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) |

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.А. Соловьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| «Клиентские языки сценариев. Javascript» |
| по дисциплине: Web-технологии |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4236 |  |  |  | Л. Мвале |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

**Цель работы**: получение опыта написания и применения функций на языке javascript.

**1 Варианты задания**

Таблица 1 – Вариант задания

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Событие** | **Способ включения сценария** | **Поиск элемента** | **Оформление функции** | **Сценарий** |
| 21 | dblclick | файл | getElementsByName | с именем | 5 |

Задание для сценария:

Часы, которые показывают постоянно час и минуты. Секунды показываются после нажатия на кнопку.

**2 Выполнение задания**

Таблица 2 – Использованные функции

|  |  |  |
| --- | --- | --- |
| **Название функции** | **Событие, вызывающее функцию** | **Описание действия функции** |
| **showAuthorInfo** | dblclick на name="author-info" | Показывает всплывающее окно с именем автора и номером группы. |
| **updateClock** | setInterval (каждую секунду) | Обновляет отображение времени, показывая только часы и минуты. |
| **showSecondsForLimitedTime** | dblclick на name="clock-button" | Показывает секунды в часах на 10 секунд, затем скрывает их. |
| **addRow** | dblclick на name="add-row-btn" | Добавляет строку в таблицу с данными, введенными пользователем. |
| **eraseRow** | dblclick на name="erase-row-btn" | Удаляет последнюю добавленную строку в таблице. |
| **changeColor** | dblclick на name="change-color-btn" | Изменяет цвет фона второго <p> в разделе "История". |
| **changeSize** | dblclick на name="change-size-btn" | Увеличивает размер шрифта второго <p> в разделе "История". |
| **styleElement** | Вызывается внутри DOMContentLoaded | Применяет заданный цвет, размер и рамку к элементу <h2>. |
| **addItem** | dblclick на name="add-item" | Добавляет элемент в список с заголовком и описанием. |
| **removeItem** | dblclick на name="remove-item" | Удаляет последний элемент из списка. |
| **scroll** | scroll в окне браузера | Выводит в консоль сообщение при прокрутке страницы. |
| **mousemove** | mousemove в окне браузера | Отображает координаты мыши в консоли при её перемещении. |
| **Loading Animation for Links** | click на <a> в nav | Показывает "Loading..." перед переходом по ссылке (задержка 2 сек). |
| **Mouse Move Effect (circle)** | mousemove в окне браузера | Перемещает красный круг за курсором мыши. |

**3 Листинг функций на языке javascript**

document.addEventListener("DOMContentLoaded", () => {

    console.log("Page loaded successfully!"); // Browser event: load

    // 1️⃣ Show Author Info Popup

    const showAuthorInfo = () => alert("Автор: Лисон Мвале\nГруппа: 4236");

    // 2️⃣ Clock Function: Shows HH:MM, shows seconds for 10 sec on dblclick

    let clockElement = document.getElementsByName("clock-display")[0];

    let clockInterval, showSecondsInterval;

    const updateClock = (showSeconds = false) => {

        let now = new Date();

        let time = now.toLocaleTimeString("ru-RU", { hour: "2-digit", minute: "2-digit", second: showSeconds ? "2-digit" : undefined });

        if (clockElement) clockElement.innerText = time;

    };

    const showSecondsForLimitedTime = () => {

        clearInterval(clockInterval);

        updateClock(true);

        showSecondsInterval = setInterval(() => updateClock(true), 1000);

        setTimeout(() => {

            clearInterval(showSecondsInterval);

            updateClock(false);

            clockInterval = setInterval(updateClock, 1000);

        }, 10000);

    };

    clockInterval = setInterval(updateClock, 1000);

    document.getElementsByName("clock-button")[0]?.addEventListener("dblclick", showSecondsForLimitedTime);

    // 3️⃣ Add a row to the table with user input

    function addRow() {

        const industryInput = document.getElementsByName("industry")[0];

        const robotTypeInput = document.getElementsByName("robot-type")[0];

        const applicationsInput = document.getElementsByName("applications")[0];

        const tbodyRef = document.getElementsByTagName("tbody")[0];

        const newRow = tbodyRef.insertRow();

        // Add the "Added" class to dynamically added rows

        newRow.classList.add("Added");

        // Check if inputs are empty, if so add empty cells

        const industryValue = industryInput.value || " ";

        const robotTypeValue = robotTypeInput.value || " ";

        const applicationsValue = applicationsInput.value || " ";

        // Add data to the row

        const industryCell = newRow.insertCell(0);

        industryCell.innerText = industryValue;

        const robotTypeCell = newRow.insertCell(1);

        robotTypeCell.innerText = robotTypeValue;

        const applicationsCell = newRow.insertCell(2);

        applicationsCell.innerText = applicationsValue;

        // Clear the input fields after adding the row

        industryInput.value = "";

        robotTypeInput.value = "";

        applicationsInput.value = "";

    }

    // Function to erase the last dynamically added row

    function eraseRow() {

        const rows = document.getElementsByClassName("Added");

        if (rows.length > 0) {

            // Remove the last row added

            rows[rows.length - 1].remove();

        } else {

            console.log("No rows to delete.");

        }

    }

    // 5️⃣ Change color of the 2nd paragraph in "history"

    const changeColor = () => {

        let p = document.getElementById("history")?.getElementsByTagName("p")[1];

        if (p) p.style.backgroundColor = "#ffcccb";

    };

    // 6️⃣ Change size of the 2nd paragraph in "history"

    const changeSize = () => {

        let p = document.getElementById("history")?.getElementsByTagName("p")[1];

        if (p) p.style.fontSize = "1.2em";

    };

    // 7️⃣ Function with multiple parameters

    const styleElement = (element, color, size, border) => {

        if (!element) return;

        element.style.color = color;

        element.style.fontSize = size;

        element.style.border = border;

    };

    // Apply the function

    let heading = document.querySelector("h2");

    styleElement(heading, "blue", "22px");

    // 9️⃣ Browser Events

    window.addEventListener("scroll", () => console.log("Scrolling..."));

    window.addEventListener("mousemove", (e) => console.log(`Mouse moved: (${e.clientX}, ${e.clientY})`));

    // Add loading animation for navigation links

    // Loading animation for navigation links

    const navLinks = document.getElementsByTagName("nav")[0].getElementsByTagName("a");

    for (let link of navLinks) {

        link.addEventListener("click", (e) => {

            e.preventDefault(); // Prevent default navigation

            // Create a centered loading message

            const loadingDiv = document.createElement("div");

            loadingDiv.style.position = "fixed";

            loadingDiv.style.top = "50%";

            loadingDiv.style.left = "50%";

            loadingDiv.style.transform = "translate(-50%, -50%)";

            loadingDiv.style.fontSize = "2em";

            loadingDiv.style.color = "#3498db";

            loadingDiv.style.fontWeight = "bold";

            loadingDiv.innerText = "Loading...";

            // Add the loading message to the page

            document.body.innerHTML = ""; // Clear the page

            document.body.appendChild(loadingDiv);

            // Simulate a 2-second loading delay

            setTimeout(() => {

                window.location.href = link.href; // Navigate to the link

            }, 2000);

        });

    }

    // Scroll effect: Change background color based on scroll position

    window.addEventListener("scroll", () => {

        const scrollPosition = window.scrollY;

        const maxScroll = document.documentElement.scrollHeight - window.innerHeight;

        const scrollPercentage = (scrollPosition / maxScroll) \* 100;

        // Change background color based on scroll percentage

        //document.body.style.backgroundColor = `hsl(${scrollPercentage}, 50%, 50%)`;

    });

    // Mousemove effect: Move a circle with the mouse

    const circle = document.createElement("div");

    circle.style.width = "20px";

    circle.style.height = "20px";

    circle.style.backgroundColor = "red";

    circle.style.borderRadius = "50%";

    circle.style.position = "fixed";

    circle.style.pointerEvents = "none"; // Ensure the circle doesn't interfere with clicks

    document.body.appendChild(circle);

    window.addEventListener("mousemove", (e) => {

        circle.style.left = `${e.clientX - 25}px`; // Center the circle on the mouse

        circle.style.top = `${e.clientY - 25}px`;

    });

    // Add and Remove List Item

    const addItem = () => {

        const titleInput = document.getElementsByName("item-title")[0];

        const descriptionInput = document.getElementsByName("item-description")[0];

        const ul = document.getElementById("dynamic-list");

        // Validate inputs

        if (titleInput.value.trim() === "" || descriptionInput.value.trim() === "") {

            alert("Пожалуйста, заполните оба поля.");

            return; // Exit the function if inputs are empty

        }

        // Create a new list item

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

        li.style.padding = "10px";

        li.style.marginBottom = "10px";

        li.style.backgroundColor = "#f9f9f9";

        li.style.border = "1px solid #ddd";

        li.style.borderRadius = "4px";

        li.innerHTML = `

            <strong>${titleInput.value}</strong>: ${descriptionInput.value}

        `;

        // Add the new item to the list

        ul.appendChild(li);

        // Clear the input fields

        titleInput.value = "";

        descriptionInput.value = "";

    };

    const removeItem = () => {

        const ul = document.getElementById("dynamic-list");

        const lastItem = ul.lastElementChild;

        if (lastItem) {

            ul.removeChild(lastItem); // Remove the last item

        } else {

            alert("Нет элементов для удаления.");

        }

    };

     const removeButton = document.getElementsByName("remove-item")[0];

    // Attach double-click event listeners

    document.getElementsByName("add-item")[0]?.addEventListener("dblclick", addItem);

    document.getElementsByName("remove-item")[0]?.addEventListener("dblclick", removeItem);

    // Assign events

    document.getElementsByName("author-info")[0]?.addEventListener("dblclick", showAuthorInfo);

    document.getElementsByName("add-row")[0]?.addEventListener("dblclick", addEmptyRow);

    document.getElementsByName("change-color-btn")[0]?.addEventListener("dblclick", changeColor);

    document.getElementsByName("change-size-btn")[0]?.addEventListener("dblclick", changeSize);

    // Assign double-click events to buttons using getElementsByName

    const addButton = document.getElementsByName("add-row-btn")[0];

    if (addButton) {

        addButton.addEventListener("dblclick", addRow);

    }

    const eraseButton = document.getElementsByName("erase-row-btn")[0];

    if (eraseButton) {

        eraseButton.addEventListener("dblclick", eraseRow);

    }

});

**4 Скриншоты страниц сайта с демонстрацией работы функций**

Базовое задание:

1) Вызов всплывающего окна с ФИО и группой автора сайта

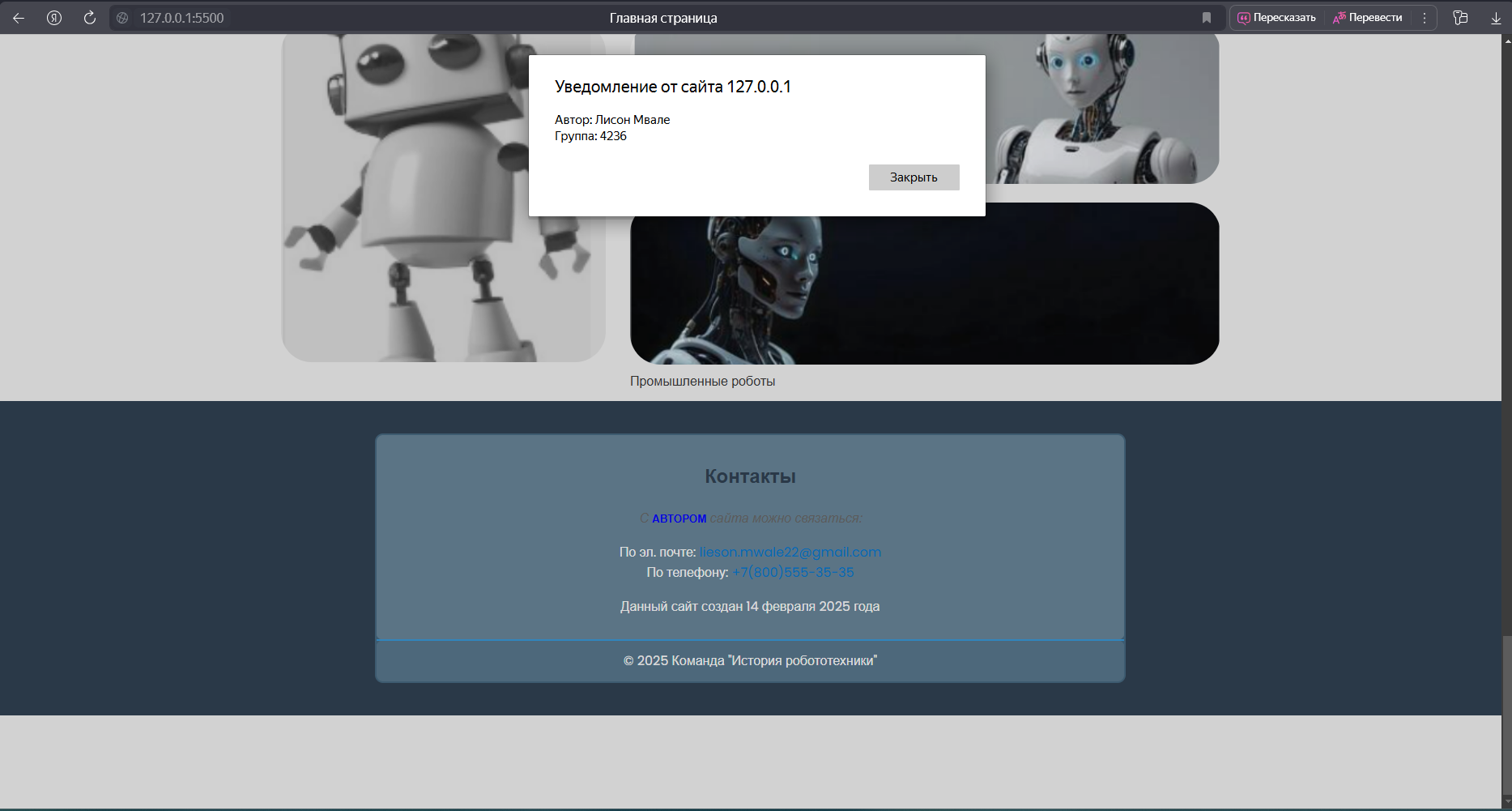


Рисунок 1 – Демонстрация работы функции showAuthor

2) Добавить сценарий по варианту из Таблицы 1. Часы, которые показывают постоянно час и минуты. Секунды показываются после нажатия на кнопку.

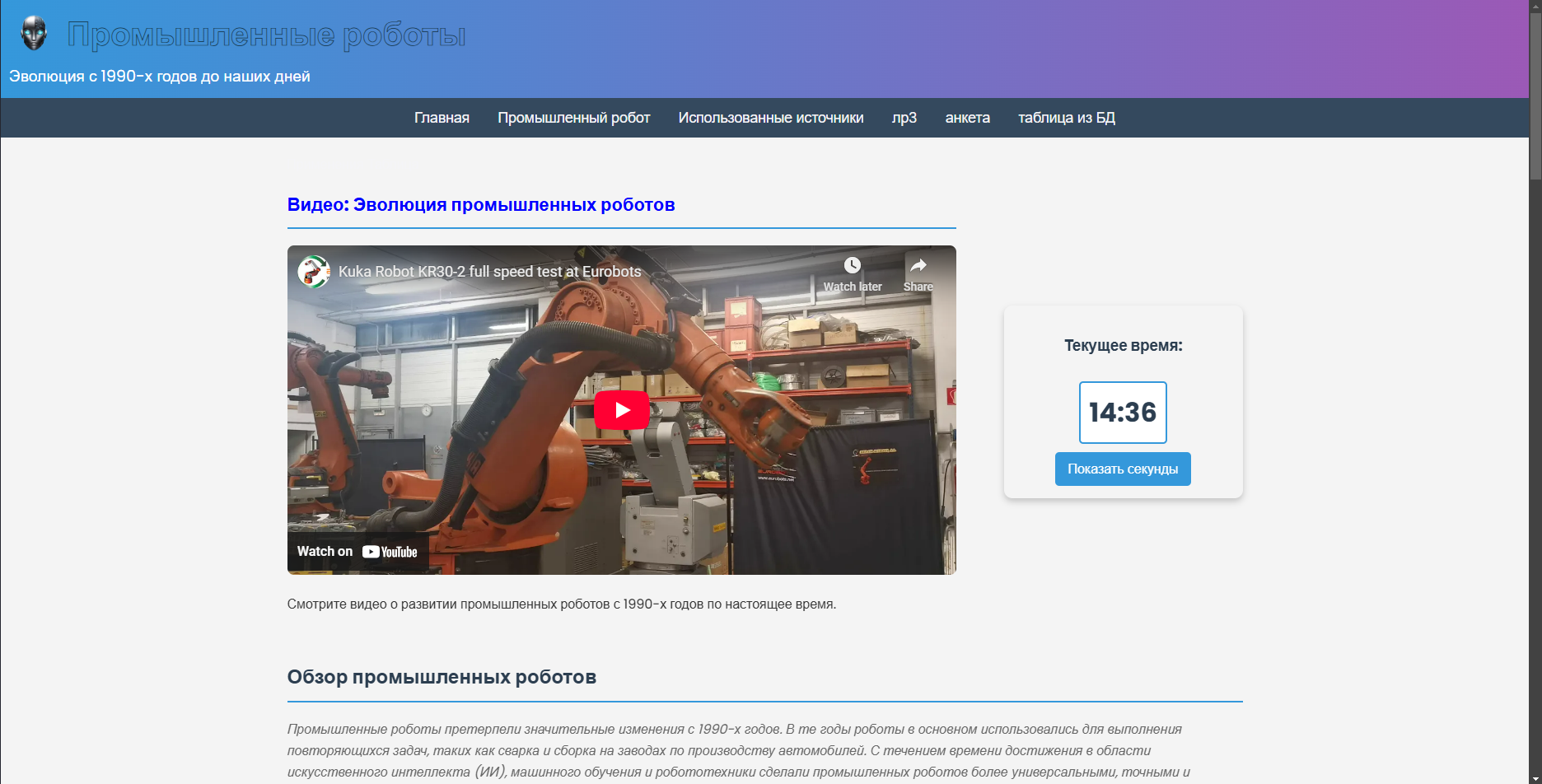


Рисунок 2- Демонстрация работы функции **updateClock**, время по Москве

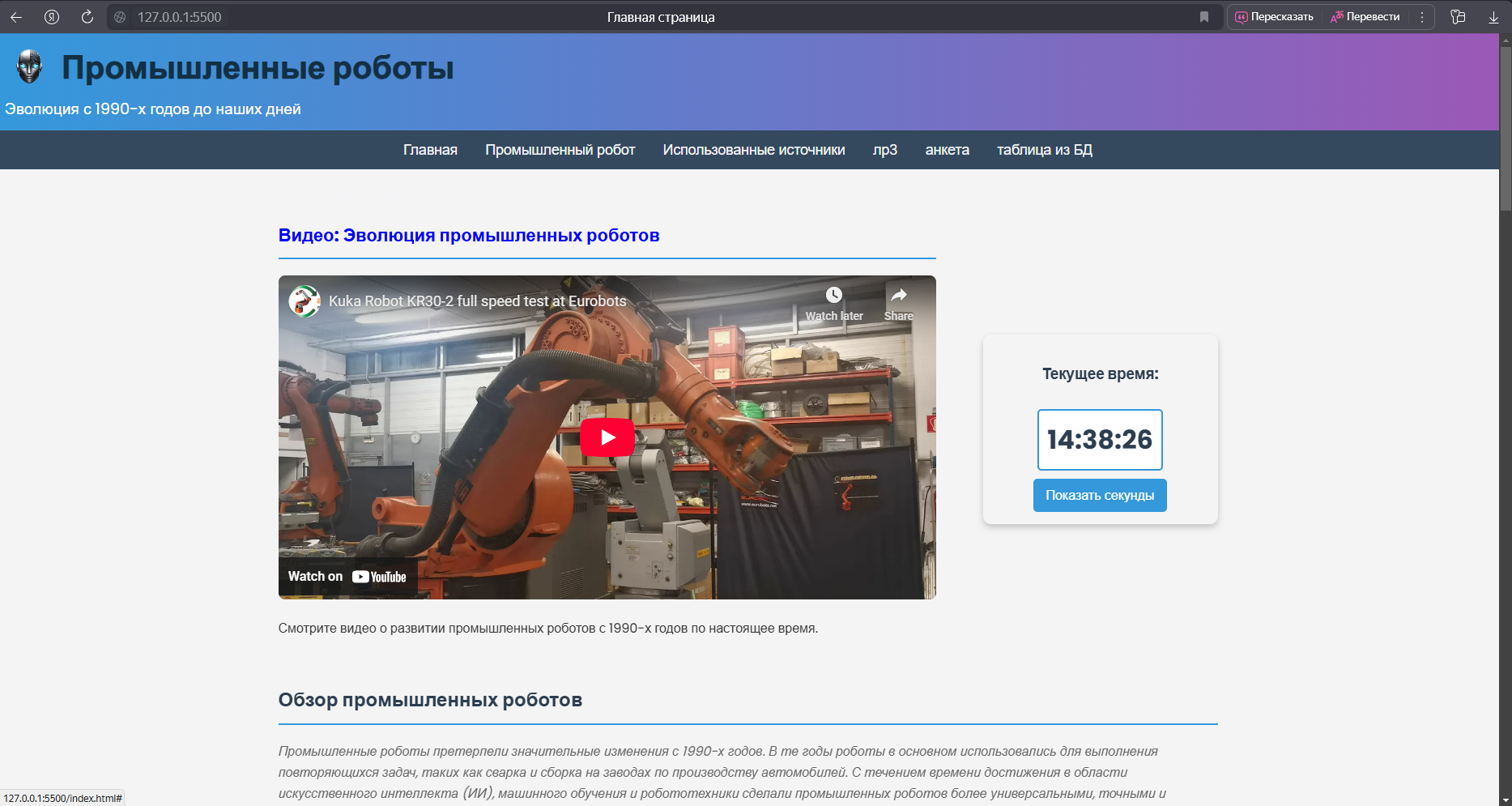


Рисунок 3- Демонстрация работы функции **showSecondsForLimitedTime**

3) Добавление пустой строки в таблицу, подготовленную в работе 1

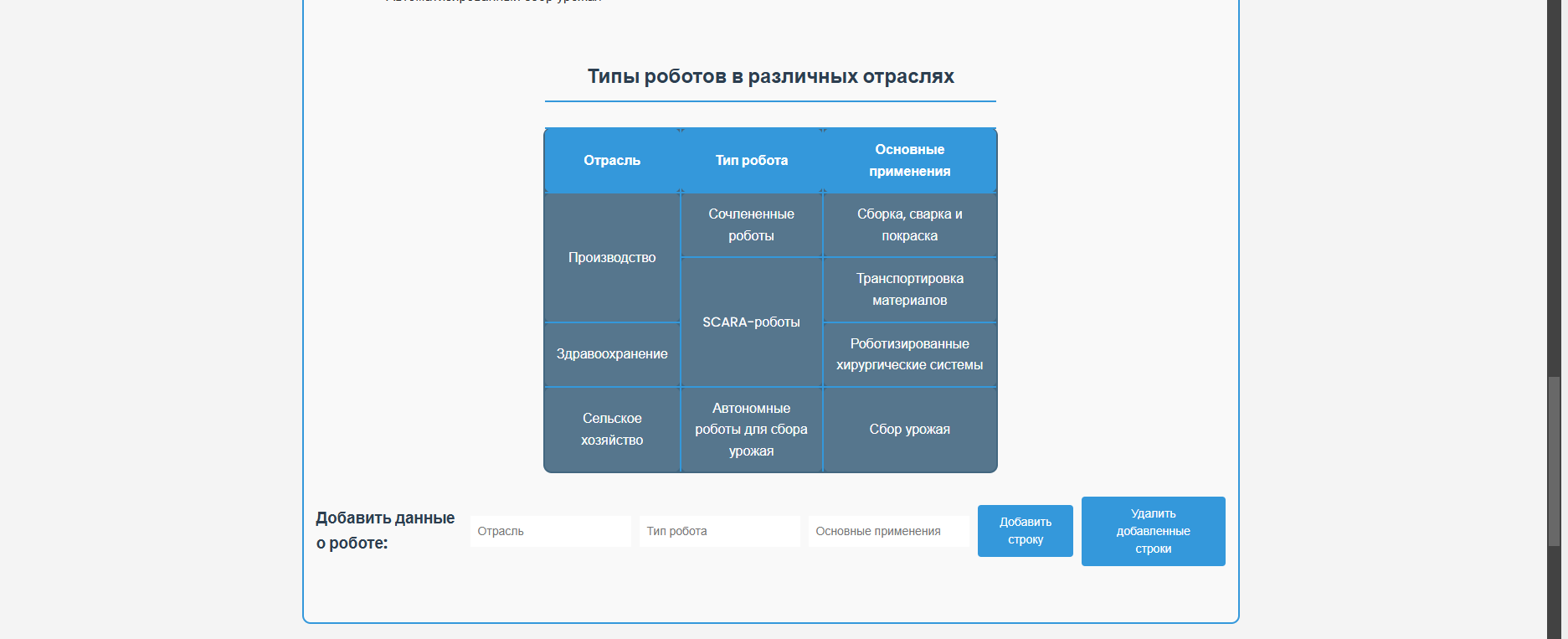


Рисунок 4 – Таблица до добавления новой строки



Рисунок 5 – Таблица после добавления строк с помощью функции addRow

4) Изменение цвета и размера элемента (применяет **changeColor**)

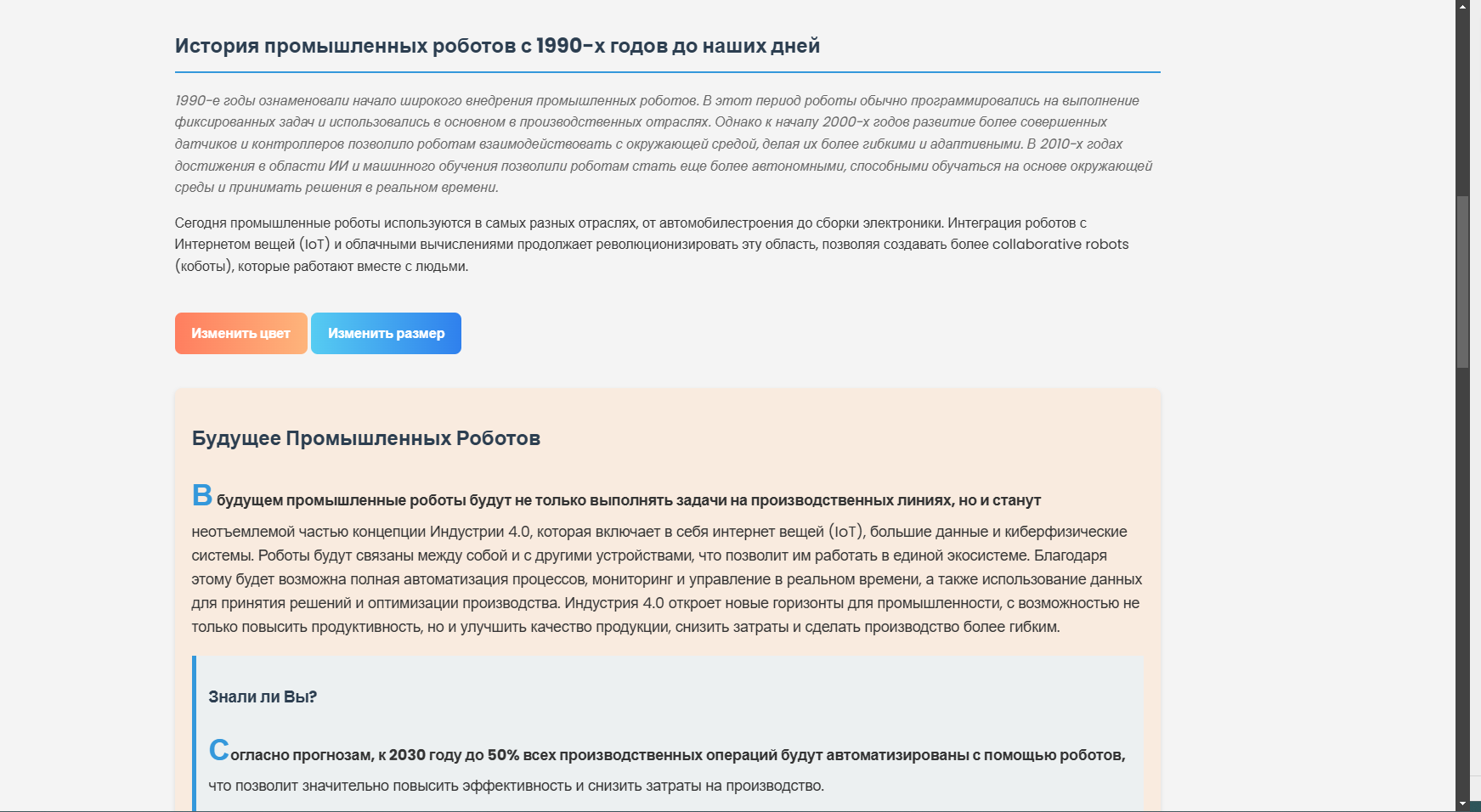


Рисунок 6 – Элемент абзаца перед применением функций изменить цвет и изменить размер

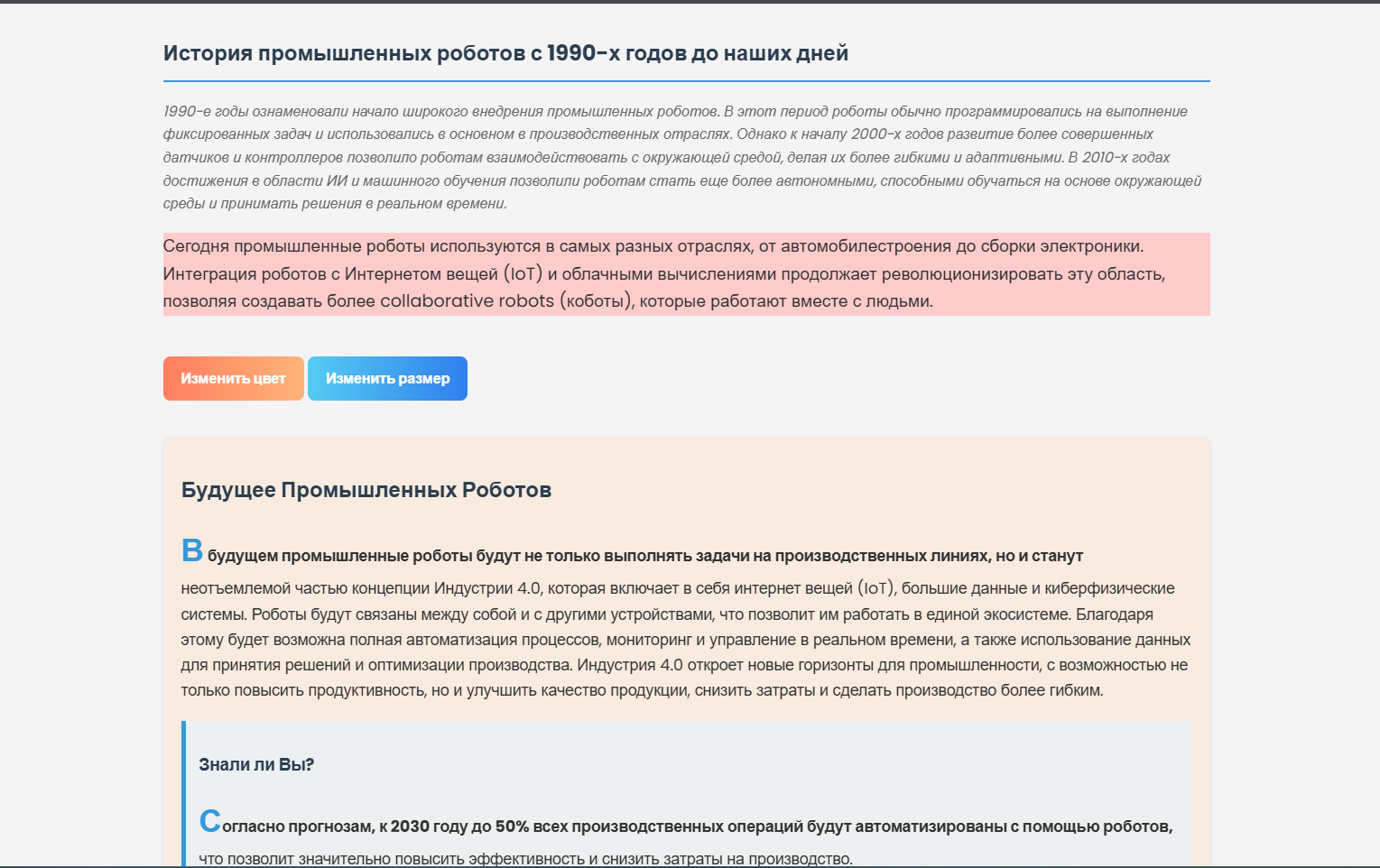


Рисунок 7 – Элемент абзаца после применения функции changeColor и changeSize

Расширенное задание:

1) При добавлении строки в таблицу для заполнения ячеек брать данные, введенные пользователем в элементы интерфейса(input)

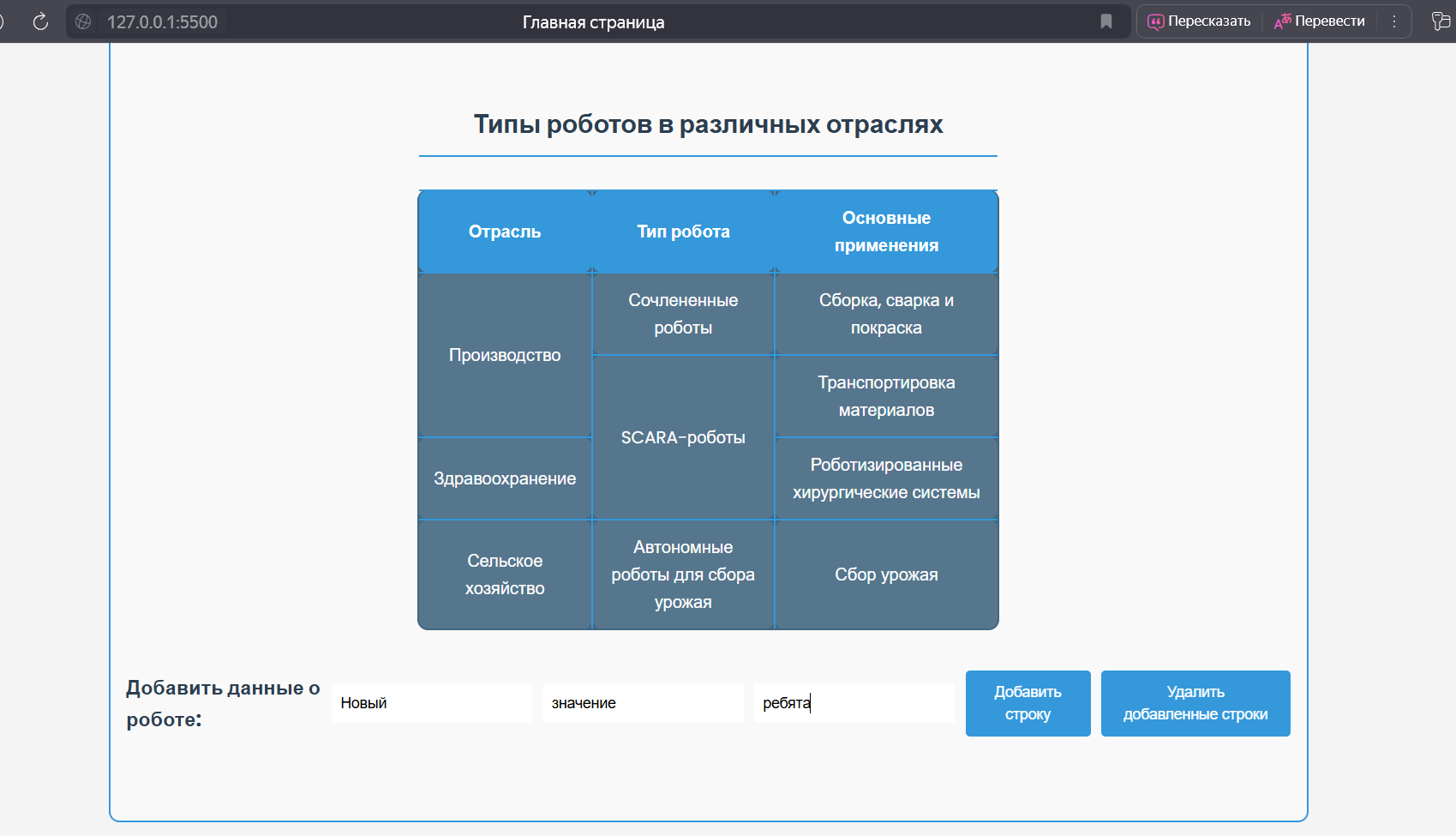


Рисунок 8 – Демонстрация при попытке добавить данные



Рисунок 9 – Демонстрация работы функции addRow с заданным



Рисунок 10 – Демонстрация работы функции addRow с другим заданным , часть 2

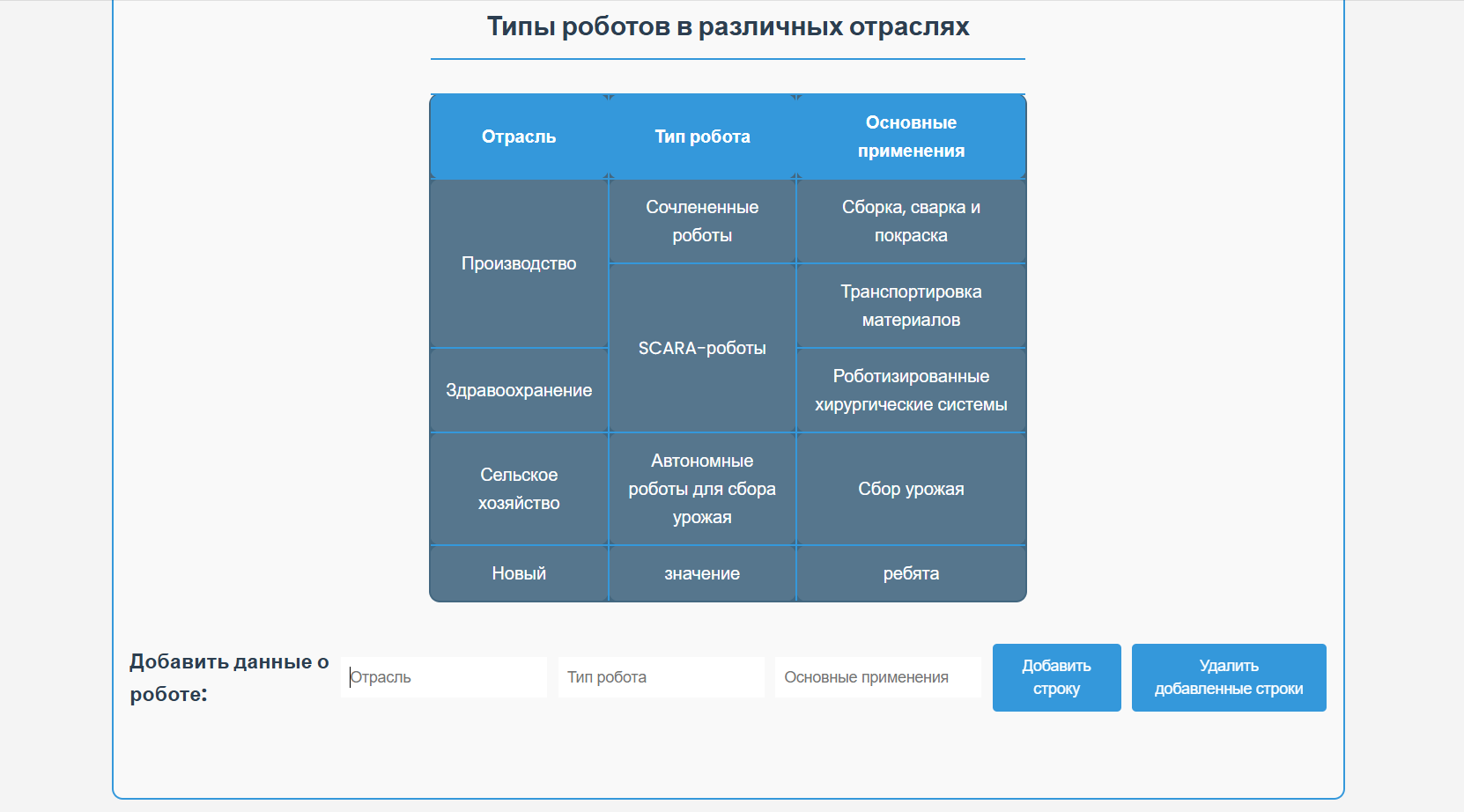


Рисунок 10 - Удалять только добавленные через сценарий строки таблицы

3)Функции со стрелками используются во всем коде, например:

const showAuthorInfo = () => alert("Автор: Лисон Мвале\nГруппа: 4236");

const updateClock = (showSeconds = false) => { ... };

const addListItem = () => { ... };

4) Использовать события окна браузера (загрузка, перемещение, скроллинг)

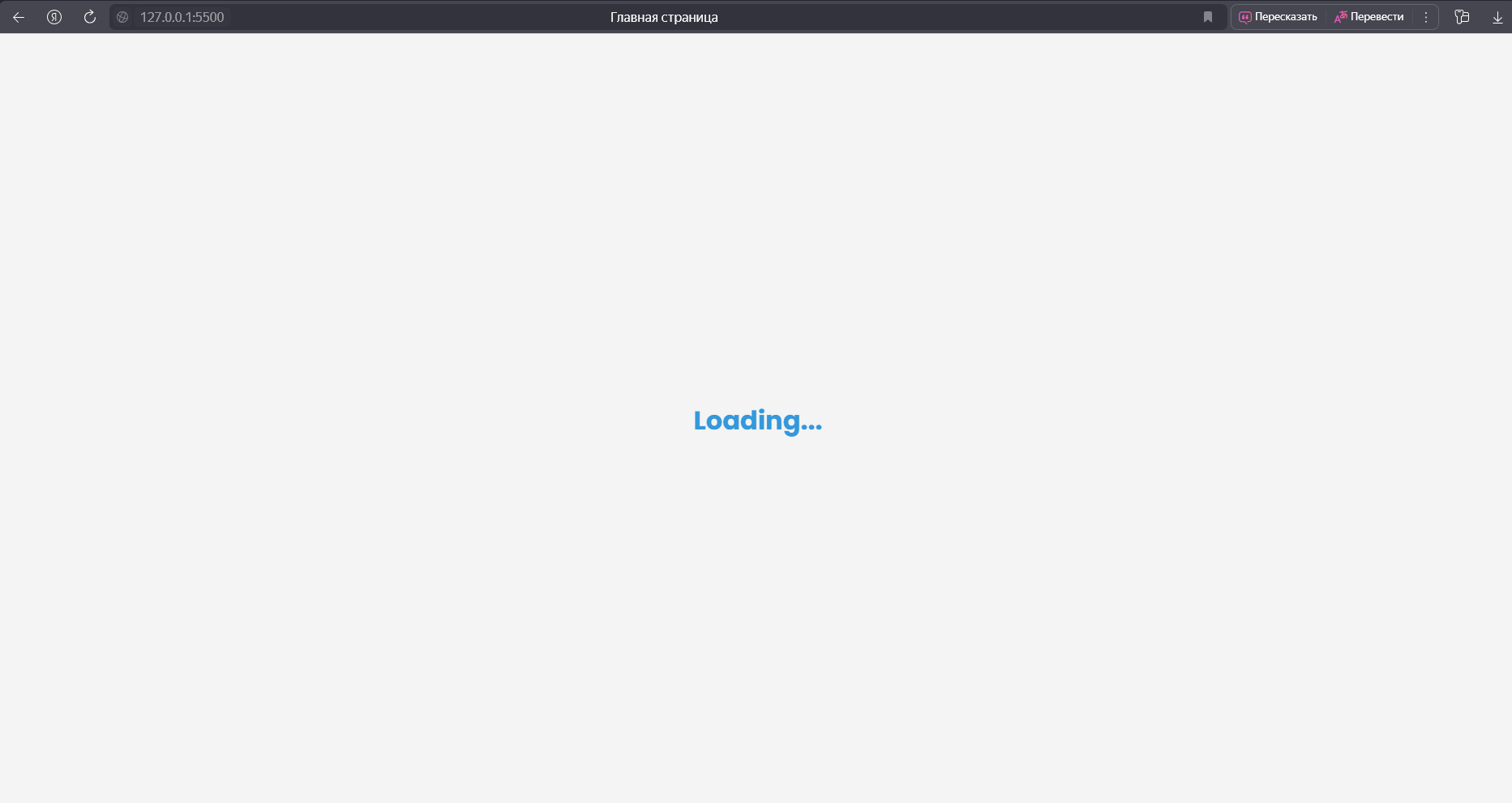


Рисунок 11 - При загрузке страницу

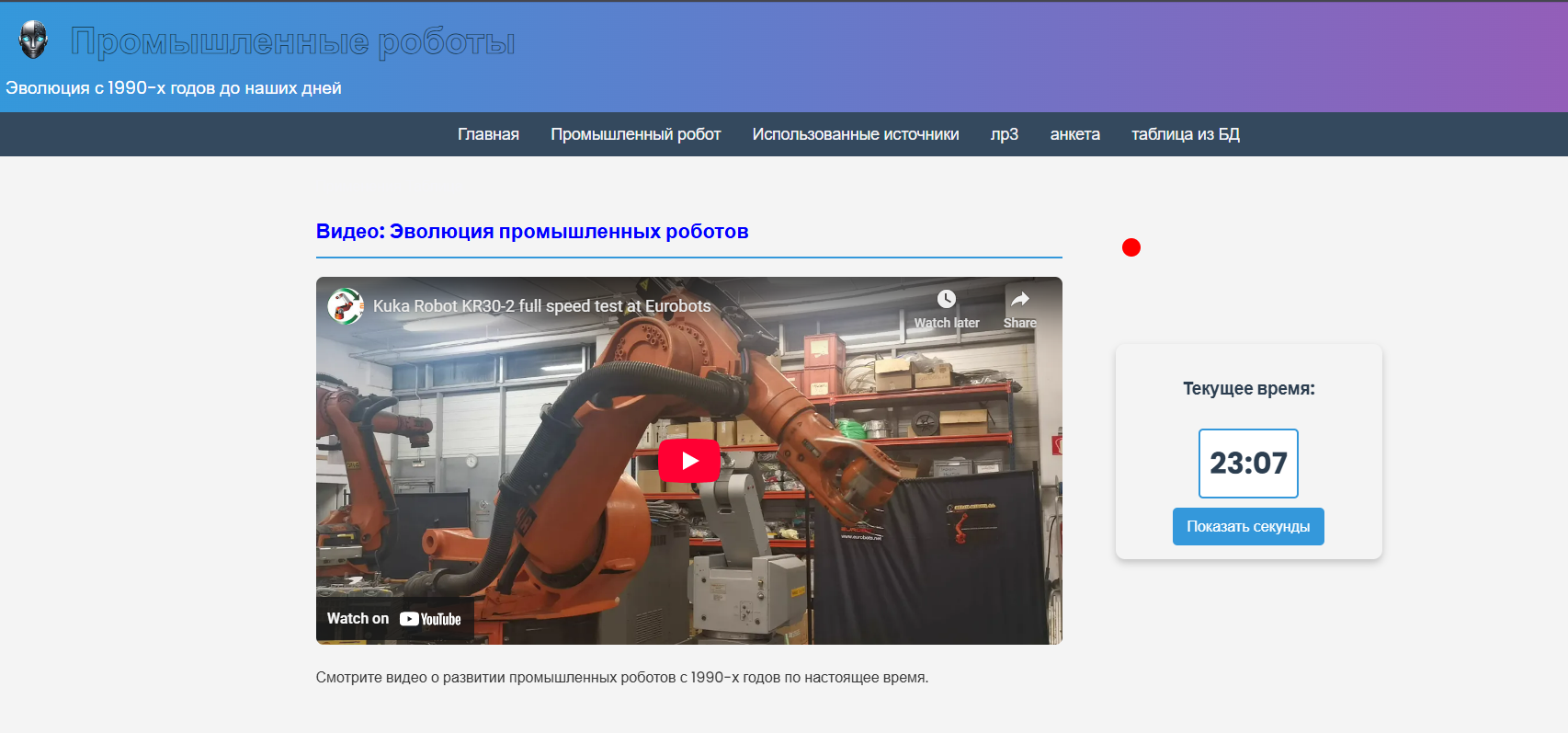


Рисунок 12 - Движение мыши обозначается красной точкой

5) Сделать функцию с несколькими параметрами и показать ее применение

const styleElement = (element, color, size, border) => {

        if (!element) return;

        element.style.color = color;

        element.style.fontSize = size;

        element.style.border = border;

    };

    // Apply the function

    let heading = document.querySelector("h2");

    styleElement(heading, "blue", "22px");

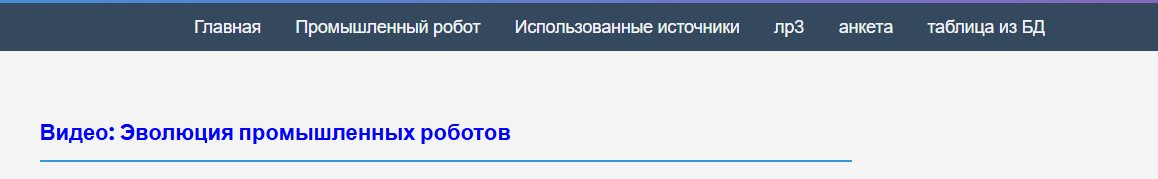


Рисунок 13 – Демонстрация работы функции styleElement

6) Использовать addEventListener

addEventListener использует добавления двух событий на один обработчик dblclick для кнопки изменения цвета и размера текстового элемента часов, представленных в базовом задании. ([рисунок 7](#рисунок7), [рисунок 8](#рисунок8))

7) Добавить и удалить элемент списка

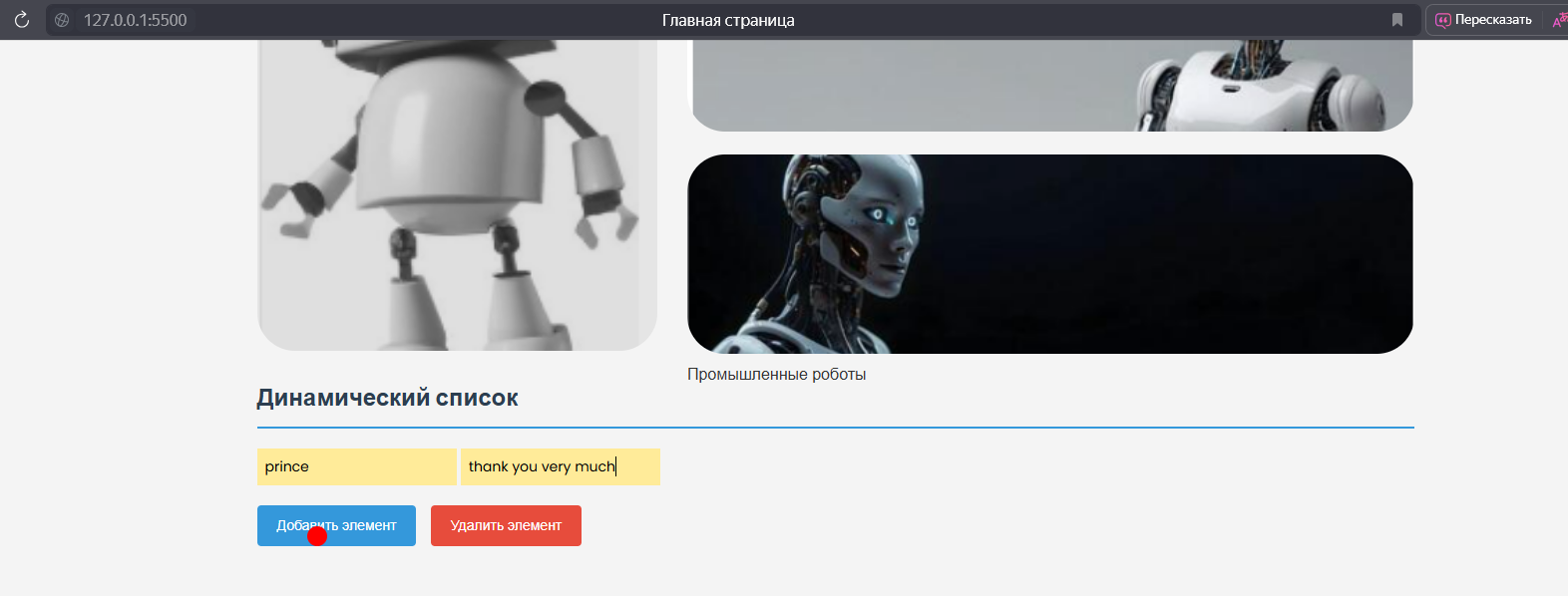


Рисунок 14 – Демонстрация работы функции addElemList

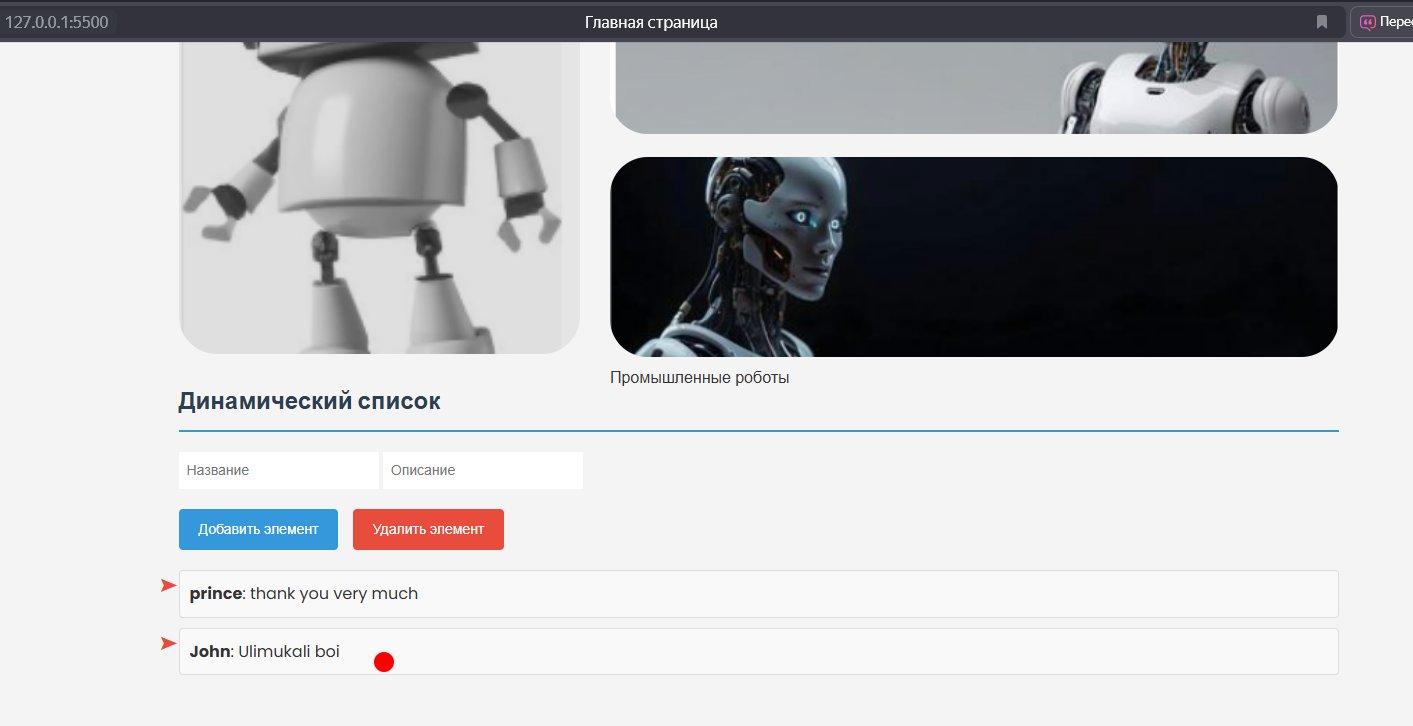


Рисунок 15 – Демонстрация работы функции addElemList

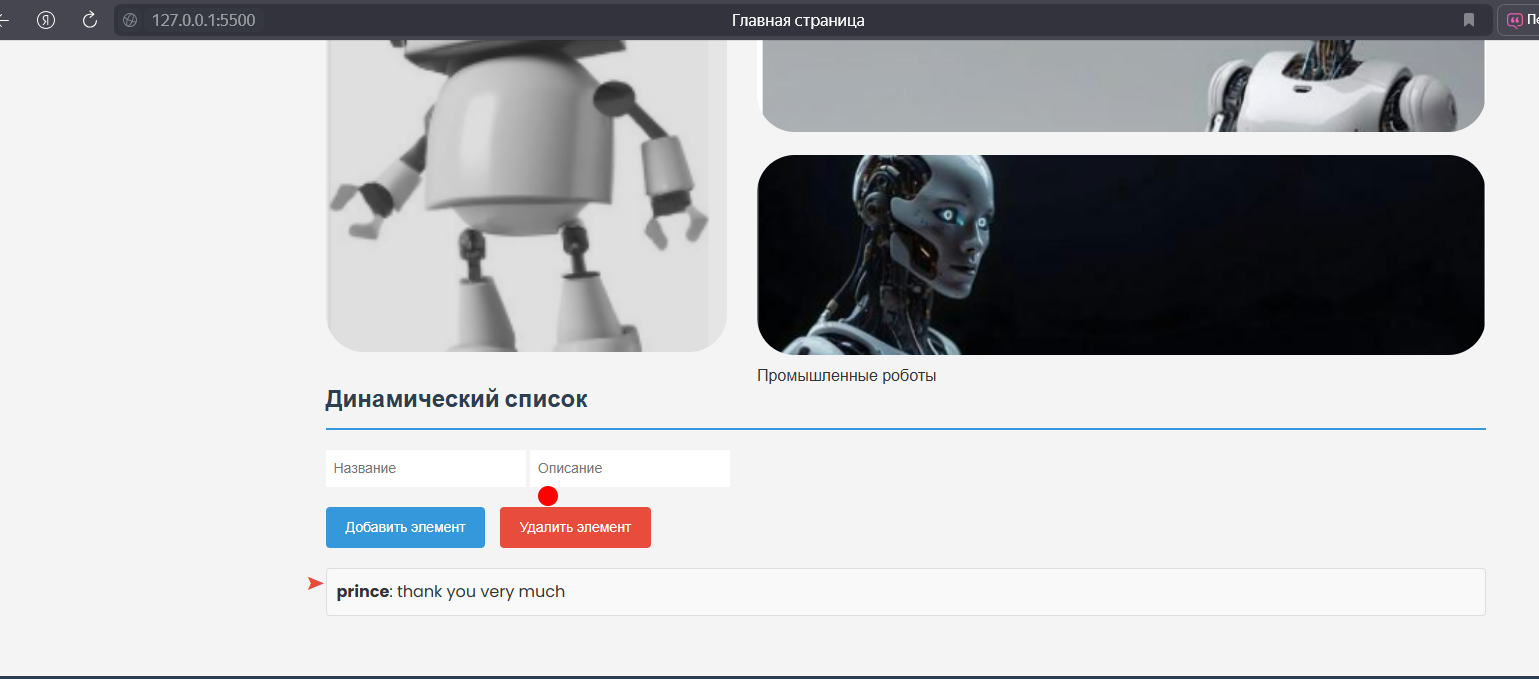


Рисунок 14 – Демонстрация работы функции eraseElemList

**5 Листинг HTML кода**

**Листинг index.html**

<!DOCTYPE html>

<html lang="ru">

    <!-- Использованы средства HTML5 -->

<head>

    <!-- Расширенное использование тега meta для указания кодировки и адаптивности -->

    <meta

        charset="UTF-8"

        name="viewport"

        content="width=device-width, initial-scale=1.0">

    <title>Главная страница</title>

    <!-- Подключение css-->

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

 <!-- box icons -->

 <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>

 <style>

    /\* Global CSS \*/

    .highlight {

        background-color: #e0f7fa; /\* Light blue background \*/

        border-color: #3498db; /\* Blue border \*/

    }

    .author-button {

        background-color: transparent !important;

        color: blue !important;

        font-weight: bold !important;

        border: none;

        outline: none;

        cursor: pointer;

        padding: 0;

        margin: 0;

    }

</style>

</head>

<body>

     <!-- google translator-->

     <div id="google\_translate\_element"></div>

    <!-- Шапка сайта с названием и логотипом -->

    <header>

        <a href="#" class="logo-container">

            <img src="images/logo.png" alt="Logo Image" class="logo-image">

            <span class="logo-text">Промышленные роботы</span>

        </a>

        <p>Эволюция с 1990-х годов до наших дней</p>

    </header>

    <!-- Горизонтальная линия для разделения контента -->

    <!-- Навигационная панель с ссылками на разделы сайта -->

    <nav align="center" >

        <a href="index.html">Главная</a>

        <a href="second.html">Промышленный робот</a>

        <a href="source.html">Использованные источники</a>

        <a href="#">лр3</a>

        <a href="#">анкета</a>

        <a href="#">таблица из БД</a>

    </nav>

    <!-- Основной контент страницы -->

    <main>

        <a href="#applications">Применения</a>

        <a href="#table-caption">Таблица</a>

       <!-- Секция с видео и часами -->

<section id="video-container">

    <div id="video">

        <h2>Видео: Эволюция промышленных роботов</h2>

        <iframe width="50%" height="400"

            src="https://www.youtube.com/embed/EZQLMLux\_FI"

            title="Видео про эволюцию промышленных роботов"

            frameborder="0"

            allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"

            allowfullscreen>

        </iframe>

        <p>Смотрите видео о развитии промышленных роботов с 1990-х годов по настоящее время.</p>

    </div>

    <!-- Часы -->

    <div id="clock-container">

        <h3>Текущее время:</h3>

        <div id="clock-display" name="clock-display">--:--</div>

        <button id="clock-button" name="clock-button">Показать секунды</button>

    </div>

</section>

        <!-- Секция с обзором промышленных роботов -->

        <section id="overview">

            <h2>Обзор промышленных роботов</h2>

            <p>Промышленные роботы претерпели значительные изменения с 1990-х годов. В те годы роботы в основном использовались для выполнения повторяющихся задач, таких как сварка и сборка на заводах по производству автомобилей. С течением времени достижения в области искусственного интеллекта (ИИ), машинного обучения и робототехники сделали промышленных роботов более универсальными, точными и способными выполнять сложные задачи.</p>

            <br>

        </section>

        <!-- Секция с историей промышленных роботов -->

        <section id="history">

            <h2>История промышленных роботов с 1990-х годов до наших дней</h2>

            <p>1990-е годы ознаменовали начало широкого внедрения промышленных роботов. В этот период роботы обычно программировались на выполнение фиксированных задач и использовались в основном в производственных отраслях. Однако к началу 2000-х годов развитие более совершенных датчиков и контроллеров позволило роботам взаимодействовать с окружающей средой, делая их более гибкими и адаптивными. В 2010-х годах достижения в области ИИ и машинного обучения позволили роботам стать еще более автономными, способными обучаться на основе окружающей среды и принимать решения в реальном времени.</p>

            <p>Сегодня промышленные роботы используются в самых разных отраслях, от автомобилестроения до сборки электроники. Интеграция роботов с Интернетом вещей (IoT) и облачными вычислениями продолжает революционизировать эту область, позволяя создавать более collaborative robots (коботы), которые работают вместе с людьми.</p>

            <br>

            <!-- Buttons for modifying <p> elements in the history section -->

                <button name="change-color-btn"

                style="background: linear-gradient(to right, #ff7e5f, #feb47b);

                       border: none;

                       color: white;

                       padding: 12px 20px;

                       font-size: 16px;

                       font-weight: bold;

                       border-radius: 8px;

                       cursor: pointer;

                       transition: transform 0.2s, box-shadow 0.2s;">

                Изменить цвет

            </button>

            <button name="change-size-btn"

                style="background: linear-gradient(to right, #56CCF2, #2F80ED);

                       border: none;

                       color: white;

                       padding: 12px 20px;

                       font-size: 16px;

                       font-weight: bold;

                       border-radius: 8px;

                       cursor: pointer;

                       transition: transform 0.2s, box-shadow 0.2s;">

                Изменить размер

            </button>

        </section>

        <article class="highlight" style="background-color: #f9ebdf; border-color: #e67e22;">

            <h1>Будущее Промышленных Роботов</h1>

            <p>

                В будущем промышленные роботы будут не только выполнять задачи на производственных линиях, но и станут неотъемлемой частью концепции

                Индустрии 4.0, которая включает в себя интернет вещей (IoT), большие данные и киберфизические системы.

                Роботы будут связаны между собой и с другими устройствами, что позволит им работать в единой экосистеме.

                Благодаря этому будет возможна полная автоматизация процессов, мониторинг и управление в реальном времени,

                а также использование данных для принятия решений и оптимизации производства.

                Индустрия 4.0 откроет новые горизонты для промышленности, с возможностью не только повысить продуктивность, но и улучшить качество

                продукции, снизить затраты и сделать производство более гибким.

            </p>

            <aside>

                <h3>Знали ли Вы?</h3>

                <p>

                    Согласно прогнозам, к 2030 году до 50% всех производственных операций будут автоматизированы с помощью роботов,

                    что позволит значительно повысить эффективность и снизить затраты на производство.

                </p>

            </aside>

        </article>

        <!-- Секция с примерами применения промышленных роботов -->

        <section id="applications">

            <h2>Применение промышленных роботов</h2>

            <p>Промышленные роботы используются в различных областях в разных отраслях. Ниже приведены некоторые из основных применений:</p>

            <ol>

                <li>Производство:

                    <ol>

                        <li>Работа на сборочной линии</li>

                        <li>Сварка и покраска</li>

                        <li>Упаковка и транспортировка материалов</li>

                    </ol>

                </li>

                <li>Здравоохранение:

                    <ol>

                        <li>Роботизированная хирургия</li>

                        <li>Сборка медицинских устройств</li>

                    </ol>

                </li>

                <li>Сельское хозяйство:

                    <ul>

                        <li>Точное земледелие</li>

                        <li>Автоматизированный сбор урожая</li>

                    </ul>

                </li>

            </ol>

            <!-- Таблица с объединенными строками -->

            <table border="1" width="60%" align="center">

                <caption id="table-caption"><h2>Типы роботов в различных отраслях</h2></caption>

                <thead>

                    <tr>

                        <th>Отрасль</th>

                        <th>Тип робота</th>

                        <th>Основные применения</th>

                    </tr>

                </thead>

                <tbody> <!-- Ensure tbody exists -->

                    <tr>

                        <td rowspan="2">Производство</td>

                        <td>Сочлененные роботы</td>

                        <td>Сборка, сварка и покраска</td>

                    </tr>

                    <tr>

                        <td rowspan="2">SCARA-роботы</td>

                        <td>Транспортировка материалов</td>

                    </tr>

                    <tr>

                        <td>Здравоохранение</td>

                        <td>Роботизированные хирургические системы</td>

                    </tr>

                    <tr>

                        <td>Сельское хозяйство</td>

                        <td>Автономные роботы для сбора урожая</td>

                        <td>Сбор урожая</td>

                    </tr>

                </tbody>

            </table>

            <!-- Input Section for Data -->

            <div id="input-section">

                <h3>Добавить данные о роботе:</h3>

                <input type="text" name="industry" placeholder="Отрасль">

                <input type="text" name="robot-type" placeholder="Тип робота">

                <input type="text" name="applications" placeholder="Основные применения">

                <button name="add-row-btn">Добавить строку</button>

                <button name="erase-row-btn">Удалить добавленные строки</button>

            </div>

            <!-- External delete button -->

            <!-- CSS for Styling the Input Section -->

            <style>

                #input-section {

                    margin: 20px 0;

                    display: flex;

                    gap: 10px;

                    justify-content: center;

                    align-items: center;

                }

                #input-section input {

                    padding: 8px;

                    font-size: 14px;

                    width: 200px;

                }

                #input-section button {

                    padding: 10px 20px;

                    font-size: 14px;

                    cursor: pointer;

                    background-color: #3498db;

                    color: white;

                    border: none;

                    border-radius: 4px;

                    transition: background-color 0.3s ease;

                }

                #input-section button:hover {

                    background-color: #2980b9;

                }

            </style>

            <br>

        </section>

        <!-- Секция с изображениями промышленных роботов -->

        <section>

            <h2 align="center">Изображения промышленных роботов</h2>

            <!-- Два ряда изображений, одно изображение занимает два ряда -->

            <div style="display: flex; flex-wrap: wrap; gap: 30px;">

                <picture style="flex: 2 1 5%; height: 400px;">

                    <img src="images/robot4.png" alt="Робот 4" style="height: 105%; width: 100%; object-fit: cover;">

                </picture>

                <div style="flex: 1 1 48%; height: 200px;">

                    <picture style="height: 100%;">

                        <img src="images/robot3.png" alt="Робот 2" style="height: 100%; width: 100%; object-fit: cover;">

                    </picture>

                    <hr/>

                    <picture style="height: 100%; margin-top: 10px;">

                        <img src="images/robot.png" alt="Робот 3" style="height: 100%; width: 100%; object-fit: cover;">

                    </picture>

                    <figcaption>Промышленные роботы</figcaption>

                </div>

            </div>

            <hr/>

        </section>

        <section>

            <h2>Динамический список</h2>

            <div style="margin-bottom: 20px;">

                <input type="text" name="item-title" placeholder="Название" style="padding: 8px; font-size: 14px; width: 200px;">

                <input type="text" name="item-description" placeholder="Описание" style="padding: 8px; font-size: 14px; width: 200px;">

            </div>

            <div style="margin-bottom: 20px;">

                <button name="add-item" style="padding: 10px 20px; font-size: 14px; cursor: pointer; background-color: #3498db; color: white; border: none; border-radius: 4px; margin-right: 10px;">Добавить элемент</button>

                <button name="remove-item" style="padding: 10px 20px; font-size: 14px; cursor: pointer; background-color: #e74c3c; color: white; border: none; border-radius: 4px;">Удалить элемент</button>

            </div>

            <ul id="dynamic-list" style="list-style-type: none; padding: 0;">

                <!-- List items will be added here -->

            </ul>

        </section>

    </main>

    </div></div>

    <!-- Подвал сайта с контактной информацией -->

    <footer>

        <table width="100%" border="0">

            <tr>

                <td>

                    <address>

                        <h2>Контакты</h2>

                        <p>С

                            <button class="author-button" name="author-info">АВТОРОМ</button> сайта можно связаться:</p>

                        <address>

                            По эл. почте: <a href="mailto:lieson.mwale22.ru">lieson.mwale22@gmail.com</a><br />

                            По телефону: <a href="tel:+78005553535">+7(800)555-35-35</a>

                        </address>

                        <p>Данный сайт создан 14 февраля 2025 года</p>

                    </address>

                </td>

            </tr>

            <tr>

                <td>&copy; 2025 Команда "История робототехники"</td>

            </tr>

        </table>

    </footer>

</body>

<script src="script.js" defer></script>

</html>

**Вывод**

В ходе выполнения лабораторной работы, были получены навыки по написанию функций на языке javascript, а так же привязывания их к обработчикам событий на веб-странице.