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

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Введение в нереляционные базы данных»

Тема: ИС Справочник медицинских организаций

Студент гр. 9381 Николаев А.А.

Студент гр. 9381 Прашутинский К.И.

Преподаватель Заславский М.М.

Санкт-Петербург
2022

ЗАДАНИЕ

Студенты:

Николаев А.А.

Прашутинский К.И.

Группа 9381

Тема работы: ИС Справочник медицинских организаций

Исходные данные:

База данных медицинских организация. Необходимо разработать приложение, позволяющее выполнять поиск по медицинским организациям с фильтрацией и сортировкой, а также умеющее импортировать и экспортировать данные и собирать статистику.

Содержание пояснительной записки:

“Введение”, “Качественные требования к решению”, “Сценарии использования”, “Модель данных”, “Разработанное приложение”, “Заключение”, “Список литературы”

Предполагаемый объем пояснительной записки:

Не менее 19 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 26.12.2022

Дата защиты реферата: 26.12.2022

Студент		Николаев А.А.
Студентк		Прашутинский К.И.
Преподаватель		Заславский М.М.

АННОТАЦИЯ

Было разработано приложение, позволяющее выполнять поиск по медицинским организациям с фильтрацией и сортировкой, а также умеющее импортировать и экспортировать данные и собирать статистику. При разработке использовался React js. Создан прототип, который позволяет произвести поиск медицинских организаций по названиям с использованием фильтров и сортировок. Также была добавлена возможность импорта и экспорта данных и возможность просмотра профиля организации. Данные хранятся, обрабатываются, импортируются и экспортируются в формате json. Приложение запускается в docker.

SUMMARY

An application has been developed that allows you to search for medical organizations with filtering and sorting, as well as able to import and export data and collect statistics. React js was used during development. A prototype has been created that allows you to search for medical organizations by name using filters and sorting. The ability to import and export data and the ability to view an organization's profile has also been added. Data is stored, processed, imported and exported in json format. The application runs in docker.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель:

- Разработать справочник медицинских организаций.

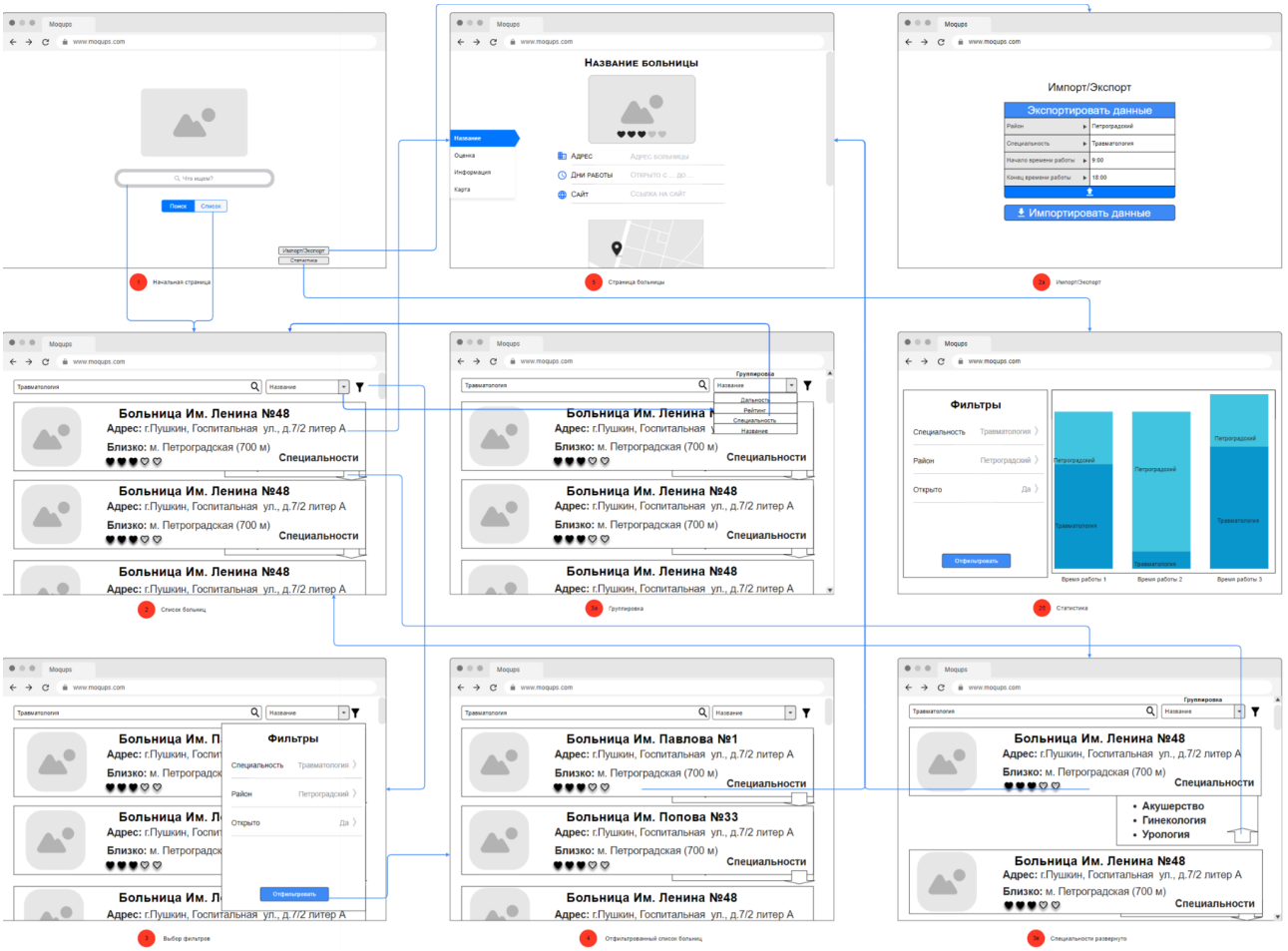
Задачи:

- Описать основные сценарии использования и макет.
- Разработать модель данных.
- Разработать прототип хранение и представление
- Разработать прототип анализ

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

1) Реализовать поиск по названию.
2) Реализовать фильтры и сортировки результатов поиска.
3) Добавить возможность просмотра профиля больницы.
4) Добавить импорт/экспорт данных в формате json.
5) Добавить возможность запуска приложения в docker.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ



1. Пользователь хочет найти больницу по названию.

Отображение: 1 -> 2

Действия пользователя: Пользователь переходит на главную страницу страницу. Основной сценарий: Пользователь вводит в строку поиска название больницы и нажимает кнопку поиска или клавишу Enter. Альтернативные сценарии:

Ошибка соединения -> 404

Результат: Пользователь видит базу данных с больницами, удовлетворяющими условию поиска. Пользователю доступны общие функции: он может искать больницы по названиям, сортировать и фильтровать базу по полям.

1.1 Пользователь хочет найти больницу из списка.

Отображение: 1 -> 2

Действия пользователя: Пользователь переходит на главную страницу. Основной сценарий: Пользователь нажимает на кнопку "список". Альтернативные сценарии:

Ошибка соединения -> 404
Результат: Пользователь видит базу данных со всеми больницами. Пользователю доступны общие функции: он может искать больницы по названиям, сортировать и фильтровать базу по полям.

2. Пользователь хочет отфильтровать больницы по Специальности/Району/Открытости.
Отображение: 2 -> 3 -> 4

Действия пользователя: Пользователь нажимает на кнопку фильтра, выбирает фильтры и нажимает кнопку "отфильтровать". Основной сценарий: Пользователь нажимает на иконку с фильтром, перед ним всплывает окно, где ему предлагается установить фильтры Специальность, Район и открыто ли сейчас. Он устанавливает фильтры, нажимает на кнопку "отфильтровать". Альтернативные сценарии:

Ошибка соединения -> 404
Результат: Пользователь видит базу данных с больницами, удовлетворяющими условию фильтрации. Пользователю доступны общие функции: он может искать больницы по названиям, группировать и фильтровать базу по полям.

2.1 Пользователь хочет сгруппировать больницы по Дальности/Рейтингу/Специальности/Названию.
Отображение: 2 -> 3а -> 2

Действия пользователя: Пользователь нажимает на кнопку "группировка", перед ним появляется список с вариантами ответа, пользователь выбирает по какому параметру группировать и база данных группируется. Основной сценарий: Пользователь нажимает на кнопку "группировка", перед ним появляется список с вариантами ответа:
Дальность/Рейтинг/Специальность/Название, пользователь выбирает по какому параметру группировать и база данных группируется. Альтернативные сценарии:

Ошибка соединения -> 404
Результат: Пользователь видит базу данных с больницами, удовлетворяющими условию группировки. Пользователю доступны общие функции: он может искать больницы по названиям, сортировать и фильтровать базу по полям.

2.2 Пользователь хочет сгруппировать больницы по Дальности/Рейтингу/Специальности/Названию.
Отображение: 2 -> 3а -> 2

Действия пользователя: Пользователь нажимает на кнопку "группировка", перед ним появляется список с вариантами ответа, пользователь выбирает по какому параметру группировать и база данных группируется. Основной сценарий: Пользователь нажимает на кнопку "группировка", перед ним появляется список с вариантами ответа:
Дальность/Рейтинг/Специальность/Название, пользователь выбирает по какому параметру группировать и база данных группируется. Альтернативные сценарии:

Ошибка соединения -> 404

Результат: Пользователь видит базу данных с больницами, удовлетворяющими условию группировки. Пользователю доступны общие функции: он может искать больницы по названиям, сортировать и фильтровать базу по полям.

3. Пользователь хочет посмотреть статистику по больницам

Отображение: 1 -> 2б Действия пользователя: Пользователь переходит по ссылке на главную страницу Альтернативные сценарии:

Ошибка соединения -> 404

Основной сценарий: Пользователь с главной страницы нажимает кнопку "Статистика". Пользователь видит станицу отображения статистики, поля выбора параметров, кнопку "Применить", поле вывода статистики. Пользователь выбирает параметры, по которым будет отобрана статистика, нажимает кнопку "Применить". Результат: Пользователь видит статистику больниц по указанным параметрам.

4. Пользователь хочет импортировать/экспортировать данные

4.1 Пользователь хочет импортировать данные

Отображение: 1 -> 2а Действия пользователя: Пользователь переходит по ссылке на главную страницу Альтернативные сценарии:

Ошибка соединения -> 404

Основной сценарий: Пользователь с главной страницы нажимает кнопку "Импорт/Экспорт". Пользователь видит станицу импорта/экспорта данных, поля выбора параметров экспорта, кнопку-значок экспорта данных, кнопку импорта данных "Импортировать данные". Пользователь нажимает на кнопку импорта данных "Импортировать данные", выбирает файл с данными, которые будут загружены. Результат: Пользователь импортирует данные.

4.2 Пользователь хочет экспортировать данные

Отображение: 1 -> 2а Действия пользователя: Пользователь переходит по ссылке на главную страницу Альтернативные сценарии:

Ошибка соединения -> 404

Основной сценарий: Пользователь с главной страницы нажимает кнопку "Импорт/Экспорт". Пользователь видит станицу импорта/экспорта данных, поля выбора параметров экспорта, кнопку-значок экспорта данных, кнопку импорта данных "Импортировать данные". Пользователь выбирает параметры, по которым будет отобраны данные для экспорта и нажимает на кнопку-значок экспорта данных. Результат: Пользователь экспортирует данные.

5. Пользователь хочет посмотреть карточку

Отображение: 2, 3в, 4 -> 5 Действия пользователя: Пользователь нажимает на карточку больницы из списка. Альтернативные сценарии:

Ошибка соединения -> 404

Основной сценарий: Пользователь переходит на страницу больницы. Пользователь видит название больницы, фотографию больницы (если имеется), адрес больницы, график работы, расположение на карте и т.п. Результат: Пользователь ознакамливается с данными, представленными на больнице.

3. МОДЕЛЬ ДАННЫХ

Схема БД и список сущностей

База данных содержит информацию о больницах. Ниже приведен список полей.

Hospitals

id
nameFull
nameShort
medicalSubjectId
medicalSubjectName
inn
kpp
ogrn
regionId
regionName
createDate
modifyDate
moAgencyKindId
moAgencyKind
addr
postIndex
cadastralNumber
latitude
longtitude
fiasVersion
aoidArea
aoidStreet
houseid
addrRegionId

addrRegionName

territoryCode

isFederalCity

areaName

prefixArea

streetName

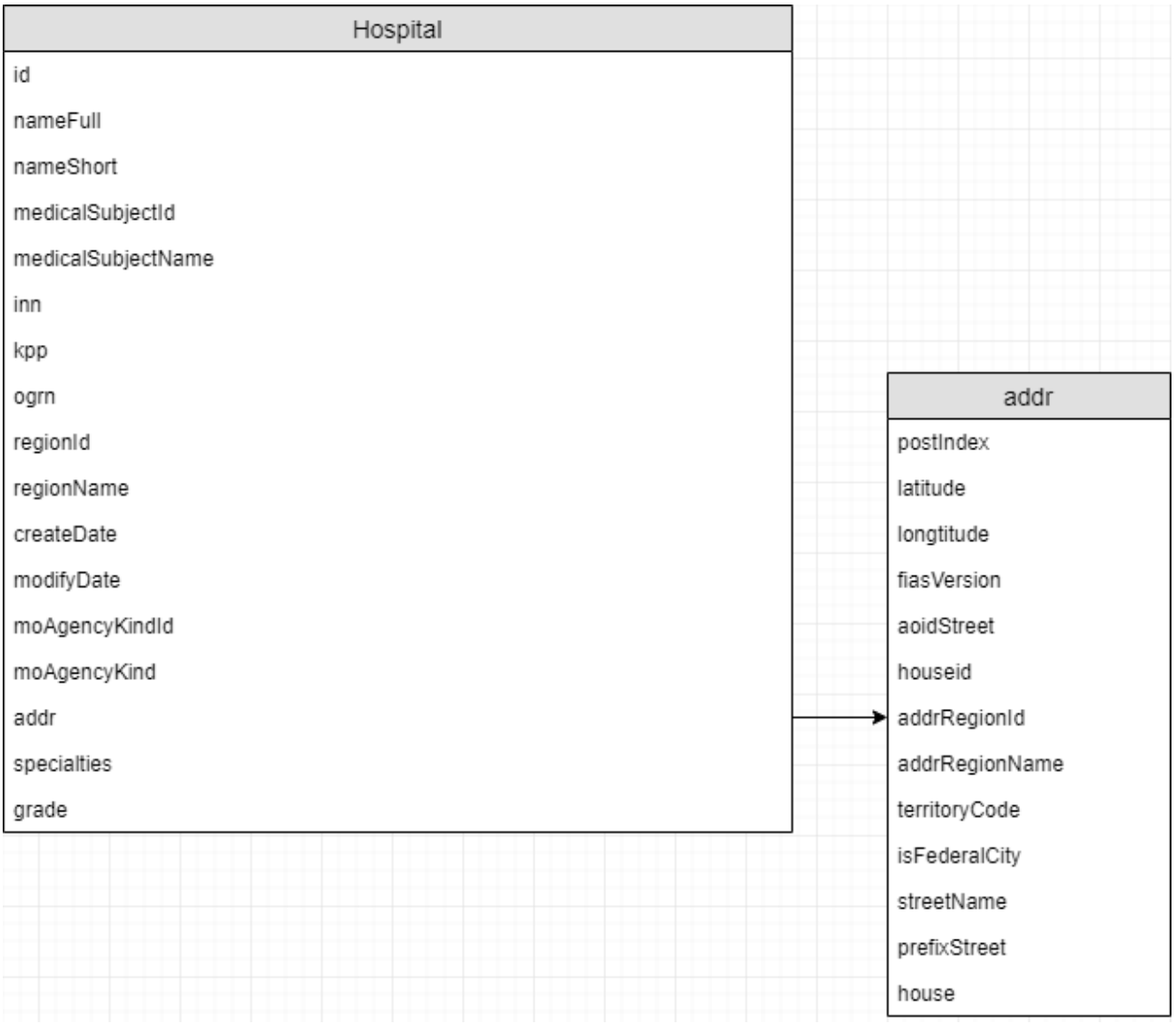
prefixStreet

house

specialties

grade

Нереляционная модель данных



Оценка удельного объема информации, хранимой в модели

Пусть в базе данных N больниц.

Максимальные размеры полей документа:

_id : 40 bytes
id : 28 bytes
nameFull : 760 bytes
nameShort : 668 bytes
medicalSubjectId : 28 bytes
medicalSubjectName : 130 bytes
inn : 61 bytes
kpp : 58 bytes
ogrn : 64 bytes
regionId : 51 bytes
regionName : 154 bytes
createDate : 59 bytes
modifyDate : 59 bytes
moAgencyKindId : 52 bytes
moAgencyKind : 232 bytes
grade : 28 bytes
specialties : 120 bytes
addr : 640 bytes

Максимальный размер базы данных: N * (40 + 28 + 760 + 668 + 28 + 130 + 61 + 58 + 64 + 51 + 154 + 59 + 59 + 52 + 232 + 28 + 120 + 640) = N * 3232

Избыточность модели

Модель не содержит дополнительных данных, помимо исходных, а значит не является избыточной.

Направление роста модели при увеличении количества объектов каждой сущности

Размер базы данных растет линейно по каждому параметру.

Запросы к модели, с помощью которых реализуются сценарии использования

Добавление

```
collection.insert_one({
   "id": 1678,
   "nameFull": "ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ 'НАЦИОНАЛЬНЫЙ МЕДИЦИНСКИЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР ОНКОЛОГИИ ИМЕНИ Н.Н. БЛОХИНА' МИНИСТЕРСТВА ЗДРАВООХРАНЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ",
   "nameShort": "ФГБУ 'НМИЦ ОНКОЛОГИИ ИМ. Н.Н. БЛОХИНА' МИНЗДРАВА РОССИИ",
   "medicalSubjectId": 2,
   "medicalSubjectName": "Медицинская организация",
   "inn": "7724075162",
   "kpp": "772401001",
   "ogrn": "1037739447525",
   "regionId": "77",
   "regionName": "г. Москва",
   "createDate": "16.10.2016",
   "modifyDate": "30.11.2022",
   "moAgencyKindId": "4",
   "moAgencyKind": "Специализированная больница",
   "grade": 5,
   "specialties": [
      "педиатрия",
      "фтизиатрия",
```

```
        "паразитология",
        "медицинская оптика",
        "ревматология",
        "психиатрия и судебно-психиатрическая экспертиза",
        "генетика"
    ],
    "addr": {
        "postIndex": "115478",
        "latitude": "55.656509",
        "longtitude": "37.643076",
        "fiasVersion": 333,
        "aoidStreet": "23667a40-8f8a-433a-9c78-78112ab013c4",
        "houseid": "39644630-a567-4794-b1fd-6b1f02bc8a43",
        "addrRegionId": 77,
        "addrRegionName": "г. Москва",
        "territoryCode": 45,
        "isFederalCity": "true",
        "streetName": "Каширское",
        "prefixStreet": "ш",
        "house": "24"
    }
})
```


Удаление

```
collection.delete_one('id' : '1678')
```

Обновление

```
collection.update_one({"_id" : "1678"}, {"$set" : {"addr" : some_addr}})
```

Поиск по полю

```
collection.find_one({"inn" : "7724075162"})
```

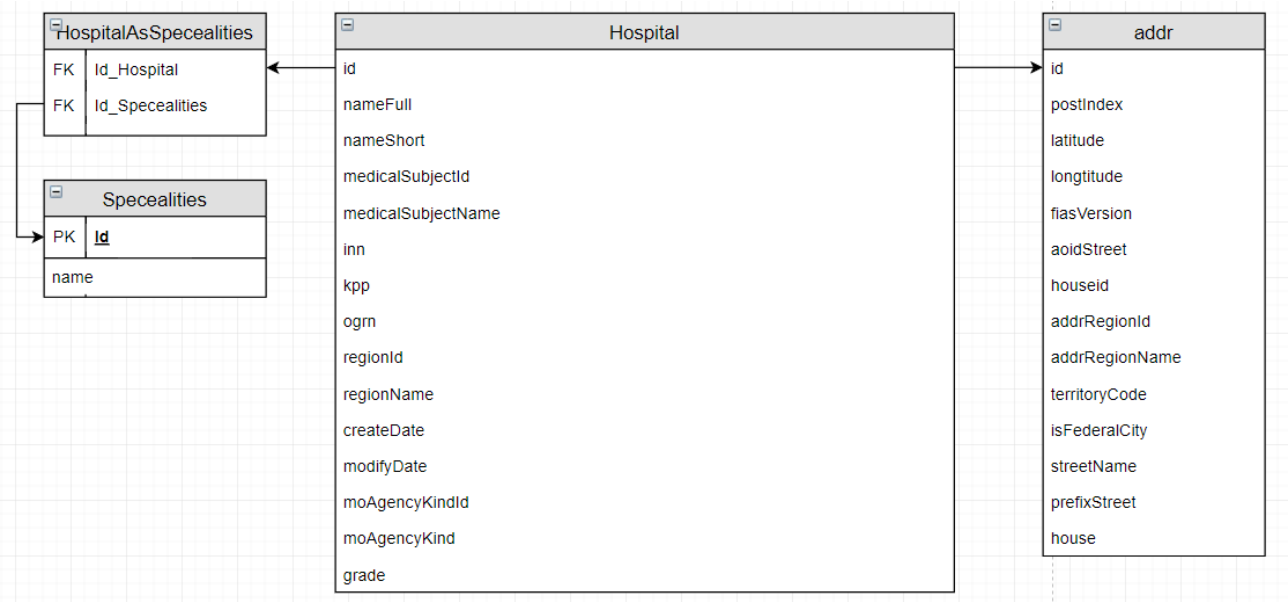
Сортировка

```
collection.find().sort("nameShort", pymongo.ASCENDING)
```

Поиск по нескольким полям

collection.find_one({"regionId" : "77", "grade": 5})

Аналог модели данных для SQL СУБД



Оценка удельного объема информации, хранимой в модели

Пусть в базе данных N больниц и 100 специальностей и в среднем 5 связей "Больница-Специальность".

N * (40 + 28 + 760 + 668 + 28 + 130 + 61 + 58 + 64 + 51 + 154 + 59 + 59 + 52 + 232 + 28 + 120) + N * 5 * (28 + 120) + 100 * (28 + 28) = N * 4000 + 5600

Избыточность модели

Модель не содержит дополнительных данных, помимо исходных, а значит не является избыточной.

Направление роста модели при увеличении количества объектов каждой сущности

Размер базы данных растет линейно по каждому параметру.

Запросы к модели, с помощью которых реализуются сценарии использования

Добавление

```
INSERT INTO Hospitals
    VALUES
    (...);
```

Удаление

```
DELETE FROM Hospitals
    WHERE id == 1678;
```

Обновление

```
UPDATE Hospitals
    SET addr = some_addr
    WHERE id == 1678;
```

Поиск по полю

```
SELECT *
    FROM Hospitals
    WHERE inn = '7724075162';
```

Сортировка

```
SELECT *
     FROM Hospitals
    ORDERED BY nameShort ASC;
```

Поиск по нескольким полям

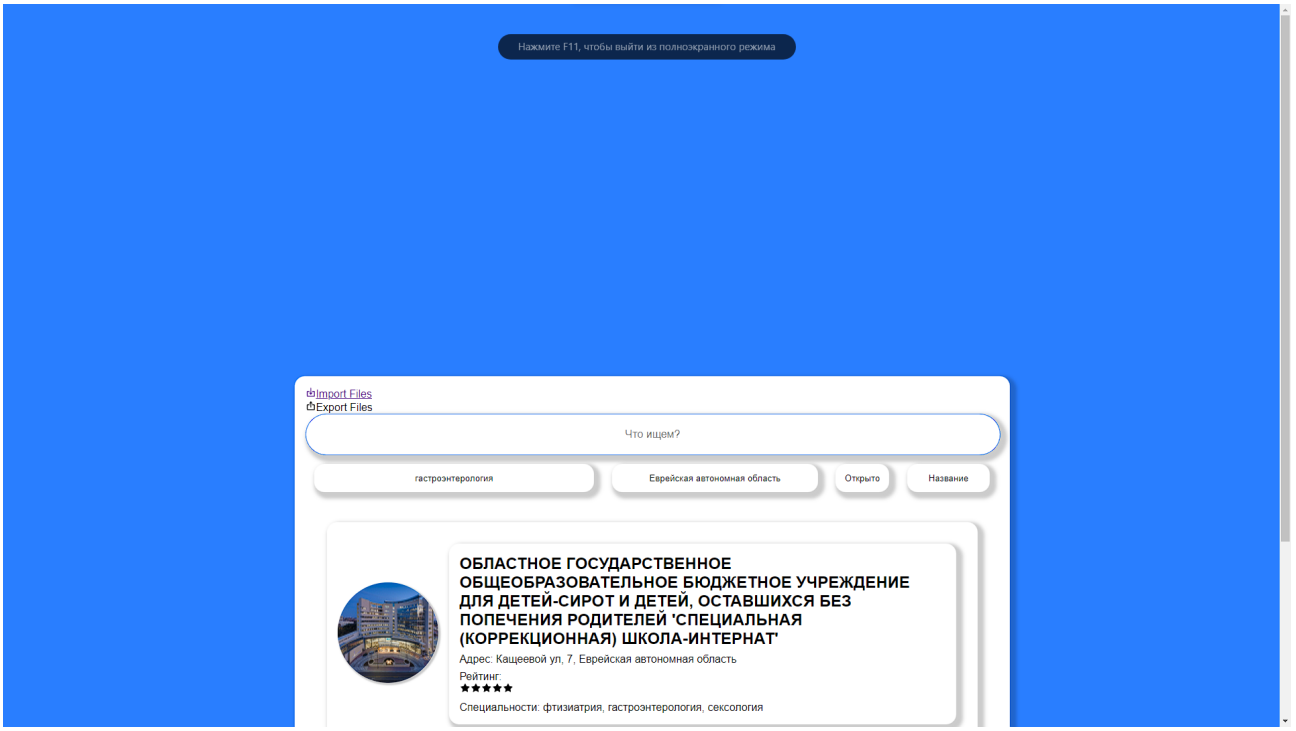
```
SELECT *
    FROM Hospitals
    WHERE grade = 5 AND actual = 'True' AND regionId = '77'
```

Сравнение моделей

В данной модели обнаружено преимущество нереляционной базы данных, она занимает в 1,04 раза меньше памяти. Обе модели покрывают сценарии use case за один запрос.

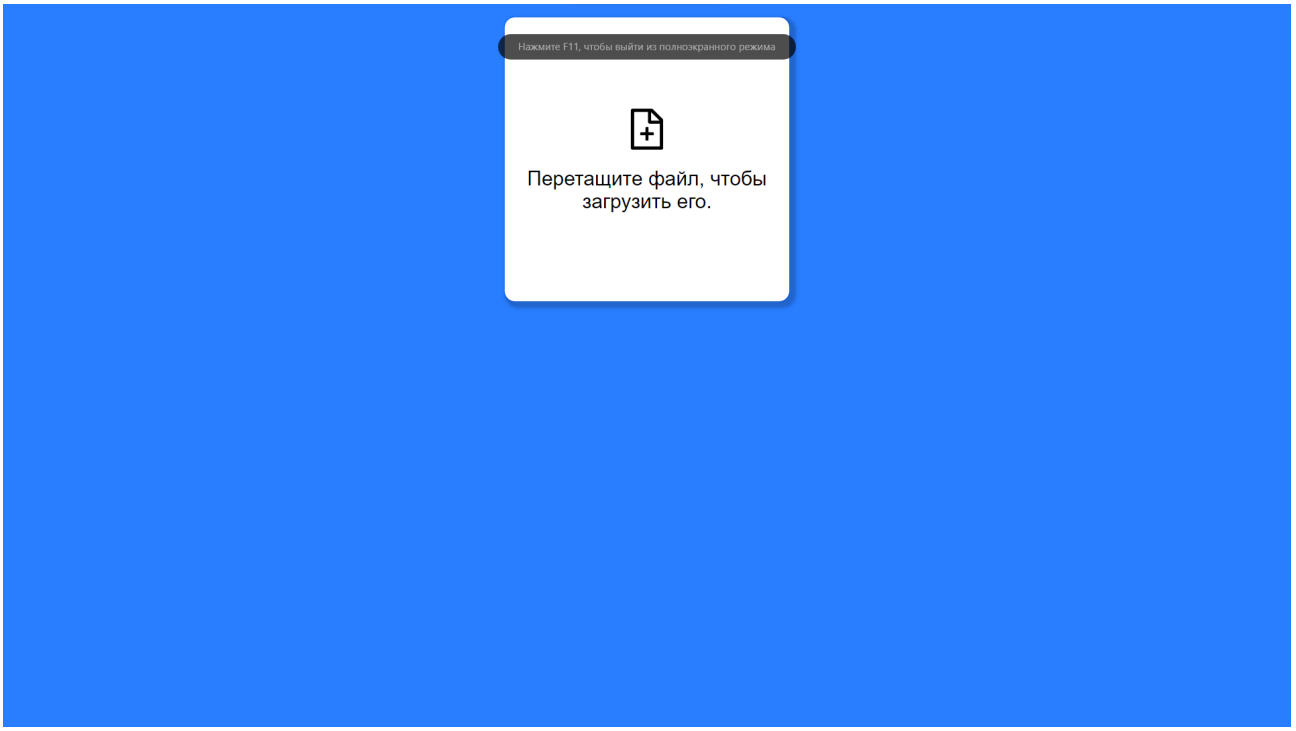
4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

При входе пользователя на сайт, его встречает вот такая страница поиска. На ней он может ввести название организации. Страница автоматически будет подгружать наиболее подходящие варианты. Также можно варианты отфильтровать. И экспортировать результаты.



При нажатии на карточку организации производится переход на её профиль.

При нажатии на кнопку импорт будет произведен переход на страницу импорта данных.



Сюда можно перенести файлы в формате json и они будут отправлены на сервер.

Использованные технологии

БД: MongoDB

Бэкенд: Python, Flask, pymongo

Фронтенд: HTML, JavaScript, React js ,CSS

5. ВЫВОДЫ

Было разработано справочное веб-приложение с медицинскими организациями, в котором возможен поиск необходимых организаций, добавление информации о них и просмотр их профиля.

6. Список литературы.

1. https://reactjs.org/docs/getting-started.html - документация React js
2. https://www.mongodb.com/docs/ - документация mongo db
3. https://www.python.org/doc/ - документация python