# Индивидуальное задание 1: функция min(a, b)

Для нахождения минимума двух чисел в JavaScript была реализована следующая функция:

function min(a, b) {  
 return (a < b) ? a : b;  
}

Функция использует тернарный оператор для сравнения значений и возвращает меньшее из них. Тестирование выполнялось в консоли браузера с различными парами чисел.

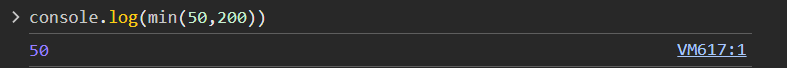
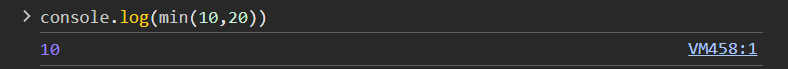
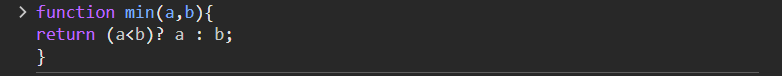


Рис. 1 - Функция и её тестирование в браузере

# Индивидуальное задание 2: переменные admin и name

Задача заключалась в объявлении двух переменных, присвоении имени и копировании значения:

let name = "Алексей";  
let admin = name;  
alert(admin);

После запуска скрипта в окне выводится имя «Алексей», что подтверждает корректное копирование значения.

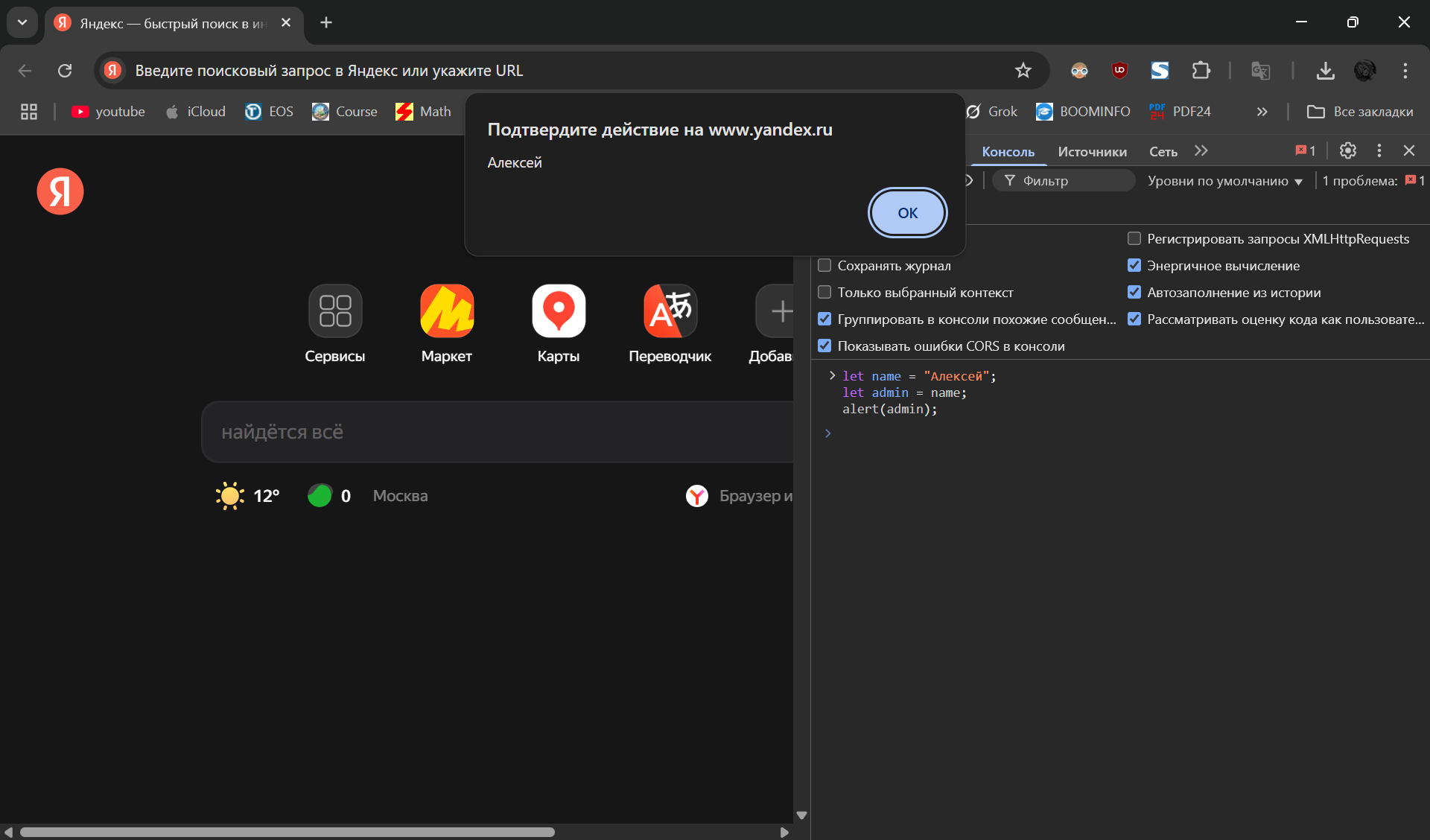


Рис. 2 - функция alert

# Индивидуальное задание 3: уточнение возраста

Реализован следующий сценарий взаимодействия с пользователем:

let age;  
while (true) {  
 age = prompt("Введите ваш возраст:");  
 if (age === null) continue;   
 if (confirm(`Ваш возраст ${age} лет?`)) {  
 alert("Спасибо за подтверждение!");  
 break;   
 }  
}

Скрипт запрашивает возраст, затем уточняет его при помощи confirm(). Если пользователь отменяет ввод, запрос повторяется.

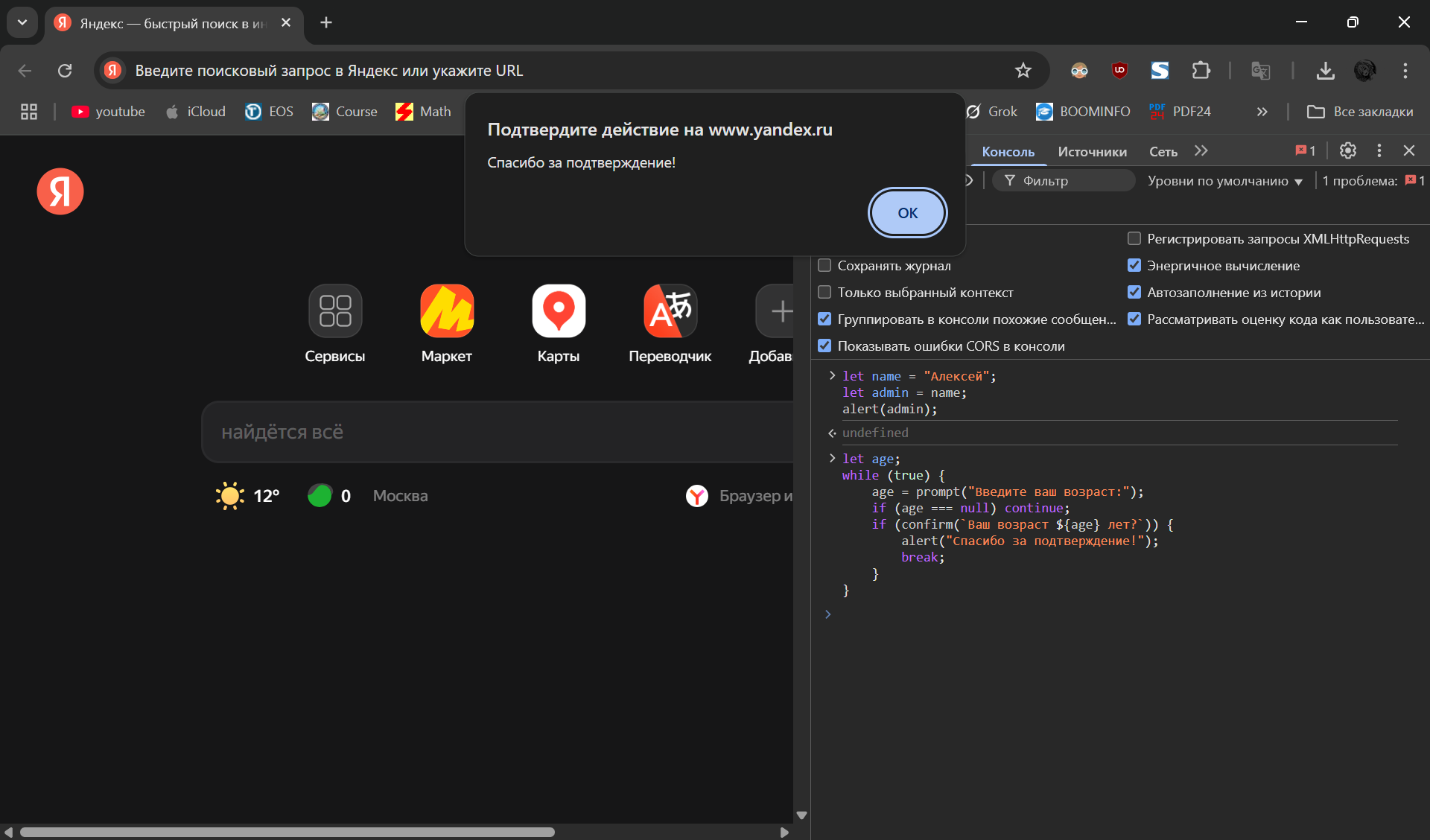
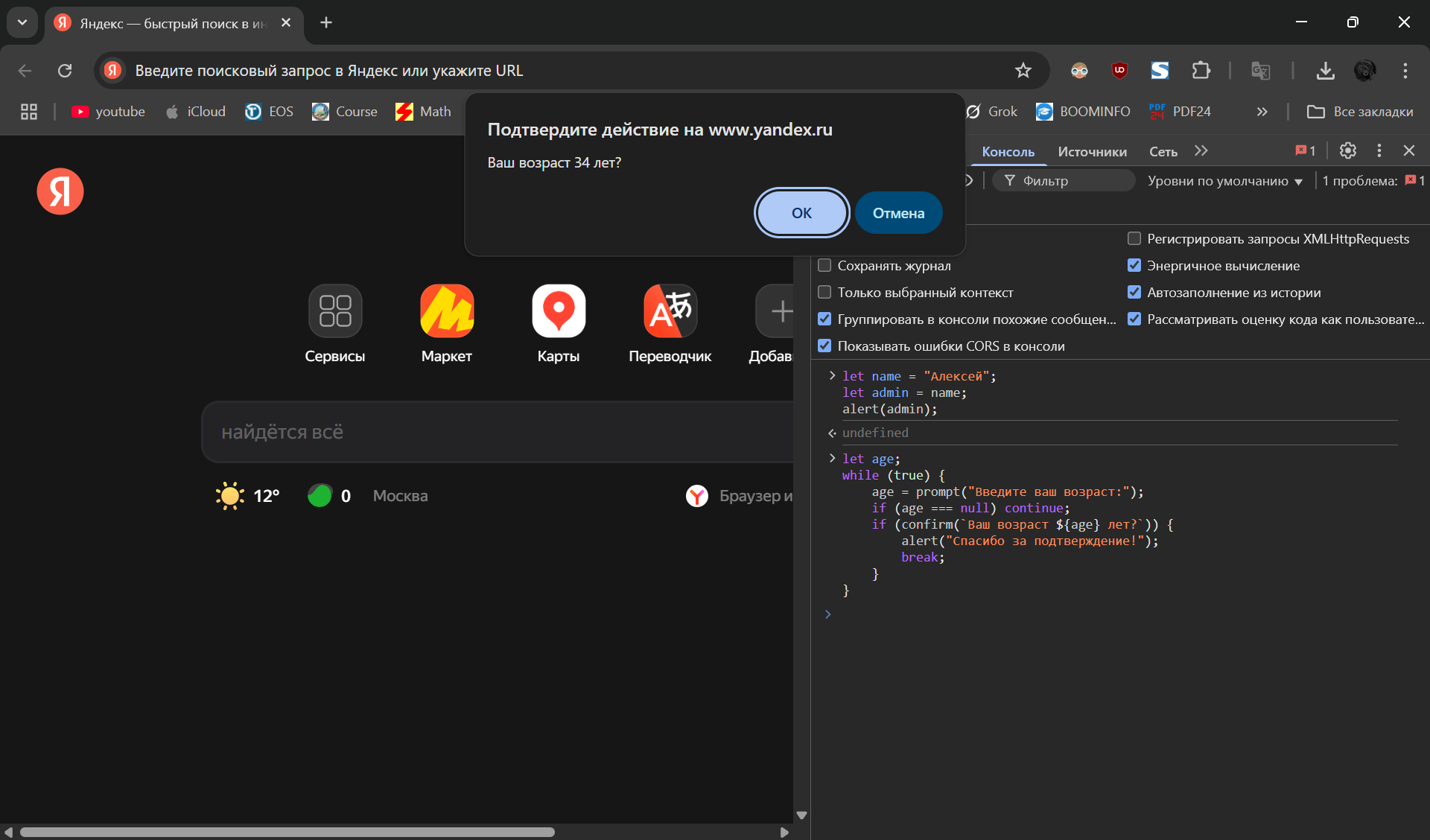
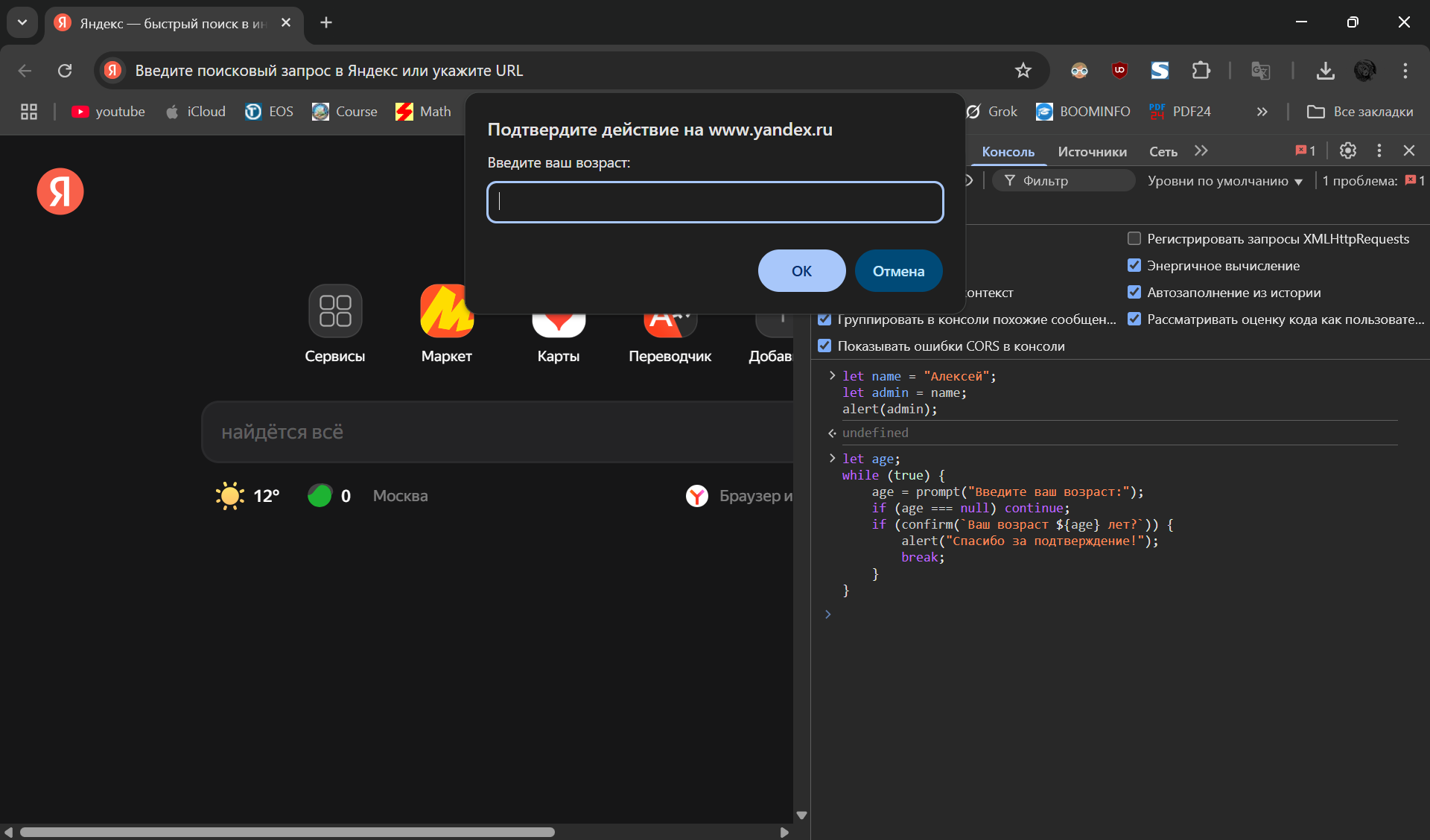
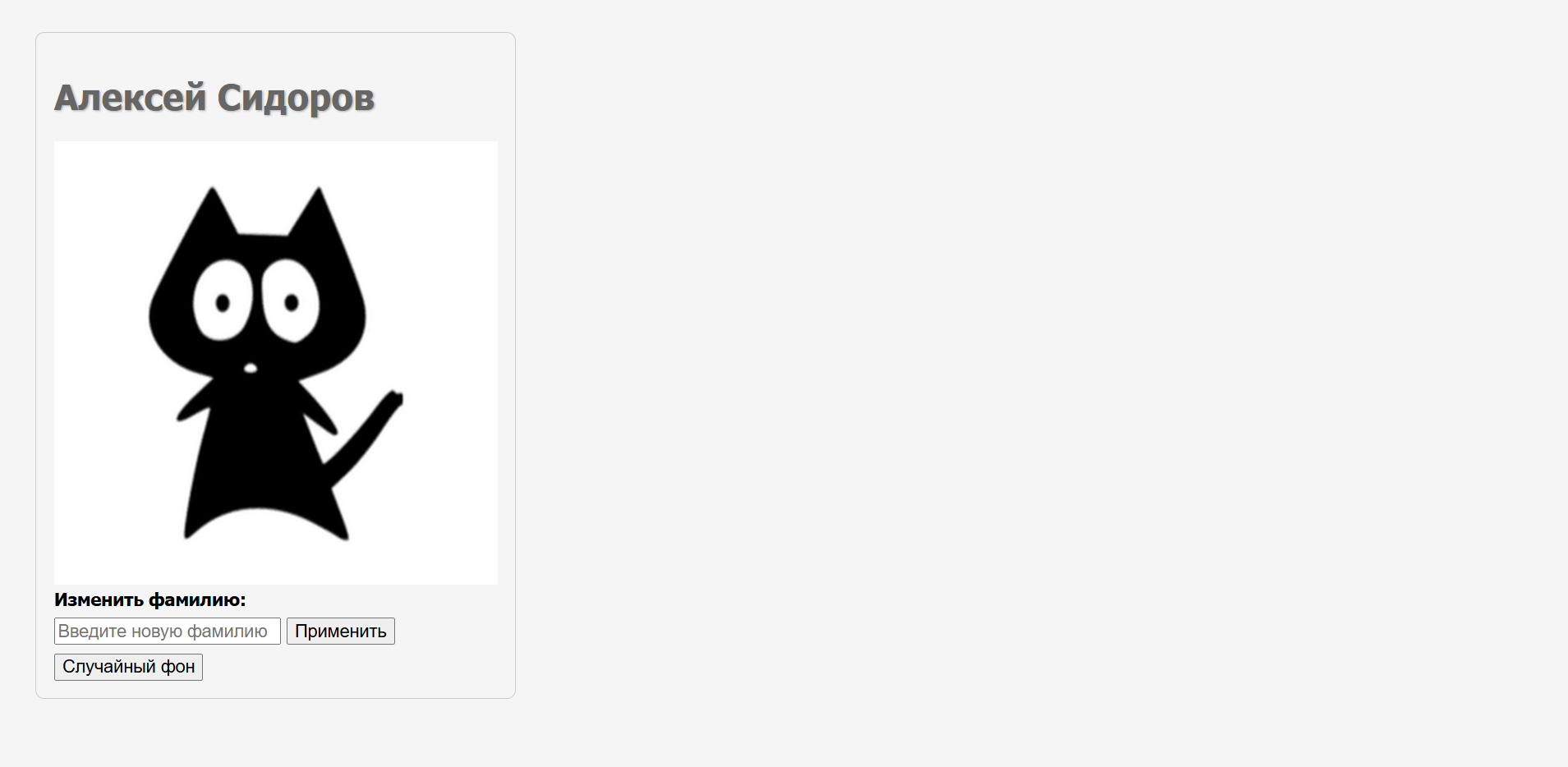


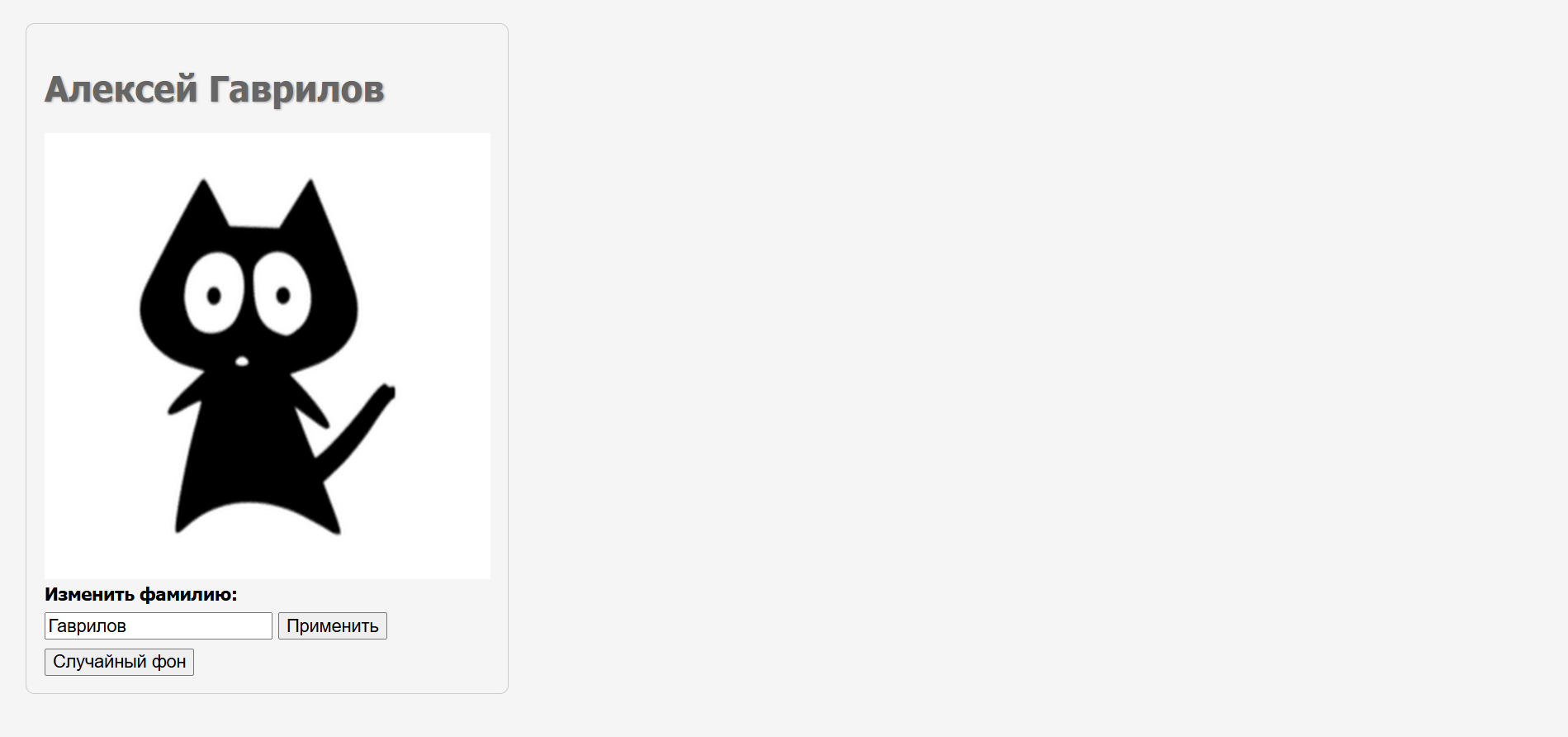
Рис. 3 - Сценарий взаимодействия с пользователем

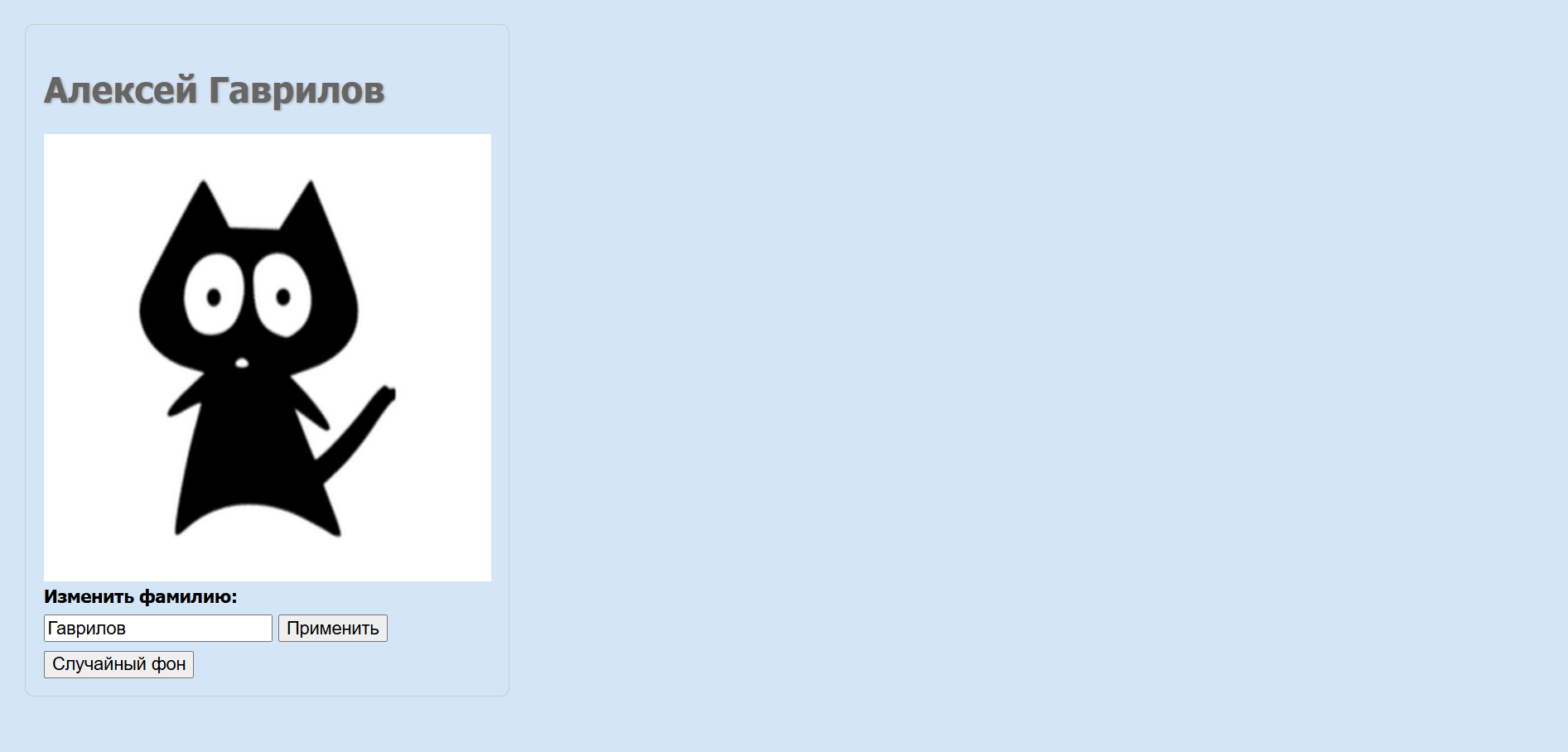
# Индивидуальное задание 4: стилизация страницы резюме

Создан файл index.html с базовой разметкой резюме, к которому добавлены CSS‑стили и интерактив на JavaScript. При наведении элементы окрашиваются, для фамилии предусмотрено динамическое изменение по тексту из поля ввода, а кнопка меняет фон страницы на случайный цвет.

<!DOCTYPE html>  
<html lang="ru">  
<head>  
<meta charset="UTF-8">  
<title>Резюме</title>  
<style>  
 body {  
 font-family: Tahoma, serif;  
 background: #f5f5f5;  
 margin: 0;  
 padding: 2rem;  
 transition: background 0.5s;  
 }  
 h1 { color: #666; text-shadow: 1px 1px 2px #aaa; }  
 .label { font-weight: bold; }  
 #surname:hover { color: crimson; cursor: pointer; }  
 button:hover { box-shadow: 0 0 10px rgba(0,0,0,0.3); }  
 input, button { font-size: 1rem; margin-top: 0.5rem; }  
 .card { border: 1px solid #ccc; padding: 1rem; border-radius: 8px; max-width: 400px;}  
</style>  
</head>  
<body>  
 <div class="card">  
 <h1>Иван <span id="surname">Иванов</span></h1>  
 <img src="photo.jpg" alt="Фото" width="100%">  
 <div>  
 <label class="label" for="newSurname">Изменить фамилию:</label><br>  
 <input id="newSurname" placeholder="Введите новую фамилию">  
 <button id="apply">Применить</button>  
 </div>  
 <button id="randBg">Случайный фон</button>  
 </div>  
  
<script>  
document.getElementById('apply').addEventListener('click', ()=>{  
 const s = document.getElementById('newSurname').value.trim();  
 if (s) document.getElementById('surname').textContent = s;  
});  
document.getElementById('randBg').addEventListener('click', ()=>{  
 document.body.style.background =  
 `hsl(${Math.random()\*360}, 70%, 90%)`;  
});  
</script>  
</body>  
</html>







Стили задают размеры шрифтов, тени, цвет текста, рамки и hover‑эффекты. JavaScript обрабатывает ввод пользователя и изменение цвета фона.

# Индивидуальное задание 6: адаптивная страница резюме

На основе требований была сгенерирована адаптивная одностраничная версия резюме. Для экономии места в отчёте приведён фрагмент, полный код размещён в репозитории GitHub и доступен по ссылке.

/\* Фрагмент CSS Grid‑разметки \*/  
.main-grid {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));  
 gap: 1.5rem;  
}  
/\* Анимации появления \*/  
.section {  
 opacity: 0;  
 transform: translateY(20px);  
 transition: all .6s ease;  
}  
.section.visible {  
 opacity: 1;  
 transform: translateY(0);  
}

Проверка адаптивности проводилась через инструменты разработчика Chrome (F12) — страница корректно отображается на ширинах от 320 px до 1440 px.