Dokumentacja

temat projektu

„Liga Mistrzów”

Konrad Pasik

Koncepcja oraz założenia:

Projekt zakładał przechowywanie oraz możliwość przeglądania wyników w rozgrywkach Ligi Mistrzów.

W lidze mistrzów bierze udział 32 drużyny, każda drużyna ma określony kraj pochodzenia oraz ligę w której grają na codzień. Każda z drużyn bierze udział w rozgrywkach na poziomie „grupa". W każdej grupie znajdują się 4 drużyny, które rozgrywają przeciwko sobie po 2 spotkania, łącznie jest 8 grup. Do następnego poziomu rozgrywek wychodzą po dwie najlepsze drużyny z każdej grupy. Administrator ma za zadanie śledzić rozgrywki sportowe, uzupełniać bazę danych nowymi wynikami, oraz po zakończeniu każdego z poziomów rozgrywek uzupełniać kolejny poziom o zakfalifikowane drużyny.

Aplikacja ma za zadanie umożliwiać śledzenie wyników dla zwykłego użytkownika oraz możliwość dodawania, usuwania oraz aktualizowania danych dla administratora.

Dostęp aplikacji możliwy jest z sieci wydziałowej po udaniu się na adres:
http://149.156.109.180:2020

Aktualne dane znajdujące się w bazie danych odzwierciedlają faktyczny stan Ligi Mistrzów na dzień 27.01.19r.

Zdefiniowanie struktury bazy danych:

Obiekty w bazie danych:

druzyny – drużyny pochodzą z różnych państw oraz grają w różnych ligach sportowych. Id_drużyny służy do odwoływania się do odpowiedniej drużyny. Rolą tabeli jest przechowywanie podstawowych informacji o drużynie.

Pola:

id_druzyny – int PK

nazwa – character(50)

kraj_pochodzenia – character(50)

liga – character(50)

grupa – każda z drużyn jest przypisana do odpowiedniej grupy, ponosi porażki, remisy oraz zwycięstwa, zdobywa oraz traci określoną liczbę bramek oraz osiąga różną liczbę punktów. Pole id_druzyny jest zarazem PK jak i FK ponieważ w tablicy grupa nie może się znaleźć 2 razy ta sama drużyna oraz w tej tabeli nie potrzebujemy ogólnych informacji o drużynie. Rolą tabeli jest przechowywanie statystyki drużyn na poziomie rozgrywek „grupa".

Pola:

id_drużyny – int PK FK

grupa – character(1)

zwyciestwa – int

remisy – int

porazki – int

zdobyte_gole – int

stracone_gole – int

punkty – int

jedna_osma_finalu, cwiercfinal, polfinal, final – tabele te są bliźniaczo podobne jednak każda z nich przechowuje statystyki drużyn na różnym poziomie rozgrywek. W przeciwieństwie do tabeli **grupa** druzyny nie uczestnicza już w żadnej z grup oraz nie zdobywają punktów. Podobnie jak w powyższej tabeli pole id_druzyny jest zarazem PK jak i FK.

Pola:

id_drużyny – int PK FK

zwyciestwa – int

remisy – int

porazki – int

zdobyte_gole – int

stracone_gole – int

spotkania_grupa, spotkania_1/8, spotaknia_cwiercfinal, spotkania_polfinal, spotkania_final – tabele te są do siebie podobne jednak przechowują informacje o spotkaniach, np. wynik spotkania, na różnych poziomach rozgrywek. Pole id_spotkania jest PK, pola id_druzyny1 oraz id_druzyny2 są FK i odwołują się do tabel ze statystykami drużyn na odpowiednim poziomie rozgrywek. Rolą tabel jest przechowywanie informacji o dacie spotkań oraz o wyniku spotkania. Data spotkania przechowywana jest jako ciąg znaków w celu łatwiejszego pobrania danych do aplikacji.

Pola:

id_spotkania - int PK

id_druzyny1 – int FK

id_druzyny2 – int FK

wynik – character(10)

data_spotkania – character(10)

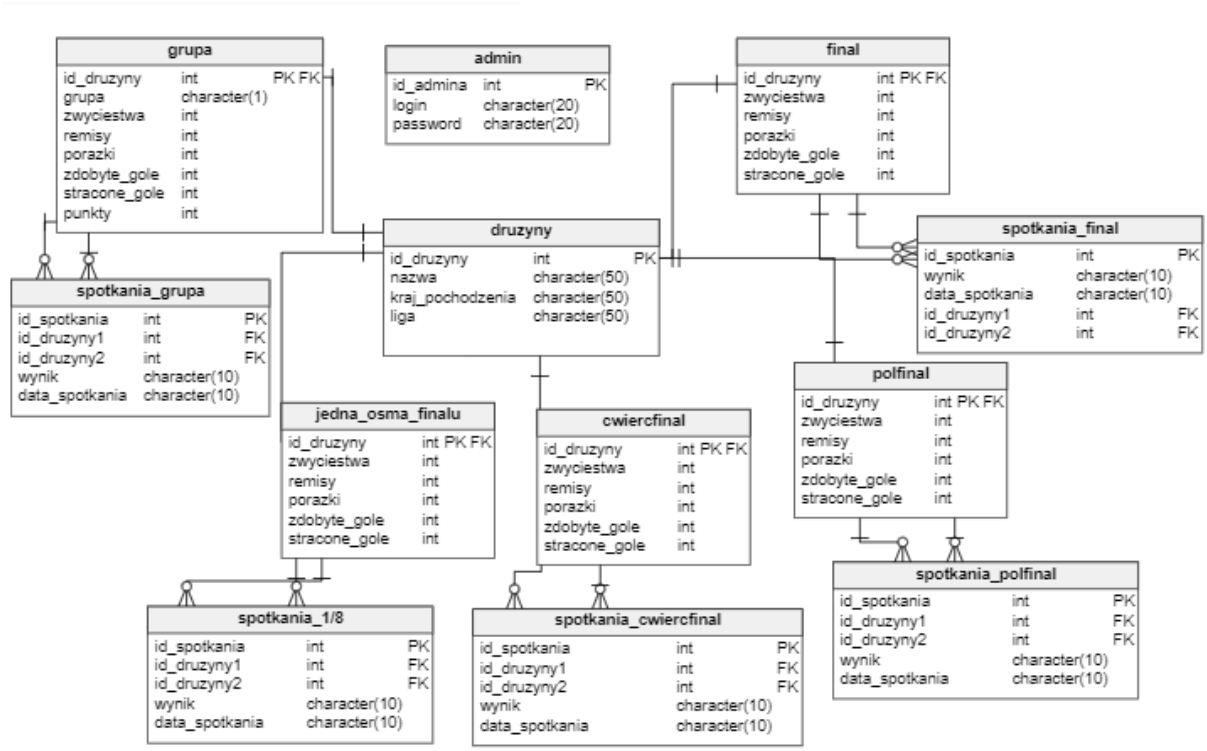
admin – tabela ta służy wyłącznie do przechowywania danych administratora bazy w celu możliwości jej modyfikacji.

Pola:

id_admina – int PK

login – character(20)

password – character(20)



Rysunek 1 Diagram ERD

Analiza zależności funkcyjnych w bazie danych:

Większość tabel ma za klucz pojedynczy parametr, a ich pola składają się z wartości atomowych więc spełniona jest zarówno pierwsza jak i druga postać normalna. Zastrzeżenia co do powyższych reguł można mieć do tabeli „admin”. Jest to tabela w zasadzie niepowiązana z żadnymi innymi tabelami, służy jedynie jako miejsce do przechowywania dancyh administratora, który może modyfikować dane w bazie danych.

Postać BCNF również jest spełniona dla wszystkich tabel.

Funkcjonalności bazy danych:

Ogółem administrowaniem danymi w bazie danych zajmuje się wyłącznie administrator, który po wpisaniu swojego loginu oraz hasła w Panelu Administratora

otrzymuje możliwość dodawnia oraz usuwania danych. Logowanie jest po strownie serwera NODE który korzysta z danych zamieszczonych w bazie danych. Każdy niezalogowany użytwkonik może korzystać z funkcji SELECT jednak nie ma żadnych praw do modyfikacji danych.

Funkcje w bazie danych:

druzyny() – funkcja ta buduje tabelę informacji o drużynie biorącej udział w rozgrywkach, która potem jest przetwarzana i wyświetlana z poziomu node.js.

grupastat() - funkcja ta buduje tabelę informacji o statystykach drużyn biorącej udział w rozgrywkach na poziomie grupowym, która potem jest przetwarzana i wyświetlana z poziomu node.js.

jedenosiemstat () - funkcja ta buduje tabelę informacji o statystykach drużyn biorącej udział w rozgrywkach na poziomie 1/8, która potem jest przetwarzana i wyświetlana z poziomu node.js.

cwiercfinal() - funkcja ta buduje tabelę informacji o statystykach drużyn biorącej udział w rozgrywkach na poziomie ćwierćfinału, która potem jest przetwarzana i wyświetlana z poziomu node.js.

polfinal() - funkcja ta buduje tabelę informacji o statystykach drużyn biorącej udział w rozgrywkach na poziomie półfinału, która potem jest przetwarzana i wyświetlana z poziomu node.js.

final() - funkcja ta buduje tabelę informacji o statystykach drużyn biorącej udział w rozgrywkach na poziomie finału, która potem jest przetwarzana i wyświetlana z poziomu node.js.

spotkaniagrupa() - funkcja ta buduje tabelę informacji spotkaniach drużyn na poziomie rozgrywek grupa, która potem jest przetwarzana i wyświetlana z poziomu node.js.

spotkaniajedenosiem() - funkcja ta buduje tabelę informacji spotkaniach drużyn na poziomie rozgrywek 1/8, która potem jest przetwarzana i wyświetlana z poziomu node.js.

spotkaniacwiercfinal() - funkcja ta buduje tabelę informacji spotkaniach drużyn na poziomie rozgrywek ćwierćfinał, która potem jest przetwarzana i wyświetlana z poziomu node.js.

spotkaniapolfinal() - funkcja ta buduje tabelę informacji spotkaniach drużyn na poziomie rozgrywek półfinał, która potem jest przetwarzana i wyświetlana z poziomu node.js.

spotkaniafinal() - funkcja ta buduje tabelę informacji spotkaniach drużyn na poziomie rozgrywek finał, która potem jest przetwarzana i wyświetlana z poziomu node.js.

Projekt aplikacji

Aplikacja napisana została przy pomocy NODE.js języka Java-Script z wykorzystaniem HTML i CSS.

W pliku app.js możemy wyróżnić funckję:

app.get('/druzyny', ...) – pobiera informacje o drużynach z bazy danych i renderuje widok druzyny.dust, która przetwarza dane i za pomocą layout.dust wyświetla je w przyjaznej formie dla użytkownika

app.get('/statGrupa', ...) – pobiera informacje o statystykach drużyn na poziomie rozgrywek grupa z bazy danych i renderuje widok grupa.dust, która przetwarza dane i za pomocą grupaStat.dust wyświetla je w przyjaznej formie dla użytkownika

app.get('/stat18', ...) – pobiera informacje o statystykach drużyn na poziomie rozgrywek 1/8 z bazy danych i renderuje widok 18stat.dust, która przetwarza dane i za pomocą jedenosiemstat.dust wyświetla je w przyjaznej formie dla użytkownika

Trzy poniższe funkcje wykorzystują widok statinne.dust do przetwarzania danych oraz inneStat.dust do ich wyświetlania ponieważ pobrane dane są wyświetlane i przetwarzane w taki sam sposób.

app.get('/statCw', ...) – pobiera informacje o statystykach drużyn na poziomie rozgrywek ćwierćfinał z bazy danych i renderuje widoki wymienione powyżej

app.get('/statPF', ...) – pobiera informacje o statystykach drużyn na poziomie rozgrywek półfinał z bazy danych i renderuje widoki wymienione powyżej

app.get('/statF', ...) – pobiera informacje o statystykach drużyn na poziomie rozgrywek finał z bazy danych i renderuje widoki wymienione powyżej

app.get('/wynikiGrupa', ...) – pobiera informacje o wynikach na poziomie rozgrywek grupa z bazy danych i renderuje widok wynikiGrupa.dust, która przetwarza dane i za pomocą grupaWyniki.dust wyświetla je w przyjaznej formie dla użytkownika

app.get('/wyniki18', ...) – pobiera informacje o wynikach na poziomie rozgrywek 1/8 z bazy danych i renderuje widok wynikiJedenOsiem.dust, która przetwarza dane i za pomocą JedenOsiemWyniki.dust wyświetla je w przyjaznej formie dla użytkownika

Podobnie jak powyżej trzy poniższe funkcje wykorzystują ten sam widok wyniki.dust do przetwarzania danych oraz WynikiInne.dust do ich wyświetlania.

app.get('/wynikiCw', ...) – pobiera informacje o wynikach na poziomie rozgrywek ćwierćfinał z bazy danych i renderuje widoki wymienione powyżej

app.get('/wynikiPF', ...) – pobiera informacje o wynikach na poziomie rozgrywek półfinał z bazy danych i renderuje widoki wymienione powyżej

app.get('/wynikiF', ...) – pobiera informacje o wynikach na poziomie rozgrywek finał z bazy danych i renderuje widoki wymienione powyżej

app.post('/login', ...) – wysyła dane wpisane przez użytkownika w celu zalogowania się, jeżeli dane są poprawne użytkownik uzyskuje dostęp do panelu modyfikacji

app.post('/dodajDruzyne', ...) – wysyła dane wpisane przez użytkownika w celu dodania druzyny do wybranego poziomu rozgrywek

app.post('/dodajWynik', ...) – wysyła dane wpisane przez użytkownika w celu dodania spotkania do wybranego poziomu rozgrywek

app.post('/usunDruzyne', ...) – wysyła dane wpisane przez użytkownika w celu usunięcia drużyny z wybranego poziomu rozgrywek

app.post('/usunWynik', ...) – wysyła dane wpisane przez użytkownika w celu usunięcia spotkania z wybranego poziomu rozgrywek

app.post('/aktualizujDruzyne', ...) – wysyła dane wpisane przez użytwkonika w celu aktualizacji statystyki drużyny w wybranym poziomie rozgrywek

app.post('/aktualizujWynik', ...) – wysyła dane wpisane przez użytkownika w celu aktualizacji spotkania w wybranym poziomie rozgrywek

Uwagi do funkcji:

W celu działania poprawności funkcji administrator musi pamiętać, że:

- w poziomie rozgrywek dana drużyna może być zapisana tylko raz,

- dodanie wyniku jest możliwe wyłącznie po uprzednim dodaniu drużyny,

- usunięcie drużyny jest możliwe wyłącznie po usunięciu spotkań drużyn z danego poziomu rozgrywek,

- aktualizacja statystyk drużyny jest możliwa po uprezdnim dodaniu drużyny

- aktualizacja spotkania jest możliwa wyłacznie po uprzednim dodaniu spotkania

Po prawidłowym działaniu funkcji, tj. dodaniu, usunięciu lub zaktualizowaniu rekordu, aplikacja informuje użytkownika odpowiednim komunikatem.

Plik package.json – zawiera wszystkie moduły potrzebne do prawidłowego działania aplikacji, instalacja modułów jest możliwa poprzez komende *npm install package.json*

W katalogu **public** znajdują się następujące pliki .html:

index.html – jest to strona startowa naszej aplikacji

admin.html – jest to strona zawierająca panel logowania administratora

W katalogu **style** znajdują się pliki .css nadające aplikacji przyjazny wygląd.

W katalogu **js** znajduje się plik **script.js**, który odpowiada za animacje oraz wysyłanie odpowiednich żądań do serwera. Ważniejsze funkcje w pliku **script.js:**

log(form) – pobiera dane wpisane przez użytkownika, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól, lub wpisane dane są błędne, zostanie o tym poinformowany. W przeciwnym wypadku funckja wyświetla panel

administratora oraz liste drużyn znajdującą się w bazie (dodatkowe ułatwienie dla administratora).

addNewTeam(form) – funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

addNewScore(form) – funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

deleteTeam2(form) – funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

deleteScore(form) – funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

UpdateTeam(form) - funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

UpdateScore(form) - funckja pobiera wpisane dane, a następnie wysyła je do serwera z odpowienim żądaniem. Jeżeli użytkownik nie uzupełnił wszystkich pól zostanie o tym poinformowany. W przypadku pozytywnej modyfikacji bazy danych użytwkonik zostaje o tym poinformowany odpowiednim komunikatem.

W katalogu **views** znajdują się widoki, które aplikacja renderuje zgodnie z wywołanym żądaniem.

Działanie aplikacji:

Podstawowy użytkownik ma możliwość przeglądania danych za pomocą poszczególnych odnośników:

DRUŻYNY – wyświetla użytkownikowi podstawowe informacje na temat drużyn biorących udział w Lidze Mistrzów.

STATYSTYKI DRUŻYN – wyświetla poziomy rozgrywek, użytkownik ma możliwość wyboru poziomu i analizowania statystyk drużyn.

SPOTKANIA – podobnie jak odnośnik powyżej, wyświetla poziomy rozgrywek i umożliwia użytkownikowi przegląd wyników spotkań na wybranym poziomie

Panel Administratora – przekierowuje użytkownika na strone logowania administratora

Administrator ma możliwość dodania,usunięcia oraz aktualizacji drużyny lub wyniku.

Administrator musi wybrać poziom rozgrywek do którego chce dodać/usunąć/aktualizować dane. W przypadku dodawnania drużyny do poziomu Grupa, administrator musi uzupełnić pole Grupa oraz punkty. W przeciwnym wypadku oba te pola muszą pozostać puste w celu prawidłowego działania aplikacji.

POWRÓT – przekierowuje administratora na stronę główną oraz automatycznie go wylogowuje.

Uwagi:

Dane administratora:

Login: **Konrad**

Hasło: **Pasik**