САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Кудина Вероника

К3343

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо будет выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать моменты, в которых необходим JS (например, открытие модальных окон).

Доступные варианты:

Приложение для бронирования столиков в ресторанах

1. Вход
2. Регистрация
3. Личный кабинет пользователя
4. Поиск ресторанов с фильтрацией по кухне, расположению и цене
5. Страница с информацией о ресторане (меню, фото, отзывы)
6. История бронирований пользователя

Ход работы

1. Введение

Проект "РестоБронь" представляет собой веб-сервис для бронирования столиков в ресторанах. Он предоставляет пользователям удобный интерфейс для поиска ресторанов, ознакомления с их меню, чтения отзывов и последующего бронирования. Сервис ориентирован на улучшение взаимодействия между клиентами и ресторанами, предлагая современные функции фильтрации, учета личных данных и управления бронированиями.

2. Цели проекта

1. Разработать удобный и интуитивно понятный пользовательский интерфейс для взаимодействия с системой.

2. Реализовать функционал регистрации, авторизации и управления личным кабинетом.
3. Создать систему для поиска ресторанов, которая позволяет пользователям находить подходящие заведения по фильтрам (кухня, район, ценовая категория и т.д.).
4. Обеспечить возможность просмотра детальной информации о ресторанах (меню, рейтинг, отзывы).
5. Внедрить функционал бронирования столиков с учетом даты, времени и количества гостей.

3. Структура проекта

Проект состоит из следующих файлов:

3.1. HTML-файлы

- index.html – главная страница.
- Приветственный экран с кнопкой "Найти ресторан".
- Раздел с популярными ресторанами.
- Информация о компании и контактные данные.

restaurants.html – страница поиска ресторанов.

- Фильтры для поиска ресторанов по типу кухни, району, ценовой категории и названию.
- Список ресторанов с краткой информацией: название, рейтинг, отзывы, описание.

restaurant.html – страница конкретного ресторана.

Детальная информация о ресторане: меню, отзывы, описание, рейтинг.

Возможность бронирования столика.

register.html – страница регистрации.

Форма для регистрации нового пользователя с полями для имени, email, телефона и пароля.

login.html – страница входа в систему.

Форма для авторизации пользователя.

profile.html – личный кабинет пользователя.

- Отображение личных данных пользователя.
- Возможность редактирования данных.
- История бронирований с возможностью их отмены.

3.2. CSS-файл (styles.css)

Файл отвечает за оформление всех страниц проекта, включая адаптивность. Основные особенности:

Глобальные переменные: используются для цветовой схемы, теней, радиусов и шрифтов.

Навигационная панель: современный дизайн с эффектами при наведении.

Карточки ресторанов: стильное отображение ресторанов с изображением, названием, ценовой категорией и рейтингом.

Формы: минималистичный дизайн с закругленными углами и эффектами при наведении на поля ввода.

Модальные окна: оформление для бронирования столиков и галереи изображений.

Адаптивность: поддержка устройств с разной шириной экрана.

3.3. JavaScript-файл (script.js)

Файл содержит основную логику работы приложения. Ключевые функции:

Работа с данными ресторанов:

Список ресторанов хранится в массиве restaurants. Для каждого ресторана указаны:

- Название, кухня, район, ценовая категория.
- Рейтинг, количество отзывов, описание.
- Меню ресторана с названиями блюд, ценами и описанием.

Эти данные используются для отображения ресторанов на страницах.

Функциональность фильтров:

Фильтрация ресторанов по типу кухни, местоположению, цене и названию через функцию applyFilters.

Детальная информация о ресторане:

- Функция showRestaurantDetails позволяет перейти на страницу с подробной информацией о ресторане.
- Функция displayRestaurantDetails отображает меню, описание и отзывы.

Авторизация и регистрация:

- handleLogin: обработка входа пользователя.
- handleRegister: логика регистрации нового пользователя с проверкой пароля.

Личный кабинет:

- Функция displayUserProfile отображает личные данные пользователя.
- Функция displayBookingsHistory показывает историю бронирований.
- Возможность обновления профиля через updateProfile.

Бронирование столиков:

- openBookingModal: открытие модального окна для бронирования.
- submitBooking: создание нового бронирования и сохранение его в localStorage.

Логика управления данными:

- Использование localStorage для хранения информации о пользователе и его бронированиях.
- Функция initializeApp загружает данные из localStorage при загрузке приложения.

Прочие функции:

- logout: выход из системы.
- cancelBooking: отмена бронирования.
- openGallery: открытие изображений ресторана в модальном окне.

4. Основные возможности

Поиск ресторанов:

- Удобные фильтры для поиска по кухне, району, цене.
- Возможность поиска по названию ресторана.

Детальная информация:

Пользователь может увидеть меню, отзывы, описание ресторана, а также его рейтинг.

Регистрация и авторизация:

- Регистрация новых пользователей с валидацией пароля.
- Авторизация зарегистрированных пользователей.

Личный кабинет:

- Возможность редактирования данных.
- Просмотр и управление историей бронирований.

Бронирование столиков:

- Пользователь может выбрать ресторан, указать дату, время, количество гостей и оставить комментарий.
- Возможность отмены бронирования.

Адаптивность:

Сайт корректно отображается как на компьютерах, так и на мобильных устройствах.

5. Технологии

HTML: Разметка страниц.

CSS: Оформление интерфейса с использованием современных технологий (глобальные переменные, адаптивность).

JavaScript:

- Логика работы приложения.
- Использование localStorage для сохранения данных.

Bootstrap: Использован для оформления модальных окон, навигации и адаптивности.

FontAwesome: Для отображения иконок.

Описание работы:

В рамках лабораторной работы было разработано приложение для бронирования столиков в ресторанах. Основной функционал включает:

- Регистрацию новых пользователей.
- Авторизацию пользователей с использованием токенов (JWT).
- Загрузка списка ресторанов с сервера.
- Фильтрацию ресторанов по параметрам (кухня, локация, цена и т.д.).
- Отображение детальной информации о ресторане.
- Создание, отображение и удаление бронирований.

Для работы с API использовался json-server в связке с json-server-auth, который предоставляет поддержку авторизации и защищенных маршрутов.

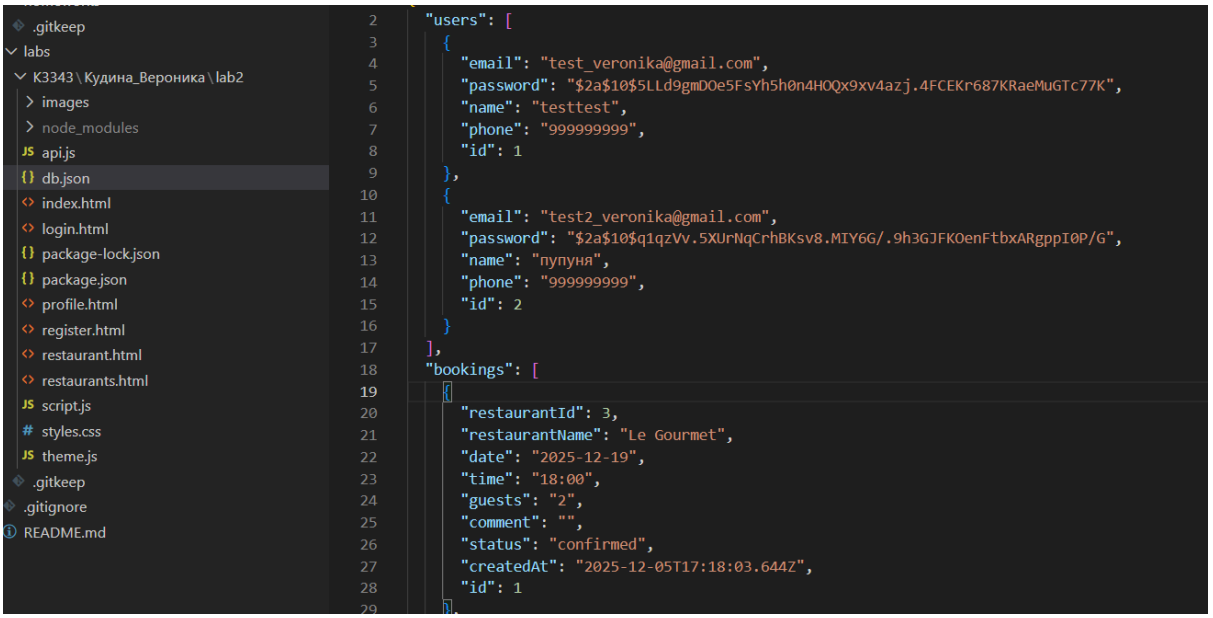

Рисунок 1 - Скриншот сервера

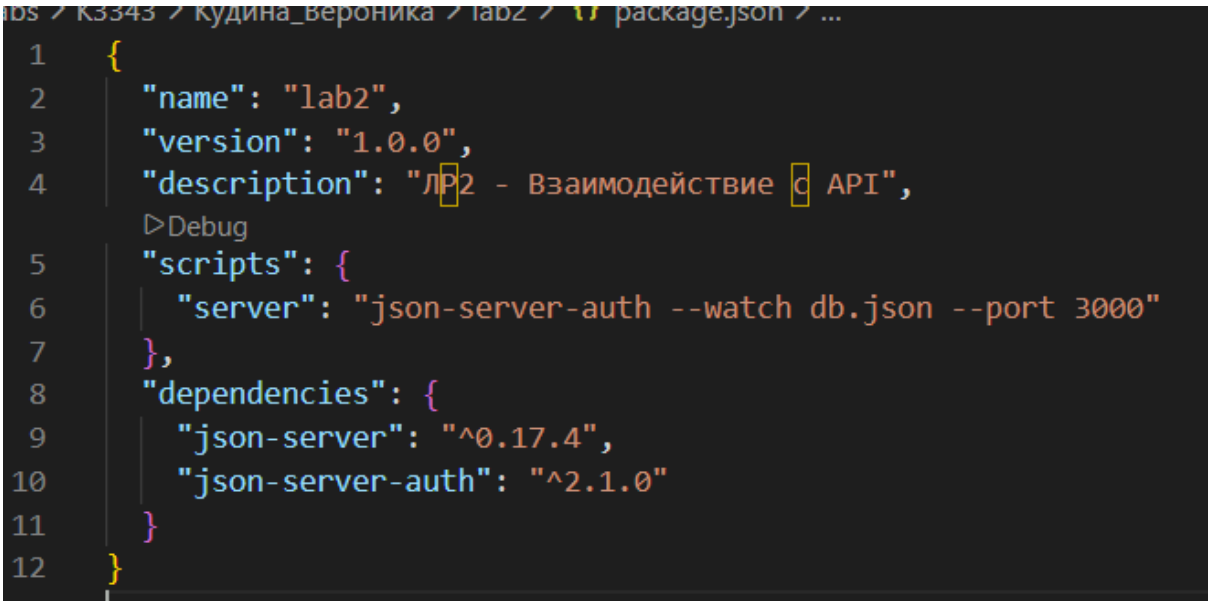

Рисунок 2 - Скриншот содержимого файла db.json

Выбор инструмента для взаимодействия с API:

Для взаимодействия с API был выбран метод fetch. Хотя существуют альтернативы, такие как axios, встроенный fetch был выбран как оптимальный инструмент для реализации данного проекта.

В файле api.js я реализовала функции, которые обеспечивают взаимодействие клиентской части приложения с API. Эти функции отвечают за отправку HTTP-запросов на сервер и обработку ответов. Вот описание каждой из них и их назначение:

1. getRestaurants

Эта функция отправляет запрос на сервер для получения списка всех ресторанов. Она запрашивает данные с общего эндпоинта /restaurants и возвращает массив объектов ресторанов. Данная функция используется для отображения всех ресторанов в приложении.

2. getRestaurantById

Эта функция предназначена для получения информации о конкретном ресторане по его ID. Она отправляет запрос на эндпоинт /restaurants/:id, где :id — идентификатор ресторана. Возвращает объект с данными ресторана.

Эта функция используется, например, при отображении детальной информации о ресторане.

3. loginUser

Данная функция выполняет авторизацию пользователя. Она отправляет POST-запрос на эндпоинт /login с email и паролем пользователя. Если данные корректны, сервер возвращает токен авторизации (JWT) и информацию о пользователе. Этот токен используется для дальнейшего доступа к защищенным маршрутам API.

4. registerUser

Эта функция отвечает за регистрацию нового пользователя. Она отправляет POST-запрос на эндпоинт /register с данными пользователя, такими как имя, email, пароль и телефон. После успешной регистрации сервер возвращает токен авторизации и данные нового пользователя.

5. createBooking

Функция отправляет POST-запрос на эндпоинт /bookings, чтобы создать новое бронирование. В запросе передаются данные бронирования: ресторан, дата, время, количество гостей и комментарий. Сервер возвращает объект с данными созданного бронирования, которое затем отображается в истории бронирований.

6. getUserBookings

Эта функция отправляет запрос на эндпоинт /bookings, фильтруя бронирования по userId. Она используется для получения списка всех бронирований текущего пользователя. Возвращаемые данные отображаются в личном кабинете.

7. deleteBooking

Функция отправляет DELETE-запрос на эндпоинт /bookings/:id, где :id — идентификатор бронирования. Она используется для удаления бронирования пользователя. После успешного удаления запись удаляется как на сервере, так и в интерфейсе клиента.

После регистрации, данные пользователю успешно записываются:

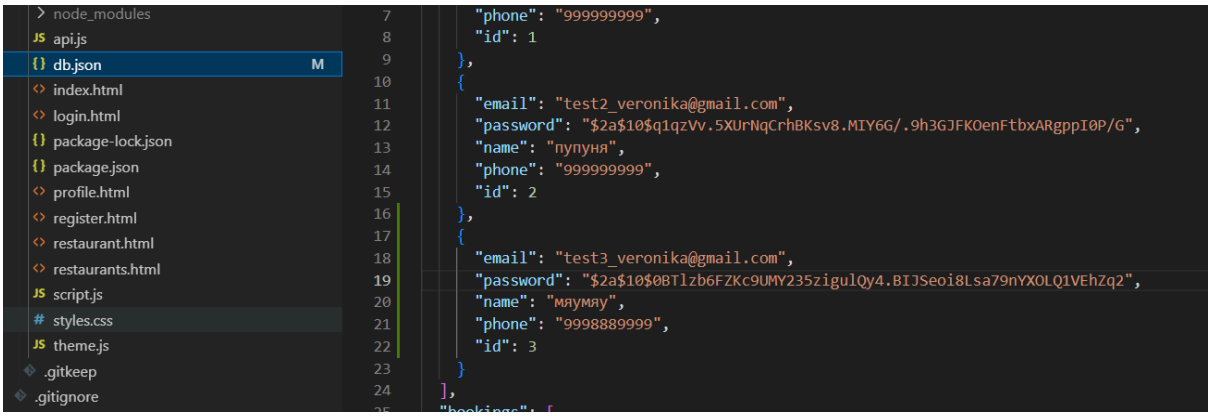

Рисунок 3 - Скриншот содержимого файла db.json

Вывод

В результате работы было создано приложение для бронирования столиков в ресторанах. Приложение взаимодействует с моковым API через HTTP-запросы с использованием fetch. Реализована регистрация, авторизация, фильтрация ресторанов, создание и удаление бронирований.