**Задача: Построение API для управления продуктами и их связями с магазинами**

**Описание**

Необходимо разработать backend сервис для управления продуктами, их ценами, остатками в магазинах и на складе. Для реализации нужно использовать Node.js с фреймворком NestJS и Prisma ORM для работы с базой данных PostgreSQL.

**Требования:**

1. **База данных**
   * Создать PostgreSQL базу данных с использованием Prisma ORM.
   * Описать модели для следующих сущностей:
     + **Product**: id, name, description, createdAt, updatedAt
     + **Category**: id, name, createdAt, updatedAt
     + **Store**: id, name, location, createdAt, updatedAt
     + **Price**: id, amount, currency, createdAt, updatedAt
     + **StoreStock**: id, quantity, storeId, productId, createdAt, updatedAt
     + **WarehouseStock**: id, quantity, productId, createdAt, updatedAt
     + Установить связи.
2. **API**
   * Создать REST API для следующих операций:
     + создать новый продукт.
     + получить информацию о продукте, включая его категории, цены и остатки.
     + обновить информацию о продукте.
     + удалить продукт.
     + создать новую категорию.
     + получить информацию о категории и связанные с ней продукты.
     + обновить информацию о категории.
     + удалить категорию.
     + создать новый магазин.
     + получить информацию о магазине, включая продукты, их цены и остатки.
     + установить цену для продукта в магазине.
     + установить остаток продукта в магазине.
     + установить остаток продукта на складе.
3. **Технические требования**
   * Использовать NestJS для построения контроллеров, сервисов и модулей.
   * Настроить Prisma для работы с PostgreSQL и миграций базы данных.
   * API должно поддерживать вышеуказанные CRUD операции.
   * Реализовать функцию по заполнению БД тестовыми данными.
   * Реализовать фильтрацию и пагинацию продуктов по категориям и магазинам.
   * Документирование API с помощью Swagger.
4. **Дополнительные требования (опционально)**
   * Реализовать тесты с использованием Jest для контроллеров и сервисов.
   * Реализовать обработку ошибок и валидацию данных с использованием class-validator.
5. **Проверка работы.**

Разместить код на github  
Отправить ссылку на задание: <https://t.me/samoryadovp>   
В случае возникновения вопросов по заданию: <https://t.me/r_mardanov_dev>