## Контакты

Ссылки на выполненные тестовые задания, а также возникшие вопросы присылать:

Telegram:

**@F1\_GTR** Евгений

## Общие требования к выполнению заданий

Язык - PHP. Используем Laravel ~~+ пакет администрирования SleepingOwl~~ (обязательное условие!).

База данных - MySQL или MariaDB.

Код должен быть прокомментирован.

Необходимо развернуть выполненные задания для общего доступа (если запаролено - выдать доступ на вход).

Исходный код необходимо разместить в GIT (любой репозиторий на выбор) и выдать доступ.

## Также нужен доступ к просмотру живой базы данных (phpMyAdmin или подключение через DBeaver). Если с последним есть сложности - быть готовым сбросить дампы базы до и во время проверки.

Возможно будут заданы вопросы по обоснованию выбранного решения задач (архитектура, структура, фреймворк, качество кода и т.д.) или требования по доработке решения.

## 

## Задание 1.

Разработать компонент “Заказы”. В админке пользователю доступна страница со списком заказов, формы создания и редактирования заказов, страница с отчетом (количество и стоимость выполненных заказов за определённую дату). Необходимо меню для перехода между разделами. Интерфейс должен быть интуитивно-понятен.

На фронт-части пользователь должен увидеть список товаров и их цены, аккуратными карточками (используем Bootstrap4).

Схема базы данных следующая:

**Таблица users**

Поля:

* id - unsigned big integer, AI, pk
* name - varchar(255)

**Таблица products**

Поля:

* id - unsigned big integer, AI, pk
* name - varchar(255)
* price - float
* stock - integer

**Таблица orders**

Поля:

* id - unsigned big integer, AI, pk
* customer - varchar(255) (имя клиента)
* phone - varchar(255) (телефон клиента)
* created\_at - timestamp
* completed at - timestamp
* user\_id - unsigned big integer, fk
* type - varchar(255) (“online”,”offline”)
* status - varchar(255) (“active”, “completed”, “canceled”)

**Таблица order\_items**

Поля:

* id - unsigned big integer, AI, pk
* order\_id - unsigned big integer, fk
* product\_id - unsigned big integer, fk
* count - integer
* discount - float, default 0
* cost - float (стоимость товаров в позиции с учётом скидки)

СХЕМУ БАЗЫ МЕНЯТЬ НЕЛЬЗЯ, НЕОБХОДИМО РАБОТАТЬ С ТЕМ, ЧТО ЕСТЬ

В заказе мы можем задавать\редактировать тип, статус, менеджера (пользователя), покупателя, контактный телефон и товары. Для выбранных товаров мы можем задать количество и скидку.

Количество больше чем stock указывать нельзя.

При создании заказа или перевода ИЗ status=canceled остатки товаров вычитаются (необходима серия проверок и сообщения с ошибками, если действие невозможно).

При переводе заказа В status=canceled товары возвращаются на склад.

Товары, цены, остатки и менеджеров (пользователей) необходимо наполнить самостоятельно.

## Задание 2.

Необходимо разработать раздел “Логистика”. В админке доступен список складов, список товаров и управление наличием. Товары можно добавить на склад (прибытие), товары можно списать со склада (продажа), товары можно переносить между складами (движение). Должна быть возможность получить остаток для любого товара на складе, а также историю движений для склада (общая и для конкретного товара). Необходимо меню для перехода между разделами. Интерфейс должен быть интуитивно-понятен. Непосредственно товары и склады необходимо наполнить самостоятельно.

На фронт-части пользователю отображаются карточки со списком складов и два счетчика: сколько товаров и сколько заказов (используем Bootstrap4).

Схему БД необходимо спроектировать самостоятельно