ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Н. И. Мирошниченко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОМ ЗАДАНИИ №1 |
| по курсу: JavaScript, его библиотеки и фреймворки в Frontend-разработке |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4217 |  |  |  | А.Н. Медянкина |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. 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. Выбранные задания:

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)».

Часть 2

2. Создайте простую игру "Угадай число". Сгенерируйте случайное число и сохраните его в sessionStorage. Пользователь должен вводить числа в текстовое поле. Необходимо организовать проверку, совпадает ли введенное пользователем число с загаданным. Используйте sessionStorage для хранения текущего состояния игры.

1. Ход выполнения работы

Сперва в разделе <body> было необходимо создать две группы студентов и указать им результат прохождения тестирования. Полный код можно посмотреть в Приложении А.



Рисунок 1 – Создание массивов с данными студентов

После чего был прописан основной костыль программы – функция с фильтрацией студентов и отбором студентов, у которых итоговый балл за тест от 80 до 100.

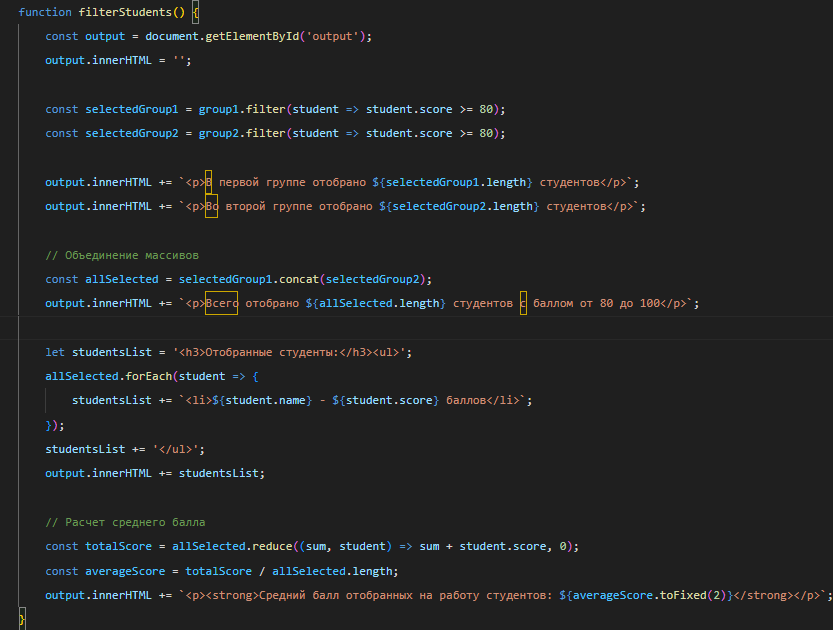


Рисунок 2 – Создание функции

Так же для этой веб-странички в отдельном файле .css был прописан код со стилем. Полный код можно посмотреть в Приложении Б.

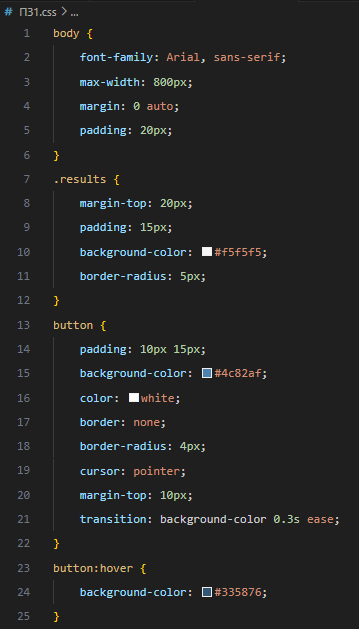


Рисунок 3 – Стиль для первой части задания

В результате получился веб-сайт, представленный на рисунках 4 и 5.

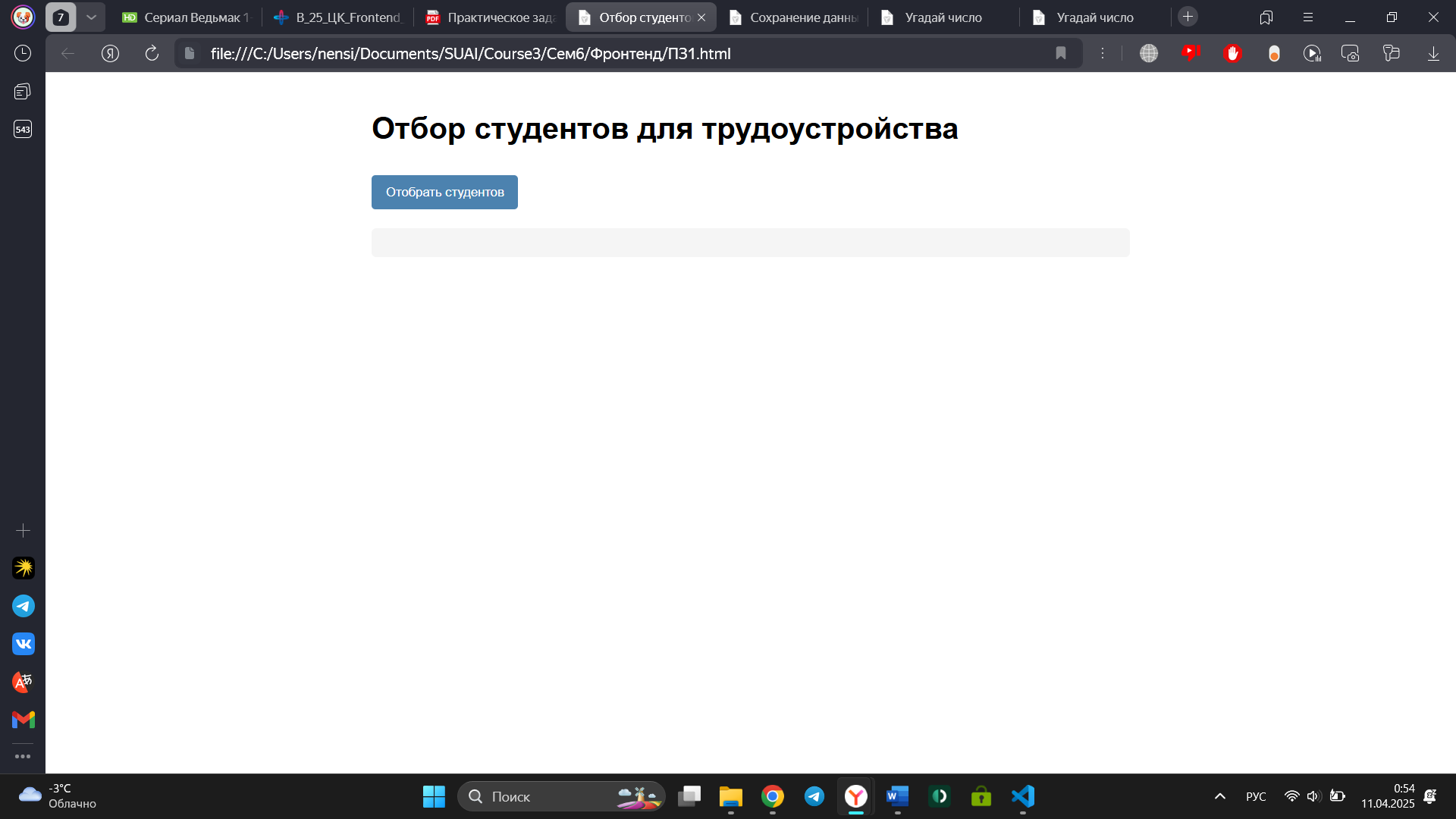


Рисунок 4 – Начальный экран сайта до нажатия на кнопку

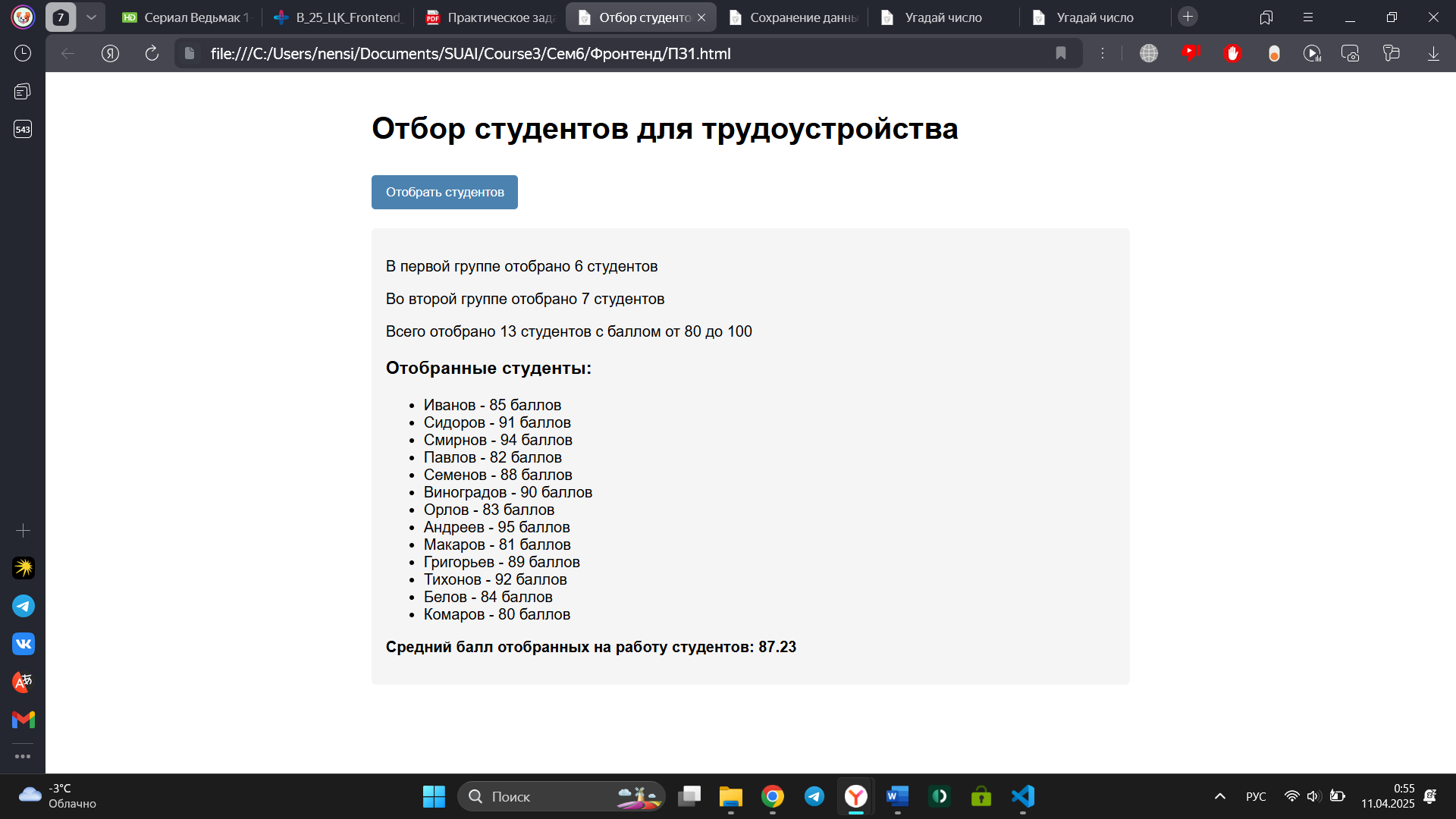


Рисунок 5 – Открытие результата после нажатия на кнопку

Затем была выполнена вторая часть заданий, где создавалась игра «Угадай число». По идее компьютер загадывает случайное число от 1 до 10, а игрок должен его отгадать. При загрузке страницы генерируется и сохраняется в sessionStorage загаданное число, чтобы при обновлении страницы игра не сбрасывалась. Пользователь вводит свои догадки в текстовое поле, нажимает кнопку "Проверить" и получает подсказки – «Загаданное число больше» или «Загаданное число меньше». Когда число угадано, игра выводит поздравление с количеством затраченных попыток. Кнопка «Начать заново» позволяет сбросить текущую игру и загадать новое число. Интерфейс игры выполнен в минималистичном стиле с зелёными кнопками и серым фоном игрового поля, адаптирован под разные размеры экранов. Все данные (загаданное число и счётчик попыток) хранятся в sessionStorage браузера и автоматически удаляются при закрытии вкладки. Результат создания игры и запуска представлен на рисунках 6, 7. Полный код, в том числе уже со стилем, можно посмотреть в Приложении В.

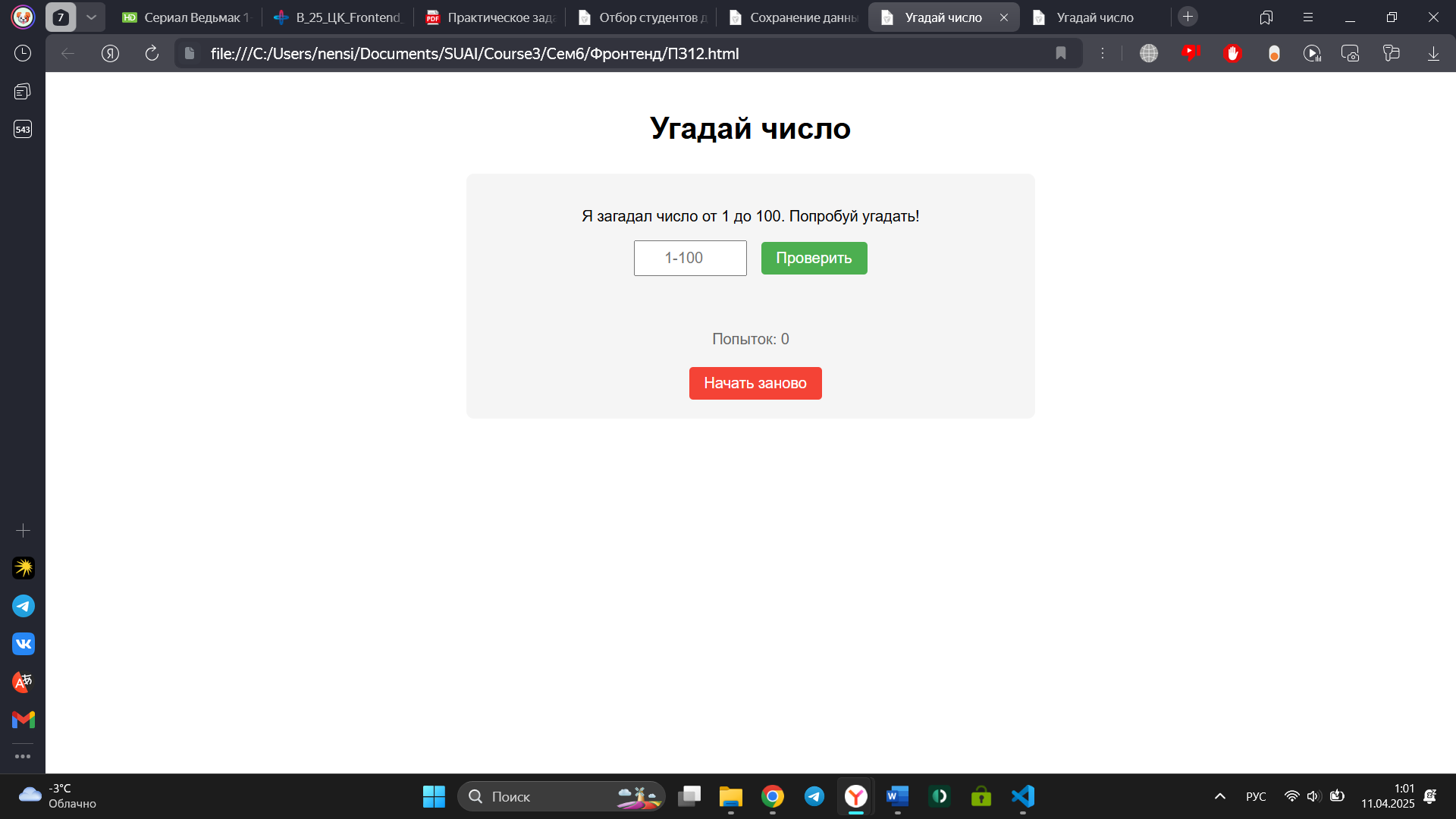


Рисунок 6 – Начало игры «Угадай число»

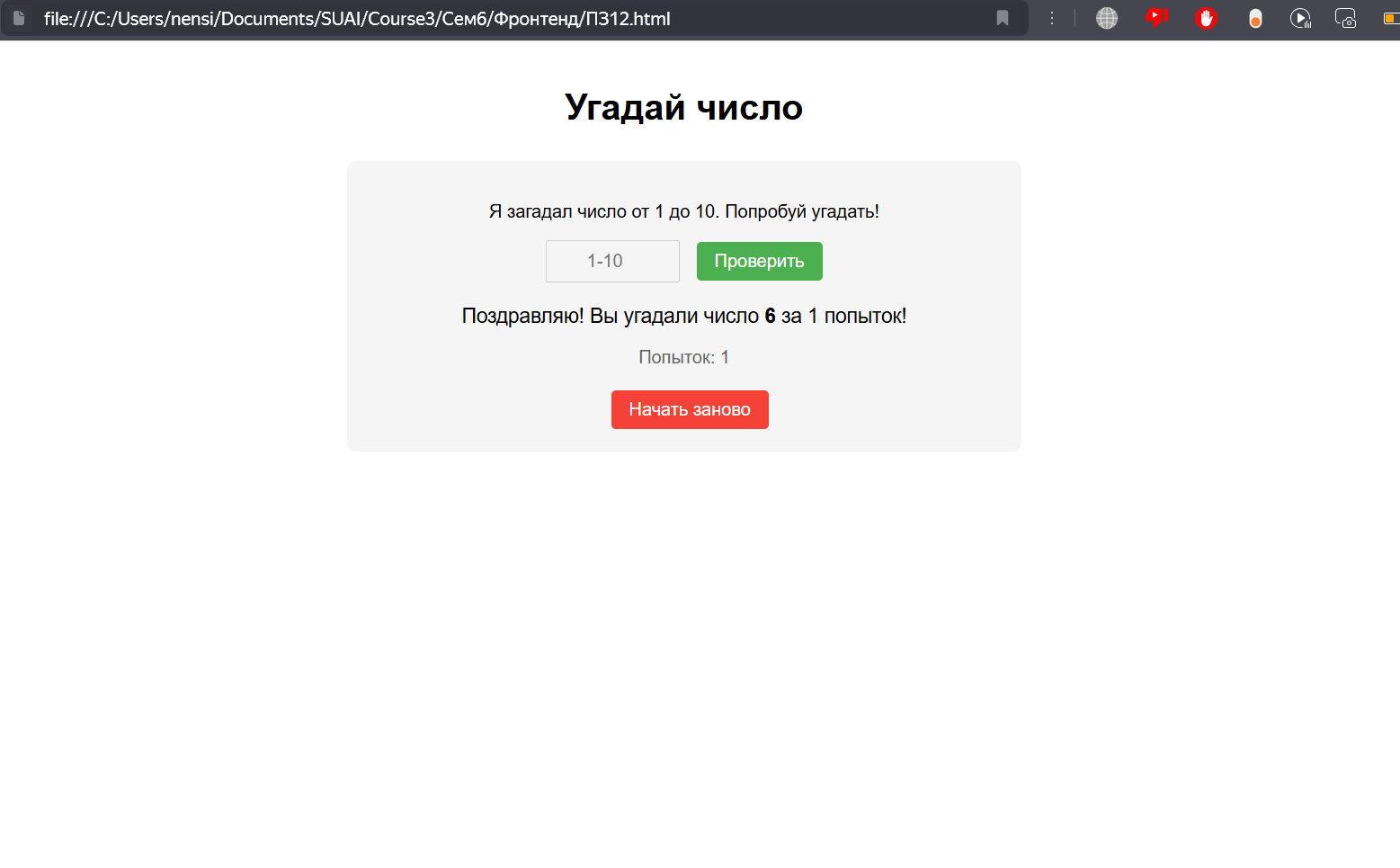


Рисунок 7 – Процесс игры «Угадай число»

1. Вывод

В ходе выполнения практического задания были успешно освоены ключевые аспекты работы с JavaScript в контексте frontend-разработки. В первой части работы реализован алгоритм обработки данных студентов с использованием современных методов работы с массивами (filter, concat, reduce), что позволило эффективно решить задачу фильтрации и анализа результатов тестирования. Разработанный интерфейс демонстрирует принципы взаимодействия JavaScript с DOM-элементами, обеспечивая интуитивно понятное отображение результатов отбора.

Во второй части задания создана интерактивная игра «Угадай число», которая наглядно иллюстрирует работу с sessionStorage для сохранения состояния приложения между перезагрузками страницы. Особое внимание уделено обработке пользовательского ввода и реализации логики игры с предоставлением обратной связи. Интерфейс игры разработан с учетом принципов адаптивного дизайна и UX-оптимизации.

Практическая значимость выполненной работы заключается в освоении технологий клиентского хранения данных и манипуляций с DOM, которые являются фундаментальными для современных веб-приложений. Реализованные решения демонстрируют понимание асинхронной природы JavaScript и принципов построения интерактивных интерфейсов.

# Приложение а

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Отбор студентов для трудоустройства</title>

<link rel="stylesheet" href="ПЗ1.css">

</head>

<body>

<h1>Отбор студентов для трудоустройства</h1>

<button onclick="filterStudents()">Отобрать студентов</button>

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

<script>

// Массивы с данными студентов

const group1 = [

{ name: "Иванов", score: 85 },

{ name: "Петров", score: 72 },

{ name: "Сидоров", score: 91 },

{ name: "Кузнецов", score: 68 },

{ name: "Смирнов", score: 94 },

{ name: "Васильев", score: 79 },

{ name: "Павлов", score: 82 },

{ name: "Семенов", score: 88 },

{ name: "Голубев", score: 65 },

{ name: "Виноградов", score: 90 }

];

const group2 = [

{ name: "Николаев", score: 78 },

{ name: "Орлов", score: 83 },

{ name: "Андреев", score: 95 },

{ name: "Макаров", score: 81 },

{ name: "Киселев", score: 76 },

{ name: "Григорьев", score: 89 },

{ name: "Тихонов", score: 92 },

{ name: "Кудрявцев", score: 77 },

{ name: "Белов", score: 84 },

{ name: "Комаров", score: 80 }

];

function filterStudents() {

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

output.innerHTML = '';

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

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

output.innerHTML += `<p>В первой группе отобрано ${selectedGroup1.length} студентов</p>`;

output.innerHTML += `<p>Во второй группе отобрано ${selectedGroup2.length} студентов</p>`;

// Объединение массивов

const allSelected = selectedGroup1.concat(selectedGroup2);

output.innerHTML += `<p>Всего отобрано ${allSelected.length} студентов с баллом от 80 до 100</p>`;

let studentsList = '<h3>Отобранные студенты:</h3><ul>';

allSelected.forEach(student => {

studentsList += `<li>${student.name} - ${student.score} баллов</li>`;

});

studentsList += '</ul>';

output.innerHTML += studentsList;

// Расчет среднего балла

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

const averageScore = totalScore / allSelected.length;

output.innerHTML += `<p><strong>Средний балл отобранных на работу студентов: ${averageScore.toFixed(2)}</strong></p>`;

}

</script>

</body>

</html>

# приложение б

body {

font-family: Arial, sans-serif;

max-width: 800px;

margin: 0 auto;

padding: 20px;

}

.results {

margin-top: 20px;

padding: 15px;

background-color: #f5f5f5;

border-radius: 5px;

}

button {

padding: 10px 15px;

background-color: #4c82af;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

margin-top: 10px;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #335876;

}

# приложение в

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Угадай число</title>

<style>

body {

font-family: Arial, sans-serif;

max-width: 600px;

margin: 0 auto;

padding: 20px;

text-align: center;

}

.game-container {

margin-top: 30px;

padding: 20px;

background-color: #f5f5f5;

border-radius: 8px;

}

input {

padding: 8px;

font-size: 16px;

width: 100px;

text-align: center;

}

button {

padding: 8px 16px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

margin-left: 10px;

font-size: 16px;

}

button:hover {

background-color: #45a049;

}

#message {

margin-top: 20px;

font-size: 18px;

min-height: 27px;

}

#attempts {

margin-top: 10px;

color: #666;

}

.restart-btn {

background-color: #f44336;

margin-top: 20px;

}

.restart-btn:hover {

background-color: #d32f2f;

}

</style>

</head>

<body>

<h1>Угадай число</h1>

<div class="game-container">

<p>Я загадал число от 1 до 10. Попробуй угадать!</p>

<div>

<input type="number" id="guessInput" min="1" max="10" placeholder="1-10">

<button onclick="checkGuess()">Проверить</button>

</div>

<div id="message"></div>

<div id="attempts"></div>

<button class="restart-btn" onclick="restartGame()">Начать заново</button>

</div>

<script>

window.onload = function() {

// Проверка на сохранение

if (!sessionStorage.getItem('secretNumber')) {

startNewGame();

}

updateAttemptsDisplay();

};

// Начать игру

function startNewGame() {

const secretNumber = Math.floor(Math.random() \* 10) + 1;

sessionStorage.setItem('secretNumber', secretNumber);

sessionStorage.setItem('attempts', '0');

document.getElementById('message').textContent = '';

document.getElementById('guessInput').value = '';

updateAttemptsDisplay();

}

// Проверка введенного с клавы

function checkGuess() {

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

const messageDiv = document.getElementById('message');

// Проверка на "число"

if (!guessInput.value) {

messageDiv.textContent = 'Пожалуйста, введите число';

return;

}

const guess = parseInt(guessInput.value);

const secretNumber = parseInt(sessionStorage.getItem('secretNumber'));

let attempts = parseInt(sessionStorage.getItem('attempts')) + 1;

sessionStorage.setItem('attempts', attempts.toString());

updateAttemptsDisplay();

// Проверка предположения

if (guess === secretNumber) {

messageDiv.innerHTML = `Поздравляю! Вы угадали число <strong>${secretNumber}</strong> за ${attempts} попыток!`;

guessInput.disabled = true;

} else if (guess < secretNumber) {

messageDiv.textContent = 'Загаданное число больше';

} else {

messageDiv.textContent = 'Загаданное число меньше';

}

guessInput.value = '';

guessInput.focus();

}

function updateAttemptsDisplay() {

const attempts = sessionStorage.getItem('attempts') || '0';

document.getElementById('attempts').textContent = `Попыток: ${attempts}`;

}

// Начать игру заново

function restartGame() {

startNewGame();

document.getElementById('guessInput').disabled = false;

document.getElementById('guessInput').focus();

}

</script>

</body>

</html>