Техническое задание для WEB-приложения Журнала

**Общие требования**

* **IDE:** Visual Studio Code
* **FrontEnd:** React
* **BackEnd:** Node.js
* **СУБД:** Oracle (бесплатная версия – Express Edition)

**Функциональные требования**

1. **Отображение данных:**
   * Отображение данных в табличной форме (например, погода в разных городах: температура, давление, влажность).
   * Данные поступают от внешнего источника в БД раз в 5 минут.
   * Обновление данных в веб-приложении каждые 5 минут.
2. **Основная экранная форма:**
   * Таблица по типу MS Excel с возможностью выгрузки в MS Excel.
   * Возможность отображения данных с историей и без истории.
   * Возможность ручного изменения данных с сохранением изменений в базе данных.
3. **Управление регионами:**
   * Разделение данных по регионам.
   * Для каждого региона отдельная таблица по типу Excel.
   * Механизм навигации по регионам и между таблицами.
4. **Печать данных:**
   * Возможность печати данных из таблиц.

**Архитектура**

* **FrontEnd:**
  + React-компоненты для отображения таблиц и управления данными.
  + Axios для выполнения запросов к API.
  + react-table для отображения таблиц.
  + react-csv для экспорта данных в Excel.
  + react-router-dom для навигации между страницами.
* **BackEnd:**
  + Node.js с Express для создания API.
  + oracledb для взаимодействия с Oracle Database.
  + cron или node-cron для периодического обновления данных.

**Сценарий работы**

1. **Первичная загрузка данных:**
   * При запуске приложения происходит первичная загрузка данных из базы данных и отображение их в таблицах.
2. **Обновление данных:**
   * Каждые 5 минут происходит запрос к базе данных для обновления данных в таблицах.
   * Данные обновляются в таблицах в реальном времени без перезагрузки страницы.
3. **Ручное изменение данных:**
   * Пользователь может изменить данные в таблице.
   * После изменения данных происходит обновление соответствующей записи в базе данных.
4. **Отображение истории:**
   * Пользователь может переключаться между режимами отображения данных с историей и без истории.
5. **Экспорт данных:**
   * Пользователь может экспортировать данные из таблицы в файл формата Excel.
6. **Печать данных:**
   * Пользователь может распечатать данные из таблицы.

**Структура проекта**

* **FrontEnd:**
  + src/
    - components/
      * TableComponent.js
      * RegionNavigation.js
      * PrintComponent.js
    - services/
      * apiService.js
    - pages/
      * HomePage.js
      * RegionPage.js
    - App.js
    - index.js
* **BackEnd:**
  + src/
    - routes/
      * dataRoutes.js
    - controllers/
      * dataController.js
    - services/
      * dataService.js
    - models/
      * dataModel.js
    - cron/
      * updateDataCron.js
    - app.js
    - index.js