**Тестовое задание для Backend-разработчика микросервисов**

**Уровень 1:**

1. **Создание микросервиса:**
   * Разработайте микросервис с использованием FastAPI, который взаимодействует с базой данных PostgreSQL.
   * В приложении должны быть следующие эндпоинты:
     1. **Получение всех товаров** (GET /products)
     2. **Добавление нового товара** (POST /products)
     3. **Получение товара по его ID** (GET /products/{id})
     4. **Получение товаров по типу** (GET /products/type/{type\_id})
2. **Схема базы данных:**
   * **Таблица product\_type:**
     1. id (INTEGER, PRIMARY KEY)
     2. name (VARCHAR)
   * **Таблица product:**
     1. id (INTEGER, PRIMARY KEY)
     2. name (VARCHAR)
     3. product\_type\_id (INTEGER, FOREIGN KEY)
3. **Миграции с Alembic:**
   * Настройте Alembic для управления миграциями базы данных.
   * Создайте миграции для создания таблиц product\_type и product.
4. **Docker и Docker Compose:**
   * Упакуйте ваше приложение и PostgreSQL в контейнеры с использованием Docker и Docker Compose.
   * Создайте Dockerfile для вашего приложения.
   * Создайте файл docker-compose.yml для запуска приложения и базы данных PostgreSQL.
5. **Управление зависимостями:**
   * Используйте Poetry для управления зависимостями.

**Уровень 2:**

1. **Написание тестов:**
   * Напишите тесты для всех 4 эндпоинтов API, используя pytest и httpx.
2. **Документация:**
   * Добавьте описание для каждого эндпоинта в FastAPI документации, используя параметры summary и description.

**Уровень 3:**

1. **Линтер:**
   * Настройте линтер для вашего проекта (например, flake8, black или другой инструмент линтинга).
   * Убедитесь, что весь код проходит проверку линтером, и исправьте все замечания.
2. **GitHub Actions Workflow:**
   * Создайте GitHub Actions Workflow для автоматического тестирования и проверки качества кода.
   * Настройте Workflow для выполнения следующих шагов:
     1. Установка зависимостей (например, Poetry).
     2. Запуск линтера.
     3. Запуск тестов.

**Требования к выполнению задания:**

1. **Создайте структуру проекта:**
   * Инициализируйте проект с помощью Poetry.
   * Настройте FastAPI, Alembic, и необходимые библиотеки.
2. **Разработайте приложение:**
   * Реализуйте модели SQLAlchemy для таблиц product\_type и product.
   * Реализуйте указанные эндпоинты API.
3. **Настройте Docker и Docker Compose:**
   * Создайте Dockerfile для вашего приложения.
   * Создайте файл docker-compose.yml для управления контейнерами.
4. **Настройте Alembic:**
   * Создайте файл конфигурации Alembic и миграции для создания таблиц.
5. **Тестирование и документация:**
   * Напишите тесты для всех эндпоинтов.
   * Добавьте описание для эндпоинтов в FastAPI документации.
6. **Уровень 3 (дополнительные задачи):**
   * Настройте линтер и убедитесь, что код проходит проверку.
   * Создайте GitHub Actions Workflow для автоматического тестирования и проверки качества кода.

**Срок выполнения:** 7 дней.