# PD2 kursā "Modernās programmēšanas tehnoloģijas" - LFL

## Risinājuma apraksts

## Izmantotās tehnoloģijas:

Uzdevuma risinājuma ietvaros tika izveidota tīmekļa vietne, izmantojot Sveltekit ietvaru 1.0.0., kas savukārt izmanto Vite koda sapakošanas rīku ("build tool") un ir balstīts uz Svelte programmēšanas valodu. Stilu veidošana tiek veikta, izmantojot Tailwind CSS pakotni.

Lai varētu palaist projektu, nepieciešams vispirms lejupielādēt NPM (vēlams jaunāko versiju, taču šī darba ietvaros tika izmantota 8.19.2. versija).

### Kā palaist programmatūru?

Informāciju par programmatūras palaišanu skatīt failā README.md (galvenajā mapē).

Vispārīgas instrukcijas no Sveltekit izstrādātāju puses ir norādītas failā SVELTEKIT INSTRUCTIONS.md (galvenajā mapē).

#### Programmatūras darbības princips:

NB! Programmatūras darbība ir parādīta arī ekrāna ieraksta failā "projekta\_demo.mp4" (galvenajā mapē), savukārt visi lietotāja saskarnes ekrānšāviņi ir parādīti mapē "screenshots" (šajā mapē ir arī mape "bez\_augsupieladetiem\_maciem", kurā ir parādīts, kā izskatās lietotāja saskarnes tad, kad vēl neviens mačs nav augšupielādēts).

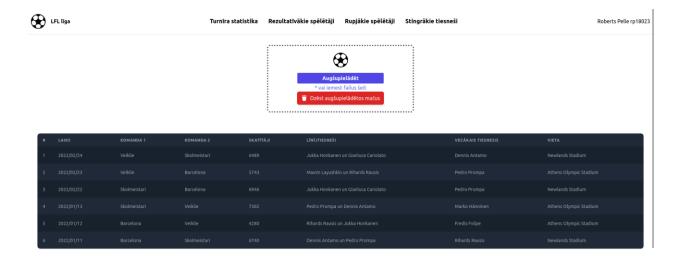
Programmatūra darbojas kā klienta pusē pieejama tīmekļa vietne, kurai var piekļūt uz tās pašas ierīces, uz kuras tā tiek palaista. Visi dati tiek glabāti un apstrādāti klienta pusē, izmantojot Svelte un JavaScript programmēšanas valodas.

1. Kad tiek palaista programmatūra (skatīt README.md), tad jāatver saiti <a href="http://localhost:5173">http://localhost:5173</a>, kurā parādās maču augšupielādes skats:



- 2. Uzspiežot uz pogu "Augšupielādēt", parādās iespēja augšupielādēt mačus. Tad, kad ir izvēlēti un augšupielādēti JSON faili, kas satur nepieciešamos maču datus, tad apakšējā tabulā parādās katra mača informācija: laiks, komandas, skatītāji, līnījtiesneši, vecākais tiesnesis un mača norises vieta.
  - a. Katru reizi, kad tiek augšupielādēti jauni mači, tiek veiktas šādas darbības:
    - Iet cauri visiem augšupielādētājiem mačiem, tos salīdzina savā starpā ja i-tais mačs atkārtojas (tas ir, ja i-tā mača vieta sakrīt ar kāda cita j-tā mača vietu UN kāda no i-tā mača komandām parādās j-tajā mačā, tad šo maču ignorē)
    - ii. Ja projektā jau ir saglabāti iepriekš augšupielādēti mači, tad šo pašu pārbaudi veic, salīdzinot katru jauno augšupielādēto maču ar jau iepriekš saglabātajiem un izņem ārā nederīgos atbilstoši prasībām

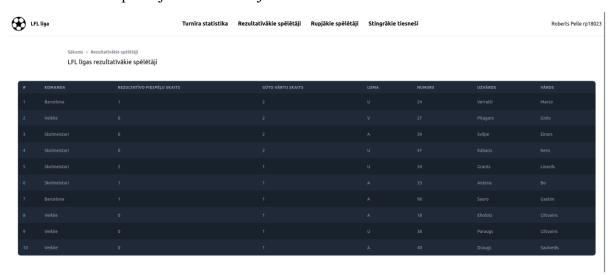
Rezultātā parādās šāds skats:



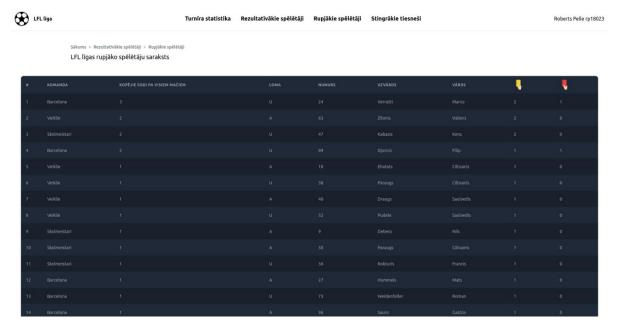
- 3. Katrā lapā augšējā sadaļā ir pieejamas vairākas saites: "Turnīra statistika", "Rezultatīvākie spēlētāji", "Rupjākie spēlētāji" un "Stingrākie tiesneši":
  - a. Turnīra statistika dotajiem JSON datiem:



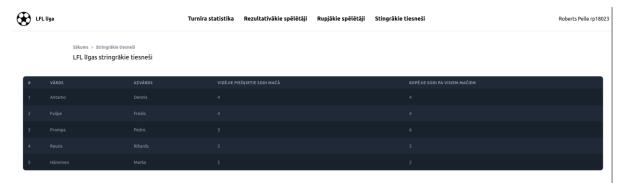
b. 10 rezultatīvāko spēlētāju saraksts dotajiem JSON datiem:



c. 20 rupjāko spēlētāju saraksts dotajiem JSON datiem:



d. Stingrāko tiesnešu saraksts dotajiem JSON datiem:



## Projekta struktūra (pašveidotās mapes / faili)

- *static*: mape, kurā glabājas statiski attēli, kuri tiek izmantoti tīmekļa galvenes sadaļā (favicon, vietnes logo)
- src: galvenā mape, no kuras tiek pasniegts viss lapas saturs:
  - assets: galveno stilu fails
  - components: dažādas vispārīgas komponentes (lietotāja saskarnē (UI) izmantoti attēli, HTML tabulas struktūras utml.)
  - helpers: mape ar palīgfailiem, kas tiek izmantoti datu grupēšanā un tās statistikas izveidē
  - *− lib*:
- *JSON\_TestData*: mape ar uzdevuma aprakstā dotajiem JSON failiem
- LFLDataFiles: mape, no kuras tiek ņemti visi JSON faili, kurus augšupielādēs
  - Oriģināli bija domāts tā, ka maču dati tiks ņemti no šajā mapē esošajiem failiem, taču pēc tam tas tika mainīts tā, ka maču datus augšupielādēs manuāli caur augšupielādes (vietnes sākuma) lapu
- stores.js: galvenais fails datu statistikas veidošanai

#### – routes:

- +layout.svelte: nosaka satura un citu HTML elementu izkārtojumu visām lapām vietnē
- +page.svelte: VIETNES SĀKUMA LAPA
- players:
  - +page.svelte: TURNĪRA 10 REZULTATĪVĀKO SPĒLĒTĀJU SARAKSTA LAPA
- fouls/+page.svelte: TURNĪRA 20 RUPJĀKO SPĒLĒTĀJU SARAKSTA LAPA
- tournament/+page.svelte: TURNĪRA STATISTIKAS LAPA
- referee-fouls/+page.svelte: TURNĪRA STINGRĀKO TIESNEŠU LAPA

## Kur un kā tiek glabāti dati?

Augšupielādētie mači tiek glabāti, izmantojot tīmeklī pieejamo localStorage (avots: <a href="https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage">https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage</a>). Ņemot vērā, ka viss projekts ir veidots tikai klienta pusē, tas nozīmē, ka principā visu laiku tiek glabāti tikai iepriekš augšupielādēti mači šajā localStorage glabātuvē zem indeksa "LFLData".

Pašā darbā netiek izmantota nedz aizmugures daļa / sistēma, nedz arī datu bāze - visi dati tiek glabāti tikai iekš localStorage (savā ziņā tas ir arī ierobežojums, bet šāda uzdevuma ietvaros glabātuve ir pietiekami laba un robusta, lai tajā varētu glabāt šāda veida JSON datus).

### Datu statistikas sagatavošana:

Datu statistika tiek sagatavota, izmantojot Svelte globālos glabātuves objektus, kurus dēvē par "*stores*" (avots: <a href="https://svelte.dev/docs#run-time-svelte-store">https://svelte.dev/docs#run-time-svelte-store</a>). Šie objekti tiek glabāti failā "*src/lib/stores.js*" un ir pieejami visās projekta vietnes lapās.

Katrs objekts satur informāciju, kas mainās atkarībā no tā, kāda ir galvenā objekta "*LFLData*" vērtība - šis objekts satur visu iepriekš augšupielādēto maču saturu.

Tālāk katra nepieciešamā statistika tiek iegūta, izveidojot *atvasināto* jeb "*derived*" objektu no galvenā "*LFLData*" objekta, kas tiks pārveidots katru reizi, kad oriģinālais "*LFLData*" objekts tiek atjaunināts. Katrā no šiem objektiem tiek apstrādāta galvenajā objektā esošā informācija (dati par mačiem).