

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: Каталог конкурсных объявлений

| | | |
|------------------|-------|------------------|
| Студент гр. 9304 | _____ | Борисовский В.Ю. |
| Студент гр. 9304 | _____ | Краев Д.В. |
| Студент гр. 9304 | _____ | Афанасьев А. |
| Преподаватель | _____ | Заславский М.М. |

Санкт-Петербург
2022

ЗАДАНИЕ НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студент Борисовский В.Ю.

Студент Краев Д.В.

Студент Афанасьев А.

Группа 9304

Тема работы: Каталог конкурсных объявлений

Содержание пояснительной записки:

- Аннотация
- Содержание
- Введение
- Качественные требования к решению
- Сценарии использования
- Модель данных
- Разработанное приложение
- Заключение
- Список использованных источников

Предполагаемый объем пояснительной записки:

Не менее 20 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 15.12.2022

Дата защиты реферата: 22.12.2022

| | | |
|---------------|--|------------------|
| Студент | | Борисовский В.Ю. |
| Студент | | Краев Д.В. |
| Студент | | Афанасьев А. |
| Преподаватель | | Заславский М.М. |

АННОТАЦИЯ

В данной работе разработано веб-приложение, представляющее каталог конкурсных объявлений, позволяющая пользователям просматривать список доступных конкурсов по научным тематикам, фильтровать их по множеству параметров, просматривать статистику конкурсов. В системе используется следующий стек технологий: язык JavaScript, библиотека React для клиентской части; mongoose и TypeScript в качестве серверной части; Docker Compose для сборки проекта.

SUMMARY

In this work, a web application has been developed that represents a catalog of competitive ads, allowing users to view a list of available contests on scientific topics, filter them by a variety of parameters, and view the statistics of contests. The system uses the following technology stack: JavaScript language, React library for the client side; mongoose and TypeScript as the server side; Docker Compose for building the project.

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение | 5 |
| 1. Качественные требования к решению | 6 |
| 1.1. Текущие требования | 6 |
| 2. Сценарии использования | 7 |
| 2.1. Макет пользовательского интерфейса | 7 |
| 2.2. Сценарии использования | 7 |
| 2.3. Преобладающие операции | 8 |
| 3. Модель данных | 9 |
| 3.1. Нереляционная модель данных | 9 |
| 3.2. Реляционный аналог модели данных | 12 |
| 3.3. Сравнение моделей | 16 |
| 4. Разработанное приложение | 18 |
| 4.1. Описание приложения | 18 |
| 4.2. Используемые технологии | 18 |
| Заключение | 19 |
| Список использованных источников | 20 |
| Приложение А. Макет пользовательского интерфейса | 21 |
| Приложение Б. Документация по сборке и развёртыванию приложения | 22 |

ВВЕДЕНИЕ

При анализе конкурсных объявлений пользователям довольно проблематично выделить из них ключевые факторы, такие как: даты подачи заявки на участие в конкурсе, какие нужны подписи, призовые и т.д. В связи с этим нужно разработать инструмент, который бы анализировал конкурсные объявления с применением средств NLP и создавал каталог конкурсных объявлений в удобном для пользователя виде.

Цель работы – разработать ИС, представляющую из себя каталог конкурсных объявлений.

Основные задачи:

1. Сформулировать основные сценарии использования приложения.
2. Разработать макет пользовательского интерфейса.
3. Разработать схему базы данных.
4. Подготовить прототип приложения.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

1.1. Текущие требования

Текущие требования к решению выглядят следующим образом:

1. Имеется главная страница, на которой можно вставить ссылку на конкурс и получить информацию о нем, а так же произвести редактирование или добавление своего конкурса в базу данных.
2. Есть возможность как импорта данных в систему, так и экспорта из системы в .json формате.
3. Модули предоставляют следующие возможности:
 - а. База данных – модуль, позволяющий просматривать набор данных. Представляет из себя таблицу, по которой можно производить поиск и фильтрацию, а также просматривать статистику конкурсов.
 - б. Главная страница – модуль, позволяющий производить поиск, редактирование и добавление конкурсов.
4. Данные лежат по адресу <http://knvsh.gov.spb.ru/contests/>.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Макет пользовательского интерфейса

Разработан макет пользовательского интерфейса, представленный в приложении А.

2.2. Сценарии использования

Сценарий использования "Загрузка нового конкурса":

- Действующее лицо: Пользователь
- Основной сценарий:
 1. Пользователь открывает web-приложение
 2. Пользователь вводит ссылку на конкурс в поле "Ссылка"
 3. Пользователь нажимает кнопку загрузить
 4. Web-приложение отображает на странице структурированную информацию о конкурсе
- Альтернативный сценарий:
 1. На шаге 2 пользователь ввел некорректную ссылку и после клика на кнопку загрузить появляется сообщение об ошибке

Сценарий использования "Поиск уже загруженного конкурса":

- Действующее лицо: Пользователь
- Основной сценарий:
 1. Пользователь открывает приложение
 2. Пользователь нажимает на кнопку "База данных"
 3. Происходит переход на страницу со списком уже загруженных конкурсов
 4. Пользователь нажимает на иконку под названием "Поиск"
 5. Появляется поле для поиска
 6. Пользователь вводит текст в появившееся поле
 7. Приложение отображает варианты, соответствующие введенному тексту

- Альтернативный сценарий:

1. Пользователь перед/после задает параметры для фильтрации и конкурсы отображаются согласно фильтру

Сценарий использования "Экспорт данных":

- Действующее лицо: Пользователь.

- Основной сценарий:

1. Пользователь открывает приложение
2. Пользователь нажимает на кнопку "База данных"
3. Происходит переход на страницу со списком уже загруженных конкурсов
4. Пользователь нажимает на иконку под названием "Экспорт"
5. Происходит загрузка отображаемых данных в csv формате

- Альтернативный сценарий:

1. Перед нажатием на иконку "Экспорт" пользователь вводит текст в поисковую строку и экспортирует данные, соответствующие введенному тексту

2.3. Преобладающие операции

В модуле «База данных» фигурируют только операции на чтение, т.к. в данном модуле данные можно просмотреть и отфильтровать.

В модуле «Главная страница» фигурируют операции как на чтение, так и на запись, т.к на этой странице можно добавить/редактировать конкурсное объявление, а также на этой странице выводится информация о добавленном конкурсе.

3. МОДЕЛЬ ДАННЫХ

3.1. Нереляционная модель данных

Разработана схема нереляционной базы данных (рисунок 1).

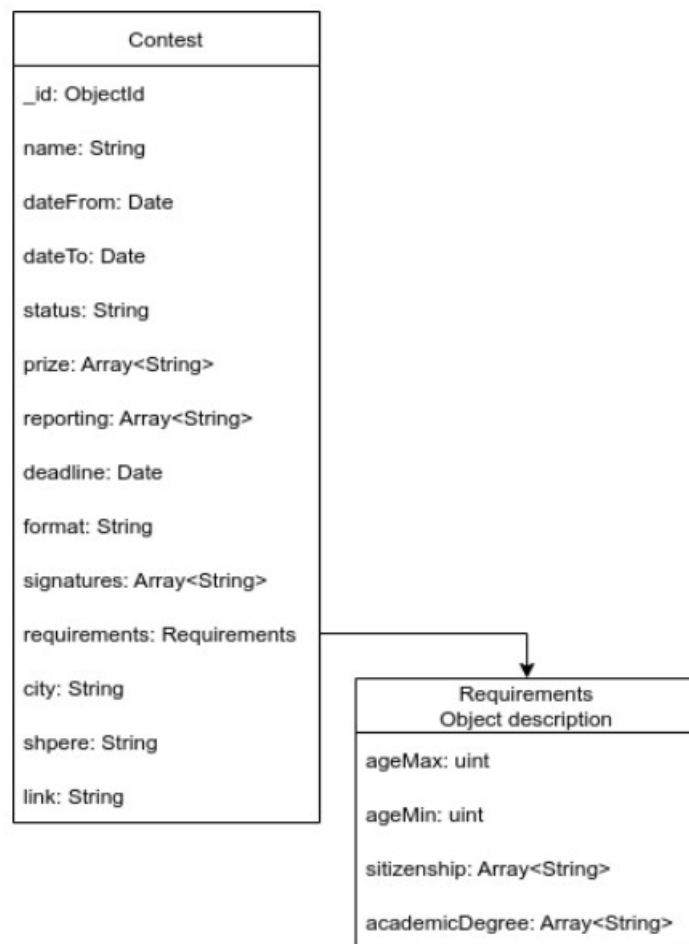


Рисунок 1 – Графическое представление нереляционной базы данных

БД содержит одну коллекцию – **Contests**. Поля данной коллекции следующие:

- `_id` - уникальный идентификатор узла
- `name` - название конкурса
- `dateFrom` - дата начала
- `dateTo` - дата окончания
- `status` - статус конкурса
- `prize` - приз

- reporting - формат подачи документов
- deadline - срок подачи документов
- format - формат конкурса(Дистанционный/Очный/Смешанный)
- signatures - необходимые подписи
- city - место проведения
- sphere - сфера конкурса

Оценим объем базы данных.

Коллекция Contests:

- _id - ObjectId. $V = 12b$
- name - String. $V = N_n$, где $N_n \sim 100$, средняя длина названия конкурса. $V = 100b$
- dateFrom - Date. $V = 8b$
- dateTo - Date. $V = 8b$
- status - String. $V = N_s$, где $N_s \sim 6$, средняя длина статуса. $V = 6b$
- prize - Array. $V = N_p$, где $N_p \sim 50$, средняя длина описания приза. $V = 50b$
- reporting - Array. $V = N_r$, где $N_r \sim 50$, средняя длина формата. $V = 50b$
- deadline - Date. $V = 8b$
- signatures - Array. $V = N_{si}$, где $N_{si} \sim 50$, средняя длина формата. $V = 50b$
- requirements - Requirements. $V = 26b$.
- format - String. $V = N_f$, где $N_f \sim 5$, средняя длина формата. $V = 5b$
- city - String. $V = N_c$, где $N_c \sim 6$, средняя длина названия города. $V = 6b$
- sphere - String. $V = N_{sh}$, где $N_{sh} \sim 8$, средняя длина названия сферы. $V = 8b$
- link - String. $V = N_l$, где $N_l \sim 50$, средняя длина ссылки. $V = 50b$

Тип Requirements

- ageMax - uint. $V = 4b$.
- ageMin - uint. $V = 4b$.
- citizenship - Array. $V = N_{ci}$, где $N_{ci} \sim 8$, средняя длина гражданства. $V = 8b$
- academicDegree - Array. $V = N_{ad}$, где $N_{ad} \sim 10$, средняя длина ученой степени. $V = 10b$
- comment - String. $V = N_{com}$, где $N_{com} \sim 60$, средняя длина комментария. $V = 60b$

В среднем на хранение одного конкурса нужно 387 байт. Для хранения N конкурсов нужно: $V(N) = 387 * N$

Запросы к БД представлены на рисунках 2-4.

- Запрос на получение конкурса по ссылке :

```
db.contests.findOne({  
  'link': link  
})
```

Рисунок 2 – Запрос на получение конкурса по ссылке

- Запрос на добавление конкурса:

```
db.contests.insertOne({  
  ...  
})
```

Рисунок 3 – Запрос на добавление конкурса

- Запрос на получение конкурсов на странице с заданным фильтром и пагинацией :

```
db.contests.find({  
  'city' : city,  
  'sphere' : shpere,  
  'dateTo' : {$gte: dateFrom}  
  'dateFrom' : {$lte: dateTo}  
}).skip(10*(n-1)).limit(10)
```

Рисунок 4 – Запрос на получение конкурсов на странице с заданным фильтром
и пагинацией

3.2. Реляционный аналог модели данных

Разработана схема реляционной базы данных (рисунок 5).

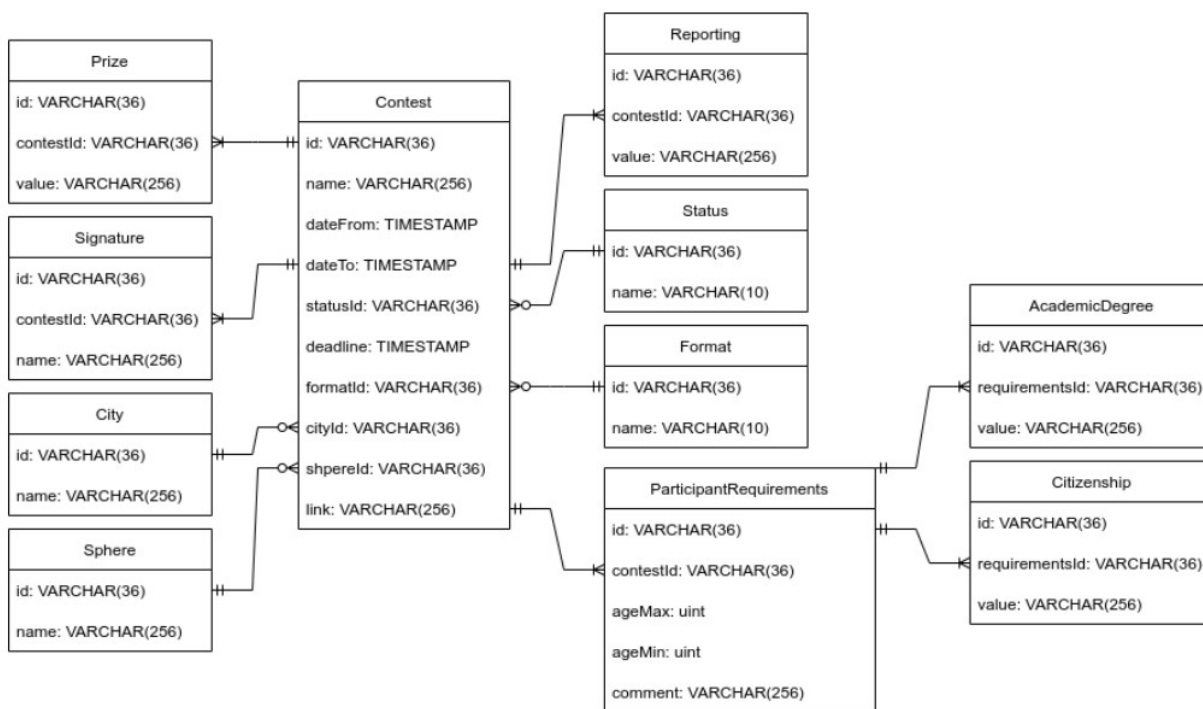


Рисунок 5 – Графическое представление реляционной базы данных

Ниже приведены описания таблиц.

Таблица "City"

- id - уникальный идентификатор города. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- name - название города. Тип VARCHAR(256). V = Nn, где Nn ~ 100, средняя длина названия конкурса. V = 100b

Таблица "Sphere"

- id - уникальный идентификатор сферы. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- name - название сферы. Тип VARCHAR(256). V = Nsh, где Nsh ~ 8, средняя длина названия сферы. V = 8b

Таблица "Status"

- id - уникальный идентификатор сферы. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- name - название статуса. Тип VARCHAR(10). V = Ns, где Ns ~ 6, средняя длина статуса. V = 6b

Таблица "Format"

- id - уникальный идентификатор формата. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- name - название формата. Тип VARCHAR(10). V = Nf, где Nf ~ 5, средняя длина формата. V = 5b

Таблица "Prize"

- id - уникальный идентификатор приза. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- contestId - название сферы. Тип VARCHAR(36). V = 36b
- value - описание приза. Тип VARCHAR(256). V = Nr, где Nr ~ 50, средняя длина описания приза. V = 50b

Таблица "Reporting"

- id - уникальный идентификатор отчетности. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- contestId - название сферы. Тип VARCHAR(36). V = 36b
- value - описание отчетности. Тип VARCHAR(256). V = Nr, где Nr ~ 50, средняя длина формата. V = 50b

Таблица "Signature"

- id - уникальный идентификатор подписи. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- contestId - название сферы. Тип VARCHAR(36). V = 36b
- name - название сферы. V = Nsi, где Nsi ~ 50, средняя длина формата. V = 50b

Таблица "ParticipantRequirements"

- id - уникальный идентификатор требований. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- contestId - название сферы. Тип VARCHAR(36). V = 36b
- ageMax - максимальный возраст. Тип uint. V = 4b

- ageMin - максимальный возраст. Тип uint. V = 4b
- comment - комментарий. Тип VARCHAR(256). V = Ncom, где Ncom ~ 60, средняя длина комментария. V = 60b

Таблица "AcademicDegree"

- id - уникальный идентификатор академической степени. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- requirementsId - уникальный идентификатор требований. Тип VARCHAR(36). V = 36b
- value - название сферы. V = Nad, где Nad ~ 10, средняя длина ученой степени. V = 10b

Таблица "Citizenship"

- id - уникальный идентификатор гражданства. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- requirementsId - уникальный идентификатор требований. Тип VARCHAR(36). V = 36b
- value - название сферы. V = Nci, где Nci ~ 8, средняя длина гражданства. V = 8b

Таблица "Contest"

- id - уникальный идентификатор конкурса. Тип VARCHAR(36). Для идентификации будет использоваться UUID. V = 36b
- name - название конкурса. Тип VARCHAR(256). V = Nn, где Nn ~ 100, средняя длина названия конкурса. V = 100b
- dateFrom - дата начала конкурса. Тип TIMESTAMP. V = 13b
- dateTo - дата окончания конкурса. Тип TIMESTAMP. V = 13b
- statusId - уникальный идентификатор текущего статус конкурса (идет / окончен / не начался). Тип VARCHAR(36). V = 36b
- deadline - срок подачи документов для участия в конкурсе. Тип TIMESTAMP. V = 13b
- formatId - уникальный идентификатор формата подачи документов для участия в конкурсе (онлайн / очно). Тип VARCHAR(36). V = 36b
- signatureId - подписи, которые требуются в документах для участия в конкурсе (только участника / ректора / руководителя...). Тип VARCHAR(36). V = 36b
- cityId - уникальный идентификатор города. Служит внешним ключом для связи с таблицей "City". Тип VARCHAR(36). V = 36b
- sphereId - уникальный идентификатор сферы. Служит внешним ключом для связи с таблицей "Sphere". Тип VARCHAR(36). V = 36b

- link - ссылка на конкурс. Тип VARCHAR(256). V = Nl, где Nl ~ 50, средняя длина ссылки. V = 50b

Для одного конкурса потребуется в среднем:

- одна запись в таблице "City" = 136b
- одна запись в таблице "Sphere" = 44b
- одна запись в таблице "Contest" = 404b
- одна запись в таблице "Format" = 41b
- одна запись в таблице "Status" = 42b
- одна запись в таблице "Prize" = 122b
- одна запись в таблице "Reporting" = 122b
- одна запись в таблице "Signature" = 122b
- одна запись в таблице "ParticipantRequirements" = 140b
- одна запись в таблице "AcademicDegree" = 82b
- одна запись в таблице "CitizenShip" = 80b

Средний объем конкурса составит $V = 1335b$, поскольку поле prize может оставаться пустым (конкурсы бывают без призов) данный объем может иногда отличаться в меньшую сторону.

Запросы к БД представлены на рисунках 6-8.

- Запрос на получение конкурса по ссылке :

```
SELECT *
from contests
JOIN cities ON(contests.cityId = cities.id)
JOIN formats ON(contests.formatId = formats.id)
JOIN statuses ON(contests.cityId = statuses.id)
JOIN spheres ON(contests.sphereId = spheres.id)
JOIN prizes ON(contests.id = prizes.contestId)
JOIN signatures ON(contests.id = signatures.contestId)
JOIN reportings ON(contests.id = reportings.contestId)
JOIN participantRequirements ON(contests.id = participantRequirements.contestId)
JOIN academicDegrees ON(participantRequirements.id = academicDegrees.requirementsId)
JOIN citizenships ON(participantRequirements.id = citizenships.requirementsId)
WHERE link = link
```

Рисунок 6 – Запрос на получение конкурса по ссылке

- Запрос на добавление конкурса:

```
INSERT contests(...)
VALUES(...)
```

Рисунок 7 – Запрос на добавление конкурса

- Запрос на получение конкурсов на странице с заданным фильтром и пагинацией :

```
SELECT *
from contests
JOIN cities ON(contests.cityId = cities.id)
JOIN formats ON(contests.formatId = formats.id)
JOIN statuses ON(contests.cityId = statuses.id)
JOIN spheres ON(contests.sphereId = spheres.id)
JOIN prizes ON(contests.id = prizes.contestId)
JOIN signatures ON(contests.id = signatures.contestId)
JOIN reportings ON(contests.id = reportings.contestId)
JOIN participantRequirements ON(contests.id = participantRequirements.contestId)
JOIN academicDegrees ON(participantRequirements.id = academicDegrees.requirementsId)
JOIN citizenships ON(participantRequirements.id = citizenships.requirementsId)
WHERE link = link
OFFSET 10(n-1)
LIMIT 10
```

Рисунок 8 - Запрос на получение конкурсов на странице с заданным фильтром и пагинацией

3.3. Сравнение моделей

Избыточными полем в NoSQL модели является поле `_id`. Тогда объем избыточных данных равен 12 байтам, а чистых $401 \cdot N$.

Для NoSQL: $(413 / 401) \sim 1.03$.

Для SQL: $(1335 / 401) \sim 3.33$.

Значения размера моделей SQL и NoSQL являются средними и посчитаны в пунктах описания моделей данных.

SQL модель является избыточной с точки зрения памяти, чего не скажешь про NoSQL.

SQL модель данных требует больше места по сравнению с MongoDB, но в MongoDB присутствует дублирование информации. Также SQL модель требует больше времени на выполнение запросов из-за соединения таблиц.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Описание приложения

Серверная часть в основе своей использует mongoose и TypeScript и взаимодействует с mongodb.

Клиентская часть в основе своей использует библиотеку React.

Для просмотра информации о каталоге конкурсных объявлений достаточно просто зайти на сайт и перейти на страницу «База данных».

Снимки экранов разработанного приложения представлены в приложении Б.

4.2. Используемые технологии

Для клиентской части используется следующий стек технологий:

- язык программирования JavaScript;
- библиотека React;

Для серверной части используется следующий стек технологий:

- язык программирования TypeScript
- платформа mongoose
- база данных MongoDB.

Для сборки проекта используется Docker Compose.

ЗАКЛЮЧЕНИЕ

В ходе данной работы был разработан макет веб-приложения каталога конкурсных объявлений, на основании которого был реализован прототип такого приложения, содержащий основной функционал. Приложение позволяет просматривать список конкурсов, фильтровать его, производить поиск, смотреть статистику по каталогу.

В дальнейшем данное приложение можно расширить, добавив новые модули.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Mongoose [Электронный ресурс] URL: <https://mongoosejs.com/>
2. JavaScript [Электронный ресурс] URL:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
3. TypeScript [Электронный ресурс] URL:
<https://www.typescriptlang.org/>
4. MongoDB Documentation [Электронный ресурс] URL:
<https://www.mongodb.com/docs>
5. React Documentation [Электронный ресурс]. URL:
<https://reactjs.org/docs/getting-started.html> (дата обращения
25.10.2022)
6. Docker Documentation [Электронный ресурс]. URL:
<https://docs.docker.com>

ПРИЛОЖЕНИЕ А

МАКЕТ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Ссылка:

ЗАГРУЗИТЬ

| Название конкурса | Дата начала | Дата окончания | Место проведения | Сфера | Статус |
|-------------------|-------------|----------------|------------------|---------|---------|
| Example | Example | Example | Example | Example | Example |

При нажатии появится информация о конкурсе

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Ссылка:

ЗАГРУЗИТЬ

Начальная страница с поиском

При нажатии происходит переход в страницу базы данных

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Поиск:

| НАЗВАНИЕ КОНКУРСА | ДАТА НАЧАЛА | ДАТА ОКОНЧАНИЯ | МЕСТО ПРОВЕДЕНИЯ | СФЕРА | Статус |
|---|-------------|----------------|------------------|------------|-----------|
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Предстоит |
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Завершен |

Rows per page: 10 1 of 2

Статистика

Популярный город: Москва

Редкий город: Татарстан

Страница с базой данных и статистикой

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Ссылка:

ЗАГРУЗИТЬ

| Название конкурса | Дата начала | Дата окончания | Место проведения | Сфера | Статус |
|-------------------|-------------|----------------|------------------|---------|---------|
| Example | Example | Example | Example | Example | Example |

Начальная страница с поиском

Кнопка для скачивания

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Contests

| НАЗВАНИЕ КОНКУРСА | ДАТА НАЧАЛА | ДАТА ОКОНЧАНИЯ | МЕСТО ПРОВЕДЕНИЯ | СФЕРА | Статус |
|---|-------------|----------------|------------------|--------------|-----------|
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Предстоит |
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Завершен |
| <input type="checkbox"/> Олимпиада по русскому языку | 2022.12.12 | 2022.12.15 | Татарстан | Русский язык | Идет |

1 of 3

Статистика

Популярный город: Москва

Редкий город: Татарстан

Страница с базой данных и статистикой

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Contests

| НАЗВАНИЕ КОНКУРСА | ДАТА НАЧАЛА | ДАТА ОКОНЧАНИЯ | МЕСТО ПРОВЕДЕНИЯ | СФЕРА | Статус |
|---|-------------|----------------|------------------|--------------|-----------|
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Предстоит |
| <input type="checkbox"/> Московская олимпиада по математике | 2022.12.12 | 2022.12.15 | Москва | Математика | Завершен |
| <input type="checkbox"/> Олимпиада по русскому языку | 2022.12.12 | 2022.12.15 | Татарстан | Русский язык | Идет |

Rows per page: 10 1 of 3

Статистика

Популярный город: Москва

Редкий город: Татарстан

Страница с базой данных и статистикой

Пагинация

Окно фильтров

БАЗА ДАННЫХ Конкурсы и гранты НА ГЛАВНУЮ

Contests

Фильтры

Название конкурса:

Дата начала:

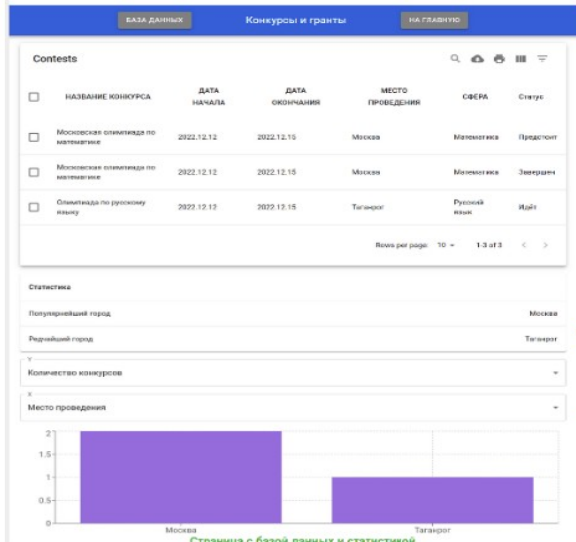
Дата окончания:

Место проведения:

Сфера:

Статус: ☐ Завершен ☐ Идет ☐ Предстоит

Страница с базой данных и статистикой



Страница с базой данных и статистикой

Статистика

Диаграмма на основе фильтров таблицы с выбором осей

ПРИЛОЖЕНИЕ Б

СНИМКИ ЭКРАНА ПРИЛОЖЕНИЯ

База данных

Конкурсы и гранты

На главную

Ссылка

Загрузить

| Атрибут | Значение |
|---------------------|----------|
| Название | |
| Дата начала | |
| Дата конца | |
| Статус | |
| Награда | |
| Отчетность | |
| Формат | |
| Требования | |
| Город | |
| Ссылка | |
| Ссылка на документы | |

Очистить или добавить конкурс

База данных

Конкурсы и гранты

На главную

Contests

🔍

🔼

| ИЗВАНМЕ | ДАТА НАЧАЛА | ДАТА КОНЦА | СТАТУС | НАГРАДА | ОТЧЕТНОСТЬ | ФОРМАТ | ТРЕБОВАНИЯ | ГОРОД |
|---|-------------|------------|--------|---------|------------|--------|------------|-------|
| К сожалению, подходящие записи не найдены | | | | | | | | |
| Открыть страницу: 1 | | | | | | | | |
| Строк на странице: 10 | | | | | | | | |
| 0-0 из 0 | | | | | | | | |

Искать

Выберите файл

Файл не выбран

UPLOAD

Список

Популярный город

Разный город

Не выбрано

0

count

auto