**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Индивидуальное Домашнее Задание

**по дисциплине «Разработка ПО информационных систем»**

**Тема: Статистика футбольных матчей**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 6303 |  | Архипцев Е.Д. |
|  |  | Матюшина М.Е. |
|  |  | Эвергрин П. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2019

**ЗАДАНИЕ**

|  |  |  |
| --- | --- | --- |
| Студенты Архипцев Е.Д., Матюшина М.Е., Эвергрин П. | | |
| Группа 6303. | | |
| Тема проекта: Разработка приложения «Статистика футбольных матчей». | | |
| Исходные данные:  Необходимо реализовать приложение, позволяющее  1)осуществлять сбор статистики футбольных матчей  2)осуществлять поиск матчей по выбранным параметрам  3)производить сравнение двух команд, в том числе графически | | |
| Содержание пояснительной записки:  «Содержание», «Введение», «Качественные требования к решению», «Сценарии использования», «Модель данных», «Разработанное приложение», «Выводы», «Приложение», «Список использованных источников» | | |
| Предполагаемый объем пояснительной записки:  Не менее 15 страниц. | | |
| Дата выдачи задания: | | |
| Дата сдачи задания: | | |
| Дата защиты задания: | | |
| Студенты гр. 6303 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Архипцев Е.Д. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Матюшина М.Е. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Эвергрин П. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Заславский М.М. |

**Аннотация**

В рамках данного курса предполагалось разработать какое-либо приложение в команде на одну из поставленных тем. Была выбрана тема создания приложения «Статистика футбольных матчей», ориентированного на хранение, анализ и использование данных футбольных матчей. Было проведено исследование возможностей **нереляционных** баз данных на примере **NoSQL** решения *MongoDB.* Найти исходный код и всю дополнительную информацию можно по ссылке: https://github.com/moevm/nosql1h19-football-stats

**Summary**

As part of this course, it was planned to develop an application as a team from one of the given topics. The topic of creation of the application "Statistics of football matches" focusing on storage, analysis and use of data of football matches was chosen. A study of the possibilities of non-relational databases on the example of NoSQL solutions MongoDB was conducted. You can find the source code and all additional information here through the link: https://github.com/moevm/nosql1h19-football-stats

**Содержание.**

[1. Введение. 5](#_Toc7541262)

2. [Качественные требования к решению. 5](#_Toc7541263)

3. [Сценарии использования. 5](#_Toc7541264)

3.1. [Mакет UI. 5](#_Toc7541265)

3.2. [Описание сценариев использования. 8](#_Toc7541266)

4. [Модель данных. 10](#_Toc7541277)

4.1.  [NoSQL модель данных. 10](#_Toc7541265)

4.1.1.  [Описание и расчет объема.](#_Toc7541265) 10

4.1.2.  [Запросы.](#_Toc7541264) 11

4.2.  [SQL модель данных.](#_Toc7541264) 12

4.2.1. [Описание и расчет объема 13](#_Toc7541284)

4.2.2. [Запросы. 14](#_Toc7541286)

4.3.  [Сравнение SQL и NoSQL. 15](#_Toc7541265)

5. [Разработанное приложение. 15](#_Toc7541288)

5.1. [Краткое описание. 15](#_Toc7541289)

5.2. [Схема экранов приложения. 16](#_Toc7541290)

5.3. [Использованные технологии. 1](#_Toc7541291)7

5.4. Ссылки на Приложение[. 1](#_Toc7541290)7

6. [Выводы. 17](#_Toc7541293)

6.1. [Результаты. 17](#_Toc7541294)

6.2. [Недостатки и пути для улучшения полученного решения. 17](#_Toc7541295)

7. [Приложение. 18](#_Toc7541296)

7.1. [Документация по сборке и развертыванию приложения. 18](#_Toc7541297)

7.2. [Снимки экранов приложения. 19](#_Toc7541298)

8. Список литературы[.](#_Toc7541296) 22

1. **Введение.**

Цель работы – создать удобное решение для хранения, анализа и использования информации о футбольных матчах.

Было решено разработать веб-приложение, которое позволит хранить в электронном виде всю информацию о матчах, осуществлять сбор статистики футбольных матчей, осуществлять поиск матчей по выбранным параметрам матча, а также производить сравнение двух команд, в том числе графически.

# **Качественные требования к решению.**

Требуется разработать приложение с использованием СУБД MongoDB.

1. **Сценарии использования.**
   1. **Макет UI.**

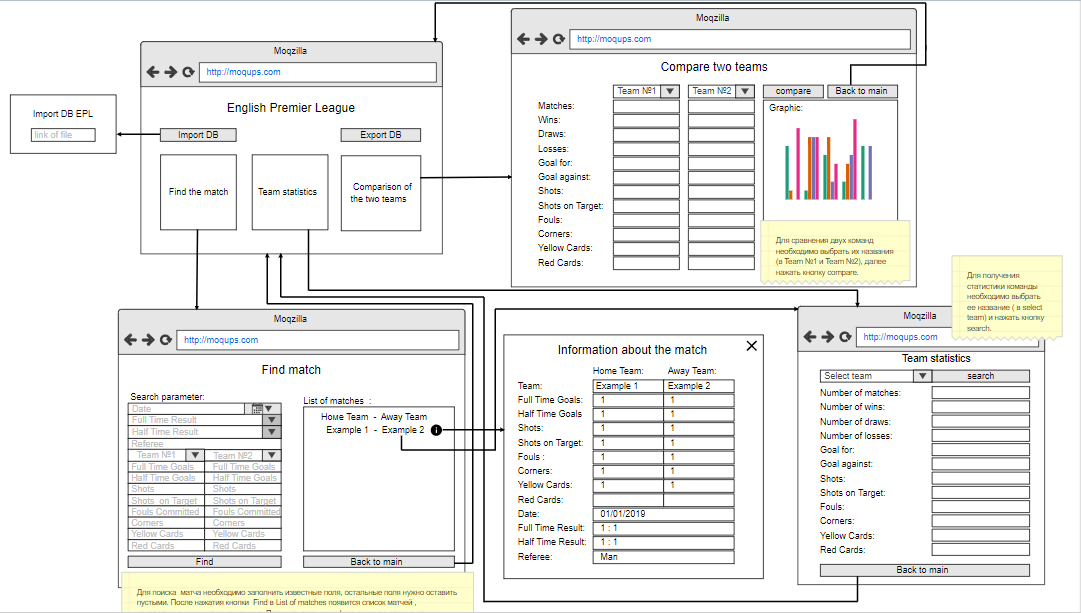


Рисунок 1 – Общий вид.

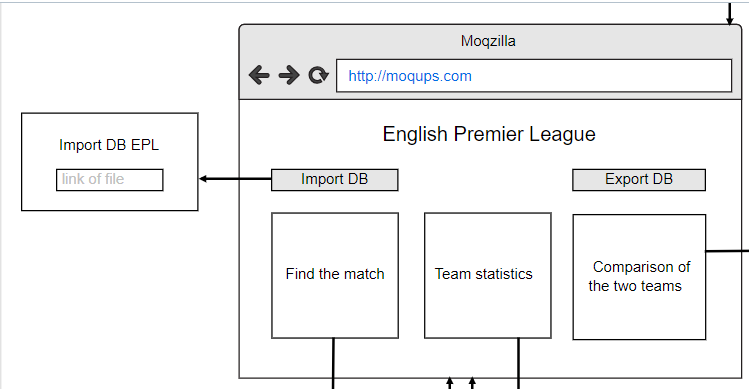


Рисунок 2 – Главная страница

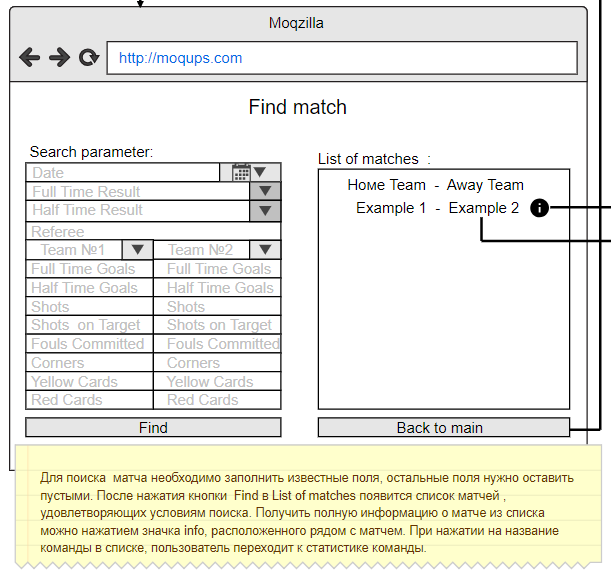


Рисунок 3 – Поиск матча

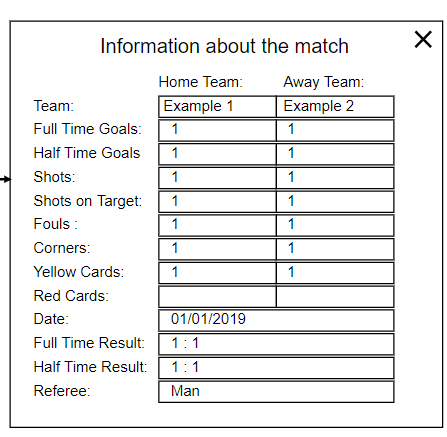


Рисунок 4 – Информация о матче

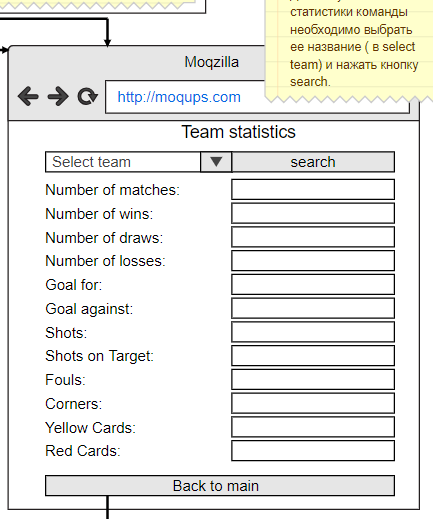


Рисунок 5 – Статистика команды

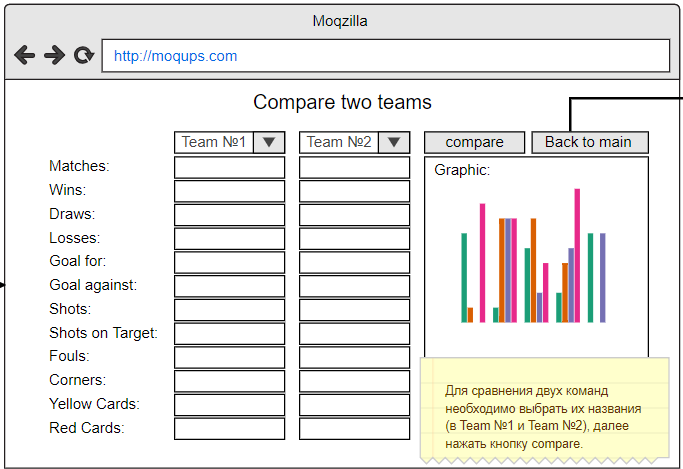


Рисунок 6 – Сравнение двух команд

* 1. **Описание сценариев использования.**

Пользователь заходит на главную страницу сайта, который предназначен для получения информации о матчах Английской Премьер-лиги. Он видит меню навигации, состоящее из трех кнопок:

* Find match - поиск матча,
* Team Statistics – статистика команды,
* Compare two teams – сравнение двух команд.

Также на главной странице расположены кнопки для экспорта и импорта базы данных матчей. Нажимая на кнопку импорта, пользователь получает возможность ввести путь до импортируемого файла. Нажимая на кнопку экспорта, база данных матчей экспортируется в файл, происходит скачивание.

Пользователь переходит в интересующий его раздел меню навигации сайта.

**Поиск матчей.**

Пользователь вбивает известные ему параметры матча в соответствующие поля, остальные поля не заполняются. После нажатия кнопки *Find* в *List of matches* появляется список матчей, удовлетворяющих условиям поиска. Список представляет собой пары *Домашняя Команда (Home Team) – Гостевая Команда (Away Team)*. Чтобы получить подробную информацию о матче, пользователь нажимает на значок *info* рядом с матчем. При нажатии на название какой-либо команды, пользователь переходит к ее статистике в раздел *Статистика команды (Team statistics)*, где ее название уже подставлено в select. Чтобы вернуться в меню, следует нажать кнопку *Back to main*.

Общие параметры матча:

* Date
* Full Time Result
* Half Time Result
* Referee
* Team №1
* Team №2

Параметры для каждой команды:

* Full Time Goals
* Half Time Goals
* Shots
* Shots on Target
* Fouls Committed
* Corners
* Yellow Cards
* Red Cards

**Статистика команды.**

В разделе *Team statistics* пользователь в *select team* выбирает название команды, статистику которой хочет получить и нажимает кнопку *search*. Пользователь получает **следующие данные**:

* Number of matches – общее количество матчей команды
* Number of wins – количество побед
* Number of draws – количество игр, сыгранных в ничью
* Number of losses – количество проигрышей
* Goal for – количество голов
* Goal against – количество пропущенных голов
* Shots – количество ударов
* Shots on Target – количество ударов в цель
* Fouls – количество фолов
* Corners – количество угловых
* Yellow Cards – количество желтых карточек
* Red Cards – количество красных карточек

Чтобы вернуться в главное меню, следует нажать кнопку *Back to main*.

**Сравнение двух команд**

Для сравнения двух команд, в разделе *Compare two teams*, пользователь выбирает названия команд *Team1* и *Team2* и нажимает на кнопку *compare*. После этого появляются статистики двух команд, а также график сравнения. Чтобы вернуться в главное меню, следует нажать кнопку *Back to main*.

1. **Модель данных.**
   1. **NoSQL модель данных.**
      1. **Описание и расчет объема.**

Данные хранятся в **MongoDB**. В БД одна коллекция – *Matchs*. В коллекции – документы, содержащие описание матча.

***Структура документа***

Документы имеют следующие поля:

* **\_id** - идентификатор документа, генерируется автоматически. **Тип** – ObjectID. **V** = 12b
* **Date** – дата матча. **Тип** – date **V** = 8b
* **HomeTeam** – домашняя команда. **Тип** - string. V = 1b \* N, где N - средняя длина строки, пусть N = 12. Тогда **V** = 12b.
* **AwayTeam** – гостевая команда. **Тип** - string. V = 1b \* N, где N - средняя длина строки, пусть N = 12. Тогда **V** = 12b.
* **FTHG** – голы домашней команды за Full Time. **Тип** – integer. **V**= 4b.
* **FTAG** – голы гостевой команды за Full Time. **Тип** – integer. **V**= 4b.
* **FTR** – результат за Full Time. **Тип** - string. **V** = 1b, т.к. строка состоит из одной буквы (H=Home Win, D=Draw, A=Away Win).
* **HTHG** – голы домашней команды за Half Time. **Тип** – integer. **V**= 4b.
* **HTAG** – голы гостевой команды за Half Time. **Тип** – integer. **V**= 4b.
* **HTR** – результат за Full Time. **Тип** - string. **V** = 1b, т.к. строка состоит из одной буквы (H=Home Win, D=Draw, A=Away Win).
* **Referee** – судья матча. **Тип** - string. V = 1b \* N, где N - средняя длина строки, пусть N = 10. Тогда **V** = 10b.
* **HS** – удары домашней команды. **Тип** – integer. **V**= 4b.
* **AS** – удары гостевой команды. **Тип** – integer. **V**= 4b.
* **HST** – удары в цель домашней команды. **Тип** – integer. **V**= 4b.
* **AST** – удары в цель гостевой команды. **Тип** – integer. **V**= 4b.
* **HF** – фолы домашней команды. **Тип** – integer. **V**= 4b.
* **AF** – фолы гостевой команды. **Тип** – integer. **V**= 4b.
* **HC** – угловые домашней команды. **Тип** – integer. **V**= 4b.
* **AC** – угловые гостевой команды. **Тип** – integer. **V**= 4b.
* **HY** – желтые карточки домашней команды. **Тип** – integer. **V**= 4b.
* **AY** – желтые карточки гостевой команды. **Тип** – integer. **V**= 4b.
* **HR** – красные карточки домашней команды. **Тип** – integer. **V**= 4b.
* **AR** - красные карточки гостевой команды. **Тип** – integer. **V**= 4b.

***Объем документа в среднем:*** 120b.

**Объем всей БД:** V = 120b \* N, где N количество документов в БД (N=4028).

***V = 483360b = 472kb.***

***Избыточность*** данных за счет того, что такие поля, как название команды 1, название команды 2, имя рефери, хранятся несколько раз. Пусть средняя длина строки названия команды - 12, количество команд - 20, длина строки, содержащей имя судьи - 10, количество судей - 7. Тогда объем, необходимый для хранения этих данных V = 12 \* 20 + 10 \* 7 = 310b.

* + 1. **Запросы.**

***Запрос на добавление матча***

db.matchs.insert\_one({ "AC": 5,"AF": 11, "AR": 0, "AS": 10, "AST": 9, "AY": 2, "AwayTeam": "Crystal Palace", "Date": "2018-08-11", "FTAG": 2, "FTHG": 0, "FTR": "A", "HC": 5, "HF": 9, "HR": 0, "HS": 15, "HST": 6, "HTAG": 1, "HTHG": 0,"HTR": "A", "HY": 1, "HomeTeam": "Fulham", "Referee": "M Dean" },)

Для добавления матча нужен один запрос.

***Запрос на поиск матча***

Для поиска матча нужен один запрос.

db.matchs.find( {"AC": 9, "AF": 11, "AR": 0, "AS": 17, "AST": 8, "AY": 3, $or:[{"AwayTeam" : "West Ham","HomeTeam" : "North Ham"},{"AwayTeam" : "North Ham","HomeTeam" : "West Ham"}], "Date": "2014-08-23", "FTAG": 3, "FTHG": 1, "FTR": "A", "HC": 6, "HF": 14, "HR": 0} )

***Запрос на получение статистики команды***

db.matchs.aggregate([ {"$match": {"$or": [{"AwayTeam":"Chelsea"}, {"HomeTeam": "Chelsea"}]}}, {"$group": {"\_id": "null","Matches": {"$sum": 1},"FTHG": {"$sum": "$FTHG"},"HS": {"$sum": "$HS"},"AS": {"$sum": "$AS"},"AC": {"$sum": "$AC"}, "AF": {"$sum": "$AF"},"AR": {"$sum": "$AR"}, "AS": {"$sum": "$AS"}, "AST": {"$sum": "$AST"}, "AY": {"$sum": "$AY"}, "FTAG": {"$sum": "$FTAG"}, "FTHG": {"$sum": "$FTHG"}, "FTR": {"$sum": "$FTR"},"HC": {"$sum": "$HC"},"HF": {"$sum": "$HF"},"HR": {"$sum": "$HR"},"HS": {"$sum": "$HS"},"HST": {"$sum": "$HST"},"HTAG": {"$sum": "$HTAG"},"HTHG": {"$sum": "$HTHG"},"HTR": {"$sum": "$HTR"},"HY": {"$sum": "$HY"},}} ])

Для получения статистики команды нужен один запрос.

***Запрос на сравнение двух команд***

db.matchs.find({"$or": [ {"$and": [{"AwayTeam": teamA}, {"HomeTeam": teamB}]}, {"$and": [{"AwayTeam": teamB}, {"HomeTeam": teamA}]} ]}, {"\_id": 0}).sort("Date")

Для сравнения двух команд нужен один запрос.

* 1. **SQL модель данных.**

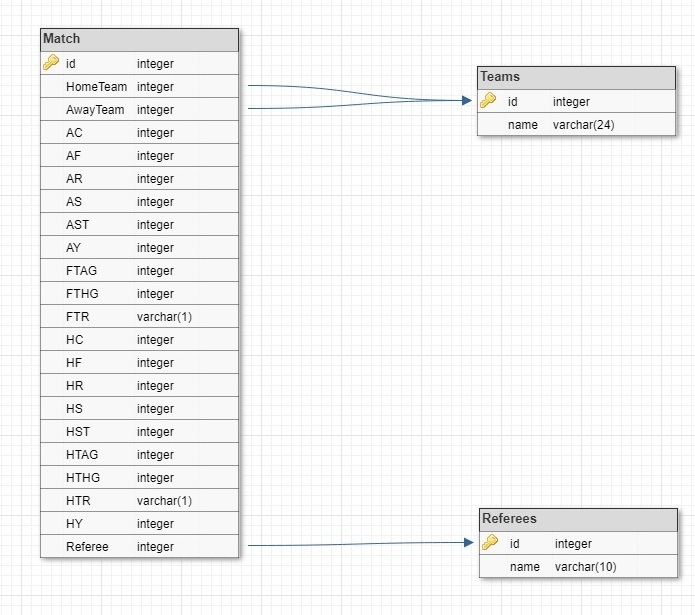


Рисунок 7 – SQL модель данных.

* + 1. **Описание и расчет объема.**

***База данных состоит из 3 таблиц:***

**1) Match** – таблица, содержащая информацию о матче. Включает следующие поля:

* **Id** - порядковый номер матча (является Primary Key). **Тип** – integer. **V**= 4b.
* **HomeTeam** и **AwayTeam** – идентификаторы команд (являются Foreign Key) **Тип** каждого – integer. **V**= 4b.
* **FTHG** – голы домашней команды за Full Time. **Тип** – integer. **V**= 4b.
* **FTAG** – голы гостевой команды за Full Time. **Тип** – integer. **V**= 4b.
* **FTR** - результат за Full Time. **Тип** - string (varchar(N)). **V** = 1b, т.к. N=1 - строка состоит из одной буквы (H=Home Win, D=Draw, A=Away Win).
* **HTHG** - голы домашней команды за Half Time. **Тип** – integer. **V**= 4b.
* **HTAG** - голы гостевой команды за Half Time. **Тип** – integer. **V**= 4b.
* **HTR** -результат за Half Time. **Тип** - string (varchar(N)). **V** = 1b, т.к. N=1 - строка состоит из одной буквы (H=Home Win, D=Draw, A=Away Win).
* **HS** - удары домашней команды. **Тип** – integer. **V**= 4b.
* **AS** - удары гостевой команды. **Тип** – integer. **V**= 4b.
* **HST** - удары в цель домашней команды. **Тип** – integer. **V**= 4b.
* **AST** - удары в цель гостевой команды. **Тип** – integer. **V**= 4b.
* **HF** - фолы домашней команды. **Тип** – integer. **V**= 4b.
* **AF** - фолы гостевой команды. **Тип** – integer. **V**= 4b.
* **HC** - угловые домашней команды. **Тип** – integer. **V**= 4b.
* **AC** - угловые гостевой команды. **Тип** – integer. **V**= 4b.
* **HY** - желтые карточки домашней команды. **Тип** – integer. **V**= 4b.
* **AY** - желтые карточки гостевой команды. **Тип** – integer. **V**= 4b.
* **HR** - красные карточки домашней команды. **Тип** – integer. **V**= 4b.
* **AR** - красные карточки гостевой команды. **Тип** – integer. **V**= 4b.
* **Referee** – идентификатор судьи матча. (является Foreign Key). **Тип** – integer. **V**= 4b.

***Итог: V = 82b.***

**2) Teams** – таблица, содержащая названия команд.

Включает следующие поля:

* **Id** - порядковый номер команды (является Primary Key) – 4b
* **Name** – название команды. **Тип** - string (varchar(N)). **V** = varchar(24) = 24b.

***Итог: V = 28b.***

**3) Refeeres** – таблица, содержащая имена судей.

Включает следующие поля:

* **Id** - идентификатор судьи (является Primary Key). **Тип** – integer. **V**= 4b.
* **Name** – имя судьи. **Тип** - string (varchar(N)). **V** = varchar(10) = 10b.

***Итог: V = 14b.***

Рассчитаем суммарный объем для хранения записей: V = N \* 88 + Nt \* 28 + Nr \* 14, где N = 4028 - количество матчей, \_Nt = 20 \_ - среднее количество команд, Nr = 7 - среднее количество судей.

Таким образом, объем для хранения N записей:

**V = N \* 88 + 658**

***Результат:* V = 4028 \* 88 + 658 = 355122b = 346,8kb**

* + 1. **Запросы.**

Для добавления записи в таблицу используется один запрос.

*Примеры:*

* Запрос на добавление команд:

INSERT INTO Teams (“Man United - Leicester”)

* Запрос на добавление судьи:

INSERT INTO Referees(“A Marriner”)

* Запрос на добавление матча:

INSERT INTO Match(1,7,2,5,3,2,5,2,4,1,5,5,1,1,0,4,1,0,1)

При поиске создается вложенность запросов, поэтому их количество увеличивается.

*Пример:*

* Запрос на получение всех матчей домашней команды по названию команды:

SELECT \* FROM Matchs(WHERE HomeTeam = (SELECT id FROM Teams WHERE name = “Man United - Leicester”) )

Также количество запросов будет расти за счет подготовленных запросов, которые используются для предотвращения возникновения sql- инъекций.

* 1. **Сравнение SQL и NoSQL.**

**1.** SQL модель выигрывает по количеству занимаемого объема *(346,8kb - SQL vs 472kb - NoSQL)*.

**2.** В NoSQL происходит избыточность данных, за чёт того, что мы повторно храним такие поля, как название команды 1, название команды 2, имя рефери.

**3.** В SQL потребовалось создать целых 3 таблицы. В NoSQL используется всего одна коллекция. Выигрывает NoSQL.

**4.** Количество запросов для NoSQL модели данных гораздо меньше, чем для SQL. Однако запросы SQL модели более читабельны, в них проще вносить изменения.

**5.** NoSQL реализация будет выигрывать в структуре и скорости сложных запросов.

Выбор сделан в пользу NoSQL модели.

1. **Разработанное приложение.**
   1. **Краткое описание.**

Веб-приложение. В БД хранятся матчи Английской Премьер-лиги. Пользователь приложения может осуществлять поиск матча по выбранным параметрам, производить сравнение двух команд, а также получить информацию о статистике каждой команды.

* 1. **Схема экранов приложения.**

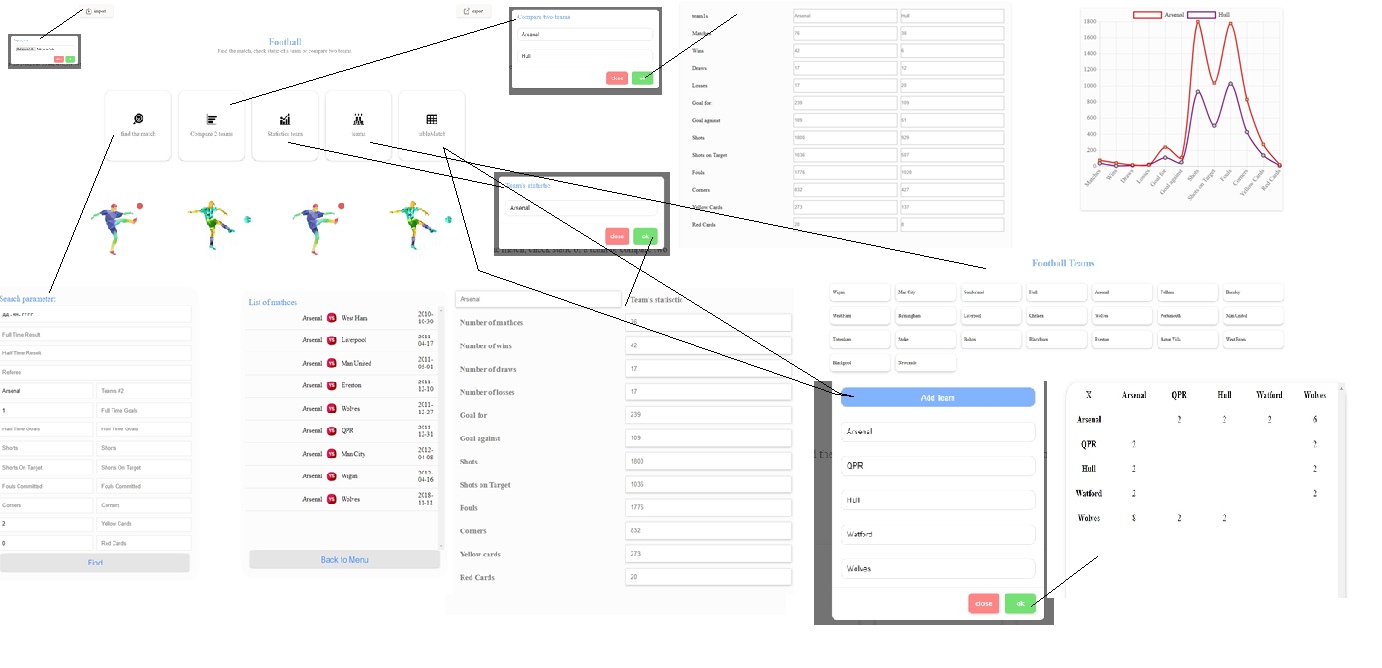
****

Рисунок 8 – Схема экранов приложения

* 1. **Использованные технологии.**
* Python 3.7.
* MongoDB
* Flask- Python microframework based on Werkzeug, Jinja 2 and good intentions
* Html
* Сss
* Java script
  1. **Ссылки на приложение.**

Ссылка на приложение:

[https://github.com/moevm/nosql1h19-football-stats](https://github.com/moevm/nosql1h19-plot-anaysis)

1. **Вывод.**
   1. **Результаты.**

В ходе работы было разработано приложение для хранения, анализа и использования данных футбольных матчей. В учебных целях были добавлены импорт и экспорт датасетов. Также была добавлена сводная таблица с результатами матчей выбранных команд, а также список со всеми футбольными командами, участвующими в Английской Премьер-лиге. Была предусмотрена возможность добавлять новые команды.

* 1. **Недостатки и пути для улучшения полученного решения.**

На данный момент import больших датасетов может занимать длительное время. Одним из способов решения может быть использование другого алгоритма импорта, с написанием python скриптов для обработки датасета.

# **ПРИЛОЖЕНИЕ.**

**Документация по сборке и развертыванию приложения.**

Инструкция по локальному запуску:

1. установить виртуальную среду

python -m venv venv

1. создать виртуальную среду

virtualenv venv

1. активировать виртуальную среду

venv\Scripts\activate

1. установка нужных пакетов

pip install flask

pip install pymongo

pip install numpy

1. импортирование flask

python

import flask

exit()

1. установка переменной среду FLASK\_APP

set FLASK\_APP=nosql.py

1. запуск приложения

flask run

1. Перейти по адресу <http://localhost:5000/>

**Снимки экранов приложения.**

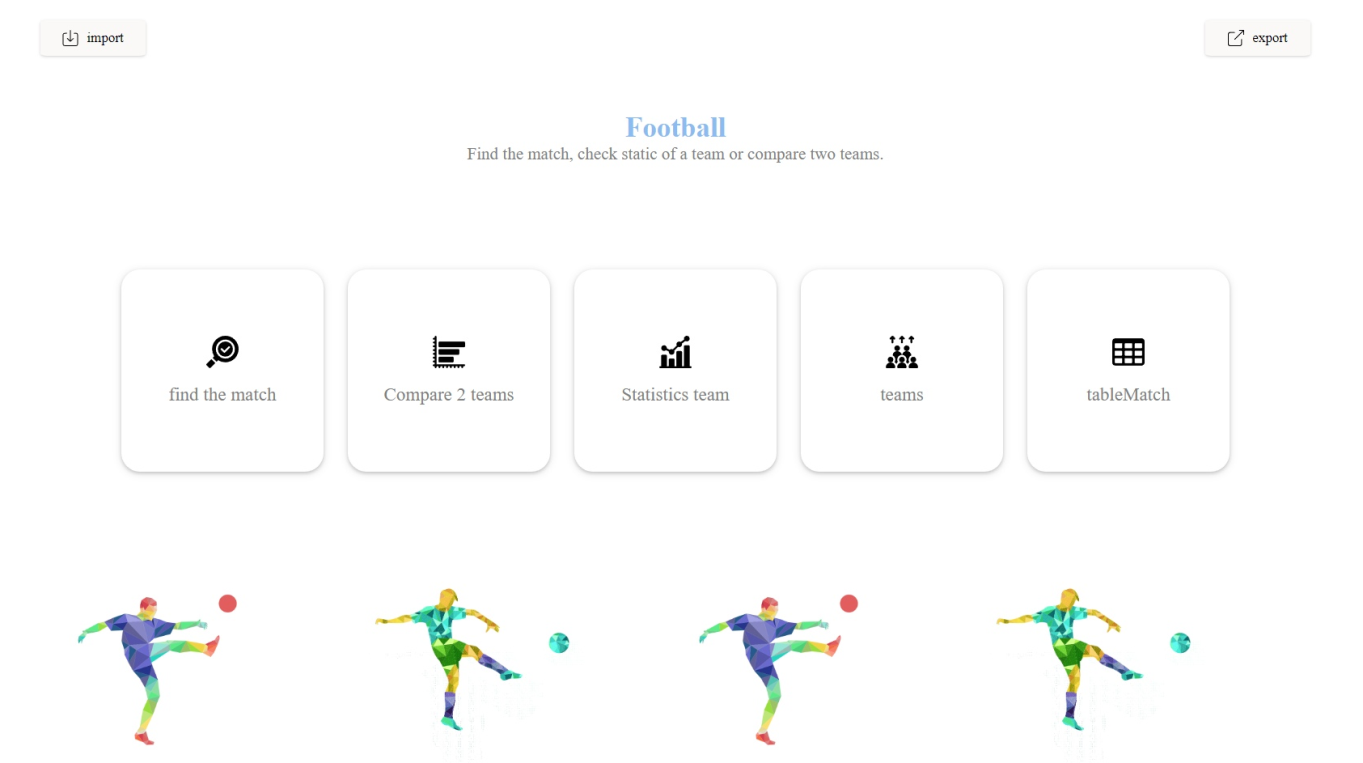


Рисунок 9 – Главная страница.

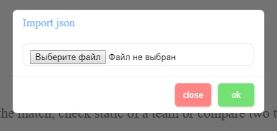


Рисунок 10 – Импорт базы данных.

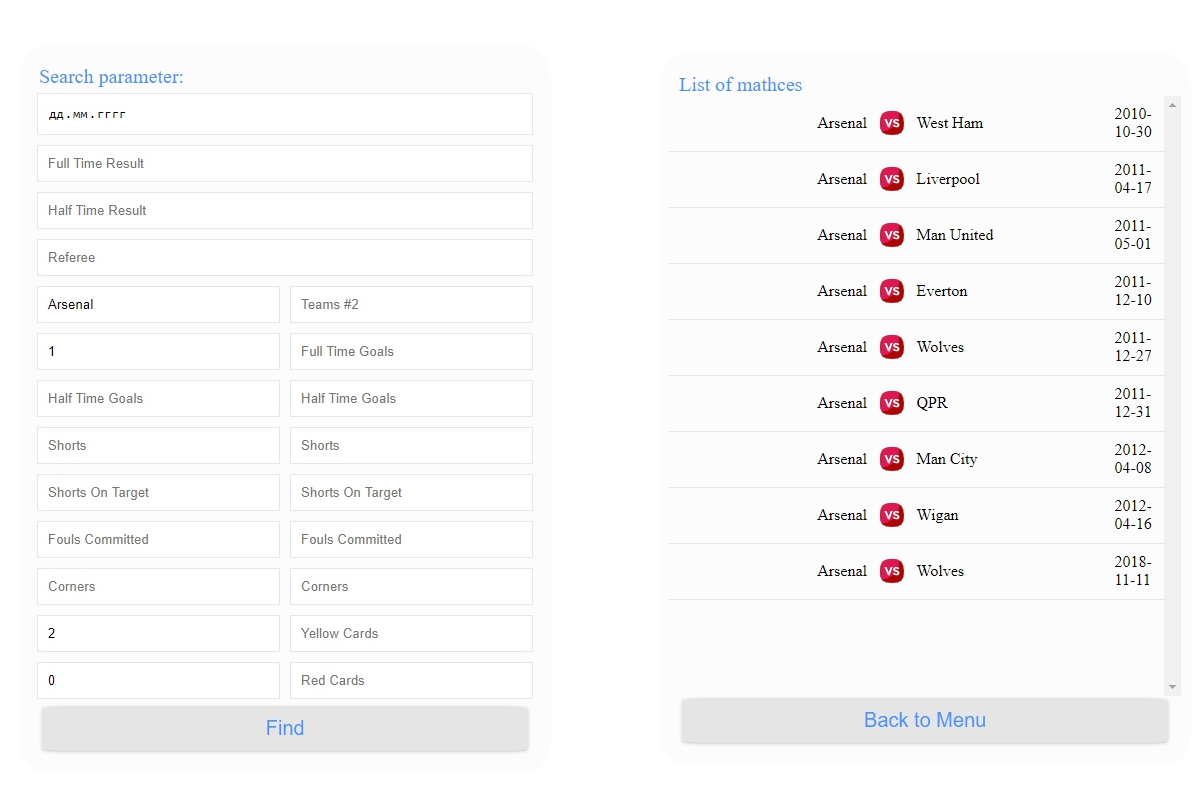


Рисунок 11-12 – Поиск матча.

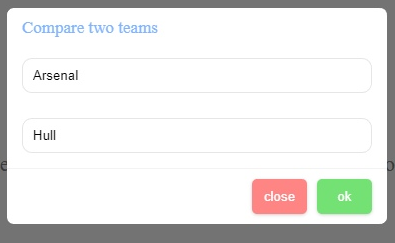




Рисунок 13-14 – Сравнение двух команд.

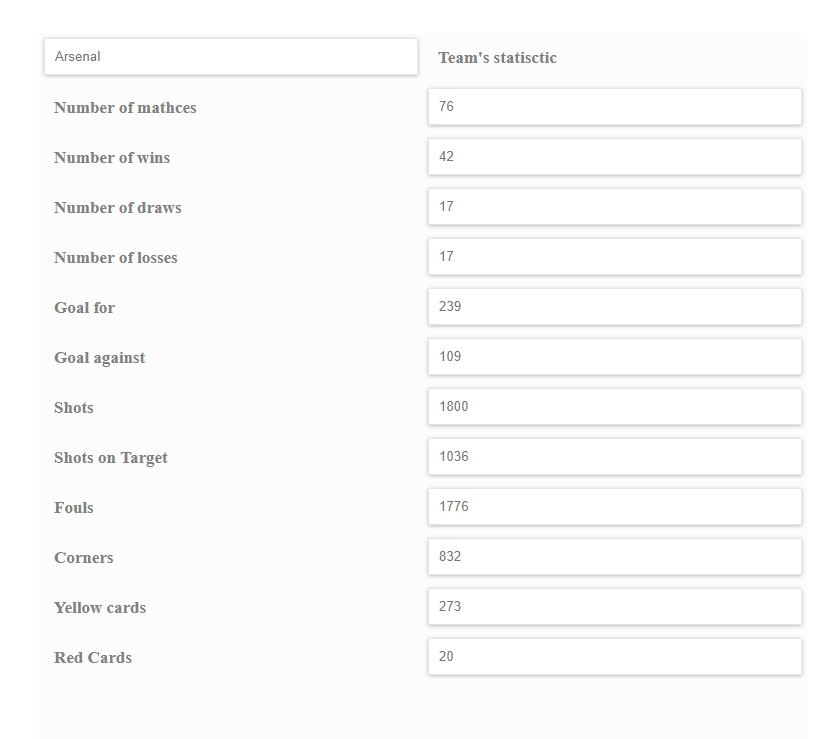
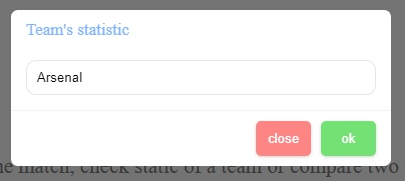
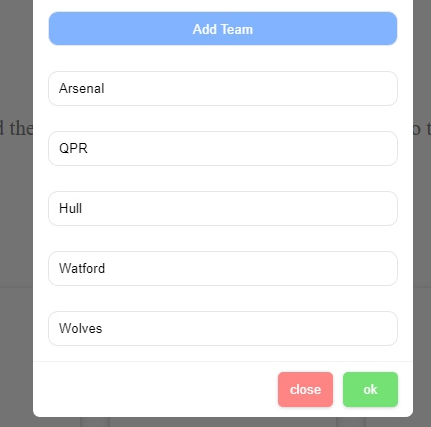


Рисунок 15-16 – Статистика команды.



Рисунок 17 – Список команд.



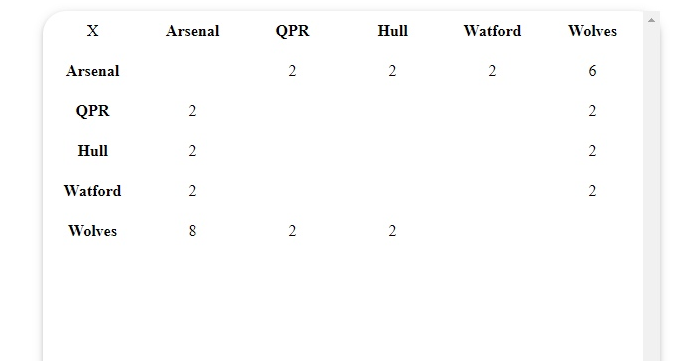


Рисунок 18-19 – Таблица с результатами матчей выбранных команд.

**Список литературы.**

* 1. Макет UI: <https://app.moqups.com/jNMJTa8zST/view/page/ad64222d5>
  2. Репозиторий проекта: <https://github.com/moevm/nosql1h19-football-stats>
  3. Документация MongoDB: <https://docs.mongodb.com>