**Тестовое задание**   
**Платформа:** ASP.NET Core, Razor Pages, микросервисная архитектура

**Цель задания**

Оценить навыки проектирования, выбора технологий, построения масштабируемой и безопасной системы, а также умение документировать архитектурные решения.

**Задача**

Спроектировать архитектуру системы управления лабораторными данными (LIMS)

Объекты системы:

Проба – объект исследования – (обладает свойствами: наименование, шифр и др. Например, проба воды, отобрана с колодца №1 01.01.2025 в 10:00, шифр «В122», для пробы требуется определить значения показателей: pH и плотность);

Результаты исследований **–** полученные значения экспериментов над пробами (например, масса, плотность, pH, цвет и т.д. Значения в виде числа либо текста);

Пользователи;

Роли;

Отчеты.

**Основные требования:**

1. Поддержка управления образцами (Samples), Результатами исследований (TestResults), пользователями (Users) и отчетами (Reports).
2. Интеграция с внешними инструментами (например, аналитическое оборудование через REST API).
3. Масштабируемость для работы с 10 000+ операций в день.
4. Реализация ролевой модели доступа (руководители, лаборанты, администраторы). Например, руководитель назначает исполнителя и просматривает результаты. Лаборант регистрирует пробы и вводит результаты, администратор может выполнять все действия.
5. Аудит действий пользователей.

**Часть 1: Проектирование архитектуры**

1. **Схема системы**
   * Нарисуйте высокоуровневую диаграмму компонентов системы с использованием микросервисов.
   * Укажите, какие сервисы будут выделены.
   * Опишите, как микросервисы будут взаимодействовать (API Gateway, брокер сообщений и т.д.).
2. **Выбор технологий**
   * Обоснуйте выбор фреймворков и инструментов
   * Предложите решение для контейнеризации
3. **Безопасность**
   * Опишите схему аутентификации/авторизации.
   * Как будет организована защита данных (HTTPS, шифрование чувствительных полей)?

**Часть 2: Реализация прототипа**

1. **Микросервис для управления образцами (Sample Tracking)**
   * Напишите на ASP.NET Core REST API с методами:
     + POST /samples (создание образца).
     + GET /samples/{id} (получение данных).
   * Используйте Dapper для работы с базой данных.
2. **Интерфейс на Razor Pages**
   * Создайте страницу для просмотра списка образцов с пагинацией.
   * Реализуйте форму добавления нового образца с валидацией.
3. **Интеграция с брокером сообщений**
   * При создании образца отправляйте событие в RabbitMQ/Kafka (например, «SampleCreated»).

**Часть 3: Документация**

1. Напишите краткое описание архитектуры (1-2 страницы).
2. Объясните, как система будет масштабироваться при росте нагрузки.
3. Укажите возможные узкие места и способы их устранения.

**Срок выполнения:** 7 дней.  
**Формат ответа:** Архив с исходным кодом, диаграммы ( Draw.io, PDF-документ).