САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Попов Никита

К3344

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Тема: Взаимодействие с внешним API (JSON-server)

Вариант: Приложение для бронирования столиков в ресторанах.

Цель работы

● Научиться работать с внешним API с помощью fetch/axios.

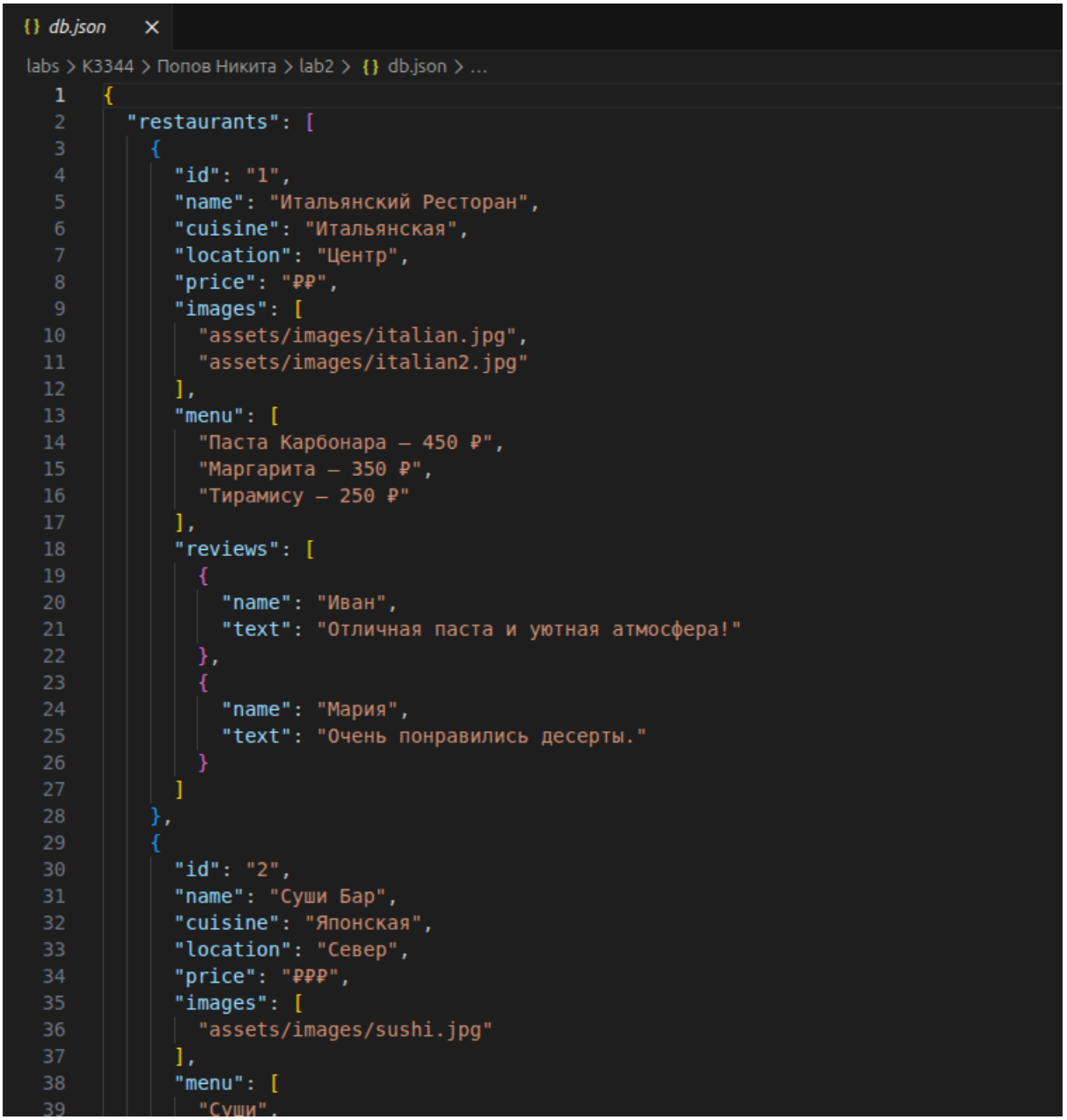
● Реализовать моковое REST-API через JSON-server и json-server-auth для авторизации.

● Перевести функционал ЛР1 на получение данных из API.

Ход работы:

Взаимодействие с внешним API:

Создан файл db.json

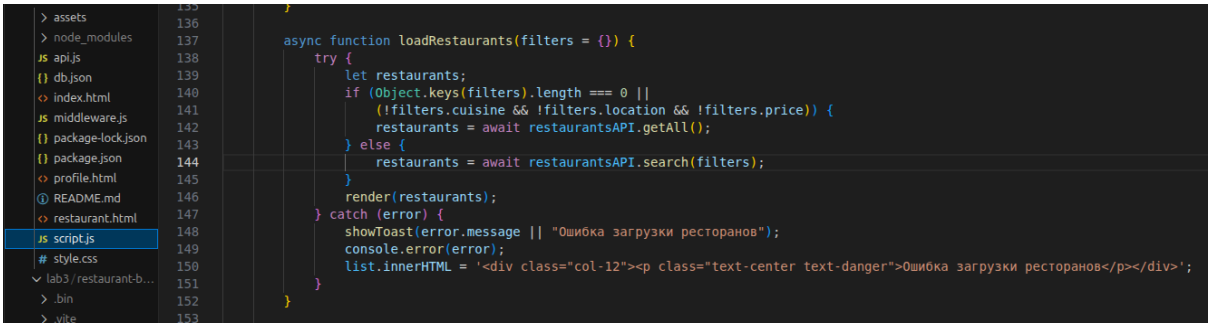


db.json

labs > K3344 > Попов Никита > lab2 > {} db.json > ...

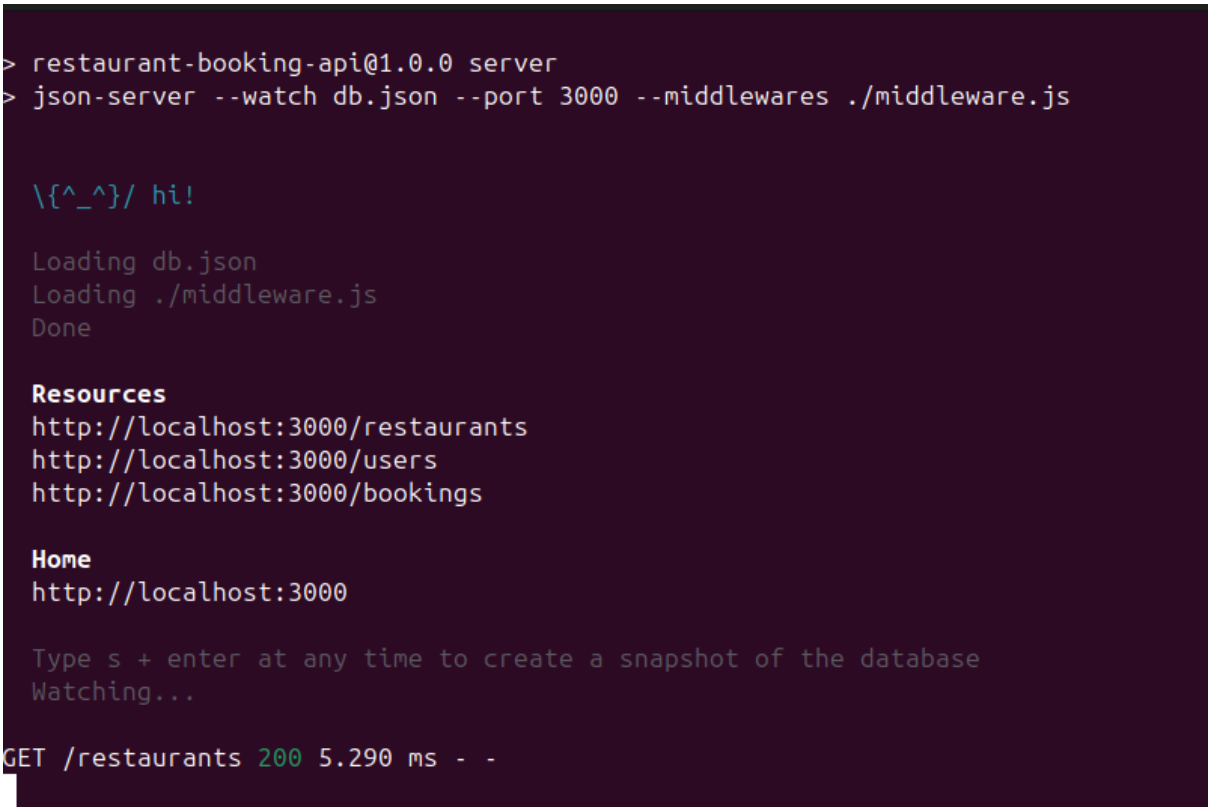
```
{
  "restaurants": [
    {
      "id": "1",
      "name": "Итальянский Ресторан",
      "cuisine": "Итальянская",
      "location": "Центр",
      "price": "₽₽",
      "images": [
        "assets/images/italian.jpg",
        "assets/images/italian2.jpg"
      ],
      "menu": [
        "Паста Карбонара – 450 ₽",
        "Маргарита – 350 ₽",
        "Тирамису – 250 ₽"
      ],
      "reviews": [
        {
          "name": "Иван",
          "text": "Отличная паста и уютная атмосфера!"
        },
        {
          "name": "Мария",
          "text": "Очень понравились десерты."
        }
      ]
    },
    {
      "id": "2",
      "name": "Суши Бар",
      "cuisine": "Японская",
      "location": "Север",
      "price": "₽₽₽",
      "images": [
        "assets/images/sushi.jpg"
      ],
      "menu": [
        "Суши",
```

В script.js реализована вся логика работы с сервером json



```
async function loadRestaurants(filters = {}) {
    try {
        let restaurants;
        if (Object.keys(filters).length === 0 ||
            (!filters.cuisine && !filters.location && !filters.price)) {
            restaurants = await restaurantsAPI.getAll();
        } else {
            restaurants = await restaurantsAPI.search(filters);
        }
        render(restaurants);
    } catch (error) {
        showToast(error.message || "Ошибка загрузки ресторанов");
        console.error(error);
        list.innerHTML = '<div class="col-12"><p class="text-center text-danger">Ошибка загрузки ресторанов</p></div>';
    }
}
```

Сам сервер поднимается через npm с помощью команды npm run server



Заключение

В рамках данной лабораторной работы была реализована интеграция ранее разработанного интерфейса с внешним API. Для имитации серверной части использовался JSON-сервер с настроенной авторизацией. Взаимодействие с API осуществлялось с помощью средств работы с HTTP-запросами (fetch/axios/xhr). В результате была отработана логика получения и отправки данных, а также закреплены навыки работы с клиент-серверным взаимодействием и имитацией реального API.