Компания занимается разработкой софта для авиапредприятий. Тестовое задание тематическое - необходимо сделать простое одно-двух страничное FullStack приложение для отображения общей и детальной информации по налету командира воздушного судна.

Исходные данные:

JSON файл с данными о налете сотрудника в составе экипажа. Это массив с информацией о выполненных рейсах. Информация по рейсу представлена объектом:

```
{
  "dateFlight": "2017-06-21T19:00:00.000Z", // Дата рейса
  "flight": "AB-3377", // Номер рейса
  "plnType": "B-757-200", // Тип воздушного судна
  "pln": "XXXAK", // Бортовой номер воздушного судна
  "timeFlight": 26100, // Время налета (с момента взлета до посадки)
  "timeBlock": 27000, // Полетное время (время налета + руление и работа двигателя на земле)
  "timeNight": 16200, // Ночное летное время (является частью налета)
  "timeBiologicalNight": 28200, // Биологическая ночь
  "timeWork": 37800, // Рабочее время (полетное время + предполетная и послеполетная подготовка)
  "type": 0, // Тип записи (0 - фактический налет, 1 - плановый)
  "takeoff": {
    "name": "Томск(Богашево)", // Аэропорт вылета
    "lat": 56.38333333, // Широта
    "long": 85.2 // Долгота
  },
  "landing": {
    "name": "Нячанг(Камрань Интл)", // Аэропорт посадки
    "lat": 11.99805555, // Широта
    "long": 109.21944444 // Долгота
  }
},
```

Задание:

- Создать пустое приложение на основе Create React App.
- Получить JSON файл с данными при загрузке страницы.
- Реализовать исходный экран с общей статистикой по годам/месяцам. Статистика должна учитывать общие за период налет и рабочее время по факту и по плану. Под фактическими понимаются данные по уже выполненным рейсам, под плановыми – будущие рейсы.
- Для планового времени предусмотреть отдельное отображение. Учесть, что возможен период, где есть и плановые и фактические показатели.
- При нажатии на год или месяц должна открываться страница с детальной информацией за период. Информация должна содержать общую сводную информацию и список рейсов со всей информацией из исходных данных.
- Адаптация под мобильные устройства.

Дополнительно (будет плюсом):

- Можно сделать backend, отдающий по запросу требуемый JSON файл.
- Реализовать поиск (по номеру рейса и/или дате).
- Реализовать графическое представление (в виде графика) с использованием любых инструментов.

Не функциональные требования:

- Разрешено использовать React/Redux, TypeScript.
- Разрешено использовать ECMAScript 2016-2018.
- Желательно, чтобы информация была максимально компактна, наглядна и удобна для использования. Дизайн на усмотрение соискателя.
- В случае реализации backend использовать NodeJS.