**Esports betting**

име: Димитър Кюртов

фн: 81797

Въведение

Esporst betting е уеб приложение, което позволява на потребителите да виждат мачовете, които се провеждат в основните лиги по League of legends и да залагат на тях. Разликата с традиционните сайтове за залагане е в това, че коефициентите на игрите не се поставят от игралната къща, а от самите потребители. Всеки може да публикува залог върху определен мач с определен коефициент. Всеки може да разглежда залозите в системата и да играе срещу някой на предложеният коефициент. Идеята е да се децентрализира традиционната идея за залагане и потребителите да играят един срещу друг вместо срещу казиното.

Технологии

Фронтендът е написан на React.js. Redux библиотека за централизиране на състоянието на приложението. За рутиране между страниците е използван React Router. Използвани bulma.css и material.ui за генериране на готови компоненти.

Node.js и express за бекенд на база REST/JSON API принципите. Базата данни е mongoDB, използвана е mongoose библиотеката за комуникация с нея.

Роли

Анонимен потребител - може да вижда мачовете и публикуваните залози

Регистриран потребител - може да вижда мачовете и публикуваните залози както и да създава нови залози и да играе срещу публикувани.

Админ - може да вижда мачовете и публикуваните залози както и да създава нови залози и да играе срещу публикувани. Може да въвежда резултатите от мачовете.

Функционалност

Логване в системата със съществуващ акаунт.

Разглеждане на мачове от основните лиги.

Разглеждане на публикуваните залози.

Филтриране на мачове по лиги.

Филтриране на залози по лиги.

Разглеждане на залозите, които си публикувал.

Разглеждане на залозите, срещу които играеш.

Публикуване на нов залог.

Играене срещу вече съществуващ залог.

Депозиране на тестови пари в акаунта.

Публикуване на резултат от мач, което разрешава всички текущи залози, в които участва този мач.

Архитектура

Aрхитектура на базата данни - <https://dbdiagram.io/d/62b2f9e869be0b672c1c8435>

Основните модели са:

Професионални играчи

Професионални отбори

Водещите лиги в играта

Мачове между различните отбори

Потребители на системата

Залози върху мачове

Връзките между тях са описани диаграмата

Single-Page Application

Frontend:

App.js - съдържа всички останали компоненти и тяхната подредба на екрана.

Navbar.js - Визуализира логото, както и връзки за навигиране към изгледа за мачове и залози. Съдържа формата за логин.

Leagues.js - изпраща заявка за всички лиги в базата, после ги визуализира като ги предава на League.js компонент

League.js - визуализира една лига като съдържа checkbox за филтриране по тази лига

Matches.js - изпраща заявка за всички мачове в базата, после ги визуализира като ги предава на Match.js компонент

Match.js - визуализира един мач. Има бутон за публикуване на резултат от мача, видим само за админ. Има възможност за избиране победител от мача, което стартира форма за публикуване на залог чрез PendingBet.js компонента.

Bets.js - изпраща заявка за всички залози в базата, после ги визуализира като ги предава на AllBet.js компонент

AllBet.js - визуализира един залог. Има форма за играене срещу този залог.

Cart.js - изпраща заявка за всички залози в базата на текущият потребител както и на всички залози срещу които играе после ги визуализира като ги предава на Bet.js и BetTaken.js компонентите. Ако има текуща форма за публикуване на залог, се визуализира тук с приоритет спрямо останалите.

Bet.js - визуализира един залог. Показва колко хора играят срещу него и колко от капацитета на залога е запълнен.

BetTaken.js - визуализира един залог. Показва каква печалба би получил ако го спечелиш.

PendingBet.js - визуализира форма за публикуване на залог.

Notification.js - Визуализира резултатите от заявките към бекенда. (Например при неуспешен логин или при успешно публикуване на залог и тн.)

Backend:

GET /${entity}/ - Връща всички инстанции на entitу

GET /${entity}/${id} - Връща инстанцията на entity с идентификатора

POST/${entity}/ - Публикува нова инстанция на entitу, която се съдържа в тялото на заявката

PUT /${entity}/${id} - Заменя инстанцията на entity с идентификатора с инстанция в тялото на заявката

DELETE /${entity}/${id} - Изтрива инстанцията на entity с идентификатора.

entity може да е

users - потребителите на системата

players - професионалните играчи, регистрирани в системата

teams - отборите от играчи

leagues - лигите от отбори

matches - мачовете между отборите

bets - залозите

POST /login - логва потребител като му връща jwt token при успешно влизане

POST /bets/leagues - връща всички залози филтрирани по лигите, в тялото на заявката

GET /bets/:userId - връща всички залози на потребителят с userId

GET /bets/taken/:userId - връща всички залози срещу, които е заложил потребителят с userId

PUT /bets/take\_bet/:id - Потребителят, който праща заявката залага срещу залога с id, информацията за залогът е в тялото на заявката.

POST /bets/resolve\_match/:id - Въвежда резултатът от мач с id и се разрешават всички залози, в които участва този мач.

Конфигурация

Нов терминал в руут папката на проектът

cd backend

npm i

npm start

Нов терминал в руут папката на проектът

cd frontend

npm i

npm start