**Сопроводительная записка**

В каталоге “backend” размещены исходные коды серверной части.

Файловая структура серверной части имеет следующую схему:

* Каталог controllers – содержит реализации классов контроллеров, методы которых отвечают за обработку запросов HTTP;
* Файл error/ApiError.cjs – экспортирует класс, наследующий класс Error (ошибка). Содержит статические методы для удобного объявления типовых ошибок взаимодействия с сервером;
* Каталог helpers – предоставляет экспортируемые модули, отвечающие за взаимодействие с базой данных, подготовку и обработку данных, получаемых или извлекаемых из неё;
* Каталог middleware – хранит функции промежуточной обработки входящих запросов HTTP;
* Файл models/models.cjs – внутри описывается модель данных при помощи средств sequelize и экспортируются сущности для доступа к таблицам базы данных;
* Каталог routes – содержит файлы, в которых описываются маршруты к конечным точкам внешнего интерфейса сервера. Для каждой конечной точки применяются специальные методы обработки данных входящих и исходящих запросов;
* Каталог static – предназначается для хранения статических файлов скомпилированной клиентской части;
* Файл utils/initFunc.cjs – файл содержит экспортируемую функцию для инициализации базы данных. При её вызове, старые значения будут стёрты. Вызывается при первом старте сервера и при сбросе настроек;
* Файл .env – служит для хранения константных значений, таких, как сведения для доступа к СУБД, доменное имя сервера, номер порта;
* Файл db.cjs – внутри объявляется экспортируемый метод создания класса sequelize для подключения к СУБД и создания модели данных;
* Файл index.cjs – основной файл серверного приложения. Внутри него подключаются внешние и сторонние модули; определяется экземпляр класса express для обеспечения серверного функционала; задаётся корневой путь к конечным точкам внешнего интерфейса; выполняется запуск сервера;
* Файл package-lock.json – автоматически генерируемый файл, отражающий изменения от любых операций, влияющих на дерево модулей node\_modules или package.json;
* Файл package.json – документ, описывающий проект, его требования и дополнительные настройки.

В каталоге “frontend” содержатся исходные коды клиентской части.

Клиентская часть включает следующие элементы:

* каталог public – содержит документ index.html, внутрь которого React встроит разработанные компоненты, манифест manifest.json, описывающий параметры для PWA-приложения, и иконки приложения;
* каталог src – здесь содержатся все основные файлы разработки, классы, компоненты, утилиты, стили и прочее:

1. каталог components – содержит компоненты, из которых строится графический интерфейс веб-приложения;
2. каталог modals – содержит компоненты, представляющие собой модальные окна, которые всплывают поверх основных страниц интерфейса;
3. каталог pages – в нём располагаются компоненты страниц интерфейса;
4. каталог css – хранит файлы CSS стилей компонентов страниц;
5. каталог hooks – содержит так называемые хуки, представляющие собой функции, обеспечивающие компоненты состояниями и методами жизненного цикла;
6. каталог png – хранит изображения формата png;
7. каталог svg – хранит векторные изображения формата svg;
8. файл utils/constants.js – содержит константные значения путей в клиентском интерфейсе;
9. каталог http – хранит файлы, содержащие функции взаимодействия с сервером;
10. файл App.css – файл CSS стилей, общих для всего веб-приложения;
11. файл App.jsx – содержит первичный компонент приложения, именуемый App, который является функцией, возвращающей пользовательские элементы интерфейса;
12. файл routes.js – содержит экспортируемые массивы объектов, представляющих собой описание маршрутов для различных уровней доступа;
13. файл index.js – начальная точка приложения React, где инициируется отрисовка DOM-элементов интерфейса.

* файл .env – позволяет хранить константные значения и обращаться к ним из любого места проекта;
* файл package-lock.json – автоматически генерируемый файл, отражающий изменения от любых операций, влияющих на дерево модулей node\_modules или package.json;
* файл package.json – как в случае с серверной частью, этот документ описывает проект, его требования и прочие настройки.