

**Дисциплина:** Технологии программирования

**Выполнил(а):** Мулдабекова Амина

**Преподаватель:** Рашидинов Дамир

**Лабораторная работа №9**

## **Цель работы:**

Освоить работу с асинхронными запросами в веб-приложениях, научиться получать и обрабатывать данные из внешних REST API, отображать их в динамическом интерфейсе и обрабатывать ошибки сети.

## **Краткий теоретический обзор:**

Fetch API — это современный интерфейс JavaScript, который позволяет выполнять HTTP-запросы к внешним ресурсам (серверу, API и т.д.) асинхронно, то есть без перезагрузки страницы.

Асинхронность реализуется через промисы (Promises) и ключевые слова async/await, что делает код более читаемым.

Основные особенности:

* Позволяет отправлять и получать данные в формате JSON.
* Работает через промисы, что упрощает обработку ошибок.
* Используется для взаимодействия клиента с сервером в реальном времени.

Пример синтаксиса:

fetch('https://example.com/api')

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error('Ошибка:', error));

## **Контрольные вопросы:**

**1. Какие современные технологии лежат в основе развития информационных систем?**

Современные ИС строятся на основе облачных вычислений, больших данных (Big Data), искусственного интеллекта (AI), Интернета вещей (IoT) и блокчейн-технологий.

**2. В чём суть технологии «облачных вычислений»?**

Облачные вычисления позволяют хранить и обрабатывать данные на удалённых серверах, предоставляя пользователю доступ через интернет. Это снижает затраты на оборудование и делает работу гибкой.

**3. Какие преимущества и риски несёт использование искусственного интеллекта?**

Преимущества: автоматизация процессов, повышение точности решений, анализ больших данных.

Риски: потеря рабочих мест, ошибки алгоритмов, этические и конфиденциальные проблемы.

**4. Как Интернет вещей изменяет повседневную жизнь?**

IoT позволяет объединять устройства в единую сеть — от бытовой техники до автомобилей. Это повышает удобство, безопасность и эффективность управления ресурсами.

**5. Почему блокчейн называют «технологией доверия»?**

Потому что данные в блокчейне защищены криптографически и хранятся в распределённой сети — их невозможно изменить без согласия всех участников.

**6. Какие аспекты безопасности необходимо учитывать при использовании ИТ в образовании и науке?**

Важно защищать персональные данные, соблюдать авторские права, предотвращать несанкционированный доступ и использовать надёжные каналы связи при работе с научными базами и образовательными платформами.

## **Листинг программы (HTML + JavaScript)**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Лабораторная №9 — Асинхронные запросы</title>

<style>

body { font-family: Arial; margin: 40px; background-color: #fafafa; }

table { border-collapse: collapse; width: 100%; margin-top: 15px; }

th, td { border: 1px solid #ccc; padding: 10px; text-align: left; }

th { background-color: #f2f2f2; }

#loader { color: blue; font-weight: bold; margin-bottom: 10px; }

button { margin-top: 15px; padding: 8px 15px; background-color: #007BFF; color: white; border: none; border-radius: 5px; cursor: pointer; }

</style>

</head>

<body>

<h2>Список пользователей</h2>

<div id="loader"></div>

<table id="userTable">

<thead>

<tr><th>ID</th><th>Имя</th><th>Email</th><th>Город</th></tr>

</thead>

<tbody></tbody>

</table>

<button id="reload">Обновить данные</button>

<script>

async function loadUsers() {

const loader = document.getElementById('loader');

const tbody = document.querySelector('#userTable tbody');

loader.textContent = 'Загрузка данных...';

tbody.innerHTML = '';

try {

const response = await fetch('https://jsonplaceholder.typicode.com/users');

if (!response.ok) throw new Error('Ошибка сети');

const users = await response.json();

users.forEach(u => {

const row = `

<tr>

<td>${u.id}</td>

<td>${u.name}</td>

<td>${u.email}</td>

<td>${u.address.city}</td>

</tr>`;

tbody.insertAdjacentHTML('beforeend', row);

});

loader.textContent = '';

} catch (error) {

loader.textContent = 'Ошибка при получении данных.';

console.error(error);

}

}

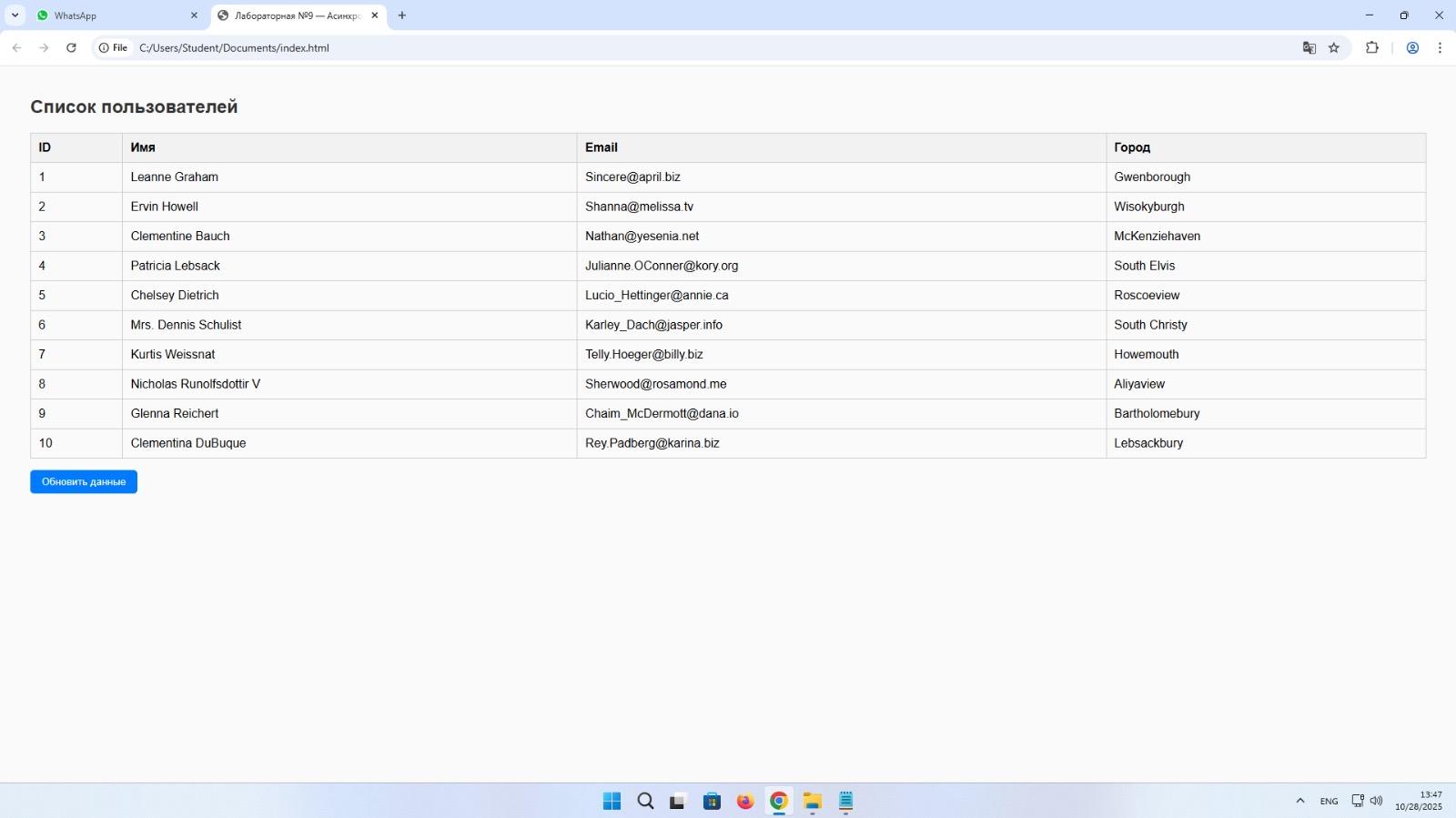
document.getElementById('reload').addEventListener('click', loadUsers);

loadUsers();

</script>

</body>

</html>



## **Вывод:**

В ходе лабораторной работы я изучила принципы выполнения асинхронных запросов с помощью Fetch API. Научилась получать данные с внешнего сервера и динамически отображать их на странице без перезагрузки.

Закрепила навыки обработки ошибок и работы с JSON-данными.

Освоила базовые приёмы интеграции REST API в веб-приложения.