**Тестовое задание**

**Цель задания:**

Необходимо написать простое **(многоуровневое, для претендующих на позицию middle developer)** CRUD приложение, для создания, удаления и редактирования заказов.

**Условия:**

1. Необходимо использовать представленную ниже схему БД.
2. Уровень представления должен быть выполнен на ASP.NET Core (MVC/API/Minimal API на выбор).
3. Ограничений по выбору ORM нет.
4. **Показать реалистичный пример использование IoC, для претендующих на позицию middle developer. Можно использовать стандартный механизм DI предоставляемый ASP.NET Core.**

**Описание приложения:**

Приложение должно содержать минимум 3 формы:

1. Главная страница

- кнопка для добавления заказов

- период в виде двух дат и мульти фильтры с выпадающим списком для фильтрации заказов

- таблица с заказами

- кнопка для применения фильтрации

1. Форма просмотра (попадаем после нажатия на строчку таблицы заказов)

- информация о заказе

- кнопка для редактирования

- кнопка для удаления

1. Форма создания/редактирования (можно раздельно)

- кнопка для сохранения

- все редактируемые поля

- кнопка для добавления новых строк в заказ

- кнопка для удаления строки из заказа

**Структура БД:**

Существуют 3 таблицы:

1. Order (заказ)

- Id (int)

- Number (nvarchar(max)) \*редактируется \*используется для фильтрации

- Date (datetime2(7)) \*редактируется \*используется для фильтрации

- ProviderId (int) \*редактируется \*используется для фильтрации

1. OrderItem (элемент заказа)

- Id (int)

- OrderId (int)

- Name (nvarchar(max)) \*редактируется \*используется для фильтрации

- Quantity decimal(18, 3) \*редактируется

- Unit (nvarchar(max)) \*редактируется \*используется для фильтрации

1. Provider (поставщик, заполнена изначально, нигде не редактируется)

- Id (int)

- Name (nvarchar(max)) \*используется для фильтрации

**Дополнительно:**

1. Поставщик выбирается для всего заказа путем выбора из выпадающего списка.
2. Фильтры представлены в виде (<select multiple>...</select>) которые содержат уникальные значения из БД (distinct).
3. Фильтр по дате представлен в виде двух input с датами, обозначающими период, за который должны отображаться заказы. Значение по умолчанию: месяц назад от сегодняшней даты – сегодняшняя дата.
4. Фильтрация должна происходить на серверной стороне приложения.
5. Пользовательский интерфейс не является ключевым аспектом данной задачи, поэтому его внешним видом можно условно пренебречь.
6. Отметки «\*редактируется» и «\*используется для фильтрации» означают, что поля можно редактировать и использовать для фильтрации на уровне представления, непосредственно в пользовательском интерфейсе.
7. Order.Number «в связке» с Order.ProviderId должен быть уникален, т.е. не должно существовать 2х заказов от одного поставщика с одинаковыми номерами. При попытке сохранения, программа должна показывать уведомление о невозможности сохранения (ограничение предметной области).
8. OrderItem.Name не может быть равен Order.Number (ограничение предметной области).