МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

(КАФЕДРА 43)

|  |
| --- |
| ОТЧЕТ ЗАЩИЩЁН С ОЦЕНКОЙ  РУКОВОДИТЕЛЬ |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ассистент |  |  |  |  |  | Ю. В. Ветрова |
| (должность, учёная степень, звание) |  | (подпись) |  | (дата защиты) |  | (инициалы, фамилия) |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРАКТИЧЕСКОМУ ЗАДАНИЮ №1

Практические навыки работы с DOM и хранением данных в JavaScript

по дисциплине: [JavaScript, его библиотеки и фреймворки в Frontend-разработке](https://lms.guap.ru/course/view.php?id=1099" \l "section-1" \o "JavaScript, его библиотеки и фреймворки в Frontend-разработке)

|  |
| --- |
| РАБОТУ ВЫПОЛНИЛ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4236 |  |  |  | Л. Мвале |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. **Цель**

Цель работы – формирование практических навыков работы с элементами DOM, с технологиями, позволяющими управлять отображением и позиционированием элементов в веб-документе, а также формирование практических навыков работы с cookie, local storage, session storage в JavaScript.

1. **Задание**

Задание состоит из двух частей. Требуется выполнить по одному из пунктов каждой части (на выбор студента).

1. **я часть задания вариант 3**

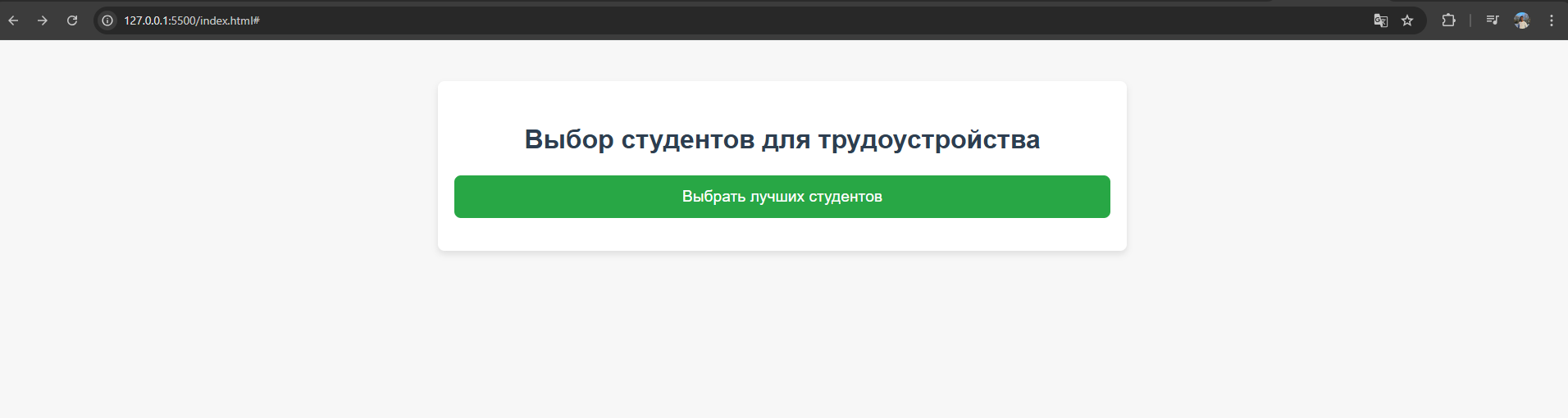
Из двух групп студентов, прошедших тестирование по некоторой дисциплине, требуется отобрать лучших для рассмотрения вопроса их дальнейшего трудоустройства.

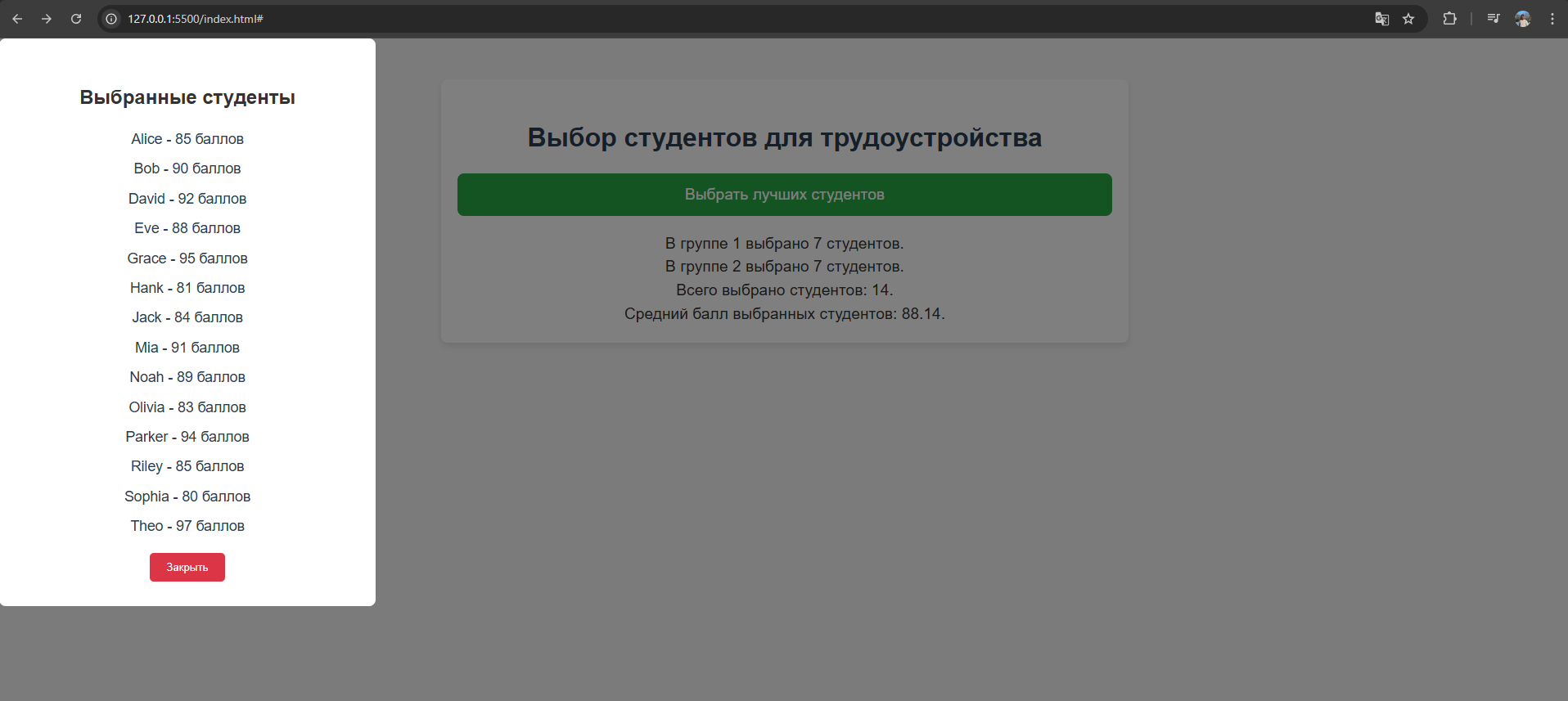
Для этого создайте 2 массива. В каждом должно быть по 10 объектов, содержащих фамилии студентов группы, прошедших тестирование по какой-то дисциплине (например, «Основы Frontend-разработки»), а также результат тестирования в баллах (от 0 до 100).

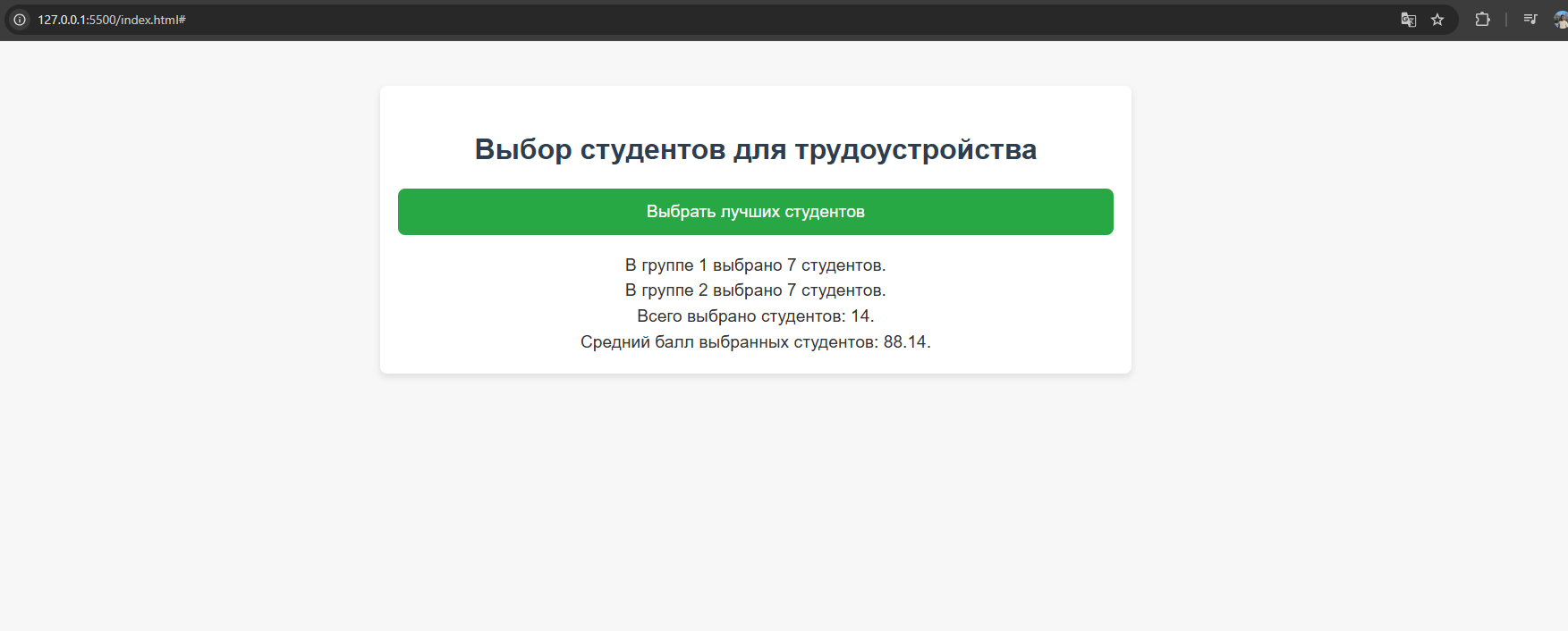
Требуется в каждом из массивов выбрать тех, кто получил не менее 80 баллов и вывести сообщение о том, сколько в каждой группе отобрано студентов. Из двух полученных в результате фильтрации массивов необходимо сформировать один массив и вывести сообщение о том, сколько всего студентов набрали от 80 до 100 баллов. Затем вывести, например, в модальных окнах фамилии отобранных студентов и рядом указать количество набранных этим студентом баллов. В завершение вывести сообщение «Средний балл отобранных на работу студентов: (указать результат)».

Подсказка: при выполнении задания пригодятся ряд известных вам методов работы с объектом Array, например: filter(), concat(), reduce(), forEach(), join().

скриншоты







Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Student Selection for Employment</title>

    <!-- Подключение CSS файла для стилизации страницы -->

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <!-- Контейнер для основного контента -->

    <div class="container">

        <h2 class="main-title">Выбор студентов для трудоустройства</h2>

        <!-- Кнопка для запуска выбора студентов -->

        <button class="select-button" onclick="selectStudents()">Выбрать лучших студентов</button>

        <!-- Контейнер для вывода результата -->

        <div id="result" class="result-container"></div>

    </div>

    <!-- Модальное окно для отображения выбранных студентов -->

    <div id="studentModal" class="modal">

        <div class="modal-content">

            <h3 class="modal-title">Выбранные студенты</h3>

            <!-- Список студентов, который будет заполняться динамически -->

            <ul id="selectedStudentsList" class="student-list"></ul>

            <!-- Кнопка для закрытия модального окна -->

            <button class="close-button" onclick="closeModal()">Закрыть</button>

        </div>

    </div>

    <!-- Подключение JavaScript файла для логики работы страницы -->

    <script src="script.js"></script>

</body>

</html>

Script.js

// Массив студентов группы 1

*const* group1 = [

    { name: "Alice", score: 85 }, { name: "Bob", score: 90 },

    { name: "Charlie", score: 78 }, { name: "David", score: 92 },

    { name: "Eve", score: 88 }, { name: "Frank", score: 79 },

    { name: "Grace", score: 95 }, { name: "Hank", score: 81 },

    { name: "Ivy", score: 74 }, { name: "Jack", score: 84 }

];

// Массив студентов группы 2

*const* group2 = [

    { name: "Liam", score: 76 }, { name: "Mia", score: 91 },

    { name: "Noah", score: 89 }, { name: "Olivia", score: 83 },

    { name: "Parker", score: 94 }, { name: "Quinn", score: 70 },

    { name: "Riley", score: 85 }, { name: "Sophia", score: 80 },

    { name: "Theo", score: 97 }, { name: "Uma", score: 65 }

];

// Функция для выбора студентов, набравших >= 80 баллов

*function* selectStudents() {

    // Фильтруем студентов, у которых оценка >= 80

*const* selectedGroup1 = group1.filter(*student* *=>* student.score >= 80);

*const* selectedGroup2 = group2.filter(*student* *=>* student.score >= 80);

    // Объединяем два массива

*const* selectedStudents = selectedGroup1.concat(selectedGroup2);

    // Вычисляем количество выбранных студентов в каждой группе

*const* totalSelectedGroup1 = selectedGroup1.length;

*const* totalSelectedGroup2 = selectedGroup2.length;

*const* totalSelected = selectedStudents.length;

    // Вычисляем средний балл для выбранных студентов

*const* totalScore = selectedStudents.reduce((*sum*, *student*) *=>* sum + student.score, 0);

*const* avgScore = totalScore / selectedStudents.length;

    // Формируем сообщение о результатах

*let* resultMessage = `

        В группе 1 выбрано ${totalSelectedGroup1} студентов.<br>

        В группе 2 выбрано ${totalSelectedGroup2} студентов.<br>

        Всего выбрано студентов: ${totalSelected}.<br>

        Средний балл выбранных студентов: ${avgScore.toFixed(2)}.<br>

    `;

    // Отображаем выбранных студентов в модальном окне

    displayModal(selectedStudents);

    // Отображаем сообщение с результатами

    document.getElementById("result").innerHTML = resultMessage;

}

// Функция для отображения модального окна с выбранными студентами

*function* displayModal(*students*) {

*const* modal = document.getElementById("studentModal");

*const* studentList = document.getElementById("selectedStudentsList");

    // Очищаем предыдущий список студентов

    studentList.innerHTML = '';

    // Создаем элементы списка для каждого выбранного студента

    students.forEach(*student* *=>* {

*const* listItem = document.createElement("li");

        listItem.textContent = `${student.name} - ${student.score} баллов`;

        studentList.appendChild(listItem);

    });

    // Показываем модальное окно

    modal.style.display = 'block';

}

// Функция для закрытия модального окна

*function* closeModal() {

    document.getElementById("studentModal").style.display = 'none';

}

styles.css

/\* Общие стили \*/

body {

*font-family*: 'Arial', sans-serif;

*background-color*: #f7f7f7;

*color*: #333;

*margin*: 0;

*padding*: 0;

}

/\* Контейнер для основного контента \*/

.container {

*max-width*: 800px;

*margin*: 50px auto;

*padding*: 20px;

*background-color*: #fff;

*border-radius*: 8px;

*box-shadow*: 0 4px 8px rgba(0, 0, 0, 0.1);

}

/\* Основной заголовок \*/

.main-title {

*text-align*: center;

*font-size*: 2em;

*color*: #2d3e50;

*margin-bottom*: 20px;

}

/\* Кнопка для выбора студентов \*/

.select-button {

*display*: block;

*width*: 100%;

*padding*: 15px;

*background-color*: #28a745;

*color*: white;

*font-size*: 1.2em;

*border*: none;

*border-radius*: 8px;

*cursor*: pointer;

*transition*: background-color 0.3s ease;

*margin-bottom*: 20px;

}

.select-button:hover {

*background-color*: #218838;

}

/\* Контейнер для результата \*/

.result-container {

*font-size*: 1.2em;

*text-align*: center;

*margin-top*: 20px;

}

/\* Модальное окно \*/

.modal {

*display*: none;

*position*: fixed;

*top*: 0;

*left*: 0;

*width*: 100%;

*height*: 100%;

*background-color*: rgba(0, 0, 0, 0.5);

*justify-content*: center;

*align-items*: center;

}

.modal-content {

*background-color*: white;

*padding*: 30px;

*border-radius*: 8px;

*max-width*: 400px;

*width*: 100%;

*text-align*: center;

}

.modal-title {

*font-size*: 1.5em;

*margin-bottom*: 20px;

}

.student-list {

*list-style-type*: none;

*padding*: 0;

*margin-bottom*: 20px;

}

.student-list li {

*font-size*: 1.1em;

*color*: #2d3e50;

*margin-bottom*: 10px;

}

.close-button {

*padding*: 10px 20px;

*background-color*: #dc3545;

*color*: white;

*border*: none;

*border-radius*: 5px;

*cursor*: pointer;

*transition*: background-color 0.3s ease;

}

.close-button:hover {

*background-color*: #c82333;

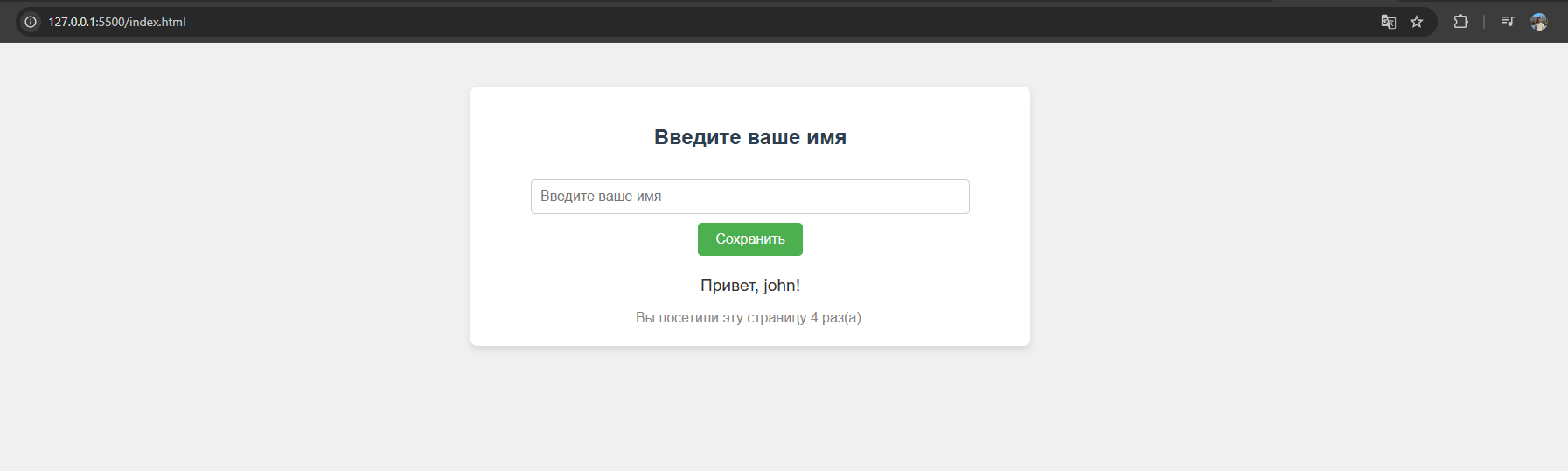
}

**2-я часть задания вариант 1**

1. Создайте веб-страницу, на которой пользователь может ввести свое имя. При нажатии кнопки "Сохранить" сохраните имя пользователя в cookies и отобразите его на странице при следующем посещении.

Напишите скрипт, который будет отслеживать, сколько раз пользователь посетил Вашу веб-страницу. Используйте localStorage для хранения этой информации и отображайте количество посещений на странице.

**Скриншоты**



Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Track User Visits</title>

    <!-- Подключаем внешний CSS для стилизации страницы -->

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <!-- Контейнер для отображения формы ввода имени -->

    <div class="container">

        <h2>Введите ваше имя</h2>

        <input type="text" id="nameInput" placeholder="Введите ваше имя">

        <button id="saveButton" onclick="saveName()">Сохранить</button>

        <!-- Сообщение с отображением имени пользователя -->

        <div id="greetingMessage"></div>

        <!-- Сообщение с количеством посещений страницы -->

        <div id="sessionCountMessage"></div>

    </div>

    <!-- Подключаем JavaScript файл для логики работы страницы -->

    <script src="script.js"></script>

</body>

</html>

script.js

// Функция для сохранения имени пользователя в cookie

*function* saveName() {

*const* userName = document.getElementById("nameInput").value; // Получаем имя из поля ввода

    if (userName) {

        // Сохраняем имя в cookie на 1 день

        document.cookie = `username=${userName}; expires=${new Date(Date.now() + 86400000).toUTCString()}; path=/`;

        displayGreeting(); // Отображаем приветственное сообщение с именем

    }

}

// Функция для получения значения из cookie

*function* getCookie(*name*) {

*const* cookieArr = document.cookie.split(";"); // Разделяем cookies

    for (*let* i = 0; i < cookieArr.length; i++) {

*const* cookie = cookieArr[i].trim();

        if (cookie.startsWith(name + "=")) {

            return cookie.substring(name.length + 1);

        }

    }

    return "";

}

// Функция для отображения приветственного сообщения

*function* displayGreeting() {

*const* userName = getCookie("username"); // Получаем имя из cookie

    if (userName) {

        document.getElementById("greetingMessage").textContent = `Привет, ${userName}!`; // Выводим имя пользователя

    } else {

        document.getElementById("greetingMessage").textContent = `Привет, незнакомец!`; // Если имя не установлено

    }

}

// Функция для отслеживания количества посещений страницы

*function* trackVisits() {

*let* visitCount = localStorage.getItem("visitCount"); // Получаем количество посещений из localStorage

    if (visitCount) {

        visitCount = parseInt(visitCount) + 1; // Увеличиваем количество посещений

    } else {

        visitCount = 1; // Если посещений еще не было, устанавливаем 1

    }

    localStorage.setItem("visitCount", visitCount); // Сохраняем количество посещений в localStorage

    // Отображаем сообщение с количеством посещений

    document.getElementById("sessionCountMessage").textContent = `Вы посетили эту страницу ${visitCount} раз(а).`;

}

// Инициализация страницы: отображение приветствия и количества посещений

displayGreeting();

trackVisits();

styles.css

/\* Общие стили для всей страницы \*/

body {

*font-family*: 'Arial', sans-serif;

*background-color*: #f0f0f0;

*color*: #333;

*margin*: 0;

*padding*: 0;

}

/\* Контейнер для централизованного отображения контента \*/

.container {

*max-width*: 600px;

*margin*: 50px auto;

*padding*: 20px;

*background-color*: #fff;

*border-radius*: 8px;

*box-shadow*: 0 4px 8px rgba(0, 0, 0, 0.1);

*text-align*: center;

}

/\* Стили для заголовка \*/

h2 {

*color*: #2d3e50;

}

/\* Стили для поля ввода имени \*/

#nameInput {

*padding*: 10px;

*font-size*: 1em;

*width*: 80%;

*margin*: 10px 0;

*border*: 1px solid #ccc;

*border-radius*: 5px;

}

/\* Стили для кнопки "Сохранить" \*/

#saveButton {

*padding*: 10px 20px;

*font-size*: 1em;

*background-color*: #4CAF50;

*color*: white;

*border*: none;

*border-radius*: 5px;

*cursor*: pointer;

*transition*: background-color 0.3s ease;

}

#saveButton:hover {

*background-color*: #45a049;

}

/\* Стили для сообщения приветствия \*/

#greetingMessage {

*margin-top*: 20px;

*font-size*: 1.2em;

}

/\* Стили для сообщения с количеством посещений \*/

#sessionCountMessage {

*margin-top*: 10px;

*font-size*: 1em;

*color*: #888;

}

1. **Выводы**

В ходе выполнения лабораторной работы были приобретены практические навыки работы с элементами DOM и технологиями хранения данных в браузере, такими как cookies, localStorage и sessionStorage. Участник научился эффективно управлять взаимодействием с пользователем, контролировать отображение и позиционирование элементов, а также сохранять и отслеживать данные о пользователе для улучшения пользовательского опыта.