

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

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**  
**по дисциплине «Введение в нереляционные СУБД»**  
**Тема: Архив наблюдений по загрязнению среды**

Студентка гр. 8382	_____	Кузина А.М.
Студентка гр. 8382	_____	Кулачкова М.К.
Студент гр. 8382	_____	Янкин Д.О.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург

2021

## ЗАДАНИЕ

Студенты Кузина А.М., Кулачкова М.К., Янкин Д.О.

Группа 8382

Тема проекта: Архив наблюдений по загрязнению среды

Исходные данные:

Необходимо реализовать приложение для просмотра данных о загрязнении окружающей среды в табличном виде и на карте, визуализации статистики и добавления новых данных. Приложение должно использовать СУБД Neo4j.

Содержание пояснительной записки:

«Содержание», «Введение», «Качественные требования к решению», «Сценарии использования», «Модель данных», «Разработанное приложение», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 06.09.2021

Дата сдачи реферата: 25.12.2021

Дата защиты реферата: 26.12.2021

Студентка гр. 8382

\_\_\_\_\_

Кузина А.М.

Студентка гр. 8382

\_\_\_\_\_

Кулачкова М.К.

Студент гр. 8382

\_\_\_\_\_

Янкин Д.О.

Преподаватель

\_\_\_\_\_

Заславский М.М.

## **АННОТАЦИЯ**

В рамках данного задания было реализовано веб-приложение для просмотра данных о загрязнении окружающей среды. Хранение и обработка данных осуществляются при помощи графовой СУБД Neo4j. Интерфейс приложения реализован с использованием фреймворка React. Созданное приложение позволяет просматривать данные на карте и в виде таблицы, добавлять записи в базу данных, визуализирует статистику, предоставляет возможность массового импорта и экспорта данных. Исходный код приложения доступен по ссылке: <https://github.com/moevm/nosql2h21-env>.

## **SUMMARY**

A web-application that visualizes environmental pollution data has been developed within the task. Data storage and management are handled by the Neo4j graph DBMS. The user interface was created using the React framework. The application allows the user to view data on the map or in a table form, to add new entries to the database, visualizes statistics, makes it possible to import and export data en masse. The source code of the application can be accessed at the following link: <https://github.com/moevm/nosql2h21-env>.

## СОДЕРЖАНИЕ

	Введение	5
1.	Качественные требования к решению	6
2.	Сценарии использования	7
2.1.	Макет UI	7
2.2.	Сценарии использования для конкретных задач	10
3.	Модель данных	18
3.1.	Нереляционная модель	18
3.2.	Реляционная модель	22
3.3.	Сравнение моделей	24
4.	Разработанное приложение	26
	Заключение	28
	Список использованных источников	29
	Приложение А. Документация по сборке и развертыванию приложения	30
	Приложение Б. Инструкция для пользователя	31
	Приложение В. Снимки экрана приложения	34

## ВВЕДЕНИЕ

Проблема загрязнения окружающей среды в последние годы привлекает внимание большого количества людей, о чем свидетельствует появление таких общественных организаций, как Sunrise Movement (2017 г.), Fridays For Future (2018 г.), Extinction Rebellion (2018 г.) и Scientists For Future (2019 г.). Приложение, которое предоставит пользователю возможность в доступной форме получить информацию о загрязнениях окружающей среды, позволит повысить осведомленность о проблеме среди населения и тем самым побудить большее число людей принять меры по сокращению вредного воздействия на окружающую среду.

Целью данной работы является реализация приложения с веб-интерфейсом для хранения, обработки и визуализации данных о загрязнениях окружающей среды. Данные должны отображаться в виде таблицы и на карте, также должна приводиться статистика. Для хранения и обработки данных будет использоваться СУБД Neo4j<sup>[1]</sup> с драйвером для JavaScript<sup>[2]</sup>.

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

Требуется разработать веб-приложение для просмотра данных о загрязнении окружающей среды. Хранение и обработка данных должны осуществляться при помощи СУБД Neo4j.

В число возможностей, предоставляемых приложением, должны входить:

- Просмотр данных на карте;
- Просмотр данных в виде таблицы;
- Фильтрация данных по одному или нескольким параметрам;
- Добавление данных;
- Просмотр статистики;
- Массовый импорт данных;
- Массовый экспорт данных.

## 2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

### 2.1. Макет UI

На рисунке 1 приведен макет приложения в виде графа. На каждой из страниц есть область навигации с четырьмя кнопками: кнопка перехода на главную страницу приложения, кнопка перехода к просмотру данных на карте, кнопка перехода к просмотру статистики и кнопка перехода к просмотру данных в таблице.



Рисунок 1 – Макет приложения в виде графа

На рисунке 2 изображена главная страница приложения. На ней расположены кнопки для массового импорта и экспорта данных.

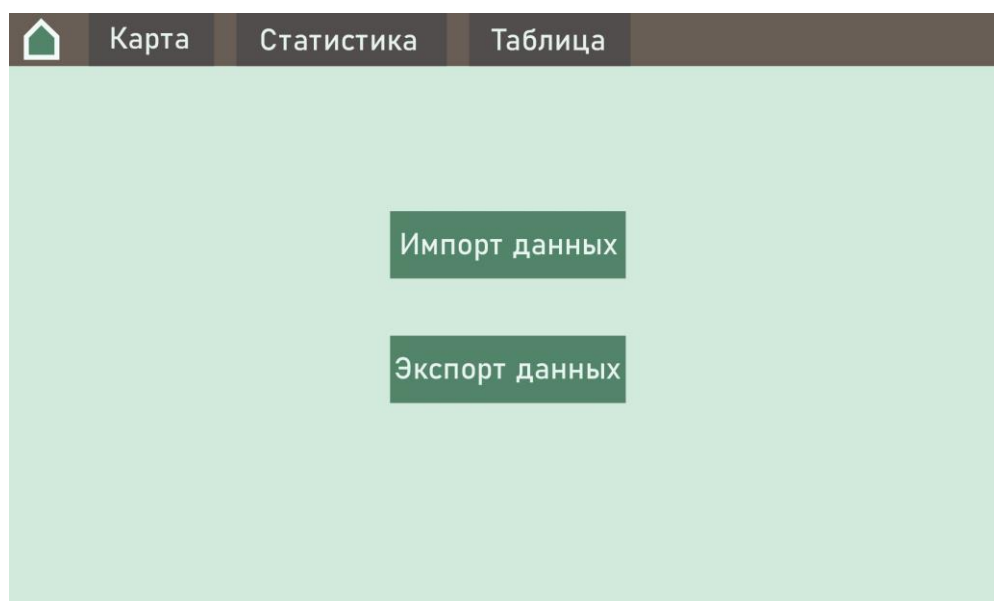


Рисунок 2 – Главная страница приложения

На рисунке 3 приведена страница для просмотра данных на карте. На этой странице можно выбрать, данные о каких веществах и за какой промежуток времени будут отображаться на карте.

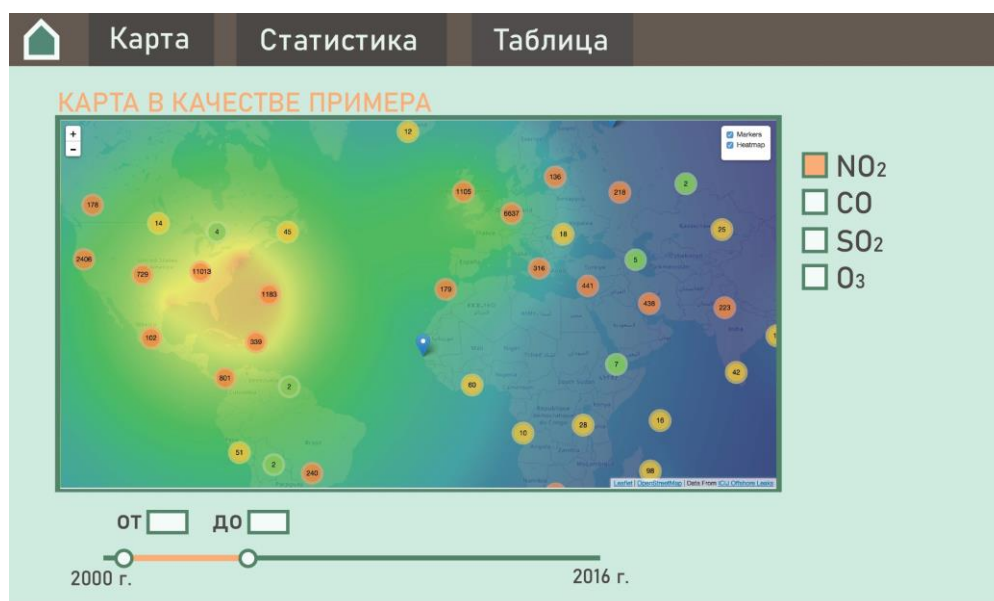


Рисунок 3 – Страница для просмотра данных на карте

На странице просмотра статистики можно просмотреть одну из двух диаграмм: график и гистограмму. График изображает зависимость средней концентрации одного из веществ от времени в заданный период в заданном штате или во всей стране. Страница с отображением графика приведена на рисунке 4.

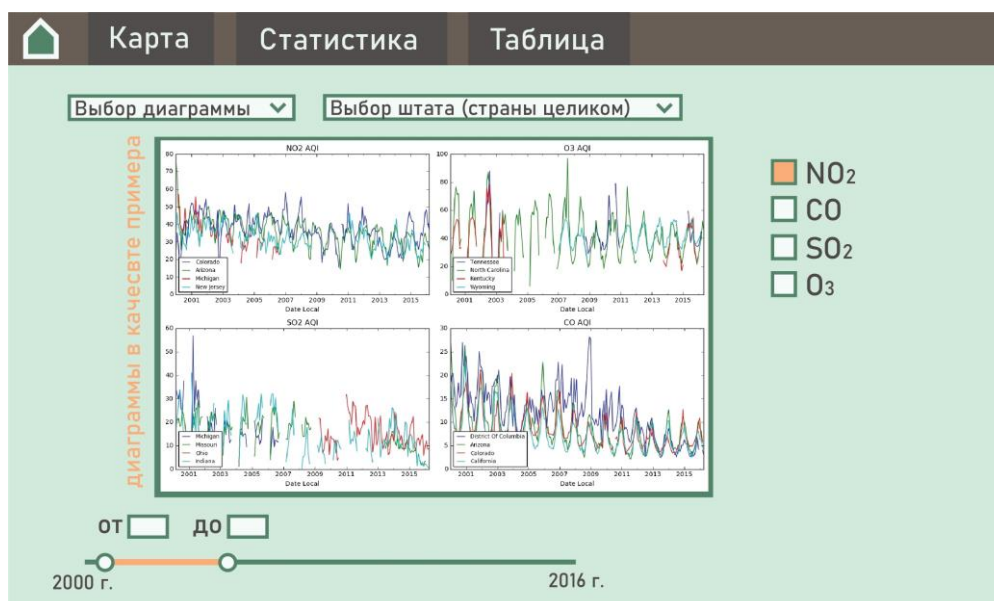


Рисунок 4 – Страница просмотра графика



Гистограмма изображает распределение средней концентрации заданного вещества по штатам в заданный период времени. Страница с гистограммой приведена на рисунке 5.

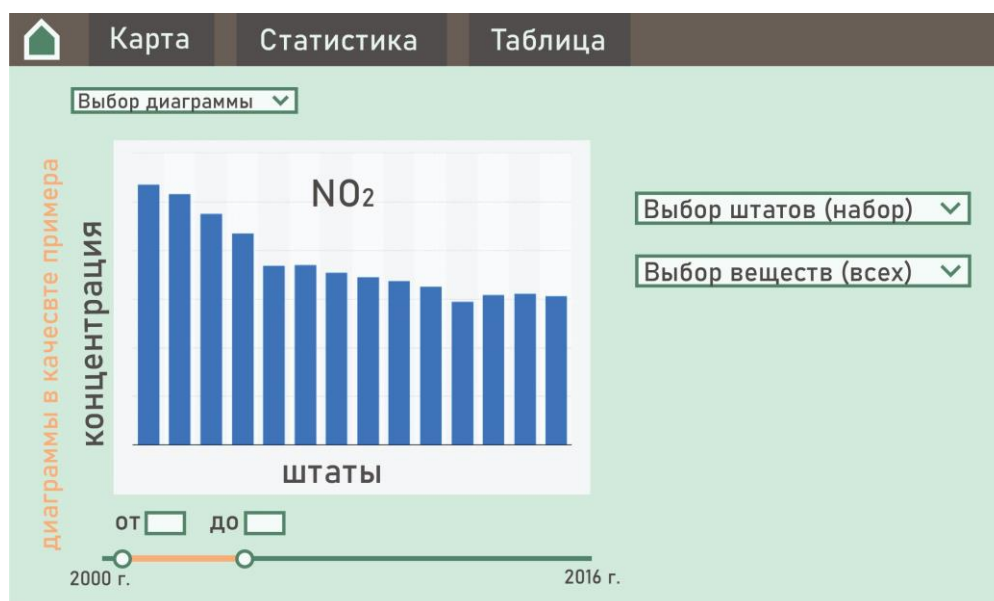


Рисунок 5 – Страница просмотра гистограммы

На рисунке 6 приведена страница просмотра данных в таблице. На странице можно выбрать, какие столбцы таблицы будут отображены, а также отфильтровать наблюдения по годам, штатам и веществам.

The screenshot shows a web interface with a navigation bar containing 'Карта', 'Статистика', and 'Таблица'. The 'Таблица' tab is active. Below the navigation bar is a 'Выбор колонок (всех)' dropdown. The main content area features a table with the following columns: #, # State Code, # County Co..., # Site Num, A Address, A State, and A County. The table contains 8 rows of data. To the right of the table are three dropdown menus: 'Выбор штатов (набор)', 'Выбор веществ (всех)', and 'Выбор годов (всех)'.

#	# State Code	# County Co...	# Site Num	A Address	A State	A County
2	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
3	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
4	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
5	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
6	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
7	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa
8	4	13	3882	1645 E ROOSEVELT ST-CENTRAL PHOENIX STN	Arizona	Maricopa

Рисунок 6 – Страница просмотра данных в таблице

В процессе разработки в макет могут быть внесены изменения и дополнения, поэтому интерфейс готового приложения может отличаться от приведенного в этом разделе.

## **2.2. Сценарии использования для конкретных задач**

### *1. Просмотр страницы с картой*

- Действующее лицо: пользователь
- Основной сценарий:
  - Пользователь нажимает на кнопку «Карта»
  - Открывается страница с картой
- Результат: открывается страница с картой; на карте изображена средняя концентрация всех вредных веществ в период от 2000 до 2016 года.
- Альтернативный сценарий:
  - Пользователь уже находится на странице с картой. Результат: ничего не происходит.
  - Ошибка соединения. Результат: система выдает сообщение об ошибке, осуществляется повторная попытка соединения.

### *2. Изменение параметров отображения карты*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице с картой
- Основной сценарий:
  - Пользователь выбирает временной интервал, на котором вычисляется средняя концентрация, с помощью ползунков
  - На карте отображается средняя концентрация выбранных веществ в период времени, заданный ползунками
  - Пользователь отмечает чекбоксы, соответствующие тем веществам, концентрацию которых он хочет отобразить на карте
  - На карте отображаются сведения о концентрации выбранных веществ
- Альтернативный сценарий:

- Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.
- Пользователь не отметил ни одного чекбокса. Результат: отображается пустая карта

### *3. Просмотр статистики*

- Действующее лицо: пользователь
- Основной сценарий:
  - Пользователь жмет на кнопку «Статистика»
  - Открывается страница со статистикой
- Результат: на странице со статистикой изображаются графики зависимости средней концентрации каждого из веществ по всей стране от времени.
- Альтернативный сценарий:
  - Пользователь уже находился на странице со статистикой. Результат: ничего не происходит.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

### *4. Просмотр графиков изменения концентрации веществ в зависимости от времени в разных штатах*

- Действующее лицо: пользователь
- Предусловие: открыта страница со статистикой, в выпадающем списке «Выбор диаграммы» выбрана диаграмма «Концентрация/время»
- Основной сценарий:
  - Пользователь в выпадающем списке «Выбор штата» выбирает штат, статистику по которому хочет посмотреть
  - Отображаются графики изменения концентрации веществ в выбранном штате/во всей стране в зависимости от времени

- Альтернативный сценарий:
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### 5. *Просмотр гистограмм средней концентрации веществ в зависимости от штата*

- Действующее лицо: пользователь
- Предусловие: открыта страница со статистикой
- Основной сценарий:
  - Пользователь в выпадающем списке «Выбор диаграммы» выбирает диаграмму «Концентрация/штат»
  - Отображаются гистограммы средней концентрации каждого вещества в зависимости от штата
- Альтернативный сценарий:
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### 6. *Изменение параметров отображения гистограмм*

- Действующее лицо: пользователь
- Предусловие: открыта страница со статистикой, в выпадающем списке «Выбор диаграммы» выбрана диаграмма «Концентрация/штат»
- Основной сценарий:
  - Пользователь в выпадающем списке «Выбор штата» выбирает штат для просмотра статистики
  - Отображается гистограмма средней концентрации текущего вещества в заданный период времени в выбранном штате
  - Пользователь в выпадающем списке «Выбор вещества» выбирает вещество, гистограмму концентрации которого желает посмотреть

- Отображается гистограмма средней концентрации выбранного вещества в заданный период времени для заданного штата
- Пользователь выбирает временной интервал, на котором вычисляется средняя концентрация, с помощью ползунков
- Отображается гистограмма средней концентрации просматриваемого вещества в заданном штате в выбранный промежуток времени
- Альтернативный сценарий:
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *7. Просмотр табличного представления данных*

- Действующее лицо: пользователь
- Основной сценарий:
  - Пользователь жмет на кнопку «Таблица»
  - Открывается страница с табличным представлением данных
- Результат: на странице отображается таблица, содержащая исходные данные целиком.
- Альтернативный сценарий:
  - Пользователь уже находился на странице с табличным представлением данных. Результат: ничего не происходит.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *8. Изменение отображаемых колонок в таблице*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице «Таблица»
- Основной сценарий:
  - Пользователь жмет на кнопку «Выбор колонок»

- Открывается выпадающее окно с набором чекбоксов, соответствующих столбцам исходной таблицы
- Пользователь выбирает столбцы, которые хочет отобразить
- В таблице отображаются только отмеченные столбцы
- Альтернативный сценарий:
  - Пользователь не отметил ни одного из чекбоксов. Результат: приложение автоматически отмечает и отображает столбец с ключом таблицы.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *9. Выбор штатов, данные о которых отображаются в таблице*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице «Таблица»
- Основной сценарий:
  - Пользователь жмет на кнопку «Выбор штатов»
  - Открывается выпадающее окно с набором чекбоксов, соответствующих штатам
  - Пользователь выбирает штаты, данные о которых хочет просмотреть
  - В таблице отображаются данные о содержании веществ в выбранных штатах
- Альтернативный сценарий:
  - Пользователь не отметил ни одного из чекбоксов. Результат: отображается пустая таблица.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *10. Выбор веществ, данные о которых отображаются в таблице*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице «Таблица»
- Основной сценарий:
  - Пользователь жмет на кнопку «Выбор веществ»
  - Открывается выпадающее окно с набором чекбоксов, соответствующих веществам
  - Пользователь выбирает вещества, данные о содержании которых хочет просмотреть
  - В таблице отображаются данные о содержании выбранных веществ
- Альтернативный сценарий:
  - Пользователь не отметил ни одного из чекбоксов. Результат: отображается пустая таблица.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *11. Выбор годов, данные за которые отображаются в таблице*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице «Таблица»
- Основной сценарий:
  - Пользователь жмет на кнопку «Выбор годов»
  - Открывается выпадающее окно с набором чекбоксов, соответствующих годам от 2000 до 2016
  - Пользователь выбирает годы, данные за которые хочет посмотреть
  - В таблице отображаются данные за выбранные годы
- Альтернативный сценарий:
  - Пользователь не отметил ни одного из чекбоксов. Результат: отображается пустая таблица.

- Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

## *12.Добавление одной записи в таблицу*

- Действующее лицо: пользователь
- Предусловие: пользователь находится на странице «Таблица»
- Основной сценарий:
  - Пользователь нажимает на кнопку «Добавить запись».
  - Открывается диалоговое окно для добавления записи.
  - Пользователь вводит информацию о наблюдении.
  - Пользователь нажимает кнопку «Добавить».
  - Система осуществляет проверку введенных данных.
  - Диалоговое окно закрывается.
- Результат: пользователь оказывается на странице с таблицей; новая запись добавляется в базу данных и отображается в таблице.
- Альтернативный сценарий:
  - Пользователь нажал кнопку «Закрыть окно» в окне добавления записи. Результат: окно добавления записи закроется, запись не будет добавлена.
  - Обязательные поля остались незаполненными. Результат: система выдаст сообщение об ошибке. Запись не будет добавлена.
  - В поле введены данные неверного типа. Результат: система выдаст сообщение об ошибке. Запись не будет добавлена.
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

## *13.Импорт данных*

- Действующее лицо: пользователь
- Основной сценарий:



- Пользователь нажимает на кнопку "Импорт данных"
- Открывается окно выбора файла с фильтром на расширение "csv"
- Пользователь выбирает желаемый файл и нажимает кнопку "Загрузить"
- Данные из загруженного файла добавляются в базу, пользователь видит радостное сообщение "Данные загружены успешно!"
- Результат: данные из файла загружены в базу, пользователь доволен
- Альтернативный сценарий:
  - Пользователь не загрузил никакой файл, закрыл окно выбора файла -> не происходит ничего
  - Формат загруженного файла или его содержимое некорректно -> открывается окно с надписью "Содержимое файла некорректно"
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

#### *14.Экспорт данных*

- Действующее лицо: пользователь
- Основной сценарий:
  - Пользователь нажимает на кнопку "Экспорт данных"
  - Начинается загрузка файла "us\_pollution\_data.csv".
- Результат: пользователь скачал датасет, используемый в приложении
- Альтернативный сценарий:
  - Ошибка соединения. Результат: система выдаст сообщение об ошибке, производится повторная попытка соединения.

### 3. МОДЕЛЬ ДАННЫХ

#### 3.1. Нереляционная модель

На рисунке 7 приведено графическое представление модели.

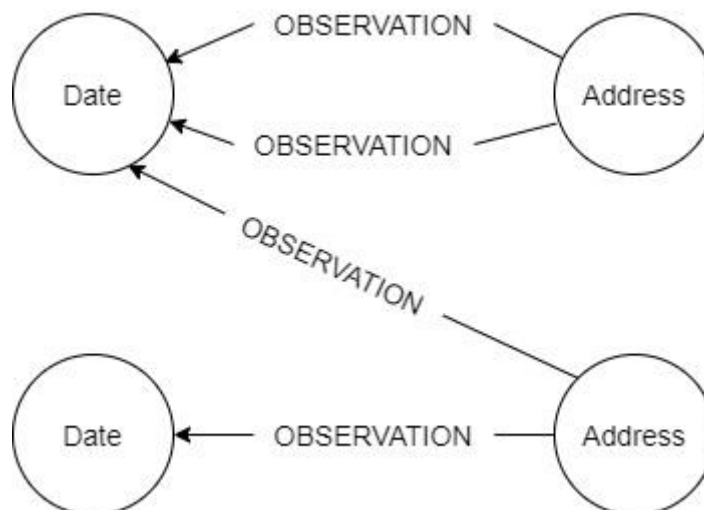


Рисунок 7 – Графическое представление нереляционной модели

Neo4j является графовой СУБД и лучше всего подходит для работы с данными, содержащими много связей. Было решено в узлах графа хранить даты и адреса наблюдений, а сами измерения веществ, загрязняющих воздух, хранить в виде отношений между датой и временем наблюдения.

Чистый объем дискового пространства, занимаемого данными, осуществлялся с учетом типа, которым представлялся каждый из атрибутов, и размера этого типа<sup>[3]</sup>. Атрибуты модели:

- Observation:
  - unit\_NO2: String(20) – 40B
  - mean\_NO2: Float – 4B
  - firstMV\_NO2: Float – 4B
  - firstMH\_NO2: Integer – 4B
  - aqi\_NO2: Float – 4B
  - unit\_O3: String(20) – 40B
  - mean\_O3: Float – 4B
  - firstMV\_O3: Float – 4B

- firstMH\_O3: Integer – 4B
- aqi\_O3: Float – 4B
- unit\_SO2: String(20) – 40B
- mean\_SO2: Float – 4B
- firstMV\_SO2: Float – 4B
- firstMH\_SO2: Integer – 4B
- aqi\_SO2: Float – 4B
- unit\_CO: String(20) – 40B
- mean\_CO: Float – 4B
- firstMV\_CO: Float – 4B
- firstMH\_CO: Integer – 4B
- aqi\_CO: Float – 4B

Чистый размер одной записи:  $V_o = 224B$

- Date:
  - date\_local: String(10) – 20B
  - year: Integer – 4B

Чистый размер одной записи:  $V_d = 24B$

Размер метки (названия узла):  $2B * 4 = 8B$

- Address:
  - state\_code: Integer – 4B
  - county\_code: Integer – 4B
  - site\_num: Integer – 4B
  - address: String(100) – 200B
  - state: String(25) – 50B
  - county: String(25) – 50B
  - city: String(50) – 100B

Чистый размер одной записи:  $V_a = 412B$

Размер метки (названия узла):  $2B * 7 = 14B$

Число записей:

- Число наблюдений:  $n = 1746661 = 291.3m = 8562.06l$

- Число уникальных дат:  $m = 5996 = 0.003n = 29.39l$
- Число уникальных адресов:  $l = 204 = 0.000117n = 0.034m$

Чистый объем модели (объем полезных данных) равен:

$$V_{clean} = nV_o + mV_d + lV_a = nV_o + 0.003nV_d + 0.000117nV_a = 224.12n \\ = 65\,286.156m = 1\,918\,928.89l$$

Расчет грязного объема осуществляется с учетом особенностей размещения данных в Neo4j<sup>[4]</sup>. В Neo4j один узел занимает 15В, отношение – 34В, атрибут – 41В, строковые атрибуты, которые не помещаются в запись атрибута – 128В. Имеем  $m + l$  узлов,  $n$  отношений,  $20n + 2m + 7l$  атрибутов и  $4n + 4l$  длинных строковых атрибутов, а также метки узлов. Тогда грязный объем равен:

$$V_{dirty} = (m + l) * 15B + n * 34B + (20n + 2m + 7l) * 41B + (4n + 4l) * 128B \\ + m * 8B + l * 14B = 1366.42n = 398\,036.87m = 11\,698\,303.8l$$

Избыточность данных равна

$$\frac{V_{dirty}}{V_{clean}} = 6.1.$$

Ниже приведены основные запросы к базе данных:

- Пример запроса на получение наблюдений за определенные года в определенных штатах:

```
MATCH (address:Address)-[observation]->(date:Date)
WHERE address.state IN ['California', 'Arizona'] AND date.year IN
[2000, 2001]
RETURN address, date, observation
```

- Пример запроса на получение координат адресов снятия измерений:

```
MATCH (address:Address)-[measurement]->(date:Date) WHERE date.year IN
[2000, 2001]
CALL apoc.spatial.geocodeOnce(address.state) YIELD location
RETURN location, date, measurement
```

- Пример запроса на массовый импорт из файла:

```
USING PERIODIC COMMIT 10000
LOAD CSV WITH HEADERS FROM 'file:///filename.csv' AS line
FIELDTERMINATOR ';'
MERGE (address:Address {state: line.state, address: line.address})
ON CREATE
```

```

        SET address.state_code = toInteger(line.state_code),
        address.county_code = toInteger(line.county_code),
        address.site_num = toInteger(line.site_num), address.county =
        line.county, address.city = line.city
    MERGE (date:Date {date_local: line.date_local, year:
    toInteger(left(line.date_local, 4))})
    CREATE (address)-[:MEASURED {
        unit_NO2: line.unit_NO2,
        mean_NO2: line.mean_NO2,
        firstMV_NO2: line.firstMV_NO2,
        firstMH_NO2: line.firstMH_NO2,
        aqi_NO2: line.aqi_NO2,
        unit_O3: line.unit_O3,
        mean_O3: line.mean_O3,      firstMV_O3: line.firstMV_O3,
        firstMH_O3: line.firstMH_O3,
        aqi_O3: line.aqi_O3,
        unit_SO2: line.unit_SO2,
        mean_SO2: line.mean_SO2,
        firstMV_SO2: line.firstMV_SO2,
        firstMH_SO2: line.firstMH_SO2,
        aqi_SO2: line.aqi_SO2,
        unit_CO: line.unit_CO,
        mean_CO: line.mean_CO,
        firstMV_CO: line.firstMV_CO,
        firstMH_CO: line.firstMH_CO,
        aqi_CO: line.aqi_CO
    }]->(date)

```

- Получение полной таблицы:

```

MATCH (address:Address)-[measurement]->(date:Date)
RETURN address.state_code AS state_code, address.county_code AS
county_code, address.site_num AS site_num, address.address AS address,
address.state AS state, address.county AS county, address.city AS
city, date.date_local AS date_local, measurement.unit_NO2 as unit_NO2,
measurement.mean_NO2 as mean_NO2, measurement.firstMV_NO2 as
firstMV_NO2, measurement.firstMH_NO2 as firstMH_NO2,
measurement.aqi_NO2 as aqi_NO2, measurement.unit_O3 as unit_O3,
measurement.mean_O3 as mean_O3, measurement.firstMV_O3 as firstMV_O3,
measurement.firstMH_O3 as firstMH_O3, measurement.aqi_O3 as aqi_O3,
measurement.unit_SO2 as unit_SO2, measurement.mean_SO2 as mean_SO2,
measurement.firstMV_SO2 as firstMV_SO2, measurement.firstMH_SO2 as
firstMH_SO2, measurement.aqi_SO2 as aqi_SO2, measurement.unit_CO as
unit_CO, measurement.mean_CO as mean_CO, measurement.firstMV_CO as
firstMV_CO, measurement.firstMH_CO as firstMH_CO, measurement.aqi_CO
as aqi_CO

```

- Экспорт в csv-файл:

```

WITH "{Выражение из прошлого пункта}" AS query
CALL apoc.export.csv.query(query, "data.csv", {})
YIELD file, source, format, nodes, relationships, properties, time,
rows, batchSize, batches, done, data
RETURN file, source, format, nodes, relationships, properties, time,
rows, batchSize, batches, done, data;

```

### 3.2. Реляционная модель

На рисунке 8 приведено графическое представление реляционной модели данных.

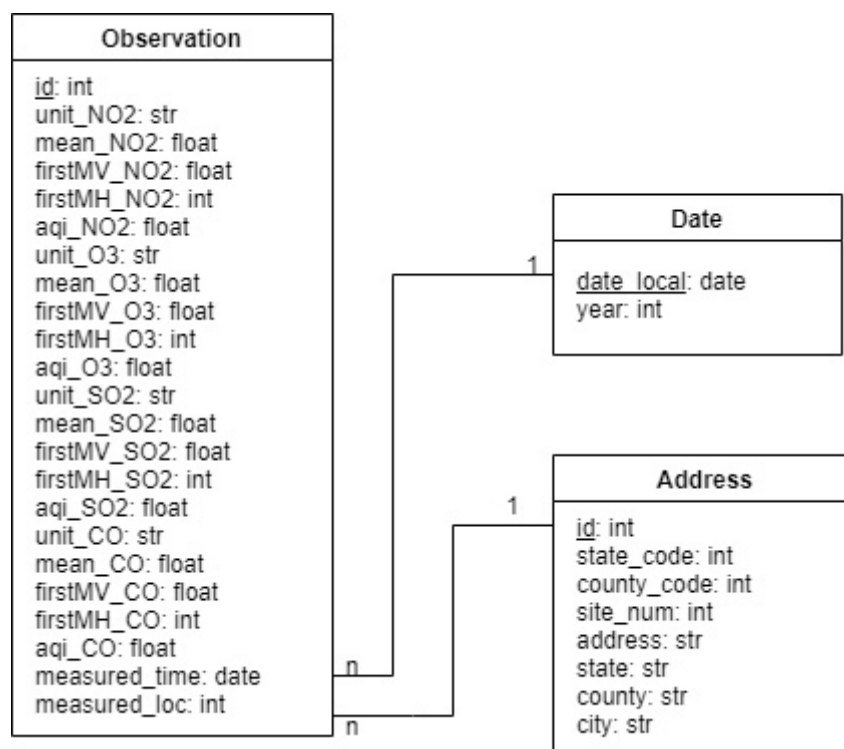


Рисунок 8 – Графическое представление реляционной модели

Рассчитывается примерный объем, занимаемый каждой из записей:

- Observation:
  - id: INT – 4B
  - unit\_NO2: VARCHAR(20) – 20B
  - mean\_NO2: REAL – 4B
  - firstMV\_NO2: REAL – 4B
  - firstMH\_NO2: INT – 4B
  - aqi\_NO2: REAL – 4B
  - unit\_O3: VARCHAR(20) – 20B
  - mean\_O3: REAL – 4B
  - firstMV\_O3: REAL – 4B
  - firstMH\_O3: INT – 4B
  - aqi\_O3: REAL – 4B
  - unit\_SO2: VARCHAR(20) – 20B

- mean\_SO2: REAL – 4B
- firstMV\_SO2: REAL – 4B
- firstMH\_SO2: INT – 4B
- aqi\_SO2: REAL – 4B
- unit\_CO: VARCHAR(20) – 20B
- mean\_CO: REAL – 4B
- firstMV\_CO: REAL – 4B
- firstMH\_CO: INT – 4B
- aqi\_CO: REAL – 4B
- *measured\_time*: DATE – 3B (поле связи с таблицей дат)
- *measured\_loc*: INT – 4B (поле связи с таблицей адресов)

Курсивом обозначены поля, предназначенные для связи записей из разных таблиц – размер этих полей не учитывается при вычислении чистого объема записи.

Чистый размер записи:  $V_o = 148B$

- Date:
  - date\_local: DATE – 3B
  - year: INT – 4B

Чистый размер записи:  $V_d = 7B$

- Address:
  - id: INT – 4B
  - state\_code: INT – 4B
  - county\_code: INT – 4B
  - site\_num: INT – 4B
  - address: VARCHAR(100) – 100B
  - state: VARCHAR(25) – 25B
  - county: VARCHAR(25) – 25B
  - city: VARCHAR(50) – 50B

Чистый размер записи:  $V_a = 216B$

Чистый объем данных равен

$$V_{clean} = V_o * n + V_d * m + V_a * l \approx V_o * n + V_d * 0.003n + V_a * 0.000117n \\ \approx 155.05n = 45\,166.065m = 1\,327\,547.4l$$

При вычислении грязного объема данных учитываются поля связей.  
Грязный объем данных равен

$$V_{dirty} = (V_o + V_{keys}) * n + V_d * m + V_a * l \\ \approx V_o * n + V_d + 0.003n + V_a * 0.000117n \approx 155.05n \\ = 45\,166.065m = 1\,327\,547.4l$$

Избыточность данных равна

$$\frac{V_{dirty}}{V_{clean}} = 1.05.$$

Ниже приведены основные запросы к базе данных:

- Пример запроса на получение наблюдений за определенные года в определенных штатах на SQL:

```
SELECT * FROM Dt INNER JOIN
(
    SELECT * FROM
    Address INNER JOIN Observation ON Address.id =
    Observation.measured_loc
    WHERE Address.state IN ('Arizona')
) AS temp
ON Dt.date_local = temp.measured_time
WHERE Dt.year IN (2000)
```

- SQL не имеет встроенных функций для геокодирования, поэтому в данном случае пришлось бы самостоятельно реализовывать запросы к интернет-сервисам вроде Google Geocoding или OpenStreetMaps

### 3.3. Сравнение моделей

- Объем данных в нереляционной модели значительно превышает объем данных в реляционной модели. Нереляционная модель также обладает большой избыточностью в силу принципа хранения данных.
- Наблюдения, удовлетворяющие заданным условиям, в Neo4j можно получить с помощью одного запроса, в реляционной СУБД – с помощью двух запросов.



- Neo4j имеет встроенную поддержку геокодирования, что облегчает работу с отображением адресов на карте. В реляционной СУБД эта задача сложнее.

## 4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

В качестве исходных данных приложение использует фрагмент набора данных о загрязнении воздуха в США, собранных организацией U.S. EPA: <https://www.kaggle.com/sogun3/uspollution>. При первом обращении к приложению осуществляется загрузка этих данных в базу данных.

Серверная часть приложения создает запросы к базе данных и обрабатывает полученные результаты.

Клиентская часть приложения состоит из четырех страниц: начальной страницы, откуда осуществляется экспорт и импорт данных, страницы с картой, страницы со статистикой и страницы с табличным отображением данных.

На рисунке 9 приведена схема экранов приложения.

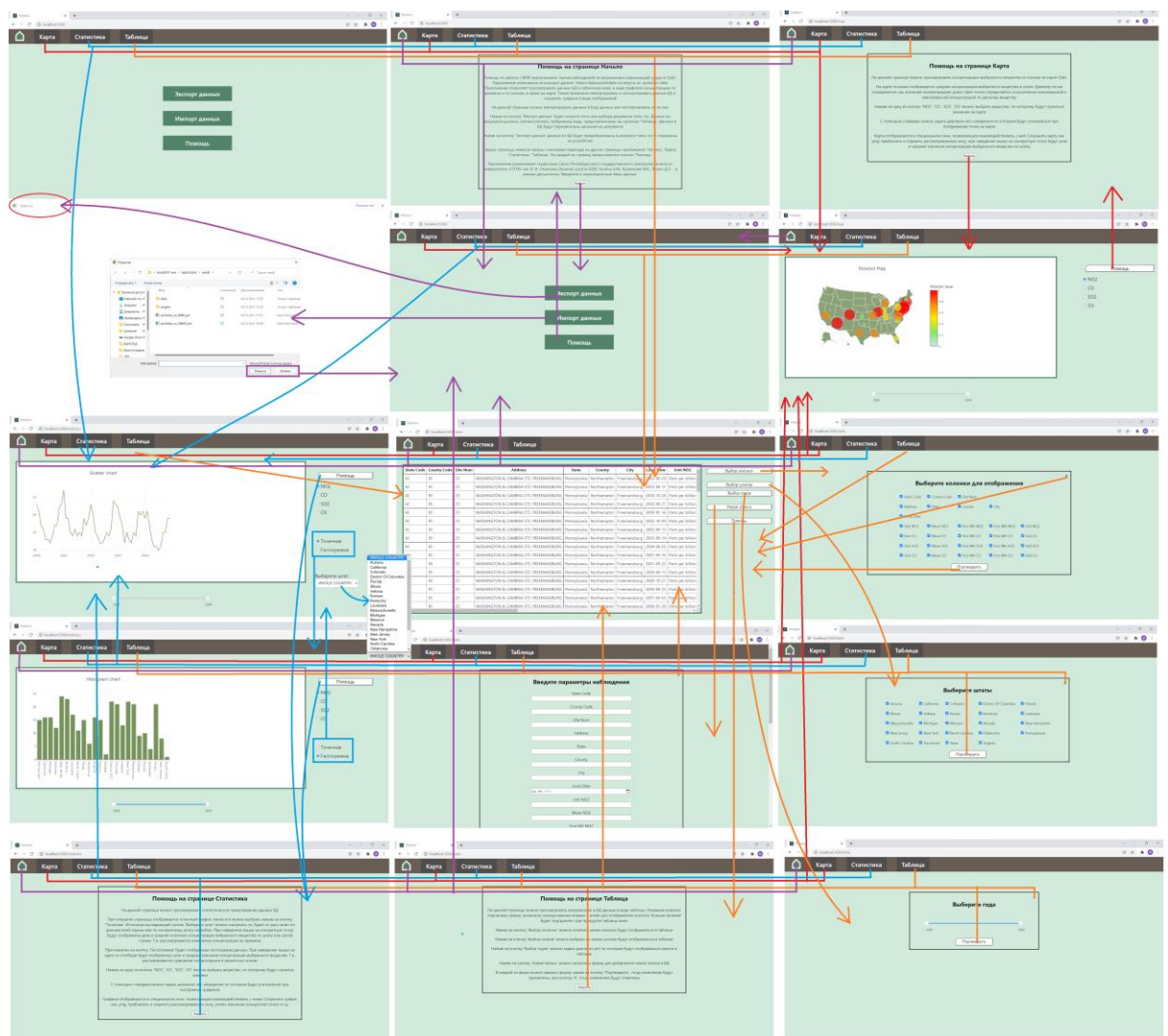


Рисунок 9 – Схема экранов приложения

#### Использованные технологии:

- Серверная часть: node.js с фреймворком express;
- Клиентская часть: React.js<sup>[5]</sup>, отображение графиков и карты при помощи библиотеки Plotly<sup>[6]</sup>;
- СУБД: Neo4j.

Ссылка на репозиторий с исходным кодом приложения:

<https://github.com/moevm/nosql2h21-env>

## ЗАКЛЮЧЕНИЕ

Было реализовано веб-приложение, предоставляющее пользователю возможность в разной форме просматривать данные о загрязнении окружающей среды. Осуществляется отображение данных на карте и в виде таблицы, а также визуализация статистики. Есть возможность добавлять записи по одной, импортировать данные из csv-файла или экспортировать данные из базы данных. Для хранения и обработки данных используется СУБД Neo4j.

Одним из недостатков приложения является объем, занимаемый базой данных, при этом фактический объем данных значительно превышает чистый объем в силу принципов хранения данных, реализованных в СУБД. Так как данные о загрязнении не являются сильно связными, возможно, было бы более эффективным использовать другую СУБД.

Другой проблемой является скорость обработки запроса на геокодирование адресов. При исполнении этого запроса СУБД обращается к внешнему сервису, на что уходит много времени. Сейчас этот запрос осуществляется при каждом новом обращении к странице с картой. Можно было бы улучшить производительность приложения, осуществляя этот запрос один раз при изначальной загрузке данных в приложение.

Также некоторые функции приложения не работают с большими наборами данных.

В дальнейшем, помимо исправления имеющихся недостатков приложения, планируется добавить возможность удалять записи по одной, импортировать данные по внешней ссылке, улучшить дизайн элементов управления.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация языка запросов Neo4j Cypher // The Neo4j Cypher Manual v4.2. URL: <https://neo4j.com/docs/cypher-manual/4.2/> (дата обращения: 19.12.2021)
2. Документация драйвера Neo4j для языка JavaScript // Neo4j Driver for JavaScript. URL: <https://neo4j.com/docs/api/javascript-driver/4.3/> (дата обращения: 28.11.2021)
3. Описание атрибутов Neo4j // The Neo4j Manual v2.1.5 3.3. Properties. URL: <https://neo4j.com/docs/2.1.5/graphdb-neo4j-properties.html> (дата обращения: 26.10.2021)
4. Описание принципов размещения данных Neo4j на диске // Understanding Neo4j's data on disk. URL: <https://neo4j.com/developer/kb/understanding-data-on-disk/> (дата обращения: 26.10.2021)
5. Документация библиотеки React // Getting Started – React. URL: <https://reactjs.org/docs/getting-started.html> (дата обращения: 13.12.2021)
6. Документация библиотеки Plotly // Plotly JavaScript Open Source Graphing Library. URL: <https://plotly.com/javascript/> (дата обращения: 24.12.2021)

## ПРИЛОЖЕНИЕ А

### ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ

Для сборки и запуска приложения необходимо:

1. Загрузить исходный код из репозитория по ссылке:  
<https://github.com/moevm/nosql2h21-env>
2. Перейти в папку nosql2h21-env/application.
3. Для сборки приложения открыть терминал и ввести команду docker-compose build.
4. Для запуска ввести в терминале команду docker-compose up.
5. Открыть браузер на странице <http://localhost:3000>.

## **ПРИЛОЖЕНИЕ Б**

### **ИНСТРУКЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ**

#### **Экспорт данных:**

1. Перейдите на начальную страницу приложения, нажав на иконку с домиком в левом углу панели навигации.
2. Нажмите на кнопку «Экспорт данных».
3. После нажатия на кнопку должна автоматически начаться загрузка csv-файла.

#### **Импорт данных:**

1. Перейдите на начальную страницу приложения, нажав на иконку с домиком в левом углу панели навигации.
2. Нажмите на кнопку «Импорт данных».
3. В открывшемся окне выберите файл формата csv, который хотите загрузить.
4. Нажмите кнопку «Открыть».
5. Для отмены импорта данных нажмите кнопку «Отмена».

#### **Просмотр и изменение параметров отображения данных на карте:**

1. Перейдите на страницу с картой, нажав на соответствующую ссылку на панели навигации.
2. Дождитесь загрузки карты.
3. Для изменения временного интервала отображаемых данных воспользуйтесь ползунком внизу страницы.
4. Можно изменить вещество, концентрация которого отображается на карте, с помощью радиокнопок в правой части страницы.
5. Взаимодействие с картой осуществляется при помощи кнопок в верхней части области отображения карты.

#### **Просмотр и изменение параметров отображения точечной диаграммы:**

1. Перейдите на страницу «Статистика», нажав на соответствующую ссылку на панели навигации.

2. При помощи радиокнопок в правой части страницы выберите режим «Точечная».
3. Для изменения временного интервала отображаемых данных воспользуйтесь ползунком внизу страницы.
4. Для выбора штата, информация о котором изображается, воспользуйтесь выпадающим списком в правой части страницы. Чтобы отобразить информацию по всей стране, выберите в списке вариант «WHOLE COUNTRY».
5. Можно изменить вещество, концентрация которого отображается на графике, с помощью радиокнопок в правой части страницы.
6. Взаимодействие с графиком осуществляется при помощи кнопок в верхней части области отображения графика.

#### **Просмотр и изменение параметров отображения гистограммы:**

1. Перейдите на страницу «Статистика», нажав на соответствующую ссылку на панели навигации.
2. При помощи радиокнопок в правой части страницы выберите режим «Гистограмма».
3. Для изменения временного интервала отображаемых данных воспользуйтесь ползунком внизу страницы.
4. Можно изменить вещество, концентрация которого отображается на графике, с помощью радиокнопок в правой части страницы.
5. Взаимодействие с графиком осуществляется при помощи кнопок в верхней части области отображения графика.

#### **Просмотр данных в таблице:**

1. Перейдите на страницу «Таблица», нажав на соответствующую ссылку на панели навигации.

#### **Изменение отображаемых колонок:**

1. На странице «Таблица» нажмите на кнопку «Выбор колонок».
2. В открывшемся окне отметьте галочками поля, которые хотите отобразить.
3. Нажмите «Подтвердить».



4. Для отмены выбора нажмите «X» в правом верхнем углу окна.

#### **Изменение отображаемых штатов:**

1. На странице «Таблица» нажмите на кнопку «Выбор штатов».
2. В открывшемся окне отметьте галочками штаты, информацию о которых хотите отобразить.
3. Нажмите «Подтвердить».
4. Для отмены выбора нажмите «X» в правом верхнем углу окна.

#### **Изменение отображаемых годов:**

1. На странице «Таблица» нажмите на кнопку «Выбор годов».
2. В открывшемся окне выберите отображаемый интервал при помощи ползунков.
3. Нажмите «Подтвердить».
4. Для отмены выбора нажмите «X» в правом верхнем углу окна.

#### **Добавление данных в таблицу:**

1. На странице «Таблица» нажмите на кнопку «Новая запись».
2. В открывшемся окне введите параметры наблюдения.
3. Нажмите «Подтвердить» для добавления записи.
4. Для отмены добавления нажмите «X» в правом верхнем углу окна.

На каждой странице приложения есть кнопка «Помощь», при нажатии на которую можно получить справочную информацию.

## ПРИЛОЖЕНИЕ В СНИМКИ ЭКРАНА ПРИЛОЖЕНИЯ

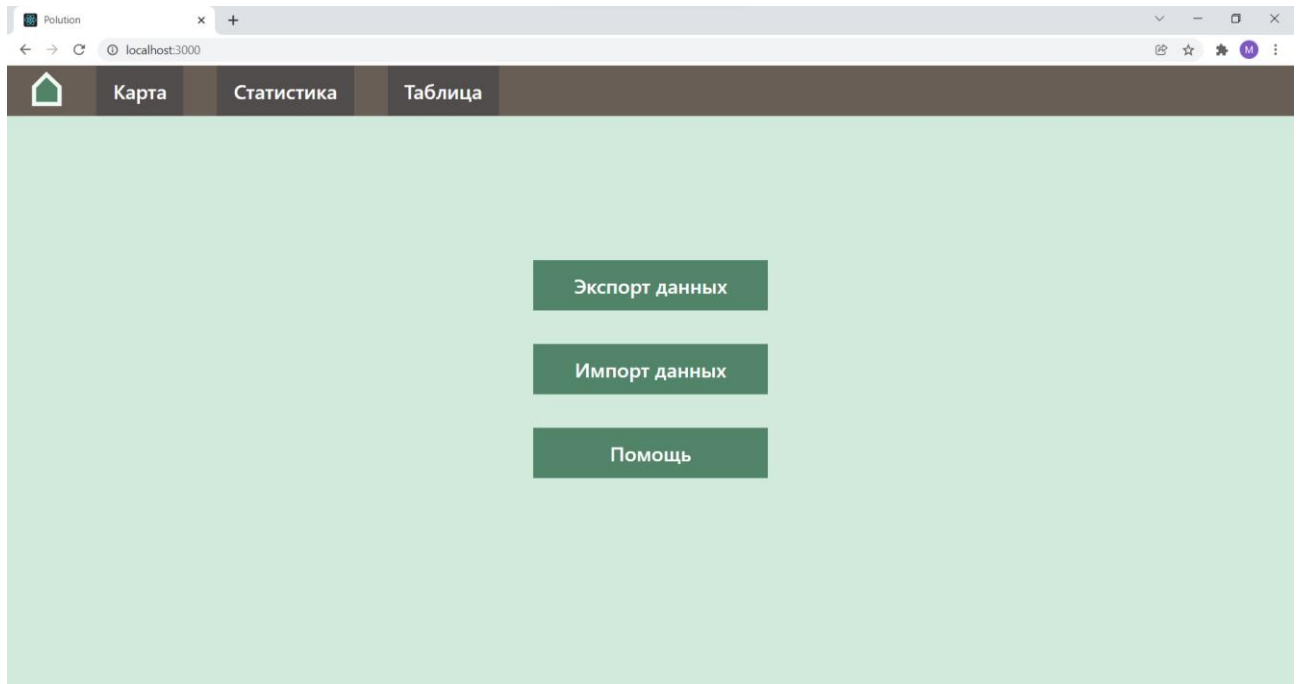


Рисунок 10 – Начальная страница

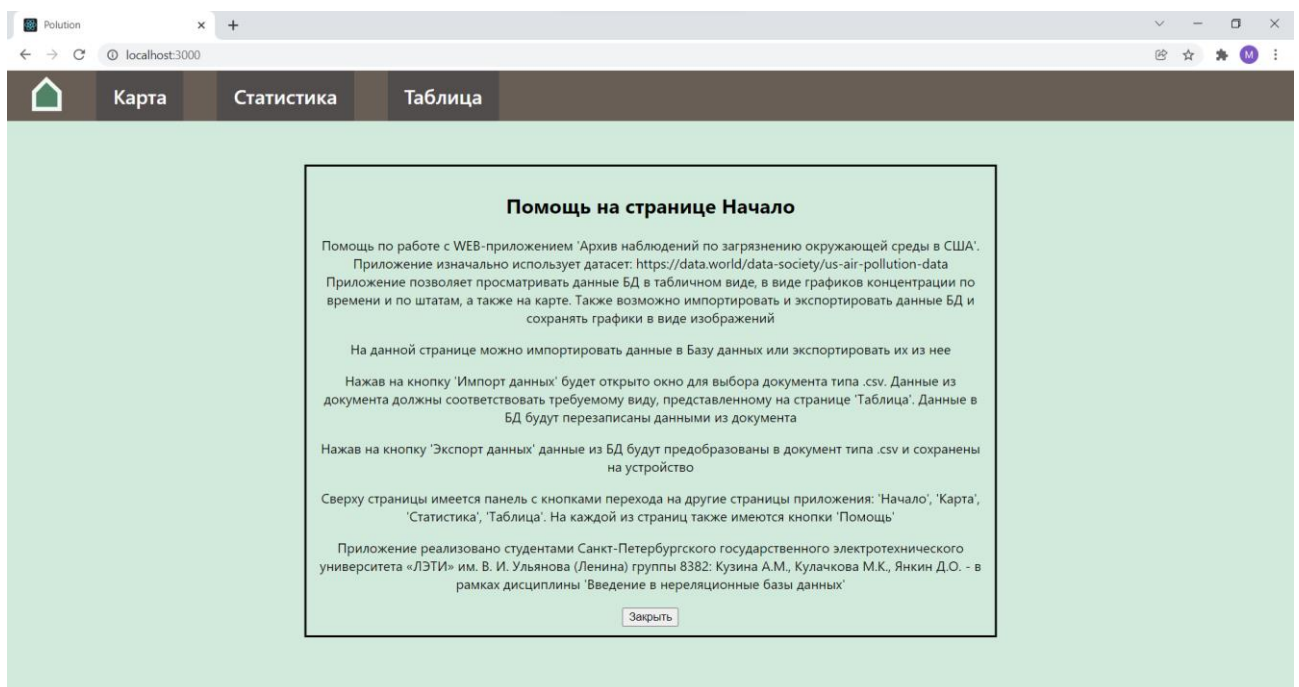


Рисунок 11 – Помощь на начальной странице

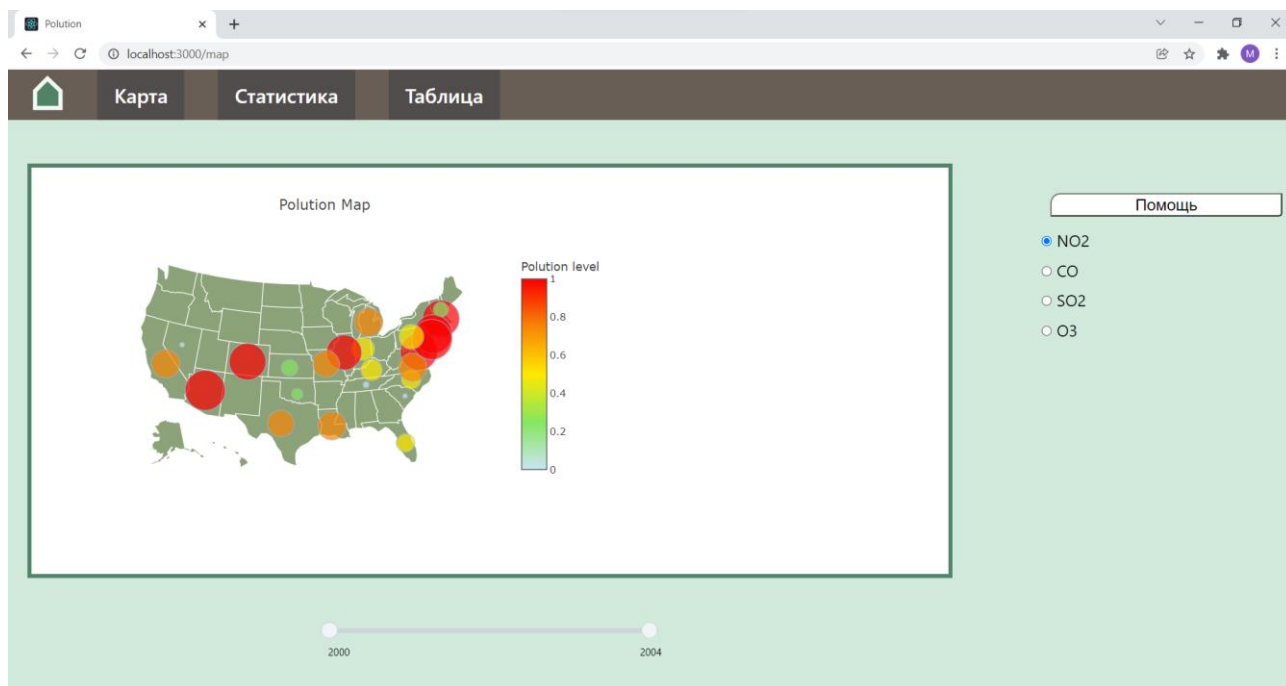


Рисунок 12 – Страница с картой

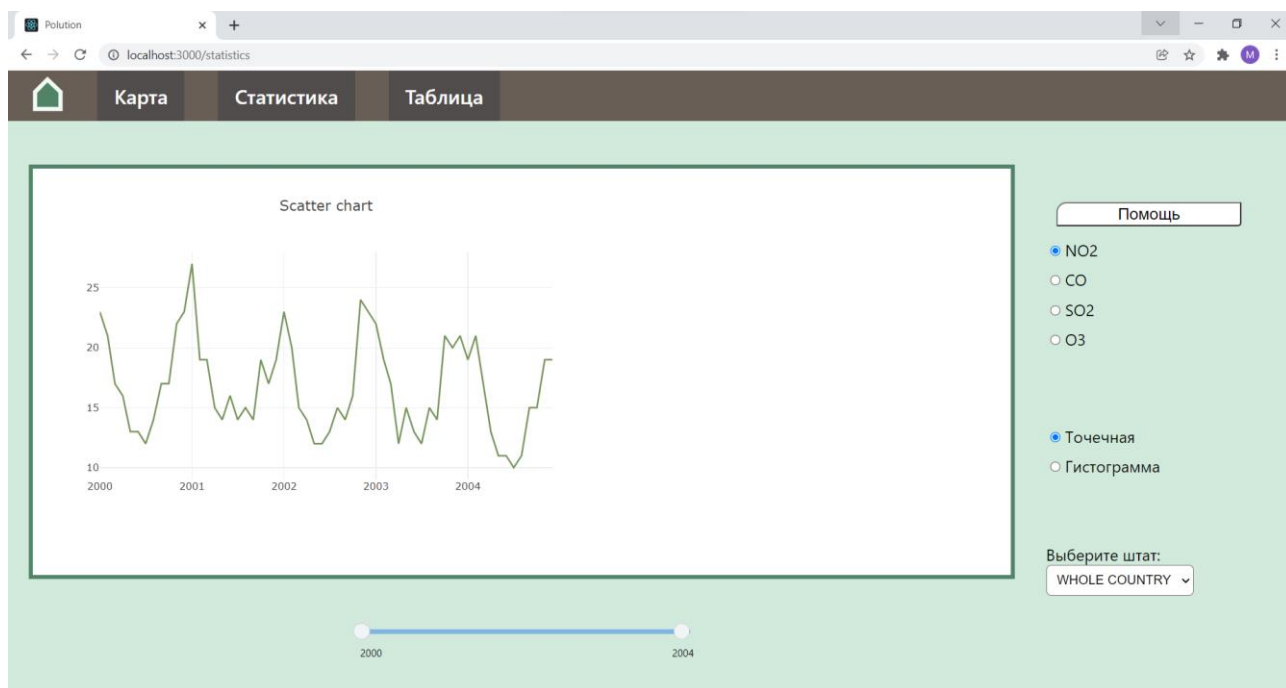


Рисунок 13 – Страница с точечной диаграммой

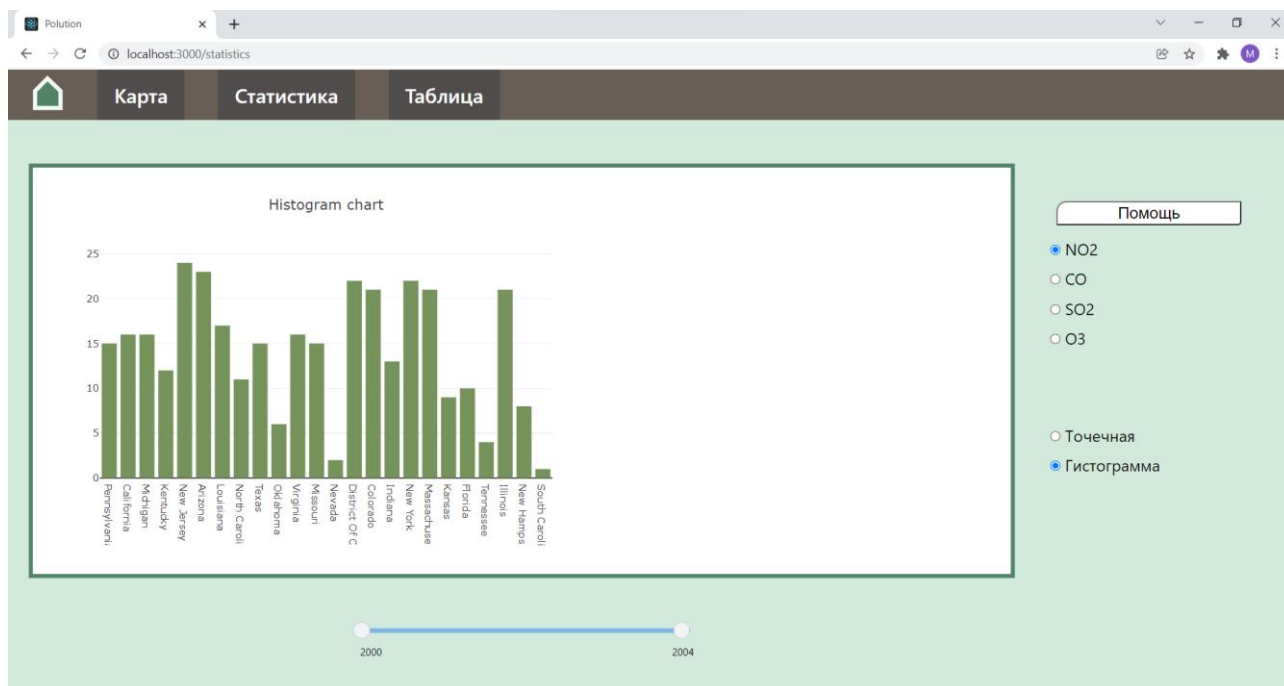


Рисунок 14 – Страница с гистограммой

Polution

localhost:3000/table

Карта Статистика Таблица

State Code	County Code	Site Num	Address	State	County	City	Local Date	Unit NO2
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2002-06-29	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2003-04-11	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2004-10-26	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2002-08-27	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2000-05-16	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2002-10-09	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2002-09-13	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2003-08-14	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2004-08-03	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2001-09-16	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2001-09-22	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2000-06-13	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2000-10-27	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2000-08-25	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2001-04-03	Parts per billion
42	95	25	WASHINGTON & CAMBRIA STS. FREEMANSBURG	Pennsylvania	Northampton	Freemansburg	2000-05-28	Parts per billion

Выбор колонок

Выбор штатов

Выбор годов

Новая запись

Помощь

Рисунок 15 – Страница с таблицей

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/table'. The application has a dark header with three tabs: 'Карта' (Map), 'Статистика' (Statistics), and 'Таблица' (Table). The 'Таблица' tab is active. A modal window titled 'Введите параметры наблюдения' (Enter observation parameters) is open. It contains the following input fields: 'State Code', 'County Code', 'Site Num', 'Address', 'State', 'County', 'City', 'Local Date' (with a date picker icon), 'Unit NO2', 'Mean NO2', and 'First MV NO2'.

Рисунок 16 – Окно добавления записи

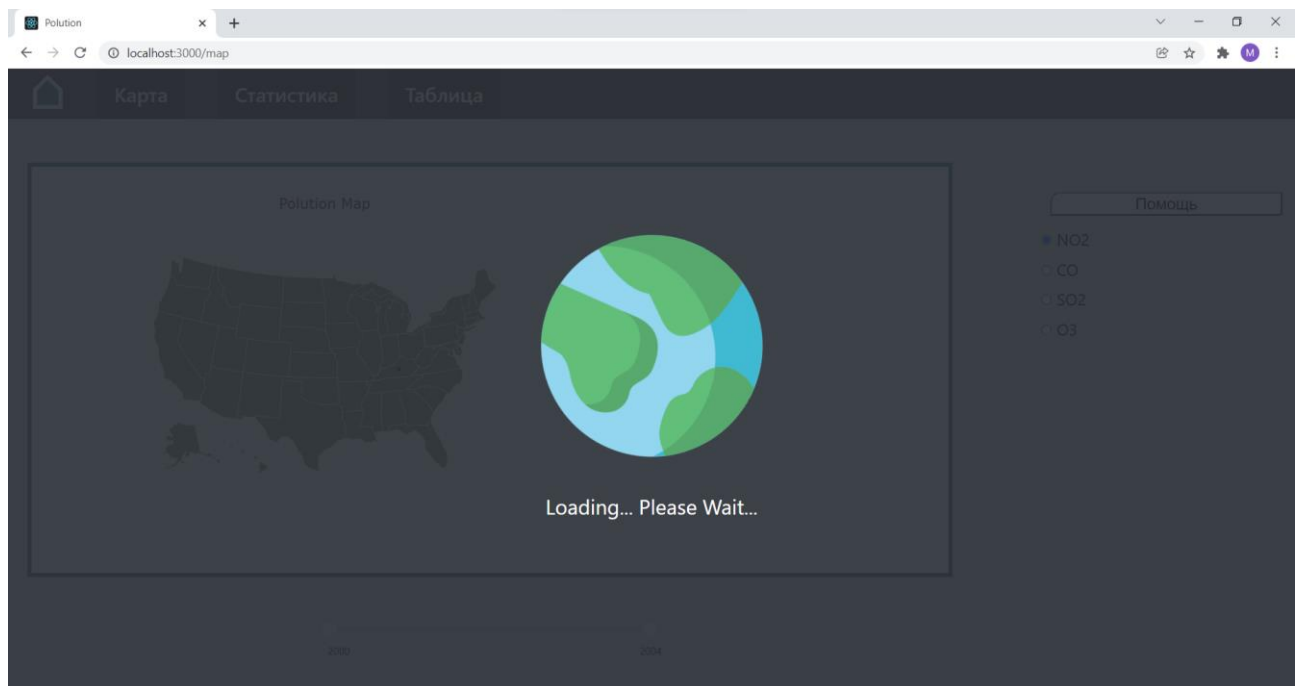


Рисунок 17 – Анимация загрузки