

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

**ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ**  
**по дисциплине «Введение в нереляционные базы данных»**  
**Тема: Каталог упражнений йоги**

Студентка гр. 1303	_____	Куклина Ю.Н.
Студентка гр. 1303	_____	Сырцева Д.Д.
Студент гр. 1303	_____	Самохин К.А.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург  
2024

## ЗАДАНИЕ

Студентка Куклина Ю.Н.

Студентка Сырцева Д.Д.

Студент Самохин К.А.

Группа 1303

Тема: Каталог упражнений йоги

Исходные данные:

Необходимо реализовать веб-приложение для каталога упражнений йоги с использованием СУБД MongoDB.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложения»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 05.09.2024

Дата сдачи реферата: 26.11.2024

Дата защиты реферата: 26.11.2024

Студентка	_____	Куклина Ю.Н.
Студентка	_____	Сырцева Д.Д.
Студент	_____	Самохин К.А.
Преподаватель	_____	Заславский М.М.

## АННОТАЦИЯ

В рамках ИДЗ разработано веб-приложение, представляющее собой каталог упражнений йоги. Приложение включает функционал для просмотра упражнений, их добавления, редактирования и удаления, а также возможность оставлять отзывы. Реализована система фильтрации и поиска упражнений по различным критериям.

Для разработки использованы технологии Vue.js, TypeScript, Nest.js, СУБД MongoDB.

Найти исходный код можно по ссылке: [yoga\\_catalog](#)

## SUMMARY

As part of the IDZ, a web application has been developed that is a catalog of yoga exercises. The application includes functionality for viewing exercises, adding, editing, and deleting them, as well as the ability to leave reviews. A system for filtering and searching exercises by various criteria has been implemented.

Vue.js, TypeScript, Nest.js, and MongoDB DBMS technologies were used for the development.

The source code can be found at the link: [yoga\\_catalog](#)

## СОДЕРЖАНИЕ

1.	Введение	6
1.1.	Актуальность проблемы	6
1.2.	Постановка задачи	6
1.3.	Предлагаемое решение	6
1.4.	Качественные требования к решению	6
2.	Сценарии использования	7
2.1.	Макет UI	7
2.2.	Сценарий использования для импорта данных	12
2.3.	Сценарий использования для представления данных	14
2.4.	Сценарий использования для анализа данных	16
2.5.	Сценарий использования для экспорта данных	17
2.6.	Вывод	17
3.	Модель данных	19
3.1.	Нереляционная модель данных	19
3.2.	Аналог модели данных для SQL СУБД	24
3.3.	Сравнение моделей	27
4.	Разработанное приложение	29
5.	Выводы	31
6.	Приложения	33
7.	Литература	35

# **1. ВВЕДЕНИЕ**

## **1.1. Актуальность проблемы**

Рост популярности йоги как практики для укрепления здоровья требует удобных инструментов для подбора упражнений, соответствующих потребностям пользователей. На текущий момент существующие решения часто не предлагают удобных механизмов для фильтрации и поиска нужных упражнений, а также не поддерживают возможность управления ими.

## **1.2. Постановка задачи**

Задача проекта заключается в разработке веб-приложения, которое позволяет пользователям:

- Находить асаны с подробным описанием, техникой выполнения и указанием пользы и противопоказаний;
- Фильтровать упражнения по различным параметрам;
- Оставлять отзывы;
- Взаимодействовать с удобным и интуитивным интерфейсом.

Также требуется обеспечить надежное хранение данных и высокую производительность приложения.

## **1.3. Предлагаемое решение**

Для реализации создается веб-приложение с использованием Vue.js, Nest.js для серверной части и MongoDB для хранения данных. Приложение поддерживает фильтрацию, поиск и публикацию отзывов.

## **1.4. Качественные требования к решению**

Решение должно быть удобным, производительным, надёжным и легко расширяемым.

## 2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

### 2.1. Макет UI

Ниже представлены макеты страниц приложения.



Рисунок 1. Каталог упражнений + Фильтрация

YOGA

to go

[Главная](#)
[Каталог упражнений](#)

Редактировать асану

Удалить асану

Халасана

Добавлена 16.06.24 в 18:31

★★★★★

Описание

Халасана, или Поза Плуга, — это классическая йогическая асана, которая укрепляет и расслабляет мышцы спины и плеч. Название асаны происходит от слова «кала», что на санскрите означает «плуг». Это символизирует то, как тело в этой позе напоминает форму традиционного сельскохозяйственного орудия. Асана улучшает гибкость позвоночника и стимулирует работу внутренних органов, особенно органов пищеварения.

Техника выполнения

- Лягте на спину, вытянув руки вдоль туловища, ладони опущены на пол.
- На вдохе поднимите ноги перпендикулярно полу, удерживая их вместе, и продолжайте поднимать таз, помогая себе руками, чтобы завести ноги за голову.
- Опустите пальцы ног на пол за головой, стараясь выпрямить ноги. Спина должна быть перпендикулярна полу, а вес тела распределён на плечах, не на шее.
- Руки можно вытянуть за спиной или согнуть в локтях и поддерживать спину.
- Держите позу в течение нескольких секунд до 1-5 минут, дыша ровно и спокойно.
- Чтобы выйти из асаны, медленно верните ноги в исходное положение, перекачиваясь позвонком за позвонком.

Противопоказания

- ✗ Заболевания сердечно-сосудистой системы (ССС);
- ✗ Критические дни;
- ✗ Травмы и заболевания опорно-двигательного аппарата (ОДА), внутренних органов в острой фазе;
- ✗ Повышенное внутричерепное давление;
- ✗ Заболевания глаз (глаукома, катаракта, отслоение сетчатки и тд);
- ✗ Гипертиреоз;

Польза

- ✓ Стимулирует венозный возврат и лимфоотток;
- ✓ Тренирует вестибулярный аппарат (улучшает чувство равновесия и координацию движений);
- ✓ Улучшает пищеварение;
- ✓ Укрепляет мышечный корсет и повышает общий тонус организма;
- ✓ Вытягивает всю заднюю линию тела и успокаивает нервную систему.

Отзывы

ОСТАВИТЬ ОТЗЫВ

Анастасия, 28 лет

18.09 20:34

★★★★★

Халасана — одно из моих любимых упражнений. После дня, проведённого за компьютером, эта асана снимает напряжение с шеи и спины. Я чувствую, как вытягивается позвоночник, и появляется лёгкость в теле. Сначала было сложно удерживать позу, но регулярная практика помогает становиться гибче.

Игорь, 35 лет

18.09 13:43

★★★★★

Халасана даёт отличное ощущение расслабления, особенно в области плеч и шеи. Правда, сначала мне было немного некомфортно из-за давления на шею, но после консультации с инструктором и использования одеяла под плечи стало значительно легче.

Мария, 40 лет

18.09 10:43

★★★★★

Эта асана буквально спасла мою спину! Сложный день на работе, и часто чувствую усталость и боли в пояснице. После практики Халасаны я ощущаю значительное облегчение и расслабление. Рекомендую тем, у кого сидячий образ жизни.

СМОТРЕТЬ ВСЕ

Рисунок 2. Страница упражнения

8





## Отзывы на асану Халасана

[ОСТАВИТЬ ОТЗЫВ](#)

### Отзывы

**Анастасия, 28 лет**

16.09 10:43



Халасана – одно из моих любимых упражнений. После дня, проведённого за компьютером, эта асана снимает напряжение с шеи и спины. Я чувствую, как выпрямляется позвоночник, и появляется лёгкость в теле. Сначала было сложно удерживать позу, но регулярная практика помогает становиться гибче.

**Игорь, 35 лет**

15.09 10:43



Халасана даёт отличное ощущение расслабления, особенно в области плеч и шеи. Правда, сначала мне было немного некомфортно из-за давления на шею, но после консультации с инструктором и использования одеяла под плечи стало значительно легче.

**Мария, 40 лет**

14.09 10:43



Эта асана буквально спасла мою спину! Сиджу целый день на работе, и часто чувствую усталость и боли в пояснице. После практики Халасаны я ощущаю значительное облегчение и расслабление. Рекомендую тем, у кого сидячий образ жизни.

**Анастасия, 28 лет**

13.09 10:43



Халасана – одно из моих любимых упражнений. После дня, проведённого за компьютером, эта асана снимает напряжение с шеи и спины. Я чувствую, как выпрямляется позвоночник, и появляется лёгкость в теле. Сначала было сложно удерживать позу, но регулярная практика помогает становиться гибче.

**Игорь, 35 лет**

12.09 10:43



Халасана даёт отличное ощущение расслабления, особенно в области плеч и шеи. Правда, сначала мне было немного некомфортно из-за давления на шею, но после консультации с инструктором и использования одеяла под плечи стало значительно легче.

**Мария, 40 лет**

10.09 10:43



Эта асана буквально спасла мою спину! Сиджу целый день на работе, и часто чувствую усталость и боли в пояснице. После практики Халасаны я ощущаю значительное облегчение и расслабление. Рекомендую тем, у кого сидячий образ жизни.

Рисунок 4. Страница всех отзывов

YOGA  
teacher

Главная Каталог упражнений

# ОСТАВИТЬ ОТЗЫВ

Имя\*

Возраст\*

Название асаны\*  
Халасана

Опишите свои ощущения от асаны\*

Сколько звёзд ставите асане?\*

★ ★ ★ ★ ★

ОТПРАВИТЬ

Рисунок 5. Страница публикации отзыва

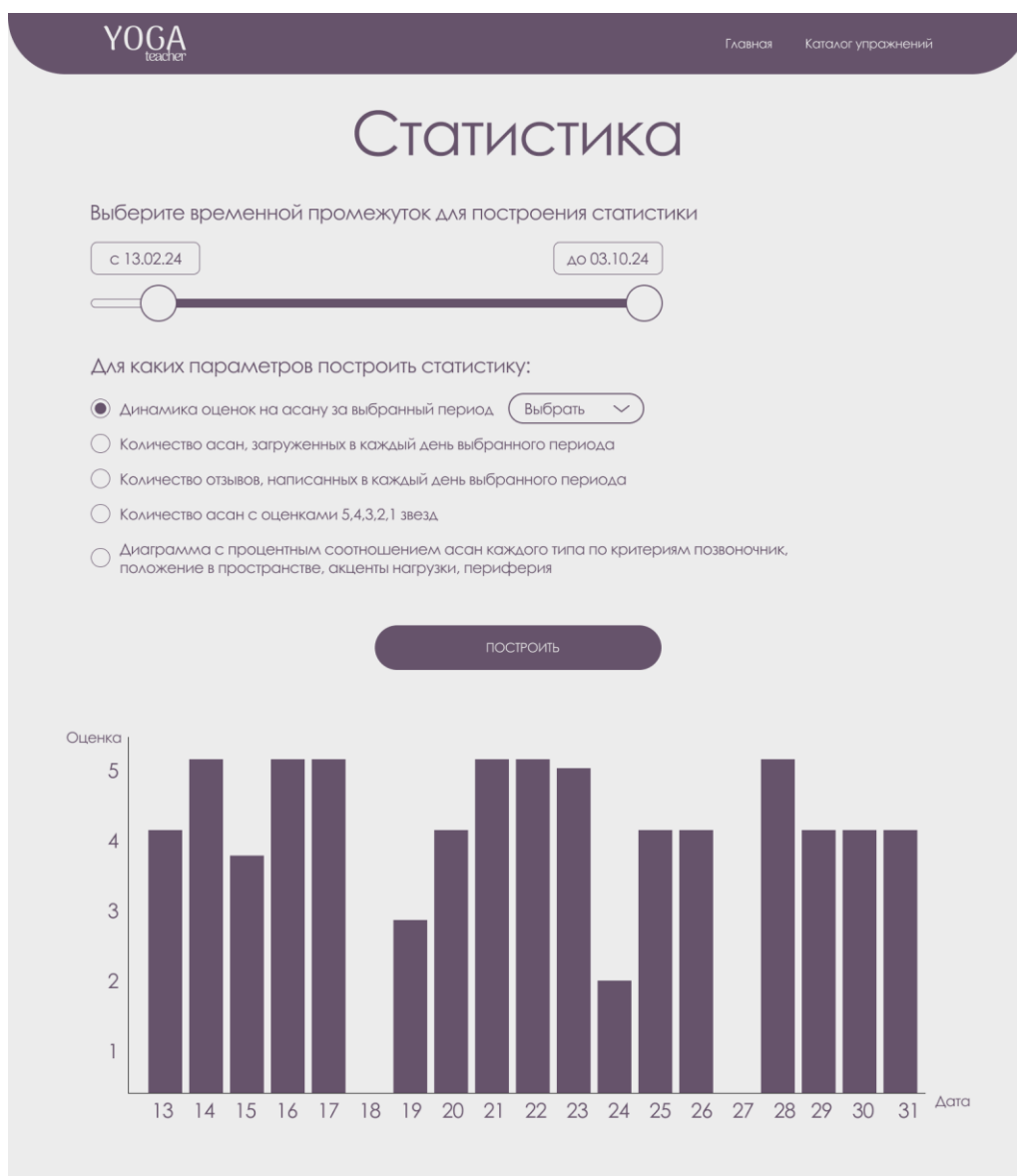


Рисунок 6. Страница построения статистики

## 2.2. Сценарий использования для импорта данных

### а) Сценарий использования: Импорт данных

#### Цель:

Обновить базу данных.

#### Действующее лицо:

Пользователь

#### Основной сценарий:

1. Пользователь попадает на главную страницу.
2. Пользователь нажимает кнопку «Импортировать данные».

3. Система отображает форму для загрузки файла.
4. Пользователь выбирает 3 файла с определенными именами и разрешением .bson, из которых хочет считать данные (exercises.bson, images.chunk.bson, images.files.bson).
5. Пользователь нажимает кнопку "Загрузить".
6. Система проверяет формат загружаемых файлов.
7. Система начинает процесс импорта данных.
8. Система сохраняет данные в базе данных.

**Альтернативные сценарии:**

7а. Неверный формат или имена файлов.

7а.1 Система выводит сообщение об ошибке и предлагает повторить попытку с файлами в правильном формате.

**Результат:**

База данных обновлена.

**б) Сценарий использования: Добавление нового упражнения**

**Цель:**

Пользователь добавляет новое упражнение.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь попадает на главную страницу.
2. Пользователь нажимает кнопку «Добавить асану».
3. Открывается страница добавления асаны.
4. Пользователь вносит информацию об асане.
5. Пользователь нажимает кнопку «Сохранить».
6. Система сохраняет данные и возвращает пользователя на главную страницу.

**Альтернативные сценарии:**

6а. Ошибка при сохранении изменений.

6а.1 Система выводит сообщение об ошибке и предлагает повторить попытку.

4а. Пользователь не заполнил все поля ввода.

4а.1 Поля ввода подсвечиваются красным, под полями ввода написано сообщение «Это обязательное поле».

**Результат:**

Упражнение добавлено в базу данных и отображается в каталоге.

### **2.3. Сценарий использования для представления данных**

Для просмотра данных в виде упражнений и отзывов на них существует три сценария использования: просмотр каталога, просмотр страницы упражнения и просмотр отзывов.

#### **1. Сценарий использования: Просмотр каталога упражнений**

**Цель:**

Пользователь просматривает каталог упражнений и применяет фильтры для поиска.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает главную страницу приложения.
2. Пользователь переходит на страницу каталога упражнений.
3. На экране отображается список упражнений и фильтры. Изначально, пока пользователь не ввел параметры фильтрации, на странице показываются «рекомендованные» упражнения.
4. Пользователь отмечает параметры фильтрации, нажимает кнопку «Применить фильтры».
5. Система обновляет список упражнений на основе выбранных фильтров, пока выполняется запрос, на экране отображается индикатор загрузки.
6. Пользователь просматривает результаты и выбирает нужное упражнение.

**Альтернативные сценарии:**

4а. Пользователь вводит название асаны в поиске и по мере ввода с некоторой задержкой выполняется запрос.

4a.1 Пока выполняется запрос, на экране отображается индикатор загрузки.

6a. Пользователь нажимает кнопку «Сбросить фильтры».

6a.1 Параметры фильтрации очищаются, снова отображаются «рекомендованные» упражнения.

5a. Нет результатов по выбранным фильтрам.

5a.1. Система выводит сообщение, что по текущим критериям ничего не найдено, и предлагает изменить фильтры.

**Результат:**

Пользователь находит интересующее упражнение.

**2. Сценарий использования: Просмотр страницы упражнения**

**Цель:**

Пользователь просматривает детальную информацию об упражнении.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Вызывается сценарий «Просмотр каталога упражнений».
2. Пользователь выбирает упражнение в каталоге – нажимает на кнопку «Подробнее».
3. Система отображает страницу с подробным описанием упражнения, изображением, техникой выполнения, показаниями и противопоказаниями, а также отзывами на упражнение.

**Альтернативные сценарии:**

3a. Ошибка при загрузке страницы.

3a.1 Система выводит сообщение об ошибке и предлагает повторить попытку.

**Результат:**

Пользователь получает полную информацию о выбранном упражнении.

**3. Сценарий использования: Просмотр отзывов на упражнение**

**Цель:**

Пользователь просматривает отзывы на выбранное упражнение.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Вызывается сценарий «Просмотр страницы упражнения».
2. Пользователь нажимает на кнопку «Смотреть все» в разделе отзывы.
3. Система отображает страницу со всеми отзывами на данную асану.

**Альтернативные сценарии:**

- 1а. Пользователь на странице каталога выбирает "Искать по отзывам".
  - 1а.1 Открывается страница, где можно по тексту отзыва найти все совпадения, а также асану, на которую эти отзывы оставлены.
- 3а. Ошибка при загрузке страницы.
  - 3а.1 Система выводит сообщение об ошибке и предлагает повторить попытку.

**Результат:**

Пользователь просматривает отзывы других пользователей.

**2.4. Сценарий использования для анализа данных****Сценарий использования: Подсчет статистики в системе****Цель:**

Посмотреть статистику данных в системе.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь попадает на главную страницу.
2. Пользователь нажимает кнопку «Кастомизированная статистика».
3. Пользователь выбирает временной промежуток для построения статистики.
4. Пользователь выбирает критерий, по которым построить статистику (например, динамика оценок на асану за выбранный период, количество асан, загруженных в каждый день и т.д.).



5. Система формирует отчет со статистикой (график, диаграмму и т.д.).
6. Система отображает отчет пользователю на экране.

**Альтернативные сценарии:**

5а. Отсутствуют данные для построения статистики.

5а.1 Система выводит сообщение об ошибке и предлагает выбрать другой критерий.

**Результат:**

Пользователь смотрит статистику, построенную на основе выбранных им данных.

## **2.5. Сценарий использования для экспорта данных**

### **Сценарий использования: Экспорт данных**

**Цель:**

Экспортировать данные в BSON файл.

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь попадает на главную страницу.
2. Пользователь нажимает кнопку «Экспортировать данные».
3. Система формирует 3 BSON файла (exercises.bson, images.chunk.bson, images.files.bson).
4. Файлы автоматически загружаются на устройство пользователя.

**Альтернативные сценарии:**

4а. Возникла ошибка при формировании данных.

4а.1 Система выводит сообщение об ошибке и предлагает повторить попытку.

**Результат:**

Данные экспортированы на устройство пользователя.

## **2.6. Вывод**

Для нашего решения в равной степени преобладают операции записи и чтения, так как пользователь как активно просматривает данные, так и добавляет, редактирует их в системе.

### 3. МОДЕЛЬ ДАННЫХ

#### 3.1. Нереляционная модель данных

##### Графическое представление модели

```
"exercises": {  
  "type": "array",  
  "items": {  
    "type": "object",  
    "properties": {  
      "id": {  
        "type": "integer",  
        "description": "Уникальный идентификатор упражнения"  
      },  
      "name": {  
        "type": "string",  
        "description": "Название упражнения"  
      },  
      "photo": {  
        "type": "string",  
        "description": "Фото упражнения в формате base64"  
      },  
      "description": {  
        "type": "string",  
        "description": "Описание упражнения"  
      },  
      "technique": {  
        "type": "string",  
        "description": "Описание техники выполнения"  
      },  
      "contraindications": {  
        "type": "array",  
        "description": "Список противопоказаний",  
        "items": {  
          "type": "string"  
        }  
      },  
      "benefits": {  
        "type": "array",  
        "description": "Список пользы от упражнения",  
        "items": {  
          "type": "string"  
        }  
      },  
      "spine": {  
        "type": "string",  
        "description": "Отдел позвоночника, на который идет нагрузка"  
      },  
    },  
  },  
}
```

```

    "position": {
      "type": "string",
      "description": "Положение в пространстве"
    },
    "load_focus": {
      "type": "string",
      "description": "Акценты нагрузки"
    },
    "periphery": {
      "type": "string",
      "description": "Периферия упражнения"
    },
    "rating": {
      "type": "number",
      "description": "Средняя оценка упражнения"
    },
    "reviews": {
      "type": "array",
      "description": "Список отзывов",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "number",
            "description": "id отзыва"
          },
          "name": {
            "type": "string",
            "description": "Имя пользователя"
          },
          "age": {
            "type": "integer",
            "description": "Возраст пользователя"
          },
          "stars": {
            "type": "integer",
            "minimum": 1,
            "maximum": 5,
            "description": "Количество звезд (оценка)"
          },
          "text": {
            "type": "string",
            "description": "Текст отзыва"
          },
          "date": {
            "type": "string",
            "format": "date",
            "description": "Дата отзыва"
          }
        }
      },
      "required": ["id", "name", "age", "stars", "date"]
    }
  },

```

```

    }
  },
  "required": ["id", "name", "technique", "contraindications", "benefits", "periphery", "spine", "position", "load_focus", "rating"]
}

```

Описание назначений коллекций, типов данных и сущностей представлено на json схеме.

## Оценка объема информации, хранимой в модели

Пусть:

N — количество упражнений в базе данных.

- Для каждого упражнения хранится:
- Название (name): строка (в среднем 25 байт).
- Описание упражнения (description): строка (в среднем 100 байт).
- Фото (photo): храним в формате base64, средний размер фото 2 Мб -> 2 МБ  
 $* 4 / 3 \approx 2,6 \text{ МБ}$  – размер файла в данном формате. ( $2,6 \text{ МБ} * 1\,048\,576 \text{ байт/МБ} \approx 2\,796\,202 \text{ байт}$ )
- Техника выполнения (technique): строка (в среднем 400 байт).
- Противопоказания (contraindications): массив строк, средняя длина строки — 50 байт, количество элементов массива — 5 (среднее число противопоказаний на упражнение).
- Польза (benefits): массив строк, средняя длина строки — 50 байт, количество элементов массива — 5 (среднее число пунктов пользы на упражнение).
- Позвоночник (spine): строка (в среднем 15 байт).
- Положение (position): строка (в среднем 15 байт).
- Акценты нагрузки (load\_focus): строка (в среднем 15 байт).
- Периферия (periphery): строка (в среднем 25 байт).
- Оценка (rating): число (8 байт).
- Дата добавления (date): строка (10 байт).
- Дата изменения (date): строка (10 байт).
- Отзывы (reviews): массив объектов. Отзыв включает:

- Имя (name): строка (в среднем 30 байт).
- Возраст (age): целое число (4 байта).
- Звезды (stars): целое число (4 байта).
- Текст отзыва (text): строка (в среднем 130 байт).
- Дата (date): строка (10 байт).

Пусть в среднем на одно упражнение приходится 17 отзывов.

Теперь объем каждого упражнения можно оценить как:

$$\begin{aligned} \text{Объем} &= \\ N * (25 + 100 + 2796202 + 400 + 50 \times 5 + 50 \times 5 + 15 + 15 + 15 + 25 + 8 + 17 \times (30 + 4 + 4 + 130 + 20)) &= \\ 2800501 * N \text{ байт} = 2734,86 * N \text{ Кб} = 2,6 * N \text{ Мб} \end{aligned}$$

Таким образом, объем данных напрямую зависит от количества упражнений N.

## Примеры запросов для совершения сценариев использования

### 1. Получение списка всех упражнений:

```
db.exercises.find({});
```

### 2. Поиск упражнения по имени:

```
db.exercises.find({ name: "Название упражнения" });
```

### 3. Получение информации о конкретном упражнении по его ID:

```
db.exercises.findOne({ _id: ObjectId("УникальныйID") });
```

### 4. Добавление нового упражнения:

```
db.exercises.insertOne({
  name: "Новое упражнение",
  technique: "Описание техники",
  contraindications: ["Противопоказание 1",
"Противопоказание 2"],
  benefits: ["Польза 1", "Польза 2"],
  spine: "Описание",
  position: "Положение",
  load_focus: "Акценты нагрузки",
  periphery: "Периферия",
  rating: 5,
```

```
    reviews: []  
  });
```

### **5. Добавление отзыва к упражнению:**

```
db.exercises.updateOne(  
  { _id: new ObjectId("УникальныйID") },  
  { $push: { reviews: { name: "Имя пользователя", age: 30,  
    stars: 4, text: "Отличное упражнение!", date: new Date() } } }  
);
```

### **6. Получение всех отзывов для конкретного упражнения:**

```
db.exercises.findOne({ _id: ObjectId("УникальныйID") }, {  
  reviews: 1 });
```

### **7. Обновление оценки упражнения:**

```
db.exercises.updateOne(  
  { _id: ObjectId("УникальныйID") },  
  { $set: { rating: 4.5 } }  
);
```

### **8. Удаление упражнения:**

```
db.exercises.deleteOne({ _id: ObjectId("УникальныйID") });
```

### **9. Экспорт данных:**

```
const collection = db.collection(collectionName);  
const documents = await collection.find({}).toArray();  
const json = JSON.stringify(documents, null, 2);  
fs.writeFileSync(outputFilePath, json, 'utf8');
```

### **10. Импорт данных:**

```
const data = fs.readFileSync('yoga_data.json', 'utf8');  
const exercises = JSON.parse(data);  
const collection = db.collection('yoga');  
await collection.insertMany(exercises);
```

### **11. Вычисление поля rating**

```
const collection = db.collection('yoga');  
const exercise = await collection.findOne({ _id: new  
  ObjectId(exerciseId) });
```

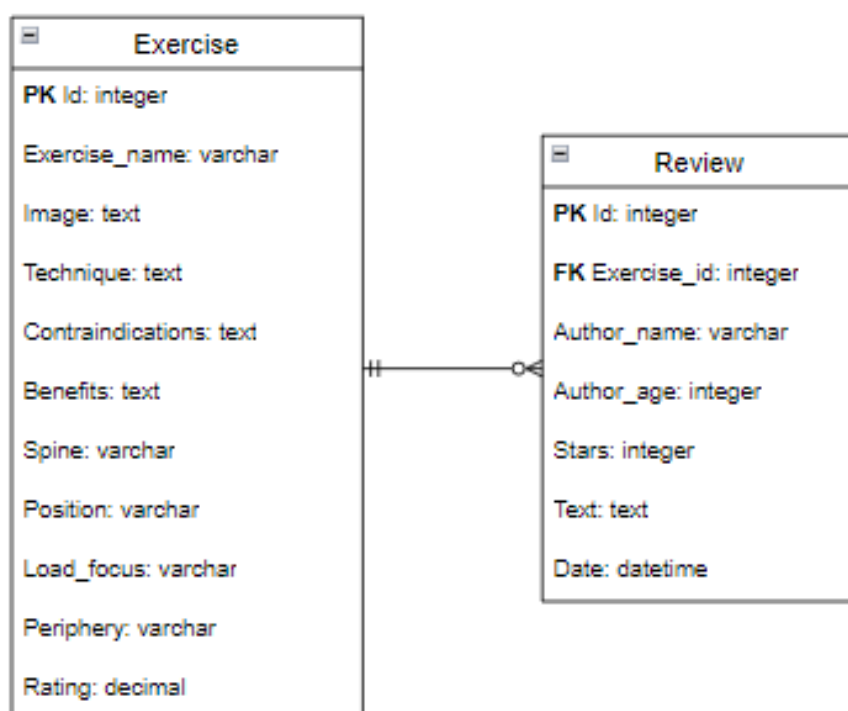
```

if (exercise && exercise.reviews && exercise.reviews.length >
0) {
  const totalStars = exercise.reviews.reduce((sum, review) =>
sum + review.stars, 0);
  const averageRating = totalStars / exercise.reviews.length;
  await collection.updateOne(
    { _id: new ObjectId(exerciseId) },
    { $set: { rating: averageRating } }
  );
}

```

## 3.2. Аналог модели данных для SQL СУБД

### Графическое представление модели



### Описание коллекций

#### Exercise

Назначение: хранение информации о существующих упражнениях.

У сущности Exercise есть следующие поля:

- Целочисленный идентификатор (Id) – serial, занимает 4 байта.
- Название асаны (Exercise\_name) – varchar(25), занимает 25 байт.



- Ссылка на изображение (Image) - text, изображение мы храним в формате base64, для фото среднего размера (2 Мб) объем файла в этой кодировке составит 2 796 202 байт.
- Техника выполнения (Technique) - text, в среднем будет занимать 400 байт.
- Противопоказания (Contraindications) – text, в среднем будет занимать 250 байт.
- Польза (Benefits) – text, в среднем будет занимать 250 байт.
- Область спины (Spine) – varchar(15), занимает 15 байт.
- Положение при выполнении (Position) – varchar(15), занимает 15 байт.
- Акцент нагрузки (Load\_focus) - varchar(15), занимает 15 байт.
- Периферия (Periphery) - varchar(25), занимает 25 байт.
- Средняя оценка по отзывам (Rating) – decimal, занимает в памяти 16 байт.
- Дата и время добавления упражнения Date – timestamp, занимает 8 байт в памяти.
- Дата и время обновления упражнения Date – timestamp, занимает 8 байт в памяти.

## Review

Назначение: хранение информации обо всех отзывах.

У сущности Review есть следующие поля:

- Целочисленный идентификатор (Id) – serial, занимает 4 байта.
- Идентификатор упражнения (Exercise\_id) - integer, занимает 4 байта.
- Имя автора (Author\_name) – varchar(30), занимает 30 байт.
- Возраст автора (Author\_age) – integer, занимает 4 байта.
- Оценка (Stars) – integer, занимает 4 байта.
- Содержимое отзыва (Text) – text, в среднем будет занимать 200 байт.
- Дата и время написания отзыва Date – timestamp, занимает 8 байт в памяти.

## Оценка объема информации, хранимой в модели

Вычислим средний размер объектов каждой сущности:

\* Exercise:  $4 + 25 * 2 + 2\,796\,202 + 400 + 250 * 2 + 15 * 3 + 16 = 2\,797\,217$  байт =  $2731.657$  Кб =  $2.6676$  Мб

\* Review:  $4 * 4 + 30 + 200 + 8 = 254$  байта.

Считаем, что у каждого упражнения в среднем имеется 17 отзывов.

Соответственно, в базе на каждую запись упражнения приходится 17 записей ОТЗЫВОВ.

$V(N) = N * (2\,797\,217 + 254 * 17) = 2\,801\,535 * N$  байт =  $2735.874 * N$  Кб =  $2.6717 * N$  Мб

## Примеры запросов

### 1. Получение списка всех упражнений:

```
SELECT * FROM Exercises;
```

### 2. Поиск упражнения по имени:

```
SELECT * FROM Exercises  
WHERE Exercise_name = @Searching_name
```

### 3. Получение информации о конкретном упражнении по его ID:

```
SELECT * FROM Exercises  
WHERE Id = @Searching_id
```

### 4. Добавление нового упражнения:

```
INSERT INTO Exercise(Exercise_name, Image_url, Technique,  
Contraindications, Benefits, Spine, Position, Load_focus,  
Periphery, Rating) VALUES  
( 'Имя', 'URL', 'Техника', 'Противопоказания', 'Польза',  
'Нижний отдел спины', 'Положение сидя', 'Акцент нагрузки',  
'Периферия', 4.93);
```

### 5. Добавление отзыва к упражнению:

```
INSERT INTO Review(Exercise_id, Author_name, Author_age, Stars,
Text, Date) VALUES
(1, 'George', 45, 5, 'Это упражнение помогло мне избавиться
от боли в спине!', '2014-04-04 20:00:00-07');
```

#### **6. Получение всех отзывов для конкретного упражнения:**

```
SELECT Review.Id, Exercise_name, Author_name, Author_age,
Stars, Text, Date FROM Review
JOIN Exercise ON Exercise.Id = Review.Exercise_id
WHERE Exercise_id = @Searching_id;
```

#### **7. Обновление оценки упражнения:**

```
UPDATE Exercise
SET Rating = New_rating
WHERE Id = @Searching_id;
```

#### **8. Удаление упражнения:**

```
DELETE FROM Exercise
WHERE Id = @Searching_id;
```

#### **9. Импорт данных:**

```
COPY Exercise (Exercise_name, Image_url, Technique,
Contraindications, Benefits, Spine, Position, Load_focus,
Periphery, Rating) FROM 'dataset.csv' DELIMITER ',' CSV HEADER;
```

#### **10. Экспорт данных:**

```
COPY (SELECT * FROM Exercise) TO 'dataset.csv' DELIMITER ','
CSV HEADER;
```

#### **11. Вычисление поля Rating:**

```
UPDATE Exercise
SET Rating = (SELECT AVG(Stars) FROM Review
WHERE Exercise_id = @Searching_id)
WHERE Id = @Searching_id;
```

### **3.3. Сравнение моделей**

#### **Удельный объем информации**

Реляционная модель имеет примерно такой же удельный объем информации, чем нереляционная. Связано это с тем, что в нереляционной модели мы не дублируем данные, поэтому она не требует больше места, чем реляционная модель.

### **Запросы по отдельным юзкейсам**

Количество запросов для каждой модели одинаково и равно 1.

### **Количество задействованных коллекций**

Обе модели используют одну коллекцию при любом сценарии использования, кроме сценария получения всех отзывов в реляционной модели, так там требуется применить JOIN для коллекций `exersices` и `reviews` (2 коллекции) и таким образом получить данные.

## **4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ**

### **4.1 Краткое описание**

Весь код приложения разделен на две части: back-end и front-end.

Back-end реализован с использованием фреймворка NestJS и служит элементом, обеспечивающим корректную передачу данных из БД на фронтенд для дальнейшего отображения. Так, бэкенд обрабатывает запросы к API и при помощи запросов достаёт из БД требуемые записи, после чего отправляет их на фронтенд.

Front-end обращается к API back-end части приложения и отображает данные в удобочитаемом виде. При разработке использовалось некое подобие FSD архитектуры, что значительно упрощает разработку и дальнейшее поддержание продукта. В проекте явно разделены страницы, на основании которых осуществляется навигация (директория pages), сущности, реализующие отдельные функциональные части интерфейса (директория entities), кастомные текстовые формы, селекторы, дэйтпикеры, чекбоксы и иконки (директория shared с поддиректориями ui и icons). Работа с данными на фронтенде реализуется через систему хранилищ и состояний при помощи библиотеки Pinia.

### **4.2 Используемые технологии**

БД: MongoDB.

Back-end: NestJS, TypeScript.

Front-end: TypeScript, Vue3, Nuxt, Pinia.

### **4.3 Схема экранов приложения**



Рисунок 7. Схема экранов приложения

## 5. ВЫВОДЫ

## **5.1 Достигнутые результаты**

В ходе работы было разработано приложение "Yoga Teacher", представляющее собой каталог асан, которое позволяет пользователям добавлять, удалять и редактировать упражнения. Приложение ориентировано на предоставление пользователям эффективного инструмента для управления и изучения асан.

Кроме того, приложение предоставляет возможность оставлять отзывы к упражнениям, осуществлять поиск отзывов по возможным свойствам и фильтровать асаны по различным характеристикам. Также реализована функция просмотра кастомизированной статистики. Для удобства работы с данными доступны функции импорта и экспорта, что позволяет переносить информацию, а также создавать бэкапы для безопасного хранения и восстановления данных.

## **5.2 Недостатки и пути для улучшения**

На данный момент приложение не предоставляет возможности авторизации и разграничения прав доступа, что делает его уязвимым к несанкционированным изменениям данных, усложняет контроль над действиями пользователей. Одним из возможных путей решения данной проблемы является внедрение системы авторизации с поддержкой ролей, которая позволит разграничивать доступ к функциям приложения в зависимости от уровня прав пользователя (например, администратор, инструктор, ученик). Это обеспечит больший контроль и защиту данных.

Также, отсутствует возможность прикрепления и просмотра видео выполнения упражнений, что могло бы улучшить практическое восприятие асан. Для решения данной проблемы можно добавить функцию загрузки видеороликов, что значительно улучшит пользовательский опыт.

Еще одним важным улучшением является многоязычность. На данный момент приложение доступно только на русском языке, что ограничивает

аудиторию пользователей. Внедрение поддержки нескольких языков позволит расширить аудиторию.

На данный момент приложение доступно только в веб-формате, что ограничивает его использование на мобильных устройствах. Необходимо создать мобильную версию приложения, адаптированную для iOS и Android. Это позволит пользователям иметь доступ к функционалу приложения в любое время и в любом месте.

Также необходимо улучшить адаптивность текущего веб-приложения, чтобы оно корректно отображалось на различных устройствах (смартфоны, планшеты, компьютеры) с разными размерами экранов. Это обеспечит комфортное использование приложения.

### **5.3 Будущее развитие решения**

В будущем планируется внедрение авторизации с разграничением прав доступа для повышения безопасности данных. Также будет добавлена функция загрузки видеороликов для улучшения восприятия асан и поддержка нескольких языков для расширения аудитории. Создание мобильной версии для iOS и Android, а также улучшение адаптивности веб-приложения сделают его доступным и удобным для пользователей на разных устройствах.



## 6. ПРИЛОЖЕНИЯ

### 6.1 Документация по сборке и разворачиванию приложения.

1. Склонировать репозиторий с проектом (ссылка указана в списке литературы) и перейти в директорию проекта.
2. Собрать контейнеры приложения командой: *docker-compose build --no-cache*.
3. Запустить контейнеры командой: *docker-compose up*.
4. Открыть приложение в браузере по адресу 127.0.0.1:3000 или нажав на порт контейнера frontend в приложении Docker Desktop.

### 6.2 Инструкция для пользователя.

#### 1. Массовый импорт-экспорт данных.

При массовом импорте-экспорте используются сразу 3 файла формата .bson: exercises.bson, images.chunks.bson и images.files.bson. При массовом экспорте эти 3 файла будут скачаны на компьютер пользователя. При массовом экспорте система уведомит пользователя о том, какие файлы должны быть выбраны, и после выбора провалидирует названия, количество и типы выбранных файлов.

#### 2. Добавление нового упражнения в каталог.

Для того, чтобы добавить новую асану, необходимо нажать на кнопку “Добавить асану” на главной странице или в каталоге. После этого осуществится переход на страницу с формой, в которую необходимо внести фото упражнения, его описание, пользу и противопоказания, а также технику выполнения. Поля формы валидируются с выводом соответствующих сообщений об ошибках.

#### 3. Оставление отзыва на упражнение.

Для того, чтобы оставить отзыв на асану, необходимо пролистать страницу с информацией об упражнении в самый низ, до блока с отзывами и нажать кнопку “Оставить отзыв”, после чего отрисовывается страница с формой.

В форме обязательно нужно ввести имя или псевдоним пользователя, его возраст, выбрать из списка асану, на которую будет оставлен отзыв, описать свои

ощущения и поставить оценку. Если какое-то из полей не будет заполнено, приложение сообщит об этом пользователю.

#### **4. Кастомизированная статистика.**

Для отображения статистики по приложению необходимо на главной странице нажать кнопку “Кастомизированная статистика”, после чего осуществится переход на соответствующую страницу. На этой странице пользователю предоставится возможность выбрать промежуток дат, для которого будет строиться статистика и непосредственной тип статистики.

Важно отметить, что некоторые типы статистики строят информационную сводку на основании всего периода существования приложения, а некоторые строят графики для выбранного промежутка времени. Система проверит, какой тип статистики выбран, и, если не заполнено одно из необходимых полей (не выбран промежуток дат или асана для статистики), графический интерфейс выведет соответствующие сообщения с предупреждением.

## 7. ЛИТЕРАТУРА

1. Ссылка на GitHub. – [Электронный ресурс]. – URL: <https://github.com/moevm/nosql2h24-yoga-cat>.
2. Vue.js. – [Электронный ресурс]. – URL: <https://vuejs.org/> (дата обращения: 26.11.2024).
3. MongoDB. – [Электронный ресурс]. – URL: <https://www.mongodb.com/> (дата обращения: 26.11.2024).
4. Chart.js: Bar Chart. – [Электронный ресурс]. – URL: <https://www.chartjs.org/docs/latest/charts/bar.html> (дата обращения: 26.11.2024).
5. Vue DatePicker. – [Электронный ресурс]. – URL: <https://vue3datepicker.com/> (дата обращения: 26.11.2024).
6. VueUse: onClickOutside. – [Электронный ресурс]. – URL: <https://vueuse.org/core/onClickOutside/> (дата обращения: 26.11.2024).
7. Pinia. – [Электронный ресурс]. – URL: <https://pinia.vuejs.org/> (дата обращения: 26.11.2024).
8. NestJS. – [Электронный ресурс]. – URL: <https://docs.nestjs.com/> (дата обращения: 26.11.2024).
9. Lodash. – [Электронный ресурс]. – URL: <https://lodash.com/docs/4.17.15> (дата обращения: 26.11.2024).
10. ESLint. – [Электронный ресурс]. – URL: <https://eslint.org/> (дата обращения: 26.11.2024).
11. Prettier. – [Электронный ресурс]. – URL: <https://prettier.io/> (дата обращения: 26.11.2024).
12. Chaturanga Yoga. – [Электронный ресурс]. – URL: <https://chaturanga.yoga/asanas/> (дата обращения: 26.11.2024).