**Документация**

**Стек**

Frontend – HTML, Javascript, CSS, SCSS.

SCSS – препроцессор, расширяющий возможности обычного CSS, например, минифицирует выходной CSS в одну строку, чтобы файл был меньше по размеру.

Backend – Node.js, Express.js, body-parser, cookie-parser, dotenv, fs, jsonwebtoken, multiparty, mysql2, path.

body-parser – парсит тело запросов клиента (frontend’а).

cookie-parser – парсит кукисы в заголовках запроса клиента, в которых хранится access-token.

dotenv – парсит файлы .env, которые хранят переменные среды. В нашем случае – секрет для генерации токена.

fs – работает с файловой системой.

jsonwebtoken – работает с токенами: генерирует, проверяет, продлевает и т.д.

multiparty – парсит тело запроса в формате FormData.

mysql2 – соединяется с БД и в дальнейшем – работает.

path – работает с путями, помогает нормально сериализовать пути по типу “C://fileSystem/…/assets/file.txt”

**Структура**

API – (сервер) хранит конечные точки к которым обращается клиент.

assets – (сервер) хранит статичные файлы для клиента: картинки, пдфки и т.п.

pages – (сервер) хранит все возможные страницы, которые сервер отдаст клиенту при запросе.

plugins – (сервер) хранит плагины, в нашем случае – роутер, позволяющий описать енд-поинты для API.

scripts – (фронт) хранит скрипты, которые использует только клиент (для каждой страницы свой скрипт).

utils – (фронт) хранит скрипты, которые используются на каждой странице.

services -- (сервер) хранит вспомогательные сервисы для API и utils(утилит).

styles – (фронт) хранит стилизацию для страниц.

css – готовый css, который подключается к страницам.

scss – css на языке препроцессора. (преобразуется в обычный css).

utils – (сервер) хранит вспомогательные утилиты для сервера: коннект к бд, парсер FormData, работу с токеном.

**Разбор кода**

package.json – содержит информацию о настройках приложения и всех используемых библиотеках.

Комментарии в самом коде, если проблемы с пониманием – созвонимся.

Алгоритм работы самой сложной страницы (News):

1. Идём по адресу /news

2. Сервер видит что к такому енд-поинту API было обращение 🡪 отдёт страницу NewsPage.html клиенту.

3. Страница подключает стили.

4. Страница подключает два основных скрипта: Authentication.js 🡪 News.js.

5. News.js формирует запрос к серверу на получение списка всех карт 🡪 сервер ловит обращение к енд-поинту 🡪 формирует запрос к базе 🡪 возвращает список новостей; News.js формирует запрос к серверу на проверку авторизован ли пользователь 🡪 сервер ловит обращение к енд-поинту 🡪 проверяет авторизован ли пользователь: если да – отдаёт true, нет – отдаёт false.

6. News.js в зависимости от того авторизован пользователь или нет – рендерит кнопку добавления новой новости + кнопку у карточки для возможности редактирования

7. News.js формирует карточку и рендерит на страницу.