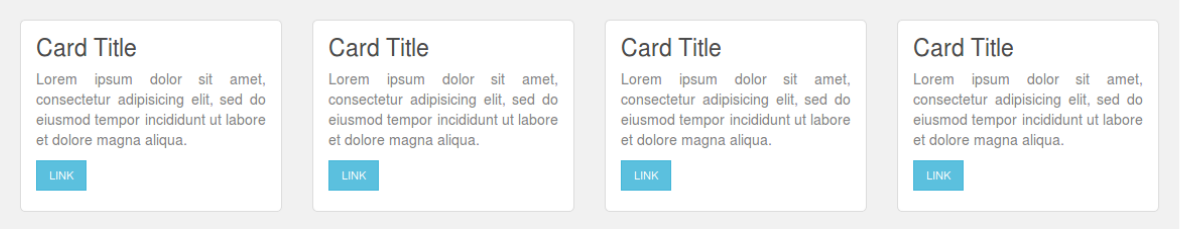
Тестовое задание frontend программист (Vue.js)



Примерный дизайн клиентского приложения.

План работы:

1. отрисовать в Figma макет создаваемого приложения
2. создать сервер приложения на Node.js + socket.io
3. создать клиент Vue.js максимально (без фанатизма) приближенный к макету

Используемые технологии, компоненты и библиотеки

- Node.js
- Socket.io
- Vue.js
- Bootstrap

Сервер

Сервер содержит замоканную базу данных для карточек:

```
let events = [
  {
    id: 1,
    title: "Card 1",
    event_date: "2022-09-01",
    guests_count: 14,
    about:
      "Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit."
  },
  {
    id: 2,
    title: "Card 2",
    event_date: "2022-09-02",
    guests_count: 12,
    about:
      "Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit."
  },
  {
    id: 3,
    title: "Card 3",
    event_date: "2022-09-04",
    guests_count: 4,
    about:
      "Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit."
  },
  {
    id: 4,
    title: "Card 4",
    event_date: "2022-09-06",
    guests_count: 1,
    about:
      "Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit."
  },
];
```

Сервер поддерживает подключение по socket и ассинхронное оповещение клиентов об изменениях в БД.
Сервер позволяет добавлять и удалять записи в БД. После перезапуска сервера, состояние БД сбрасывается.
Сервер оповещает всех подключенных клиентов об изменении карточки.

Клиент

Клиент должен быть написан на Vue + bootstrap.
Первоначальная загрузка данных для отображения карточек через отложенный запрос к серверу.
Сортировка карточек по дате по убыванию.
Кнопка добавления новой карточки “Добавить” (можно в виде иконки), расположить сверху справа над сеткой карточек

- открывает модальное окно с формой ввода всех необходимых данных по объекту кроме id
- по нажатию на кнопку “Сохранить” (можно в виде иконки) в модалке, отправляет данные на сервер
- при успешном сохранении, добавляет карточку в сетку карточек, учитывая сортировку

На каждой карточке добавить кнопку “Редактировать” (можно в виде иконки):

- открывает модалку с формой редактирования всех данных карточки
- по нажатию на кнопку “Сохранить” (можно в виде иконки) в модалке, отправляет данные на сервер
- при успешном сохранении, добавляет карточку в сетку карточек, учитывая сортировку

На каждой карточке добавить кнопку “Удалить” (можно в виде иконки):

- удаляет данную карточку из сетки, отправляя информацию об удалении на сервер

Поля карточки и валидация:

- title - “Название мероприятия” тип *текст*, обязательно для заполнения
- event_date - “Дата мероприятия”, тип *дата*, обязательно для заполнения
- guests_count - “Количество гостей”, тип *целое число*, обязательно для заполнения
- about - “Описание”, тип *текст*, не обязательно для заполнения

Клиент принимает события из socket об измененнии карточки - подсвечивем красным цветом карточку, которая изменилась. Не забыть о сортировке, она может изменится после обновления информации в карточке.
Клиент принимает события об удалении карточки - просто удаляя карточку из списка.

Дополнительно, но не обязательно

1. Рядом с кнопкой “Добавить”, разместить dropdown кнопку с списком полей карточки и возможностью настроить отображение полей карточки: event_data, guests_count, about. Сохранять состояние в cookie.

2. Добавить простую авторизацию запросов к серверу по фиксированному токену

Результат:

Оформить в виде репозитории на github.
Написать пошаговый README для запуска приложения.

Прислать 2 ссылки: figma + github