Yleiskuva

Järjstelmän backend toteutettiin Javascriptillä käyttäen TypeScript -syntaksia. Back- ja frontendin välisissä datakyselyissä käytetään GraphQL-kieltä.

Koodin muotoiluasetukset

Frontendin lähdekoodi on muotoiltu käyttäen Airbnb:n muotoiluasetuksia.

Backendissä käytettiin Eslintin Stantardia syntaksia.

Käyttöohje

Vatii NodeJS. Tehty käyttämään versiota 14.16.0 tai uudempaa.

Asenna riippuvuudet sovelluksen juurihakemistossa suorittamalla komento

npm install

Käännä backendin TypeSript -tiedostot JavaScriptiksi komennolla

npm run build-backend

Käännä frontendin lähdekoodi html-sivuksi/javascriptiksi komennolla

npm run build-frontend

Käynnistä palvelin

npm run start

Palvelin käynnistyy oletusarvoisesti porttiin 8080.

Muutokset

Tietokanta

*Index*in ollessa varattu sana MySQL:ssä, vaihdettiin taulujen Score ja Hole index-attribuutti muotoon **indeksi**

Lisätty taulukko *logs* lokitietoja varten

*Scorecard* taulu yhdistetään käyttäjään yhdistämällä playerName *alias*-taulun kautta *user*-tauluun sen sijaan että jokaiselle Scorecard riville haettaisiin *user*-avain.

*Hole*-taulua ei otettu käyttöön.

Komponentit

Backend

|  |  |
| --- | --- |
| index.ts | Käynnistystiedosto. Luo ja käynnistää palvelimen |
| types.ts | TypeScriptin tyyppejä |
| tsconfig.json | TypeScriptin asetustiedosto |
| .eslintrc.json | Eslintin asetustiedosto |

GraphQl

|  |  |
| --- | --- |
| mutations | GraphQL mutaatiot |
| queries | GraphQL kyselyt |
| resolvers | GraphQL resolverit |
| typedefs | GraphQL skeemat |

Services -kansio

|  |  |
| --- | --- |
| Komponentti | Kuvaus |
| gameServices | Yhteys tietokannan ja pelin osien (tuloskortti, rata yms.) välillä. |
| logServices | Hakee ja kirjoittaa tietokantaan lokitietoja |
| setupService | Hakee ja kirjoittaa tietokantaan asetuksia |
| statsService | Hakee tilastotietoja tietokannasta |
| userService | Hoitaa tunnuksien luomisen ja muokkaamisen sekä aliaksien luomisen ja muokkaamisen tietokantaan |

Utils

Sisltää erilaisia apu-tiedostoja

|  |  |
| --- | --- |
| csvParser | Parsii .csv tiedoston JSON muotoon sekä kirjoittaa sen tietokantaan |
| mathAndShit | Matemaattisia funktioita |
| mySqlHelpers | Apufunktioita tietokannan käsittelyyn. |
| validators | TypeScriptin string- ja int-validaattorit |

Riippuvuudet

GraphQL – GraphQ…

Jsonwebtoken – Sisäänkirjautumisessa käytettävän tokenin luomiseen/validointiin

Bcryptjs – Salasanojen tiivisteiden luomiseen/validointiin

Express – Palvelinkirjasto

Mysql – MySQL pyyntöjen lähettäminen

Frontend

|  |  |
| --- | --- |
| useAliases | Hakee, tallentaa ja poistaa aliaksia palvelimelta |
| useGetCompetitions | Hakee kilpailut palvelimelta ja parsii ne helposti käytettävään muotoon |
| useMe | Hakee kirjautuneen käyttäjän tiedot palvelimelta |
| useSetup | Hakee kilpailujen asetukset palvelimelta |

|  |  |
| --- | --- |
| Aliases | Käyttäjän aliaksien näyttäminen ja hallinnointi |
| AppBar | Sivuston yläreunassa oleva navipalkki |
| AppBarLink | Linkki |
| Competitions | Renderöi ruudulle kilpailut |
| CreateAccount | Formi ja tapahtumakäsittelijät tunnuksien luomiseen |
| FrontPage | Etusivu |
| HideIfNotLogged | Renderöi komponentin sisälltä olevat komponentit vain jos käyttäjä on kirjautunut sisään |
| LoggedUserProvider | Toimittaa lapsikomponenteille tiedon onko käyttäjä kirjautunut sisään |
| LoginForm | Kirjautumisformi sekä siihen liittyvät tapahtumakäsittelijät |
| LogoutButton | Uloskirjautumisnappi |
| MyStats | Käyttäjän ratakohtaiset tilastot |
| ReadLogs | Lokitietojen katselu |
| Settings | Käyttäjän asetuksetn (email, aliakset, salasana) |
| Setup | Peliasetukset |
| SimpleCourseStats | Renderöi yhden radan tilastot |
| SingleCompetitions | Yhden kilpailun |
| SingleLogEntry | Yhden lokitiedon |
| UploadCsv | .csv tiedoston lähetys ja tapahtumakäsittelijät |

notistack – Ilmoitusten näyttämiseen

ApolloClient – GraphQL kyselyjen suorittamiseen

Material UI – Tyylittely

Formik – Formien käsittelyyn

PropTypes – React komponenttien propsien validointiin