

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Введение в нереляционные базы данных»**

**Тема: Разработка приложения для работы с каталогом исторических битв**

Студент гр. 7303

Студент гр. 7303

Студент гр. 7303

Преподаватель

Петров С.А.

Юсковец А.В.

Швец А.А.

Заславский М.М.

Санкт-Петербург

2020

## **ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студенты

Швец А.А.

Петров С.А.

Юсковец А.А.

Группа 7303

Тема курсовой работы: Разработка приложения для работы с каталогом исторических битв.

Исходные данные:

Необходимо реализовать приложение для работы с каталогом исторических битв с использованием MongoDB.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Качественные требования к решению»

«Макет пользовательского интерфейса»

«Сценарии использования»

«Модель данных»

«Разработанное решение»

«Исследование разработанного решения»

«Заключение»

Предполагаемый объем пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания: 00.00.2000

Дата сдачи реферата: 00.00.2000

Дата защиты реферата: 00.00.2000

Студент гр. 7303

Студент гр. 7303

Студент гр. 7303

Преподаватель

Петров С.А.

Юсковец А.В.

Швец В.А.

Заславский М.М.

## АННОТАЦИЯ

В данной курсовой работе представляется решение для работы с каталогом исторических битв, использующее в качестве базы данных MongoDB. Выделяются качественные требования к решению и прорабатываются сценарии использования. Представляется спроектированная модель данных, и проводится сравнение с моделью данных для реляционной СУБД. Представляется разработанное решение: веб-приложение с клиент-серверной архитектурой. Приводятся скриншоты работы приложения, описываются использованные технологии. Приводится исследование разработанного решения и выводы по дальнейшему улучшению.

Исходный код решения: <https://github.com/moevm/nosql2h20-battles-catalog/>

## SUMMARY

In this course work, we present a solution for working with the catalog of historical battles, using MongoDB as a database. Qualitative requirements for the solution are highlighted and usage scenarios are worked out. The designed data model is presented and compared with the data model for a relational DBMS. The developed solution is presented: a web application with a client-server architecture. Screenshots of the application are provided, and the technologies used are described. A study of the developed solution and conclusions for further improvement are presented.

Source code: <https://github.com/moevm/nosql2h20-battles-catalog/>

## ОГЛАВЛЕНИЕ

	Введение	6
1.	Качественные требования к решению	7
2.	Макет пользовательского интерфейса	8
3.	Сценарии использования	9
4.	Модель данных	13
5.	Разработанное решение	21
6.	Исследование разработанного решения	25
	Заключение	28
	Список литературы	29
	Приложение А. Документация по сборке и развертыванию решения	30

## **ВВЕДЕНИЕ**

Целью данной курсовой работы является решение для работы с каталогом исторических битв, исходники которого можно найти по следующей ссылке: <https://www.kaggle.com/residentmario/database-of-battles>.

Для достижения данной цели были поставлены следующие задачи:

- выделить качественные требования к решению;
- спроектировать сценарии использования
- спроектировать модель данных нереляционной СУБД
- провести сравнение моделей данных нереляционной и реляционной СУБД
- разработать решение
- провести анализ разработанного решения

## **1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ**

Требуется разработать приложение с использованием СУБД MongoDB [1] для работы с каталогом исторических битв.

Функционал:

1. Страница со списком войн, для каждой войны: даты, количество битв, длительность, стороны, численность армий с обеих сторон, потери. По параметрам можно производить сортировку. Возможность выбора войн для сравнения, сравнение представляет из себя графики (гистограммы).

2. Страница со списком битв, для каждой битвы: название, название войны, дата, длительность, численность армий, потери, командиры. Сортировка по параметрам, фильтрация по войнам. Аналогично войнам, можно произвести сравнение.

## **2. МАКЕТ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**

Подробный макет интерфейса вместе с картой переходов находится на вики-странице репозитория проекта по следующей ссылке:

<https://github.com/moevm/nosql2h20-battles-catalog/wiki/UI-mockup-and-UC>



### 3. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

#### 3.1. Сценарии использования

Ниже приведены сценарии использования приложения разбитые на несколько логических групп

- Список войн/битв
- Сравнение войн/битв
- Импорт данных
- Экспорт данных
- Создание битвы

#### Список войн/битв

##### Сортировка

1. Нажать на колонку по которой необходимо произвести сортировку.
2. Нажать на колонку второй раз, чтобы поменять порядок сортировки.
3. Нажать на колонку третий раз, чтобы сбросить сортировку.

Сортировать можно все поля из обеих таблиц кроме поля *Actors*.

1. Нажать на иконку с изображением "перевернутой пирамиды" справа от колонки.
2. В появившемся фильтре выбрать названия, которые должны быть в полях данной колонки.
3. Нажать на кнопку "Apply".

Для сброса фильтра повторить шаг 1, нажать на кнопку "Clear all" и, затем, на кнопку "Apply".

Фильтровать можно только по полям *War* и *Actors*.

**Детальная информация по участникам(*Actors*), размерам армий(*Army size*) и потерям(*Losses*)**

1. Навести курсор на интересующее поле в таблице.
2. Ввести часть названия в инпут расположенный над таблицей.
3. Нажать на кнопку с изображением лупы.

### **Навигация по разделам приложения**

1. После нажатия на кнопку в левом меню, будет произведен переход на соответствующий раздел приложения.

## **Сравнение войн/битв**

### **Выбор войн/битв, участвующих в сравнении**

1. Выбрать с помощью нажатия по строке интересующие войны/битвы.
2. Нажать на кнопку Compare в нижнем левом углу экрана.

### **Сброс выборки**

1. Нажать на "indeterminate" чекбокс.

### **Работа с модальным окном для сравнения войн/битв**

1. При наведении на бар появится тултип со значением бара и названием войны/битвы, к которой это значение относится.
2. При клике по области вне модального окна или нажатии на кнопку Esc, оно закроется.

## **Импорт данных**

1. Нажать на кнопку "Import".
2. Ознакомиться с требованиями к данным во всплывающем окне и нажать кнопку "Choose files".
3. В системном модальном окне выбрать файлы для импорта в базу данных.
4. Отправить файл нажатием на соответствующую кнопку в системном модальном окне.

В модальном окне возникнет ошибка, при отсутствии какого-либо файла из списка: actors.csv, battles.csv.

В модальном окне возникнет ошибка, при некорректном формате в файлах.

### **Экспорт данных**

1. Нажать на кнопку "Export".
2. Ознакомиться с требованиями к данным во всплывающем окне и нажать кнопку "Choose files".
3. В системном модальном окне выбрать файлы для импорта в базу данных.

### **Создание битвы**

1. Нажать на кнопку с изображением (+) в левом меню.

### **Заполнение необходимой для битвы информации**

Все поля формы в этом разделе обязательны.

Пара полей *War* и *Name* должны быть уникальны в рамках системы.

2. Заполнить поле *War* (Война, в рамках которой была битва).
3. Заполнить поле *Name* (Название битвы).
4. Заполнить поле *Start date* (Дата сражения).
5. Заполнить поле *End date* (Дата сражения).

### **Заполнение поля *Actors***

Необходимо создать как минимум двух участников битвы.

6. Добавить сторону (написав названия и нажав Enter или введя запятую).
7. После добавления сторон под полем *Actors* появится несколько форм на каждого участника.

### **Заполнение формы для участника**

Значения полей *Size* и *Losses* должны быть не меньше нуля.

Значения полей *Size* и *Army name* обязательны.

8. Заполнить поле *Army name* (Название армии).
9. Заполнить поле *Size* (Размер армии).
10. Заполнить поле *Losses* (Потери, которые понесла армия).
11. Заполнить поле *Commanders*, создав новых командиров (см. шаг 6).
12. Заполнить поле *Winner*, если сторона стала победителем, или оставить пустым иначе (Чекбокс для победителя).

### **Завершение создания битвы**

13. Нажать на кнопку "Save" для отправки данных на сервер и создания битвы или на кнопку "Cancel" для закрытия модального окна.

## 4. МОДЕЛЬ ДАННЫХ

### 4.1 NoSQL модель

#### 4.1.1. Графическое представление

```
{
  "battle_id": 123,
  "name": "battle name",
  "war": "war name",
  "datetime_min": "ISO 8601 start datetime",
  "datetime_max": "ISO 8601 end datetime",
  "actors": [
    {
      "actor_name": "actor name",
      "is_winner": "bool: true/false",
      "army_name": "army name",
      "initial_state": "number",
      "casualties": "number/unknown(0)",
      "commander": "name",
    },
  ],
}
```

#### 4.1.2. Описание назначений коллекций, типов данных и сущностей

База данных содержит единственную коллекцию - battles, которая состоит из вышеописанных документов и представляет собой список битв. Каждый документ описывает сущность "Битва".

#### 4.1.3. Оценка удельного объема информации, хранимой модели

Объем информации грубо вычисляется по формуле  $D * N$ , где  $D$  - средний объем документа,  $N$  - количество документов, рост объема модели линейно зависит от числа документов.

С помощью средств MongoDB был получен средний размер документа - 480 байт. Формула объема коллекции от количества документов -  $480 * N$ . При 1000 битв размер коллекции будет составлять приблизительно 480 000 байт.

#### 4.1.4. Избыточность модели

Модель является избыточной, так как названия сторон, названия армий и командиры могут быть одинаковыми в различных битвах, однако из-за вложенной структуры данные будут повторяться в документах различных битв.

Существует возможность выноса полей `actor_name`, `army_name`, `commander`, `war` в отдельные сущности, что уменьшит размер документа приблизительно на  $25 + 2 * 3 * 25$  байт (в качестве значений для всех трех полей используются не очень длинные строки, + по 4 байта на поля-ссылки, которые появятся вместо полей-строк). Таким образом "чистый" размер документа  $\sim 480 - 2 * 3 * 25 - 25 = 305$ . Размер документов для коллекций вынесенных полей  $\sim 12$  (ObjectId size) + 25 = 38 байт. Предположим, что акторов = битв = 1000, войн = 100, тогда чистый объем будет равен  $N * 305 + 3 * 38 * N + 38 * N/10 = 422.8 * N$ . Отношение чистого объема к реальному:  $422.8 / 480 = 0.88$ .

#### 4.1.5. Запросы к модели, реализующие варианты использования

1. Запрос для получения записей о битвах, с фильтрацией и сортировкой. В качестве примера приведен запрос с фильтрацией по войне и названию стороны, сортировка по возрастанию начала битвы, без вывода полей `_id` и `attacker_won`.

```
db.battles.find({"war": "Thirty Years' War of 1618-1648", "actors":
{$elemMatch: {"actor_name": "Holy Roman Empire"}}}, {"_id": 0,
"attacker_won": 0}).sort({"datetime_min":
1}).skip(skip_size).limit(limit)
```

Методы `skip` и `limit` необходимы для взаимодействия с фронтендом (для реализации пагинации)

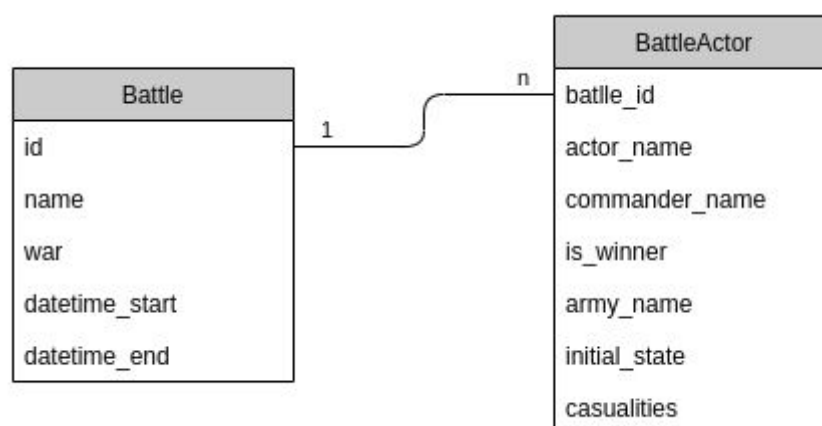
2. Запрос на получение записей о войнах (агрегация по битвам), с фильтрацией и сортировкой

```
db.battles.aggregate([
  {
    '$unwind': '$actors',
  },
  {
    '$group': {
      '_id': '$war' ,
      'actors': { '$push': '$actors' },
      'datetime_min': { '$min': '$datetime_min' },
      'datetime_max': { '$max': '$datetime_max' },
      'battles_num': { '$sum': 1 },
    },
  },
  {
    '$project': {
      '_id': 0,
      'name': '$_id',
      'datetime_max': 1,
      'datetime_min': 1,
      'battles_num': 1,
      'actors': 1,
    }
  },
  { '$match': query },
  { '$sort' : { sort : 1 } },
  {
    '$facet': {
      'wars': [
        { '$skip': limit * (page_num - 1) },
        { '$limit': limit }
      ],
      'total': [
        { '$count': 'count' }
      ]
    }
  }
])
```

Запрос агрегирует битвы по названию войн (агрегирует акторов, подсчитывает количество войн в каждой войне, собирает даты войны (самую раннюю дату, относящуюся в войне, и самую позднюю) и считает количество полученных войн. Аналогично запросу выше реализована пагинация.

## 4.2 SQL модель

### 4.2.1. Графическое представление



### 4.2.2. Описание назначений коллекций, типов данных и сущностей

Содержит следующие таблицы:

**Battle** – представляет сущность "Битва", содержит идентификатор, название, локацию, и время битвы;

**BattleActor** – представляет сущность "Сторона битвы", содержит идентификатор битвы, название стороны, имя командира. Дополнительно содержит данные о численности армии и потерях.



### 4.2.3. Оценка удельного объема информации, хранимой модели

Пусть  $B$ ,  $BA$  - размер записей в таблицах Battle, BattleActor.  $B_n$ ,  $BA_n$  - соответственно количество записей. Тогда объем модели будет рассчитываться по формуле  $B \cdot B_n + BA \cdot BA_n$ . При следующем определении таблицы battle:

```
CREATE TABLE IF NOT EXISTS battle(  
    id INTEGER PRIMARY KEY,  
    name VARCHAR(255),  
    war VARCHAR(255),  
    datetime_start DATETIME,  
    datetime_end DATETIME  
);
```

Размер записи (при условии что строковые поля заполняются данными длиной около 50 символов) приблизительно равен  $4 + 50 \cdot 2 + 23 \cdot 2 = 148$  байт (при условии что DATETIME представляется строкой вида "YYYY-MM-DD HH:MM:SS.SSS" – длина 23)

При следующем определении таблицы battle\_actor:

```
CREATE TABLE IF NOT EXISTS battle_actor(  
    id INTEGER PRIMARY KEY,  
    battle_id INTEGER KEY,  
    actor_name VARCHAR(255),  
    commander_name VARCHAR(255),  
    army_name VARCHAR(255),  
    is_winner bit,  
    initial_state INTEGER,  
    casualties INTEGER,  
    FOREIGN KEY(battle_id) REFERENCES battle(id)  
);
```

Размер записи (при условии что строковые поля заполняются данными длиной около 50 символов) приблизительно равен  $4 \cdot 2 + 50 \cdot 3 + 1 + 4 \cdot 2 = 167$  байт (при условии что DATETIME представляется строкой вида "YYYY-MM-DD HH:MM:SS.SSS" – длина 23)

Имея численные значения размеров записей, можно составить формулу зависимости объема коллекции от количества битв:

1. При условии, что акторы битв не повторяются, т.е. акторов =  $N*2$ , где  $N$  - количество битв, формула:  $148 * N + 167 * N * 2 = N * (148 + 167 * 2) = 482 * N$ . При  $N = 1000$ , объем равен 482 000 байт.

2. При условии, что акторы битв повторяются, т.е. акторов  $< N*2$ , где  $N$  - количество битв, возьмем количество акторов =  $N$  (грубо - каждый актор участвовал в двух битвах), формула:  $148 * N + 167 * N = N * (148 + 167) = 315 * N$ . При  $N = 1000$ , объем равен 315 000.

В расчетах учитывалось, что один символ в строковых типах занимает 1 байт (т.е. используются символы англ. алфавита)

#### **4.2.4. Избыточность модели**

Таблица BattleActor является избыточной, так как данные по названию армии и количеству битв будут повторяться в случаях, где у объединенной армии имеется несколько командиров.

При вынесении war, actor\_name, commander\_name, army\_name в отдельные коллекции, размер записи battle уменьшится на  $\sim 50$  байт, и станет  $147 - 50 + 4 = 101$  байт. Размер battle\_actor станет  $167 - 3 * 50 + 4 * 3 = 29$  байт. При условии, что количество акторов = количество битв = 1000, войн  $\sim 100$ , чистый размер модели  $\sim 101 * N + 29 * N + 54 * N / 1000 + 54 * 3 * N = 297.4 * N$ . Отношение чистого объема к реальному при этом:  $297.4 / 315 \sim 0.94$ .

#### 4.2.5. Запросы к модели, реализующие варианты использования

1. Запрос для получения записей о битвах, с фильтрацией и сортировкой.

```
SELECT *
FROM battle b
      LEFT JOIN battle_actor ba ON b.id = ba.battle_id
WHERE war = '<war name>' AND actor_name = '<actor name>'
ORDER BY datetime_end DESC;
```

2. Агрегация по войнам, суммирование численностей армий и потерь по акторам, с фильтрацией по войне.

```
SELECT war,
       actor_name,
       sum(initial_state),
       sum(casualties),
       min_date,
       max_date
FROM battle b
      LEFT JOIN battle_actor ba ON b.id = ba.battle_id
      LEFT JOIN (
        SELECT battle_id,
               MIN(datetime_start) as min_date,
               MAX(datetime_end) as max_date
        FROM battle b
              LEFT JOIN battle_actor ba ON b.id = ba.battle_id
        GROUP BY war
      ) dates ON b.id = dates.battle_id
WHERE war = '<war name>'
GROUP BY war, actor_name;
```

#### 4.3. Сравнение

##### Размеры моделей

NoSQL	SQL	SQL с повторением акторов (условия описаны выше)
480 * N	482 * N	315 * N

При уникальности акторов, NoSQL модель незначительно выигрывает по размеру, так как в рассматриваемых условиях связи между таблицами в SQL

модели не дают преимущества. Однако для более реального случая, в котором некоторые акторы участвуют в нескольких битвах, SQL модель является более оптимальной по памяти, так как избегает повторного включения одних и тех же данных, используя вместо этого связи между таблицами.

### **Запросы**

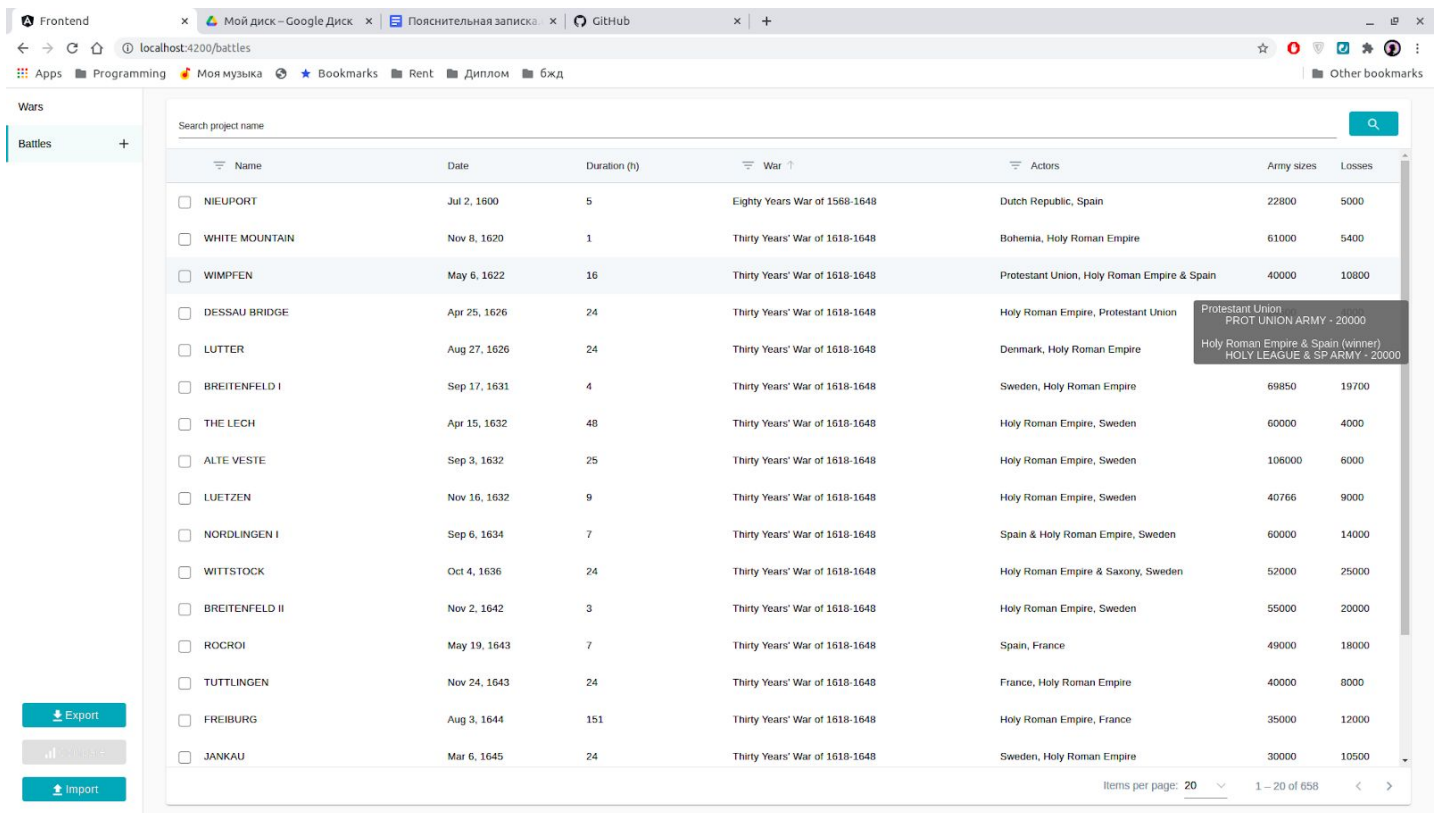
NoSQL модель выигрывает по количеству запросов к базе, необходимых для выполнения юзкейсов, так как состоит из одной коллекции, не имеет связей, не требует джойнов и т.д.

## 5. РАЗРАБОТАННОЕ РЕШЕНИЕ

Было разработано веб-приложение с классической клиент-серверной архитектурой. Приложение включает в себя:

1. Frontend (клиент) - веб-приложение, написанное на Angular [2]. Отвечает за обработку пользовательского ввода, и отображение данных, полученных с сервера с помощью REST API.
2. Backend (сервер) - веб-сервис, написанные на Python [3] с использованием асинхронного фреймворка FastAPI [4]. Отвечает за всю логику приложения, все операции с данными (сортировка, фильтрация, пагинация) осуществляются на данном сервисе.

### Работа приложения.



Name	Date	Duration (h)	War	Actors	Army sizes	Losses
<input type="checkbox"/> NIEUPORT	Jul 2, 1600	5	Eighty Years War of 1568-1648	Dutch Republic, Spain	22800	5000
<input type="checkbox"/> WHITE MOUNTAIN	Nov 8, 1620	1	Thirty Years' War of 1618-1648	Bohemia, Holy Roman Empire	61000	5400
<input type="checkbox"/> WIMPFEN	May 6, 1622	16	Thirty Years' War of 1618-1648	Protestant Union, Holy Roman Empire & Spain	40000	10800
<input type="checkbox"/> DESSAU BRIDGE	Apr 25, 1626	24	Thirty Years' War of 1618-1648	Holy Roman Empire, Protestant Union	Protestant Union PROT UNION ARMY - 20000 Holy Roman Empire & Spain (winner) HOLY LEAGUE & SP ARMY - 20000	
<input type="checkbox"/> LUTTER	Aug 27, 1626	24	Thirty Years' War of 1618-1648	Denmark, Holy Roman Empire		
<input type="checkbox"/> BREITENFELD I	Sep 17, 1631	4	Thirty Years' War of 1618-1648	Sweden, Holy Roman Empire	69850	19700
<input type="checkbox"/> THE LECH	Apr 15, 1632	48	Thirty Years' War of 1618-1648	Holy Roman Empire, Sweden	60000	4000
<input type="checkbox"/> ALTE VESTE	Sep 3, 1632	25	Thirty Years' War of 1618-1648	Holy Roman Empire, Sweden	106000	6000
<input type="checkbox"/> LUTZEN	Nov 16, 1632	9	Thirty Years' War of 1618-1648	Holy Roman Empire, Sweden	40766	9000
<input type="checkbox"/> NORDLINGEN I	Sep 6, 1634	7	Thirty Years' War of 1618-1648	Spain & Holy Roman Empire, Sweden	60000	14000
<input type="checkbox"/> WITTSTOCK	Oct 4, 1636	24	Thirty Years' War of 1618-1648	Holy Roman Empire & Saxony, Sweden	52000	25000
<input type="checkbox"/> BREITENFELD II	Nov 2, 1642	3	Thirty Years' War of 1618-1648	Holy Roman Empire, Sweden	55000	20000
<input type="checkbox"/> ROCROI	May 19, 1643	7	Thirty Years' War of 1618-1648	Spain, France	49000	18000
<input type="checkbox"/> TUTTLINGEN	Nov 24, 1643	24	Thirty Years' War of 1618-1648	France, Holy Roman Empire	40000	8000
<input type="checkbox"/> FREIBURG	Aug 3, 1644	151	Thirty Years' War of 1618-1648	Holy Roman Empire, France	35000	12000
<input type="checkbox"/> JANKAU	Mar 6, 1645	24	Thirty Years' War of 1618-1648	Sweden, Holy Roman Empire	30000	10500

Рисунок 1 - Экран битв

Frontend x Мой диск - Google Диск x Пояснительная записка x GitHub x +

localhost:4200/wars

Apps Programming Моя музыка Bookmarks Rent Диплом бжд Other bookmarks

Wars

Battles +

Search project name

Name	Dates	Duration	Battles	Actors	Army sizes	Losses
<input type="checkbox"/> Eighty Years War of 1568-1648	Jul 2, 1600 - Jul 2, 1600	<1	1	Dutch Republic, Spain	22800	5000
<input type="checkbox"/> Thirty Years' War of 1618-1648	Nov 8, 1620 - Aug 10, 1648	28	18	Bohemia, Denmark, France, Holy Roman Empire, Holy R...	847116	207100
<input type="checkbox"/> First English Civil War of 1642-1646	Oct 23, 1642 - Jun 14, 1645	3	6	Parliamentarian, Royalist	Bohemia - 21000 Denmark - 20000 France - 103000 Holy Roman Empire - 340770 Holy Roman Empire & Saxony - 30000 Holy Roman Empire & Spain - 20000 Protestant Union - 28500 Spain - 26000 Spain & Holy Roman Empire - 35000 Sweden - 222846	29507
<input type="checkbox"/> Second English Civil War of 1648-1649	Aug 17, 1648 - Sep 3, 1651	3	3	English Parliament, Scottish Royalist		
<input type="checkbox"/> The Fronde of 1648-1653	Jul 5, 1652 - Jul 5, 1652	<1	1	Conde Rebels, France		
<input type="checkbox"/> Franco-Spanish War of 1635-1659	Jun 14, 1658 - Jun 14, 1658	<1	1	France, Spain	27000	9000
<input type="checkbox"/> Austro-Turkish War of 1663-1664	Aug 1, 1664 - Aug 1, 1664	<1	1	Holy Roman Empire, Ottoman Empire	90000	10000
<input type="checkbox"/> Polish-Ottoman War of 1672-1676	Nov 11, 1673 - Nov 12, 1673	<1	1	Ottoman Empire, Poland	130000	31000
<input type="checkbox"/> Franco-Dutch War of 1672-1678	Jun 16, 1674 - Jun 28, 1675	1	5	Dutch Republic, France, Holy Roman Empire, Prussia, S...	266100	41500
<input type="checkbox"/> Great Turkish War of 1683-1699	Sep 12, 1683 - Sep 11, 1697	14	2	Holy Roman Empire, Ottoman Empire	333000	42500
<input type="checkbox"/> Monmouth Rebellion of 1685	Jul 6, 1685 - Jul 6, 1685	<1	1	England, English Rebels	7000	1800
<input type="checkbox"/> Nine Years' War of 1688-1697	Jul 27, 1689 - Oct 5, 1693	4	8	Dutch Republic, England, France, Jacobites, Savoy	580200	92230
<input type="checkbox"/> War of the Spanish Succession of 1701-1714	Aug 13, 1704 - Sep 11, 1709	5	4	England, England & Holy Roman Empire, France, France...	585000	124693
<input type="checkbox"/> Great Northern War of 1700-1721	Jun 28, 1709 - Jun 29, 1709	<1	1	Russia, Sweden	101500	10900
<input type="checkbox"/> Austro-Turkish War of 1716-1718	Aug 5, 1716 - Aug 5, 1716	<1	1	Habsburg Empire, Ottoman Empire	123000	10500
<input type="checkbox"/> War of the Austrian Succession of 1740-1748	Apr 10, 1741 - Dec 14, 1745	4	7	France, Great Britain, Habsburg Empire, Prussia, Saxony	506362	85915

Items per page: 20 1 - 20 of 64

Рисунок 2 - Экран войн

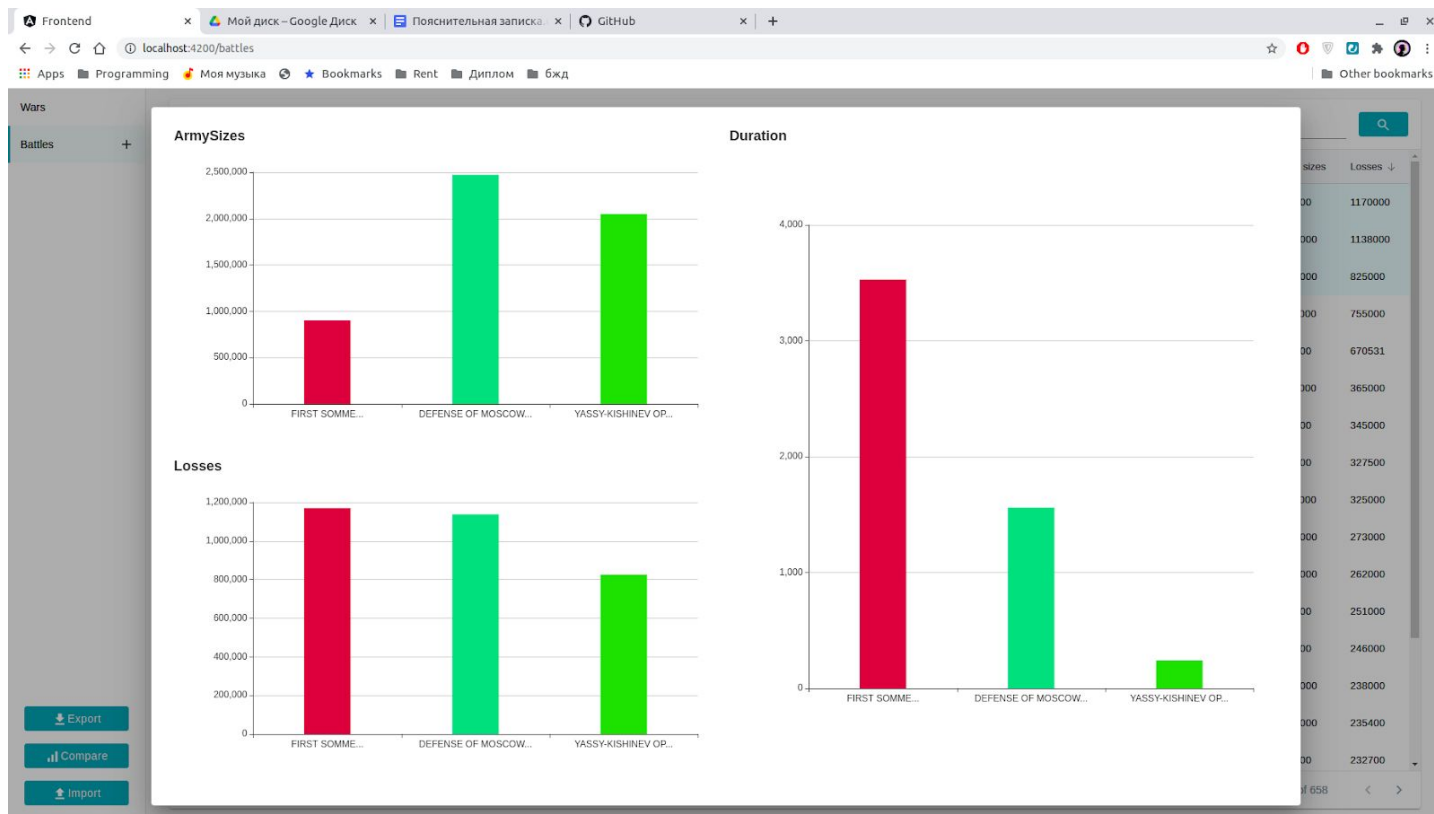


Рисунок 3 - Экран сравнения битв

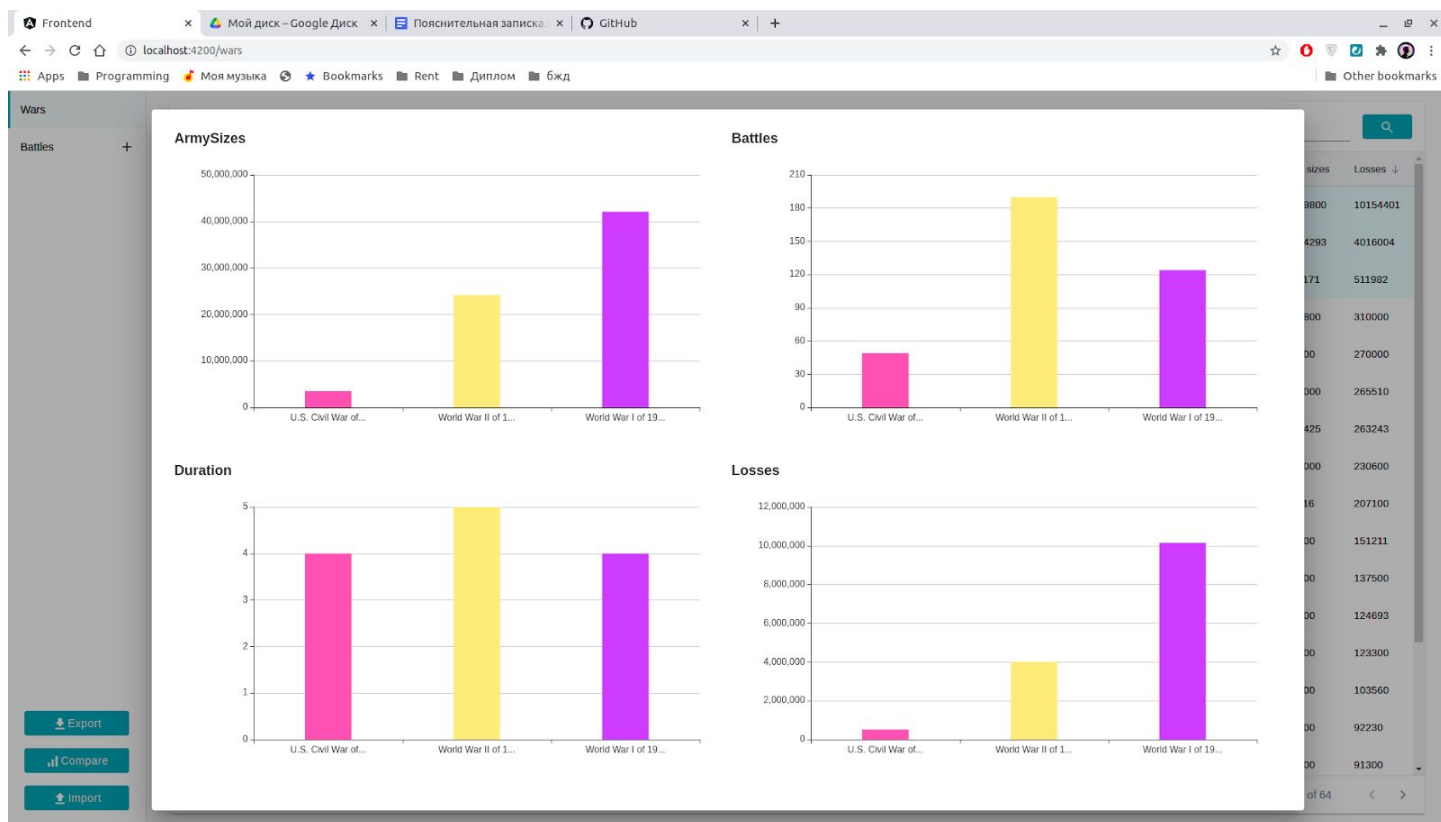


Рисунок 4 - Экран сравнения войн

The screenshot shows a web application interface with a modal dialog box titled 'Импорт данных' (Import data). The dialog provides instructions for importing CSV files and lists the required fields for the 'battles.csv' and 'actors.csv' files.

**Импорт данных**

Общие требования к файлам для импорта

1. Формат - csv, разделитель - запятая
2. Количество файлов - 2, названия - "battles.csv", "actors.csv"

Требования к формату файлов

Колонки файла battles.csv (описание битв):

1. battle\_id - уникальный id битвы
2. war - название войны, к которой относится битва - строка
3. name - название битвы - строка
4. datetime\_min - дата и время начала битвы в формате ISO 8061 - строка
5. datetime\_max - дата и время конца битвы в формате ISO 8061 - строка

Колонки файла actors.csv (описание сторон битв):

1. battle\_id - id битвы, к которой относится запись о стороне
2. actor\_name - название стороны - строка
3. is\_winner - является ли сторона победителем - number (1 - да, 0 - нет, 0 у всех акторов = ничья)
4. army\_name - название армии атакующей стороны - строка
5. initial\_state - численность армии стороны на момент начала битвы - число
6. casualties - потери стороны в битве - число
7. commander - имя командира армии стороны - строка

Пример данных

[Ссылка на подготовленные файлы](#)

Buttons: Cancel, Import

Рисунок 5 - Экран импорта данных

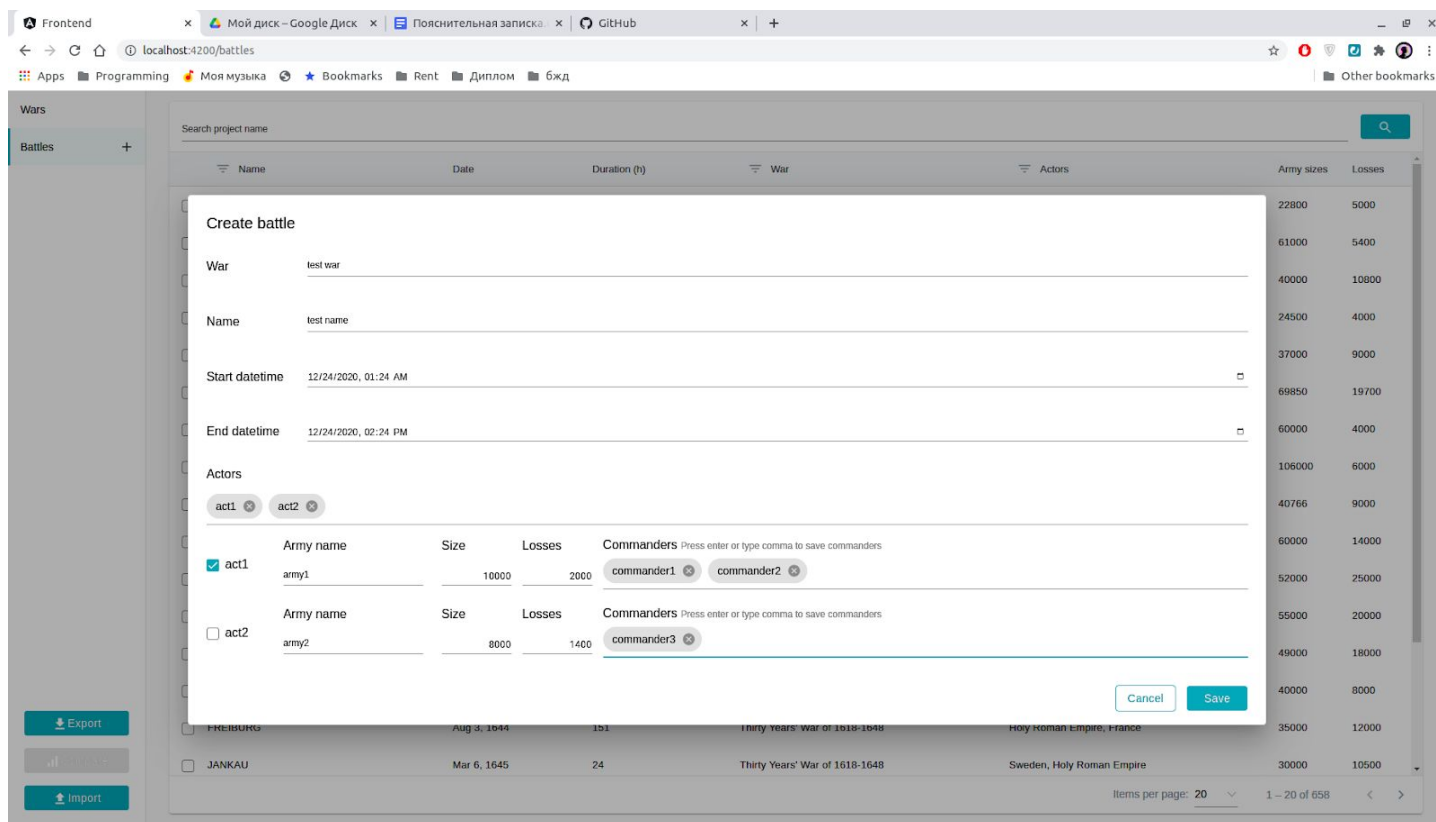


Рисунок 6 - Экран создания битвы

### Использованные технологии.

База данных: MongoDB

Back-end: Python3.6, FastAPI, Motor [5]

Front-end: JS (Angular)

### Ссылки на приложение.

Github: <https://github.com/moevm/nosql2h20-battles-catalog>



## 6. ИССЛЕДОВАНИЕ РАЗРАБОТАННОГО РЕШЕНИЯ

### 1. Размер базы данных

На рисунке 2 изображена статистика реального размера БД.

```
"ns" : "nosql2020-battles.battles",  
"size" : 324050,  
"count" : 658,  
"avgObjSize" : 492,
```

Рисунок 7 - Статистика

Из статистики видно, что реальная формула размера представляет собой  $492 \cdot N$ , в отличие от теоретической  $480 \cdot N$ . Это обусловлено тем, что на этапе разработки модель данных пришлось скорректировать, добавив два поля для суммарного количества армий и суммарных потерь (для сортировки по данным значениям). Добавленные два поля и составляют разницу между реальным и теор. значением размера бд. Однако, как и было посчитано теоретически, зависимость размера бд от числа документов линейна, разница между реальным размером и теоретическим не превышает  $2 \cdot N$  байт.

### 2. Время обработки запросов

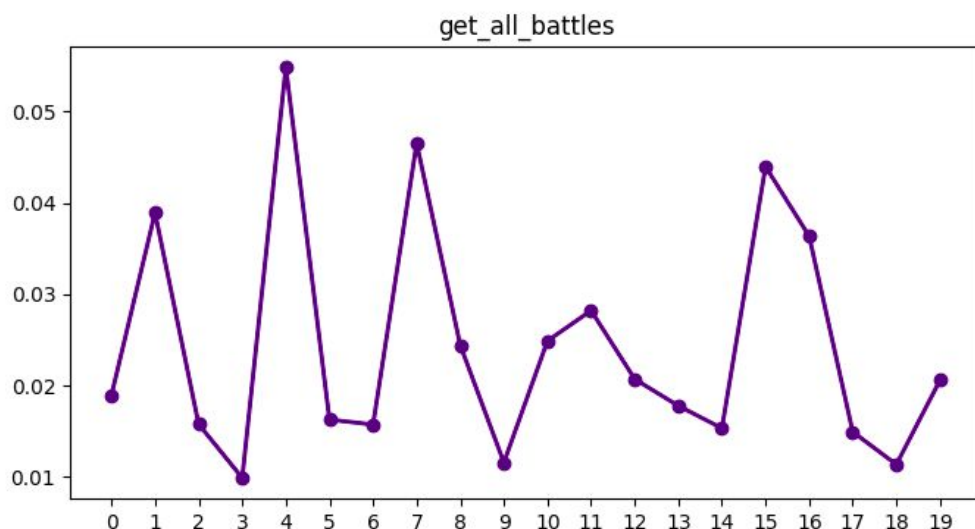


Рисунок 8 - Получение всех битв

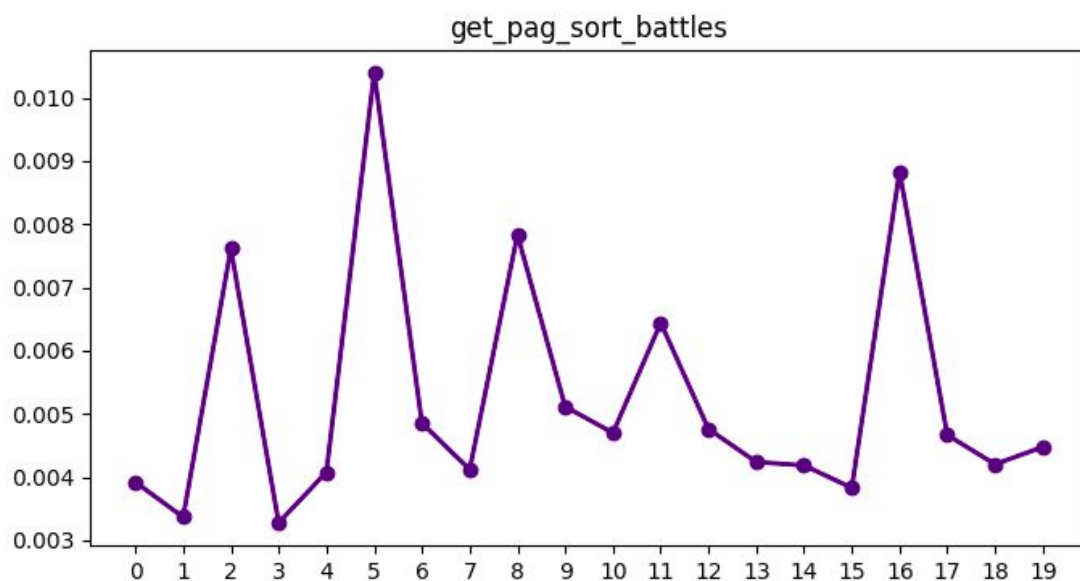


Рисунок 9 - Получение битв с пагинацией и сортировкой

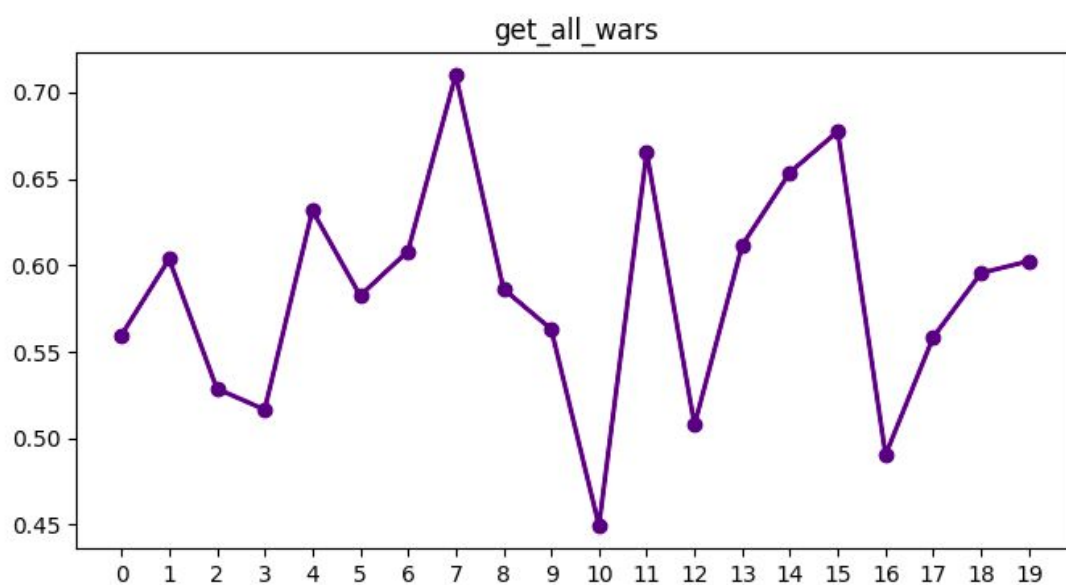


Рисунок 10 - Получение всех войн

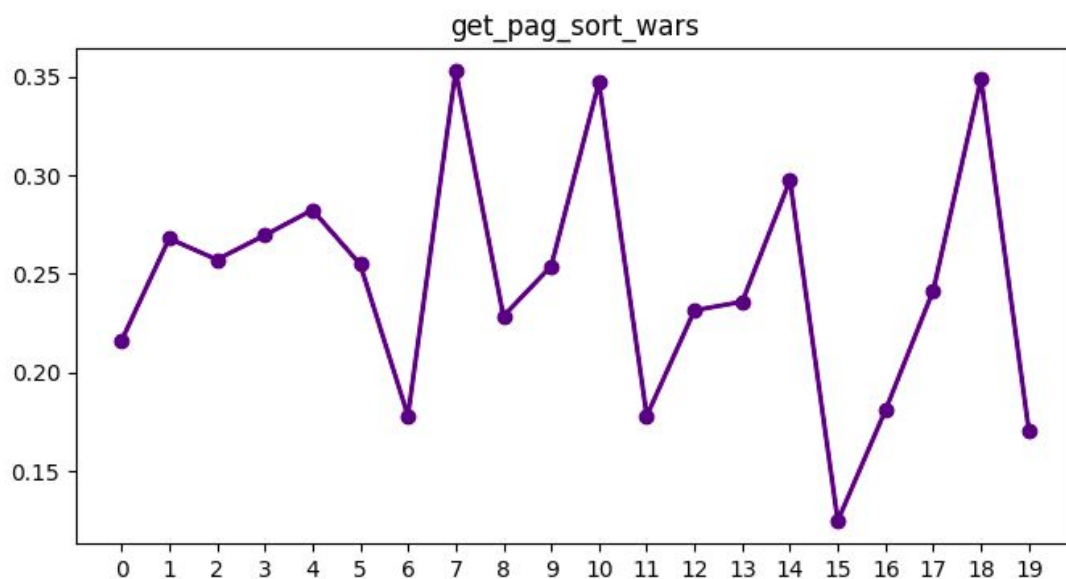


Рисунок 11 - Получение войн с пагинацией и сортировкой

По полученным результатам исследования были сделаны следующие выводы:

1. Рост объема бд соответствует формуле с погрешностью  $2 * N$  байт, где  $N$  - количество документов.
2. Запросы на получение битв обрабатывают достаточно быстро ( $\leq 0.1$  сек).
3. Запросы на получение войн требуют много времени ( $> 0.1$  сек).

Данный факт обусловлен решением не использовать коллекцию для войн: список войн составляется из коллекции битв с помощью механизма агрегации.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной курсовой работы было разработано веб-приложение для работы с каталогом исторических битв. В ходе разработки были выделены требования к решению, проработаны сценарии использования и макеты пользовательского интерфейса, спроектирована модель данных. Было проведено сравнение нереляционной модели данных с реляционным аналогом, из которого был сделан вывод о том, что NoSQL модель на реальных данных требует меньше запросов, но больше памяти для реализации требуемых сценариев использования. Приложение было разработано на основе клиент-серверной архитектуры: клиент на Angular, сервер на Python и FastAPI. Было проведено исследование разработанной решения, из которого был сделан вывод о существовании возможности для оптимизации запросов для войн к БД.

## СПИСОК ЛИТЕРАТУРЫ

1. Официальная документация MongoDB: [сайт]. URL: <https://docs.mongodb.com/manual>.
2. Официальный сайт Angular: [сайт]. URL: <https://angular.io/>.
3. Официальный сайт языка Python: [сайт]. URL: <https://www.python.org/>.
4. Официальная документация FastAPI: [сайт]. URL: <https://fastapi.tiangolo.com/>.
5. Официальная документация Motor: [сайт]. URL: <https://motor.readthedocs.io/en/stable/>.
6. Taneja S., Gupta P. R. Python as a tool for web server application development //JIMS8I-International Journal of Information Communication and Computing Technology. – 2014. – Т. 2. – №. 1. – С. 77-83.
7. Oluwatosin H. S. Client-server model //IOSR J Comput Eng (IOSR-JCE). – 2014. – Т. 16. – №. 1. – С. 67.

## ПРИЛОЖЕНИЕ А. ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ РЕШЕНИЯ

**Требования:** Python3.6+, MongoDB, nodejs, npm

Установка зависимостей

```
python3 -m venv venv
```

```
./venv/bin/activate
```

```
pip install -r ./backend/requirements.txt
```

**Запуск вручную**

Запуск дев-сервера (будет запущен на <http://localhost:3030>)

```
python -m backend
```

Запуск продакшн-сервера

```
gunicorn backend.app:app --bind=127.0.0.1:3030 -w 4 -k uvicorn.workers.UvicornH11Worker
```

Запуск фронтенда (клиента)

```
cd frontend
```

```
npm install
```

```
npm start
```

**Запуск через Docker**

```
docker-compose up -d
```

**Использование**

Перейти по адресу <http://localhost:4200>