

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет прикладной информатики

Образовательная программа Интеллектуальные системы в гуманитарной сфере

Поток Фронтэнд-разработка 1.1

О Т Ч Е Т

По лабораторной работе №2

Обучающийся: Алексеев Николай Сергеевич, К3344

Преподаватель: Добряков Давид Ильич

Санкт-Петербург,

2025

1 Задача

2 Введение

Проект "Food Source" представляет собой веб-сервис для нахождения, размещения и комментирования рецептов. Пользователи имеют возможность просматривать, оценивать и комментировать рецепты, также их можно сохранять и подписываться на авторов. На сайте присутствует система регистрации с запоминанием пользователя для повторной авторизации.

3 Цели проекта

1. Разработать удобный и интуитивно понятный пользовательский интерфейс для взаимодействия с системой.
2. Реализовать функционал регистрации, авторизации и выхода из личного кабинета.
3. Создать систему для поиска рецептов, комбинировать поиск рецептов по подстроке и фильтрам (сложность, входящие ингредиенты)
4. Предоставить возможность открывать каждый рецепт, оценивать, комментировать, сохранять.

4 Структура проекта

Index.html – главная страница (список рецептов, поиск, фильтры)

Login.html, register.html – страницы авторизации (простая форма)

Profile.html – страница профиля пользователя (сохраненные рецепты, его рецепты)

Recipe.html – страница рецепта (подробное описание, возможность поставить лайк, сохранить, комментировать)

Code.js – основной файл js (вывод рецептов, логика фильтров и поиска)

Author.js – файл класс автора рецептов.

Data.js – файл для создания тестовых рецептов.

LoginPage.html, registerPage.html – файлы с логикой страниц авторизации

Profile.js – файл с логикой профиля (выход из аккаунта, вывод сохраненных рецептов)

Recipe.js – файл класс рецепта (описание, название, картинка и т.д.)

recipePage.js – файл с логикой страницы рецепта.

User.js – файл класс пользователя (имя, сохраненные рецепты, подписки и т.д.)

API файлы:

Seed.js – файл для заполнения локального файла db.json рецептами из mealdb api.

Sever.js – файл для поднятие локального сервера на основе файла db.json.

Db.json – локальная база данных для рецептов и пользователей.

Api.js – файл для работы с локальным сервером с помощью запросов.

5 Основные функции api

Файл api.js:

- apiFetch – обертка запроса, добавляет токен авторизации, ловит ошибки,
- loginByName – формирует почту и пароль из username, возвращает объект пользователя,
- registerByName – создает пользователя с дефолтными полями, возвращает пользователя,
- getRecipes – получает список рецептов,
- getRecipe – получает рецепт по id,
- patchMe – частично обновляет пользователя,
- getMe – берет актуальные данные пользователя с сервера, возвращает пользователя,
- getUser – получает пользователя по id,
- toggleSubscription – подписывает/отписывает от автора,
- patchRecipe – частично обновляет рецепт.

Файл server.js:

- импортирует json-server,
- создает express-сервер,
- подключает роутер для работы с db.json как с базой данных,
- включает парсинг запросов,
- подключает авторизацию,
- запускает api на порту 3001.

Файл seed.js:

- pickIngredients – собирает список ингредиентов,
- mapType - переводит строковую категорию блюда в числовой тип,
- mapDifficulty – определяет сложность рецепта по количеству ингредиентов,

- `fetchMealsByLetter` – запрашивает блюда из `themealdb` api,
- `main` – загружает рецепты, нормализует данные, удаляет дубликаты, создает пользователя `TheMealDB`, записывает рецепты в `db.json`.

Выводы

В результате выполнения лабораторной работы я научился работать с моковым api. Поднимать локальный сервер с базой данных на основе json файла. Закрепил навыки работы с запросами.