САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 2

Выполнил:

Берулава Леон Алхасович

К3429

Проверил:

Добряков Д. И.

Санкт-Петербург

2026 г.

Задача

нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами fetch/axios/xhr. Реализуйте моковое API средствами JSON-сервера и подключите к нему авторизацию, как в примерах, которые мы рассматривали в рамках тем "Иммитация работы с API".

Например, для приложения для просмотра прогнозов погоды задание выглядит следующим образом:

Реализовать получение погоды (прогноз на ближайшие 7 дней) из открытого API OpenWeatherMap, взависимости от геолокации пользователя. Реализовать вывод полученного прогноза в виде 7 карточек в три ряда (первый ряд - крупная карточка, второй ряд - три карточки в меньшем размере, третий ряд - четыре карточки в маленьком размере).

Ход работы

1. Цель работы

Разработать полнофункциональное веб-приложение "Кулинарный Портал" с использованием REST API, системой аутентификации.

2. Реализованный функционал

2.1. Backend (REST API)

Технология: JSON Server + Node.js

Реализованные эндпоинты:

POST /auth/login - Авторизация пользователя

POST /auth/register - Регистрация нового пользователя

GET /auth/me - Получение данных текущего пользователя

GET /recipes - Получение списка рецептов

GET /recipes/:id - Получение конкретного рецепта

Пример кода аутентификации:

```
server.post('/auth/login', (req, res) => {
```

```
    const { email, password } = req.body;
    const user = users.find(u => u.email === email &&
u.password === password);

    if (user) {
      const token = 'token_' + Date.now() + '_' +
user.id;
      res.json({ success: true, token, user });
    } else {
      res.status(401).json({
        success: false,
        message: 'Неверный email или пароль'
      });
    }
  });
```

2.2. Frontend

Страницы:

- `index.html` - Страница входа
- `register.html` - Регистрация
- `profile.html` - Профиль пользователя
- `search.html` - Поиск рецептов
- `recipe.html` - Страница рецепта

Технологии:

- HTML5
- CSS3 (Glassmorphism дизайн)
- JavaScript (ES6+)
- Bootstrap 5.3.0

- Font Awesome 6.4.0

2.4. Функции приложения

Аутентификация:

Вход в систему

Регистрация

Хранение токена в localStorage

Защита страниц

Работа с рецептами:

Просмотр списка рецептов

Поиск по названию

Фильтрация по категориям

Просмотр детальной информации

Профиль пользователя:

Отображение данных пользователя

Статистика (количество рецептов)

Список созданных рецептов

3. Примеры работы

```
<form id="loginForm">
    <div class="mb-3">
        <label for="email" class="form-label">
            <i class="fas fa-envelope"></i> Email
        </label>
        <input   type="email"   class="form-control
form-glass"
                id="email"
placeholder="user@example.com" required>
```

```
        </div>
        <button type="submit" class="btn btn-glass-
primary w-100">
            <i class="fas fa-sign-in-alt"></i> Войти
        </button>
    </form>
```

Пример 2: API запрос (JavaScript)

```
    const response = await
fetch(`${API_URL}/auth/login`, {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ email, password })
    });

    const data = await response.json();
    if (data.success) {
        localStorage.setItem('token', data.token);
        window.location.href = 'profile.html';
    }
```

Пример 3: База данных (db.json)

```
    {
     "users": [
       {
        "id": 1,
        "email": "user@example.com",
        "password": "password123",
```

```
    "name": "Иван Иванов"
   }
 ],
 "recipes": [
   {
    "id": 1,
    "title": "Борщ",
    "category": "Супы",
    "time": "90 мин",
    "difficulty": "Средняя",
    "rating": 4.8
   }
 ]
}
```

---

4. Структура проекта

```
Berulava_LR2/
├── css/
│   └── style.css          # Glassmorphism стили
├── js/
│   └── main.js           # Логика приложения
├── index.html            # Страница входа
├── register.html         # Регистрация
```

```
├── profile.html          # Профиль
├── search.html           # Поиск рецептов
├── recipe.html           # Страница рецепта
├── server.js             # REST API сервер
├── db.json               # База данных
└── package.json          # Зависимости
```

7. Выводы

Успешно разработан кулинарный портал с использованием современных веб-технологий. Проект демонстрирует навыки работы с REST API, системой аутентификации