**Тестовое задание:**

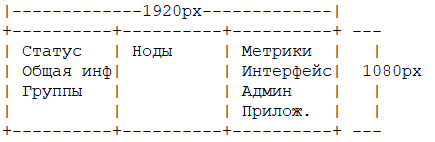
**Разработать представление отображающее состояние инфраструктуры предприятия**.

1. **SQL-запросы:**
   1. **Напишите запрос получающий все группы**

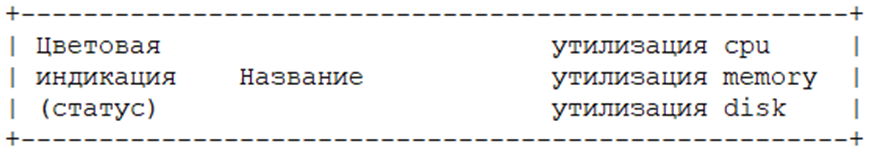
* Отредактируйте файл ./node/sql/groups.sql
* Группы должны содержать все входящие в них ноды.
* Ноды должны содержать информацию о интерфейсах.
* Ноды должны содержать информацию об приложениях.
* Ноды должны содержать информацию об администраторах.
* Ноды и интерфейсы должны содержать информацию о статусе (цвет и описание).
  1. **Напишите запрос, получающий метрики**
* Отредактируйте файл ./node/sql/metrics.sql
* Метрики должны содержать ноды.

1. **Представление vue:**
   1. **Общие требования:**

* Представление должно корректно работать как в Google Chrome, так и в Internet Explorer 11.
* Стили должны быть написаны на 'чистом' css + препроцессор less.
* Использовать плагины axios, vuex, router, vue-chartjs.
* Представление должно запрашивать новые данные с backend 1 раз в минуту.
* Стили оформления блоков придумать самостоятельно, делать адаптивную верстку не нужно.
  1. **Требования к оформлению элементов:**
* Схематичное расположение элементов, выбор дизайна представления остаётся на усмотрение разработчика.



* Представление должно быть разбито на 3 равных столбца
* Каждый столбец должен содержать блоки с данными.
* Блоки могут содержать:
* элементы (ноды)
* таблицы
* графики
* строковые элементы
  1. **Описание блоков:**
     1. Блок "Статус":
     + Необходимо реализовать статус сервиса - рассчитывается по наихудшему из статусов нод.
     + Должен отображать название и цветовой окрас текущего статуса отображаемых нод (с учетом установленного фильтра групп).
     1. Блок "Общая информация":
     + Необходимо реализовать статус сервиса - рассчитывается по наихудшему из статусов нод.
     1. Блок "Группы":
     + При клике на группу в столбце 2 и 3 должны отображаться данные только из этой группы.
     1. Блок "Ноды":
     + Должен выводить список элементов "нода"

****

* + 1. Элемент "Нодa":
    - Нода должна отображать название, последние данные метрик (метрика со значением> 85> 95 окрашивается желтым, метрика> 95 окрашивается красных), цветовую индикацию статуса ноды.
    - При клике на ноде в столбце 3 должны отображаться только метрики и данные этой ноды, а в столбце 1 группы в которые входит эта нода.
    1. Блок "Метрики":
    - Метрики должны быть реализованы в виде графика chartjs (https://www.chartjs.org/)
    1. Блок "Интерфейс":
    - Должен отображать название и статус интерфейса ноды.
    1. Блок "Администратор":
    - Должен отображаться данные администратора ноды.
    1. Блок "Приложения":
    - Должен отображать данные приложений, выполняющихся на ноде.

1. **Установка и запуск:** 
   1. **Backend:**
      1. Установить NodeJS (https://nodejs.org/ru/)
      2. Перейти в папку "node" и выполнить установку зависимостей: "npm install"
      3. Запустите backend командой: "npm run start"
      4. Проверьте работу в браузере по адресу: <http://127.0.0.1:23456/api>
   2. **Frontend:**
      1. Перейдите в каталог проекта "vue".
      2. Выполните установку зависимостей: "npm install"
      3. Запустите проект: "npm run serve"
2. **Дополнительное программное обеспечение:**

Для просмотра таблиц и выполнения тестовых запросов можно воспользоваться бесплатной программой "DB Browser for SQLite” (https://sqlitebrowser.org/)