ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №2

по дисциплине: JAVASCRIPT, ЕГО БИБЛИОТЕКИ И ФРЕЙМВОРКИ В FRONTEND-РАЗРАБОТКЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4232		Г. П. Матюшков
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2025

Цель работы:

Формирование практических навыков работы с асинхронным кодом в языке JavaScript, используя callback-функции и объект Promise.

Задание:

1. Напишите функцию `filterArray`, которая принимает массив чисел и callback-функцию. Функция `filterArray` должна вызывать callback функцию для каждого элемента массива и возвращать новый массив, содержащий только те элементы, для которых callback-функция вернула `true`.

Напишите 2 примера применения этой функции. К примеру, для фильтрации четных и нечетных значений массива.

2. Напишите асинхронную функцию `fetchData`, которая принимает URL в качестве параметра и возвращает Promise. Функция должна использовать `fetch` для получения данных с указанного URL. Если запрос прошел успешно (статус ответа 200), Promise должен быть разрешен с полученными данными в виде строки. Если запрос не удался (любой другой статус), Promise должен быть отклонен с сообщением об ошибке.

Результаты работы:

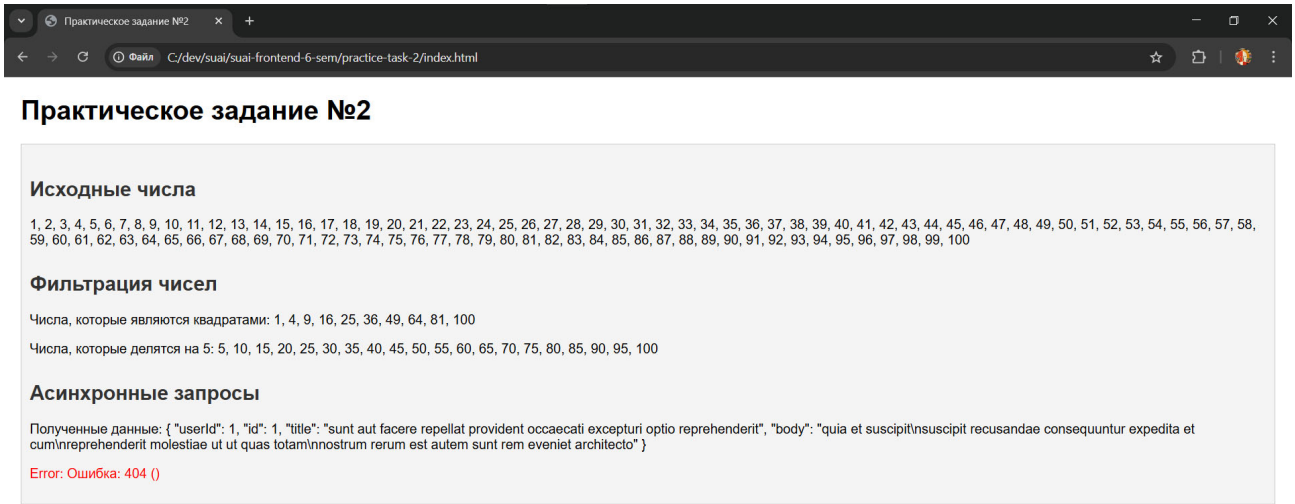

Рисунок 1 – главная страница

Листинг кода:

index.html:

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Практическое задание №2</title>
    <style>
        body {
            font-family: Arial, sans-serif;
```

```
            margin: 20px;
        }

        .output {
            margin-top: 20px;
            padding: 10px;
            background-color: #f4f4f4;
            border: 1px solid #ccc;
        }

        .error {
            color: red;
        }

        h2 {
            margin-top: 30px;
            color: #333;
        }
    </style>
</head>
<body>
    <h1>Практическое задание №2</h1>

    <div class="output" id="output"></div>

    <script>
        // Функция filterArray
        function filterArray(array, callback) {
            const result = [];
            for (let i = 0; i < array.length; i++) {
                if (callback(array[i])) {
                    result.push(array[i]);
                }
            }
            return result;
        }

        // Исходный массив чисел от 1 до 100
        const numbers = Array.from({ length: 100 }, (_, i) => i + 1);

        // Вывод исходных чисел
        const output = document.getElementById('output');
        output.innerHTML = `
                <h2>Исходные числа</h2>
                <p>${numbers.join(', ')}</p>
            `;

        // Фильтрация 1: числа, которые являются квадратами целых чисел
        const numbersThatAreSquares = filterArray(numbers, function (num) {
            return Math.sqrt(num) % 1 === 0; // Проверка, является ли число
квадратом
        });

        // Фильтрация 2: числа, которые делятся на 5
        const numbersDivisibleBy5 = filterArray(numbers, function (num) {
            return num % 5 === 0;
        });

        // Вывод результатов фильтрации
```

```
        output.innerHTML += `
                <h2>Фильтрация чисел</h2>
                <p>Числа, которые являются квадратами:
${numbersThatAreSquares.join(', ')}</p>
                <p>Числа, которые делятся на 5: ${numbersDivisibleBy5.join(',
')}</p>
            `;

        // Асинхронная функция fetchData
        async function fetchData(url) {
            return new Promise((resolve, reject) => {
                fetch(url)
                    .then(response => {
                        if (response.status === 200) {
                            return response.text();
                        } else {
                            reject(`Ошибка: ${response.status}
(${response.statusText})`);
                        }
                    })
                    .then(data => resolve(data))
                    .catch(error => reject(error));
            });
        }

        // Вывод заголовка для асинхронных запросов
        output.innerHTML += `<h2>Асинхронные запросы</h2>`;

        // Пример успешного использования fetchData
        fetchData('https://jsonplaceholder.typicode.com/posts/1')
            .then(data => {
                output.innerHTML += `<p>Полученные данные: ${data}</p>`;
            })
            .catch(error => {
                output.innerHTML += `<p class="error">Error: ${error}</p>`;
            });

        // Пример использования fetchData с ошибкой (неверный URL)
        fetchData('https://jsonplaceholder.typicode.com/nonexistent')
            .then(data => {
                output.innerHTML += `<p>Полученные данные: ${data}</p>`;
            })
            .catch(error => {
                output.innerHTML += `<p class="error">Error: ${error}</p>`;
            });
    </script>
</body>
</html>
```

Вывод:

В ходе выполнения лабораторной работы были успешно реализованы две задачи, направленные на формирование навыков работы с асинхронным кодом в JavaScript. В первой части задания была разработана функция filterArray, которая позволяет фильтровать массив чисел с использованием callback-функций. Были продемонстрированы примеры её применения для фильтрации чисел, являющихся квадратами целых чисел, и чисел, делящихся

на 5. Во второй части задания была создана асинхронная функция fetchData, использующая fetch для получения данных по указанному URL. Функция возвращает Promise, который разрешается при успешном запросе или отклоняется с сообщением об ошибке. Работа показала умение работать с callback-функциями, Promises и асинхронными запросами, а также продемонстрировала понимание принципов обработки данных в JavaScript. Все задачи выполнены в соответствии с требованиями, результаты работы корректны и наглядно представлены на веб-странице.