo principas.....	5
Registracija	5
Prisijungimas	6
Neapsaugotų puslapių peržiūra	6
Apsaugotų puslapių peržiūra	6
Veikimas iš vartotojo perspektyvos	6
Namų puslapis.....	6
Žaidimų puslapis (angl. Games page).....	6
Naujienų puslapis.....	6
Susirasti žaidimų draugą puslapis (angl. Find a playmate)	6
Profilio puslapis.....	7
Sistemos ekrano kopijos ir paaiškinimai	7
Sistemos navigacija	7
Žaidimų puslapis	7
Naujienų puslapis.....	8
Susirasti žaidimų draugą puslapis	9
Profilio puslapis.....	10

Temos pasirinkimas

Darbo pavadinimas

Informacinė sistema

Darbo aprašymas

Kuriamas informacinis puslapis kuris susideda iš esamų žaidimų ypatumų, tokių kaip PUBG, Apex Legends, Hearthstone. Jame bus talpinama įdomi ir svarbi informacija apie tuos žaidimus. Nepaisant tik informacijos bus galimybė susisiekti su kitais puslapio lankytojai, taip susirasti virtualią komandą ar varžovą (panašių sugebėjimų) jeigu yra poreikis. Bus galimybė pasidaryti mini turnyrą tuose žaidimuose kuriuose yra galimybė.

Darbo analizė

Darbo įrankiai

Projektas susideda iš dviejų dalių, vartotojo sąsajos (front-end) ir serverio f-ijų (back-end) programavimo, atitinkamai React.js (Javascript) ir Laravel (php) karkasų.

Pasirinkimui turėjo įtaka karkasų patogumas, vienas jų – laiko taupymas, nes nemažai įprastų f-ijų: prisijungimas, registracija, validacija, turinio puslapiavimas, el. paštas ir pan., yra įgyvendintos ir telieka išmokti naudotis. Antras didelis plusas renkantis tokias technologijas yra, kad šie du karkasai gali būti visiškai atskirti vienas nuo kito, realybėje du ar daugiau žmonių gali dirbti kartu prie to pačio projekto nesiaiškinant kito kodo. Trečias aspektas, kuris mane patraukė yra šių įrankių reiklumas, populiarumas darbo rinkoje, todėl tai puiki pradžia.

Be šių dviejų įrankių žinoma yra naudojamas CSS3 (SASS mano atveju), ir HTML5.

Pavyzdinės sistemos

Vienas iš, galbūt, žinomiausių pavydžių tarp žmonių, kurie mėgsta žaidimus yra <https://www.faceit.com> sistema, iš tiesų šita sistema tokia plati jau funkcionalumo atveju, kad net vargu ar galima aplenkti.

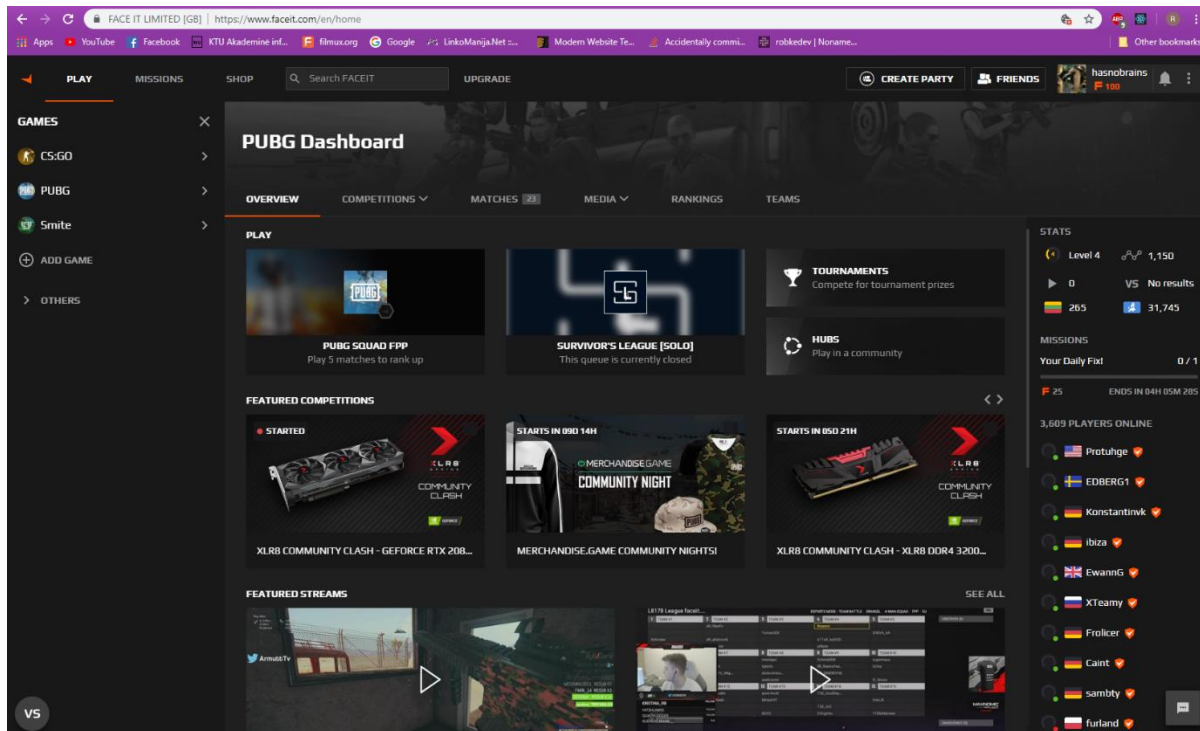
Pavyzdinės sistemos privalumai

Ši sistema turi galimybę sugrupuoti žmones įvairiuose žaidimuose, labai didelis privalumas, kad suteikia gerus serverius (128 tick) žaidimui, yra žaidėjų lygiai, pasiekimai, žaidimo statistikos, galimybė laimėti realius prizus, dieniniai ir savaitiniai turnyrai, kartais daromi išskirtiniai turnyrai su didesniais prizais, žaidėjų profiliai, draugų ryšiai. Todėl ši internetinė programa yra pavyzdys mano kuriamai programai.

Pavyzdinės sistemos trūkumai

Navigavimas puslapyje – dažnai sunku surasti norimą funkcionalumą, trūksta sistemoje aiškumo. Pirmą kartą atvykęs vartotojas sunkiai orientuojasi kas kur randasi.

Pavyzdinės sistemos ekrano kopija žemiau. (pav. 1)

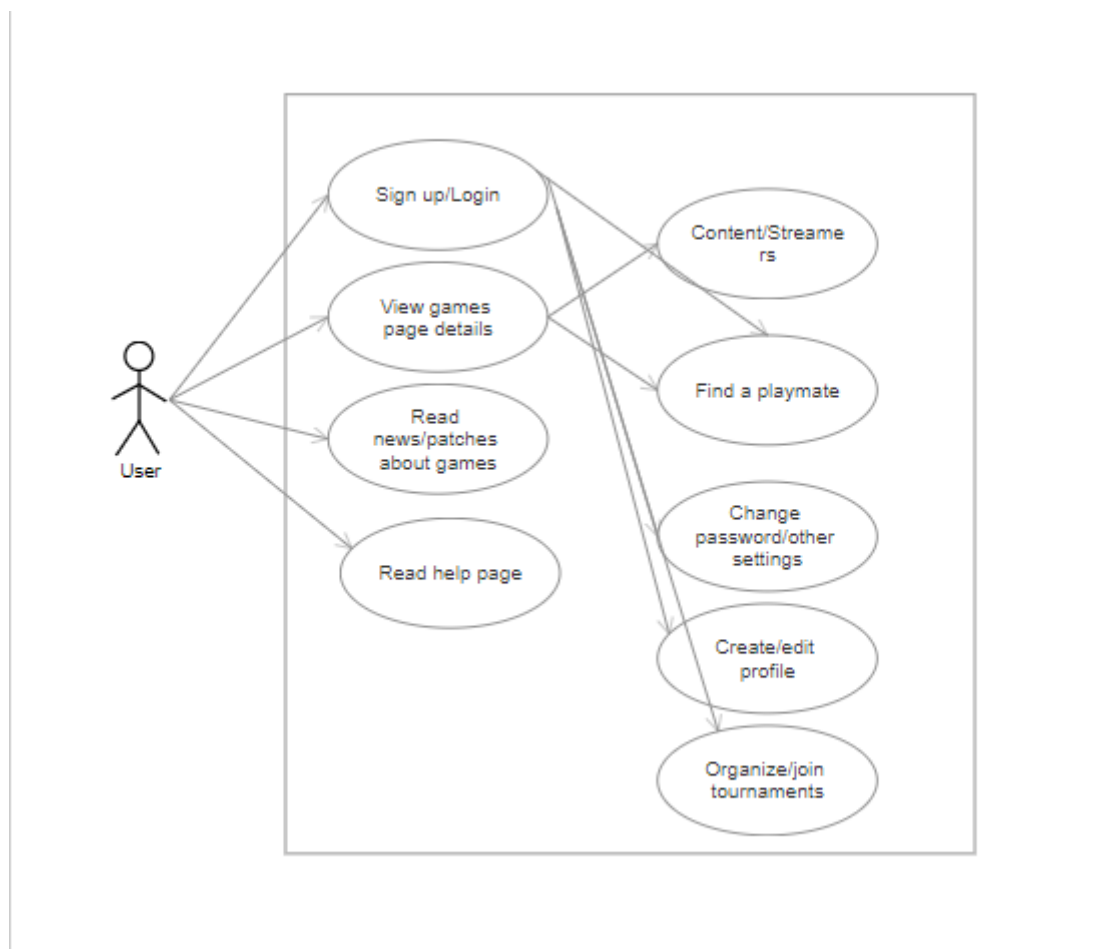


Pav 1. Pavyzdinės sistemos ekrano kopija

Kitų sistemų su tokiu pat ar bent jau panašiu funkcionalumu neradau.

Individualus projektas

Panaudojimo atvejų diagrama (pav. 1) padeda suprasti ką vartotojas gali daryti užėjęs į internetinę svetainę.



Pav. 2 Panaudojimo atvejų diagrama

Remiantis šia diagrama buvo suprogramuotos funkcijos kurios atvaizduoja tam tikrą informaciją.

Sistemos veikimo principas

Sistema suprogramuota iš dviejų skirtingų programavimo kalbų karkasų:

- „React.js“ (Javascript kalba)
- „Laravel“ karkaso (PHP kalba)

Visa grafinė sąsaja suprogramuota „React.js“ karkasu. Būtent dėl to ši sistema realiai turi tik vieną puslapį, niekada nereikia užkrauti kito failo, vietoje to kas kart yra keičiamas komponentas su bibliotekos pagalba „react-router-dom“. Keičiasi tik „URL“ puslapio adresas.

Registracija

Vartotojui įvedus reikiamus laukus yra siunčiama užklausa serveriui, kad vartotojas nori užsiregistruoti. Ši užklausa apdorojama su papildoma biblioteka „JWT authentication“, kuri vartotojui sugeneruoja ilgą žetoną (angl. token), įrašo duomenis į duomenų bazę jeigu tokio vartotojo dar nėra ir siunčia užklausa su žetonu atgal į „React.js“ su atsakymu ar pavyko/nepavyko. Sėkmingu užklauskos atveju vartotojui žetonas ir kiti jo duomenys be slaptažodžio yra išsaugomi lokaliaje naršyklės saugykloje ir vartotojas peradresuojamas į namų puslapį.

Prisijungimas

Jeigu vartotojas buvo prisijungęs jau prieš tai ir neatsijungė, atsidarius svetainę sistema patikrina ar yra vartotojo duomenys lokaliaje naršyklėje ir jeigu yra tuomet vartotojas lieka prijungtas.

Vartotojas įveda savo el. paštą/vartotojo vardą ir slaptažodį ir spaudžia mygtuką prisijungti. Po mygtuko paspaudimo iškeliauja užklausa kartu su duomenimis į serverį, ir teisingų duomenų atveju siunčiama užklausa, kad vartotojas gali gauti prieigą prie apsaugotų resursų. Grįžę duomenys yra išsaugomi lokaliaje naršyklės saugykloje.

Neapsaugotų puslapių peržiūra

Vartotojas gali apsilankyti neapsaugotuose puslapiuose ir peržiūrėti informaciją ir neprisijungus.

Apsaugotų puslapių peržiūra

Vartotojui norint atsidaryti „Find a playmate“ arba profilio puslapius reikia būti prisijungus, kitu atveju sistema neįleidžia ir peradresuoja į prisijungimo puslapį.

Jeigu vartotojas yra prisijungęs ir nori pažiūrėti/keisti duomenis kartu yra siunčiamas ir žetonas per URL, kad serveris galėtų patikrinti ar toks žetonas yra duomenų bazėje. Atėjusi užklausa pirmiausiai keliauja per „JWT“ programos kodą ir tik žetonas yra tvarkingas sistema atlieka tolesnius veiksmus. Nesėkmės atveju sistema gražina užklausą su klaidos kodu.

Veikimas iš vartotojo perspektyvos

Atidarius puslapį vartotojas gali pažiūrėti tik dalį puslapių, tokių kaip namų, naujienų, žaidimų, pagalbos puslapius, tačiau norint susirasti bendraminčių ar prisijungti prie turnyrų vartotojas turi užsiregistruoti/prisijungti.

Namų puslapis

Šioje skiltyje vartotojas gali pasiekti greitai kitus puslapius prieš tai apžvelgiant trumpus aprašymus apie tuos puslapius.

Žaidimų puslapis (angl. Games page)

Vartotojas turi galimybę peržiūrėti šiuo metu populiariausius atitinkamų žaidimų žmones, kurie kelia vaizdo įrašus arba filmuoja tiesiogiai ir rodo per platformą <https://www.twitch.tv> (angl. streamers). Paskaityti apie juos bendrą informaciją ir jeigu jie šiuo metu gyvai transliuoja – pažiūrėti transliaciją.

Naujienų puslapis

Šį puslapį gali matyti visi, jame yra atvaizduojama informacija apie žaidimų naujienas, jų patobulinimus.

Susirasti žaidimų draugą puslapis (angl. Find a playmate)

Sistema leidžia vykdyti šio puslapio funkcijas tik tokiu atveju jeigu vartotojas yra prisijungęs, kitu atveju jis yra peradresuojamas į prisijungimo puslapį.

Jeigu vartotojas yra prisijungęs jis gali pasirinkti žaidimą pagal kurį nori ieškoti draugų ir spausti mygtuką kad surastų. Sistema tikrina kitų žaidėjų profilius ir atvaizduoja žaidėjus korteles, ant kurių paspaudus galima matyti daugiau informacijos su kuria galima pasikviesti žaidėją į tam tikros platformos draugus.

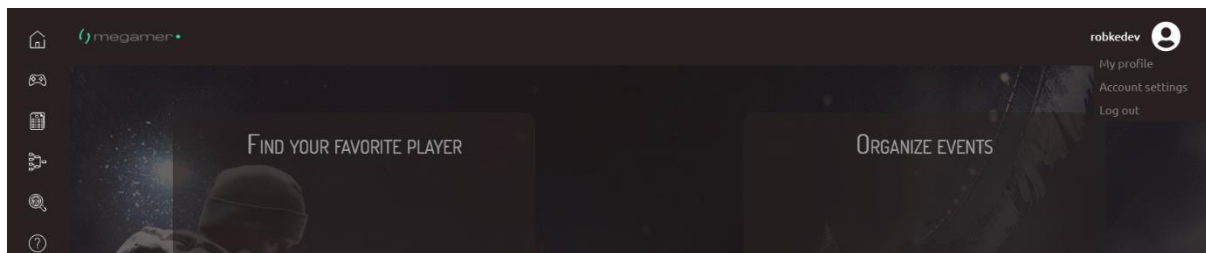
Profilio puslapis

Vartotojas gali peržiūrėti/keisti informaciją apie save, savo žaidimų pasiekimus, taip pat pridėti nuorodas, per kurias kiti žaidėjai gali pakviesti į draugus atitinkamoje žaidimų platformoje.

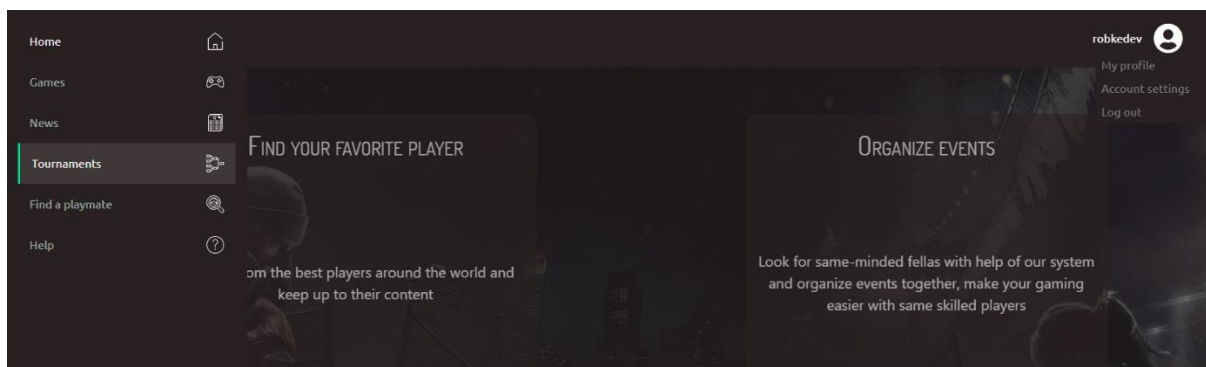
Sistemos ekrano kopijos ir paaiškinimai

Sistemos navigacija

Yra kompaktiška ir užima ne daug vietos, kairėje pusėje kur yra ikonos (2 ir 3 pav.)



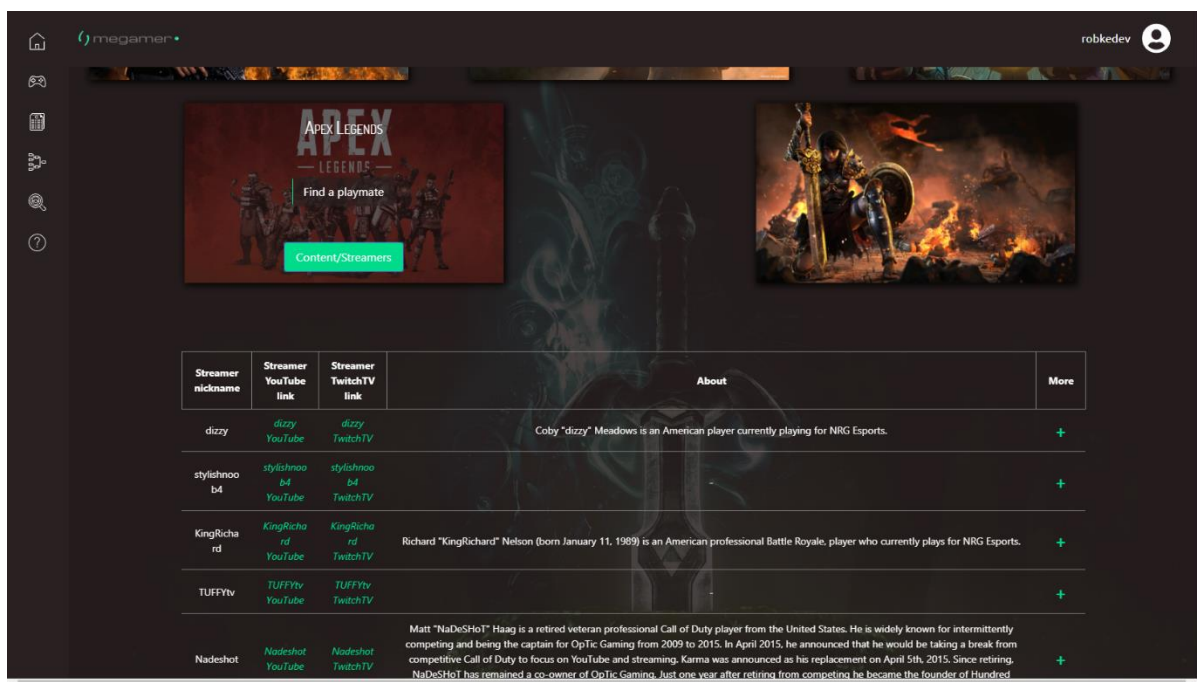
Pav. 3. Suskleista navigacija



Pav. 4. Išskleista navigacija

Žaidimų puslapis

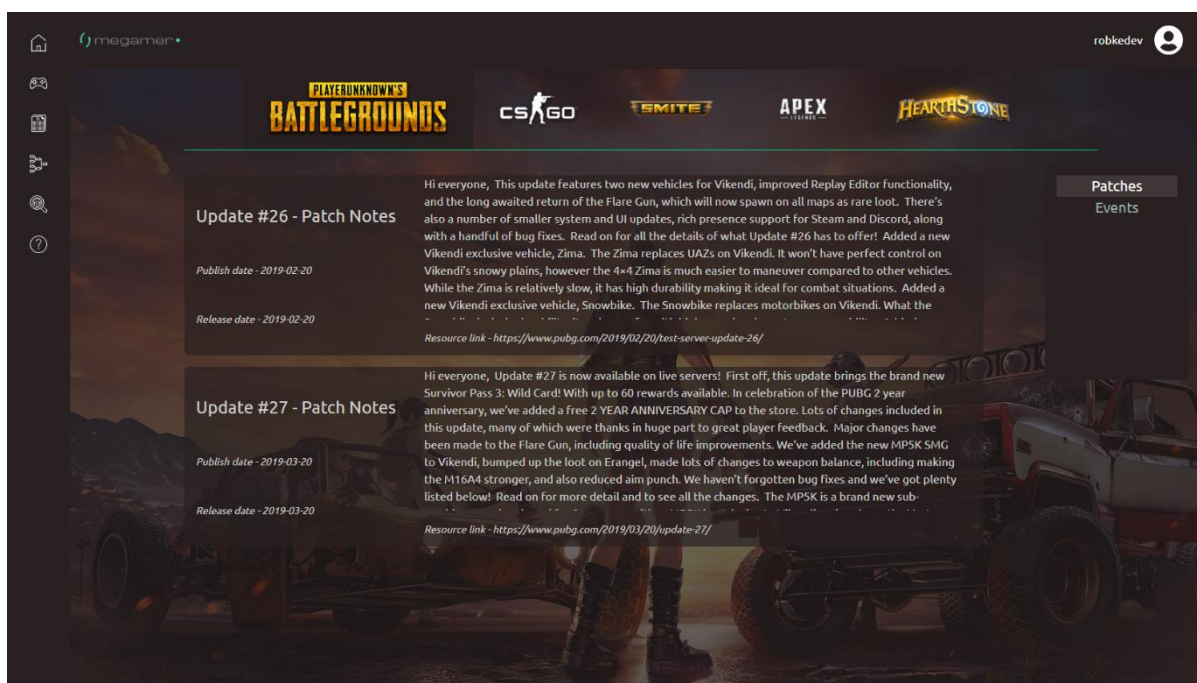
Komponentai užima ne daug vietos ir viskas yra interaktyvu (pav. 4).



Pav. 5. Žaidimų puslapis

Naujių puslapis

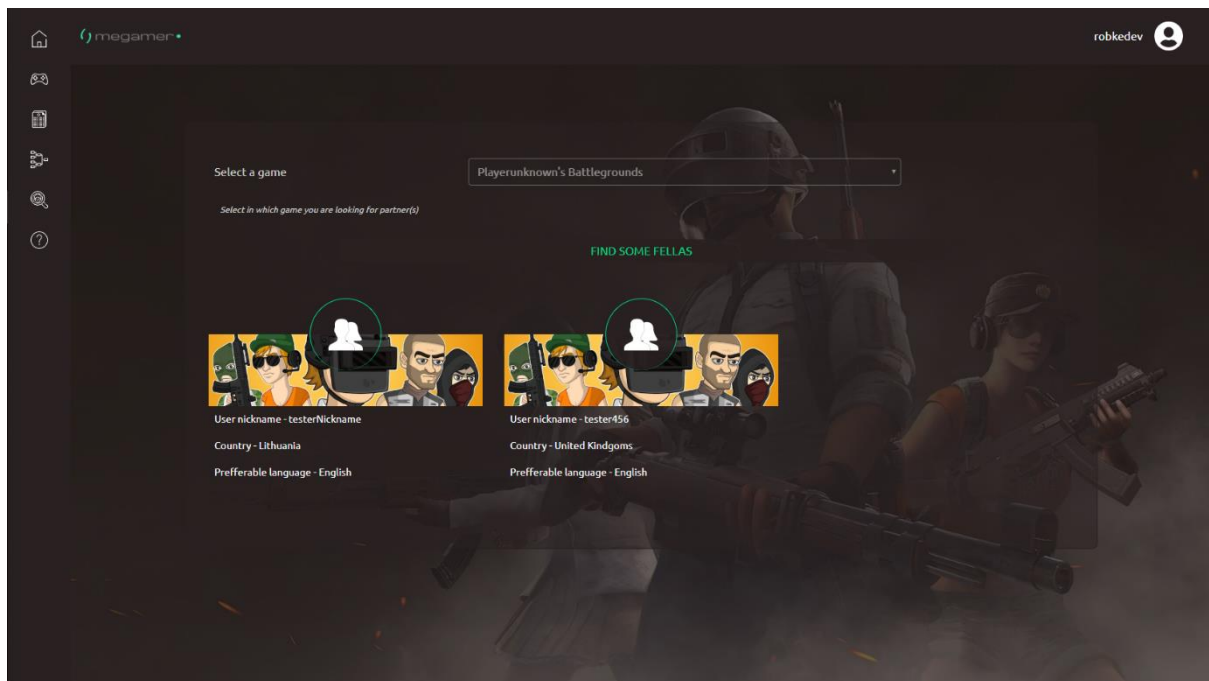
Naujių puslapyje vartotojas gali pasirinkti peržiūrėti informaciją apie žaidimo atnaujinimus arba ateinančius/buvusius renginius (pav. 5).



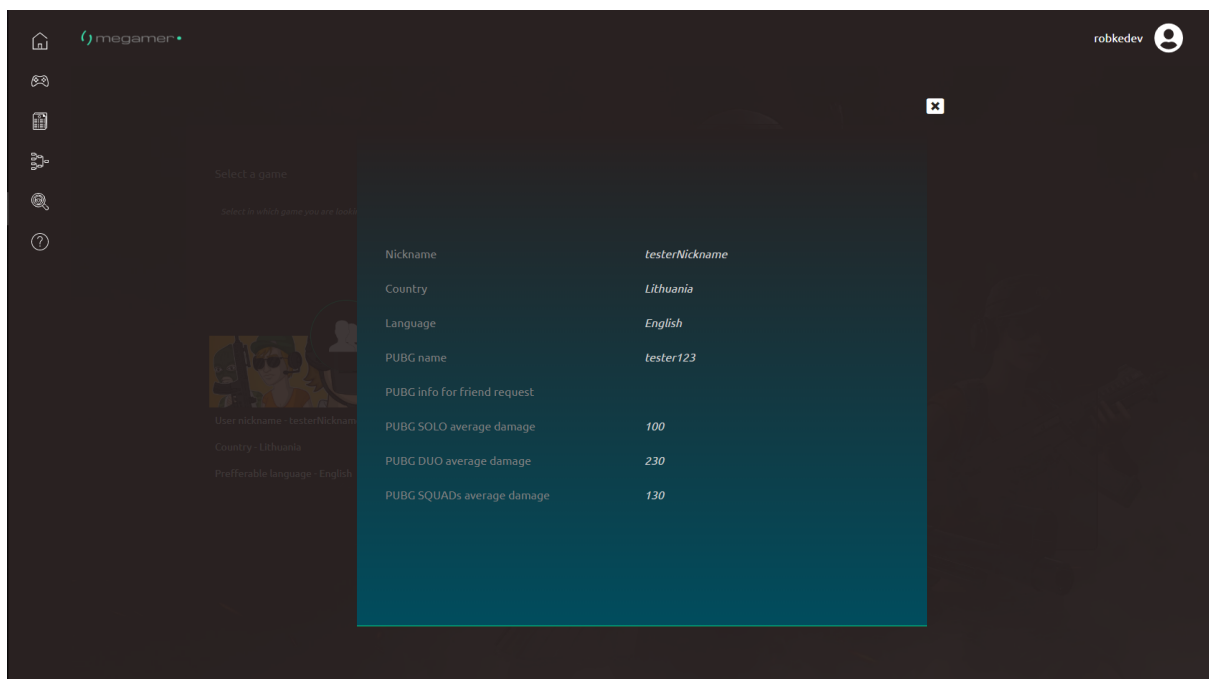
Pav. 6. Naujių puslapis

Susirasti žaidimų draugą puslapis

Šiame puslapyje vartotojas gali atlikti paiešką (pav. 6). Apie surastus žaidėjus galima pamatyti daugiau informacijos paspaudus ant žaidėjo kortelės (pav. 7).



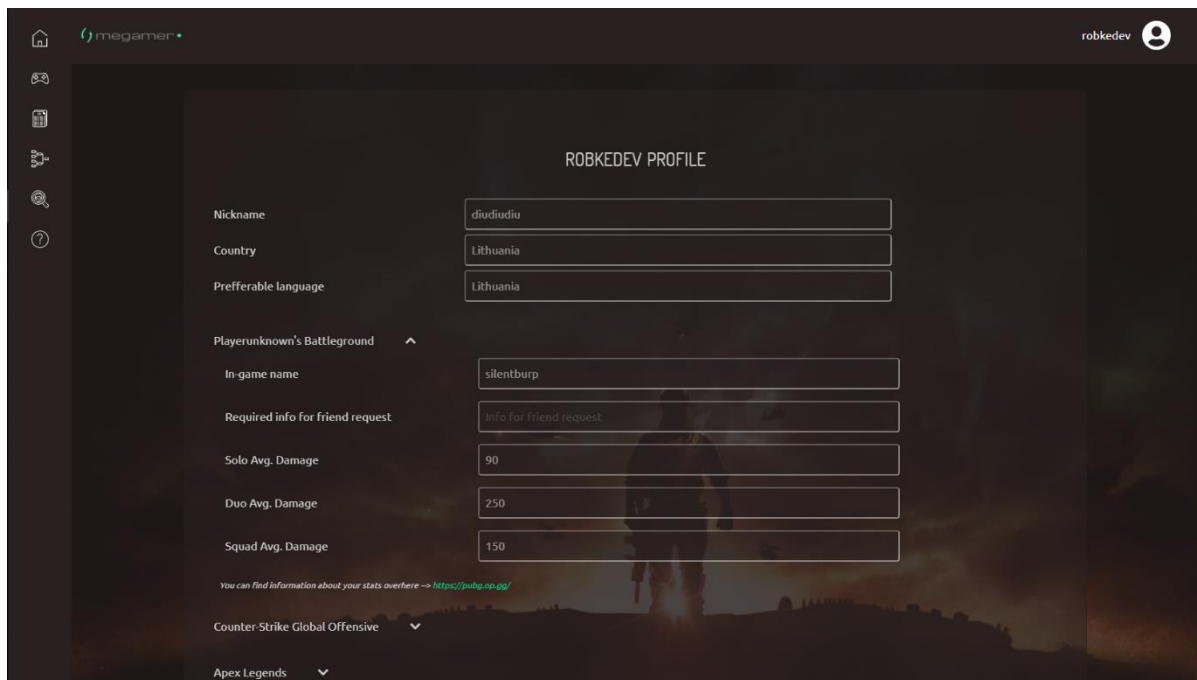
Pav. 7. "Find a playmate" puslapis



Pav 8. Daugiau informacijos apie vartotoją

Profilio puslapis

Šiame puslapyje vartotojas prideda/keičia informaciją apie save.

The screenshot shows a web application interface for a user profile. At the top, there's a navigation bar with a home icon, a 'megamer' logo, and a user profile icon labeled 'robkedev'. A sidebar on the left contains icons for home, friends, items, stats, and a help/question mark. The main content area is titled 'ROBKEDEV PROFILE' and contains a form with the following fields: Nickname (filled with 'diudiudi'), Country (filled with 'Lithuania'), Preferable language (filled with 'Lithuania'), Playerunknown's Battleground (expanded section with fields for In-game name 'silentburp', Required info for friend request, Solo Avg. Damage '90', Duo Avg. Damage '250', and Squad Avg. Damage '150'), Counter-Strike Global Offensive (collapsed), and Apex Legends (collapsed). A link at the bottom of the Playerunknown's Battleground section says 'You can find information about your stats over here -> https://pubg.gg/'.

Pav. 9 Profilio puslapis

Išvados

Kurti internetinę sistemą atsiejus „back-end“ ir „front-end“ turi savų plusų ir minusų.

Plusai

- Greitesnė navigacija tarp „puslapių“
- Puslapis interaktyvesnis negu tik su vienu įrankiu, pvz. „Laravel“.
- Gali dirbti skirtingi žmonės prie to pačio projekto visiškai paprastai netrukdam vieni kitiems.
- Lengvesnis ir aiškesnis sistemos palaikymas
- React.js turi „Virtual DOM“ (DOM - Document Object Model), kas leidžia išsaugoti pakeitimus virtualiai ir kai reikia išvesti kodą į ekraną realiai yra keičiama tik tai kas buvo perrašyta, tik ta dalis, nereikia perrašyti visos „DOM“ šakos.

Minusai

- Nėra tiesioginės prieigos prie serverio pavyzdžiui norint talpinti ir pasiimti nuotraukas, reikalaujama papildomų metodų
- Reikalauja daugiau konfigūracijų (tokių kaip „webpack“, „babel“ ir kt.) , kad viską apjungti.

Kurti API (angl. Application Programmable Interface) yra patogiu, nes visas darbas su modeliais lieka vienoje vietoje, o viskas kas susiję su grafine sąsaja – atskirai. Tą patį API galima panaudoti ir

programuojant mobilią aplikaciją. „Laravel“ karkasas labai palengvina darbą su API, vos keliomis eilutėmis kodo galima prisidėti reikiamus failus ir toliau programuoti mažiau galvojant apie struktūrą.

„React.js“ leidžia labai patogiai ir lengvai susitvarkyti su vykstančiais pakeitimais puslapyje, nes kuriant klases yra saugomos ir klasės būsenos, kurias keičiant „React.js“ per naujo išveda tos vietos kodą į tikrą „DOM“.

Galima teigti, kad šie įrankiai yra labai patogūs ir sutaupantys laiko, kur reikia programuoti interaktyvią ir greitą vartotojo sąsają.

Programos kodas

<https://github.com/loudburp/megamerweb>

Šaltiniai

Chunk, C., be datos *Free Code Camp*. [Tinkle]

Available at: <https://www.freecodecamp.org/news/these-are-the-concepts-you-should-know-in-react-js-after-you-learn-the-basics-ee1d2f4b8030/>

Laravel, be datos *Laravel docs*. [Tinkle]

Available at: <https://laravel.com/docs/5.8/>

Oni, G., 2019. *Medium*. [Tinkle]

Available at: <https://medium.com/@Gbxnga/token-based-authentication-with-react-and-laravel-restful-api-83f16581e85>

[Kreiptasi 2019].

React, be datos *React docs*. [Tinkle]

Available at: <https://reactjs.org/docs/getting-started.html>

Youtube, be datos *Traversy media*. [Tinkle]

Available at: <https://www.youtube.com/channel/UC29ju8bIPH5as8OGnQzwJyA>

[Kreiptasi 2019].