Домашнее задание: Создание простого REST API на Node.js с MongoDB

Цели задания:

- Практика работы с базой данных MongoDB без использования ORM/ODM.
- Разработка простого REST API для выполнения операций CRUD над ресурсом (например, книги) на чистом Node.js, используя встроенный модуль `http`.

Задачи:

1. **Установка MongoDB:**
    - Установите MongoDB локально на вашем компьютере или используйте MongoDB Atlas для создания облачной базы данных.
2. **Настройка проекта Node.js:**
    - Создайте новый проект Node.js, используя `npm init`.
    - Установите пакет `mongodb` для взаимодействия с вашей базой данных MongoDB.
3. **Разработка REST API:**
    - Используя встроенный модуль `http` в Node.js, создайте сервер, который будет слушать HTTP-запросы на определенном порту.
    - Определите и реализуйте следующие эндпоинты для работы с ресурсом "книги":
        - `POST /books` для добавления новой книги.
        - `GET /books` для получения списка всех книг.
        - `GET /books/:id` для получения конкретной книги по её идентификатору.
        - `PUT /books/:id` для обновления информации о книге.
        - `DELETE /books/:id` для удаления книги.
4. **Взаимодействие с MongoDB:**

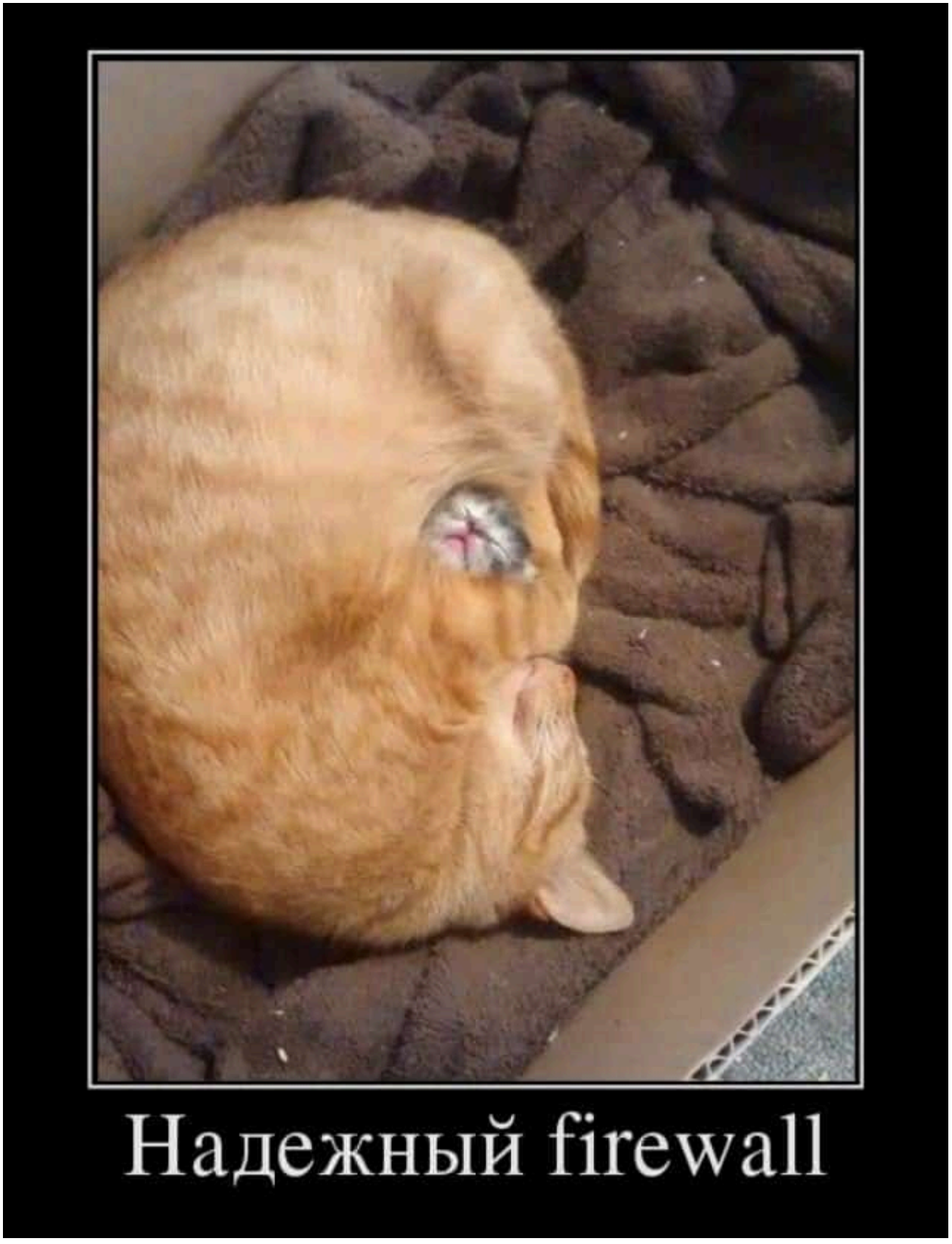
    - Подключитесь к вашей базе данных MongoDB из Node.js приложения.
    - Реализуйте логику для выполнения операций CRUD с использованием MongoDB в рамках обработчиков эндпоинтов.

5. **Тестирование:**

    - Протестируйте ваше API, используя Postman или любой другой клиент для отправки HTTP-запросов. Убедитесь, что все операции CRUD работают корректно.

Дополнительные задания (необязательные):

- Реализуйте дополнительную логику фильтрации для `GET /books`, позволяющую пользователю указывать различные параметры запроса (например, фильтрация по автору, жанру и т.д.).
- Добавьте валидацию входящих данных для эндпоинтов создания (`POST`) и обновления (`PUT`) книг.
- Исследуйте и примените в проекте возможности индексации в MongoDB для улучшения производительности запросов.


Надежный firewall