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 №7 |
| «Асинхронное клиент-серверное взаимодействие  с использованием Ajax и JSON » |
| по дисциплине: Web-технологии |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2023

**СОДЕРЖАНИЕ**

1 Варианты задания 3

2 Адаптированная формулировка задания………………………………3

3 Структура таблиц в БД 3

4 Скриншоты страниц сайта и таблиц базы данных 5

5 Код javascript…………………………………………………………….7

6 Php код 9

5 HTML код 10

Вывод 12

**Цель работы:** иизучение и применение на практике технологии AJAX

**1 Варианты задания**

Таблица 1 – Тема сайта

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Тема** | **Тип запроса** |
| 21 | Промышленные роботы | Post |

Модификация данных без перезагрузки страницы: при нажатии кнопки «сохранить», запись сохраняется в базе, но перезагрузка страницы не выполняется.

**2 Адаптированная формулировка задания**

**Цель**: Разработать AJAX-приложение для управления базой данных промышленных роботов и их производителей.

**Задачи:**

- Реализовать асинхронную загрузку данных (роботы и производители) без перезагрузки страницы.

- Использовать JSON для обмена данными между клиентом и сервером.

- Обеспечить работу в разных браузерах с выводом информации о браузере и поддержке AJAX.

- Применить jQuery для одного из вариантов AJAX-запросов.

- Использовать метод POST для всех изменяющих запросов (добавление, редактирование, удаление).

**3 Структура таблиц БД**

Таблица robots (Роботы)

| Поле | Тип данных | Описание |
| --- | --- | --- |
| id | INT | Первичный ключ (автоинкремент) |
| model\_name | VARCHAR | Название модели робота |
| type | VARCHAR | Тип робота |
| weight | FLOAT | Вес (кг) |
| payload | FLOAT | Грузоподъемность (кг) |
| manufacturer\_id | INT | Внешний ключ (производитель) |

Таблица manufacturers (Производители)

| Поле | Тип данных | Описание |
| --- | --- | --- |
| id | INT | Первичный ключ (автоинкремент) |
| name | VARCHAR | Название компании |
| country | VARCHAR | Страна производителя |
| founded\_year | INT | Год основания |
| website | VARCHAR | Веб-сайт |
| rating | FLOAT | Рейтинг (0-10) |

CREATE DATABASE IF NOT EXISTS robotics;

USE robotics;

CREATE TABLE manufacturers (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50),

country VARCHAR(50),

founded\_year INT,

website VARCHAR(100),

rating FLOAT

);

CREATE TABLE robots (

id INT AUTO\_INCREMENT PRIMARY KEY,

model\_name VARCHAR(50),

type VARCHAR(30),

weight FLOAT,

payload FLOAT,

manufacturer\_id INT,

FOREIGN KEY (manufacturer\_id) REFERENCES manufacturers(id)

);

**4 скриншоты страниц сайта и таблиц базы данных**

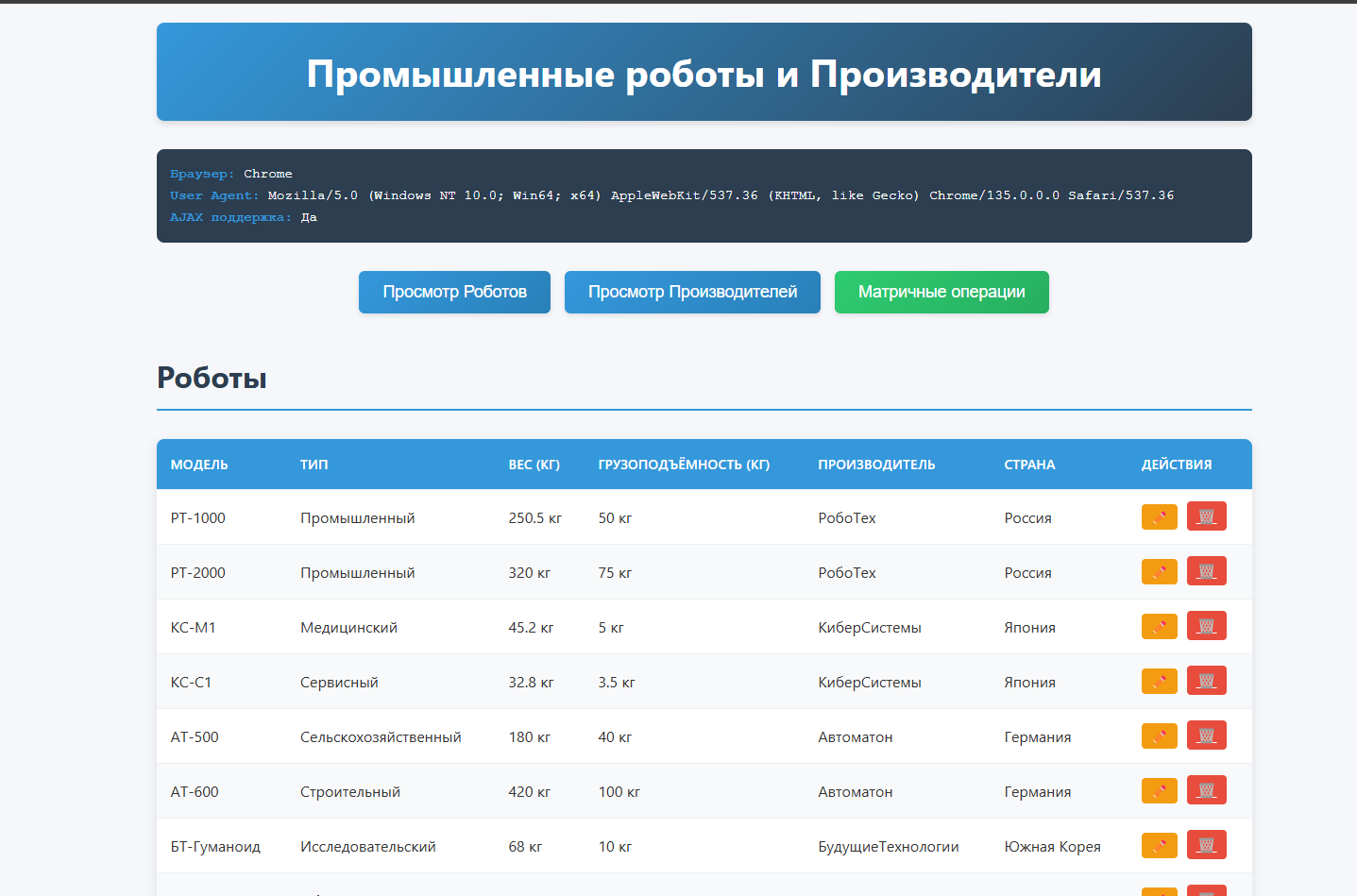


Рисунок 1 – Главная страницаp, часть роботов с таблицей базы данных

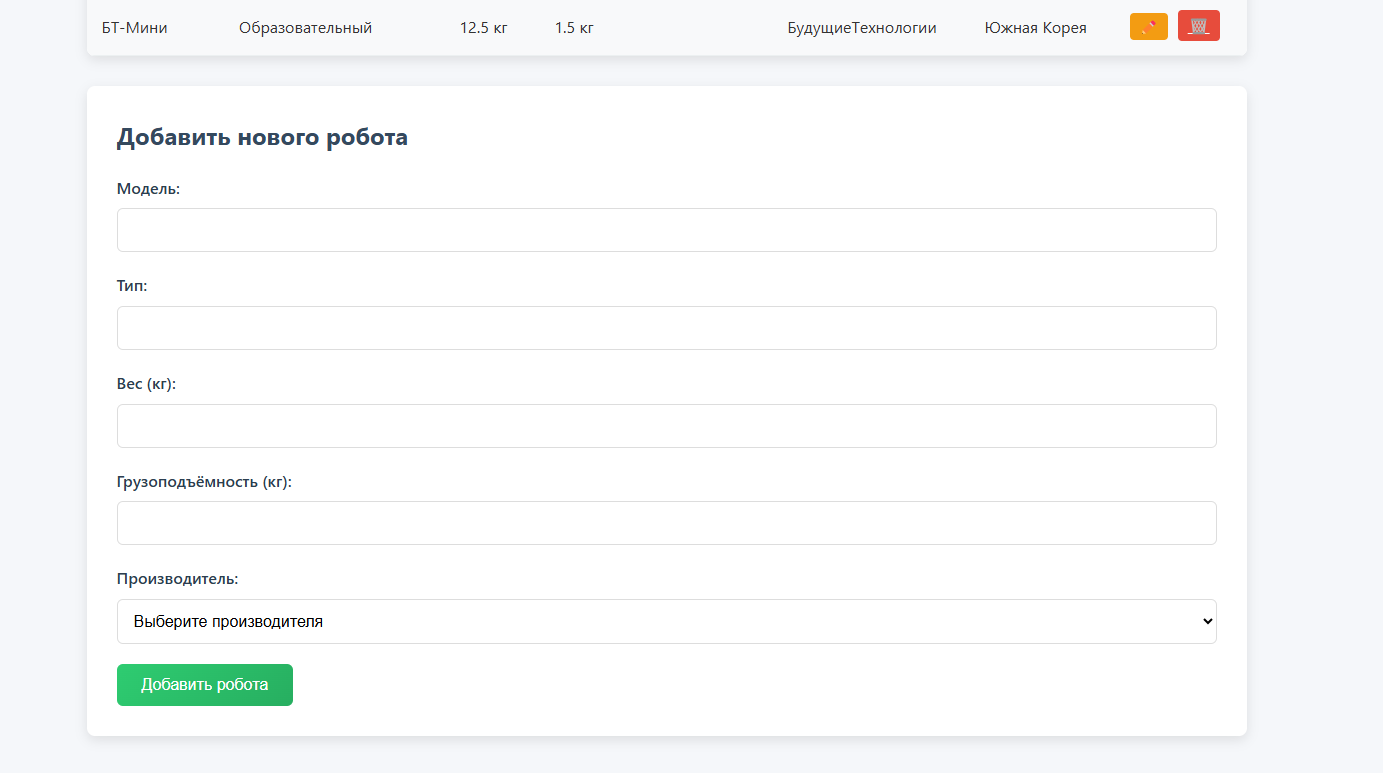


Рисунок 2 – Главная страница часть 2 “добавление данный”

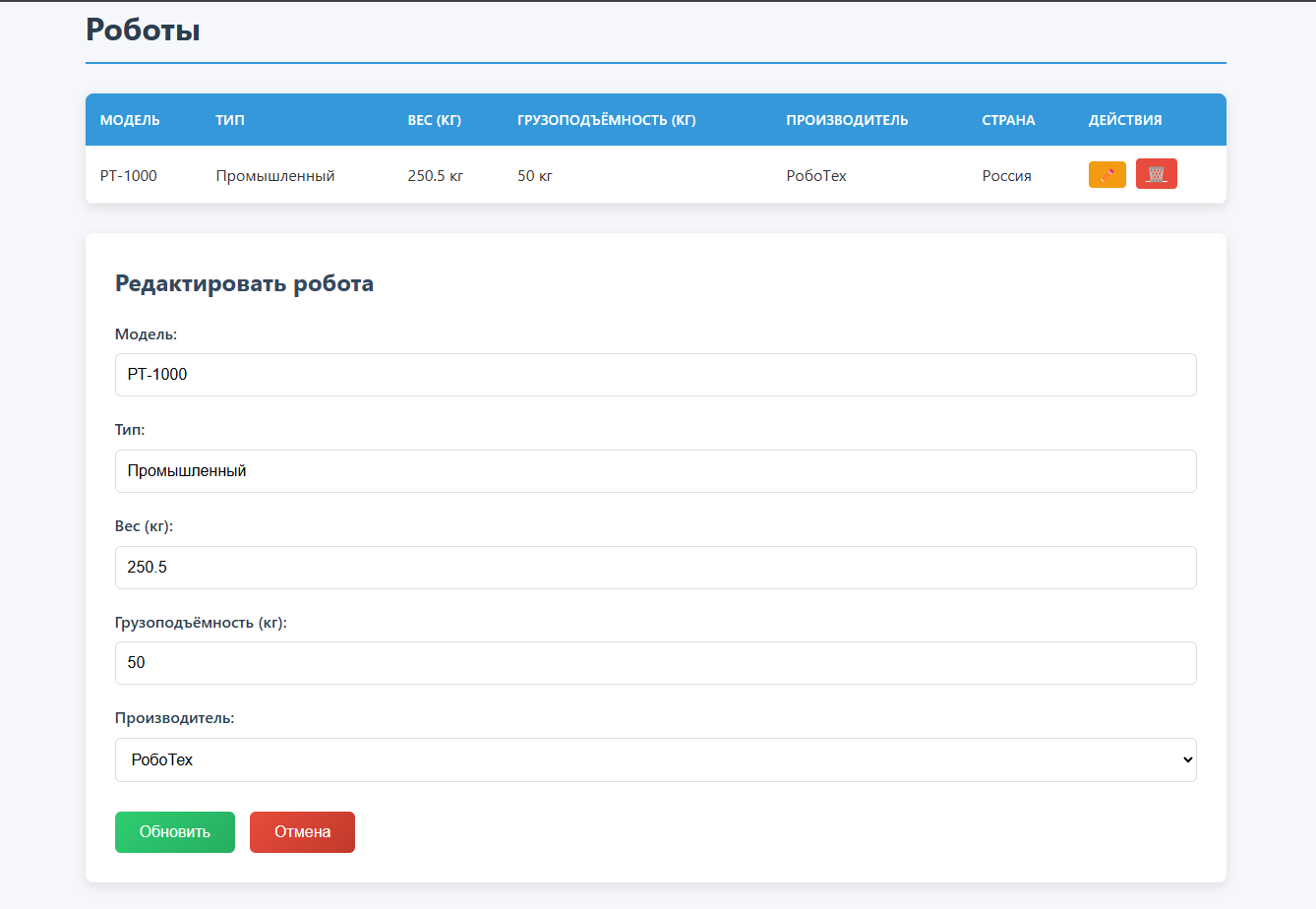


Рисунок 3 – Главная страница часть 3 “редактирование данный”

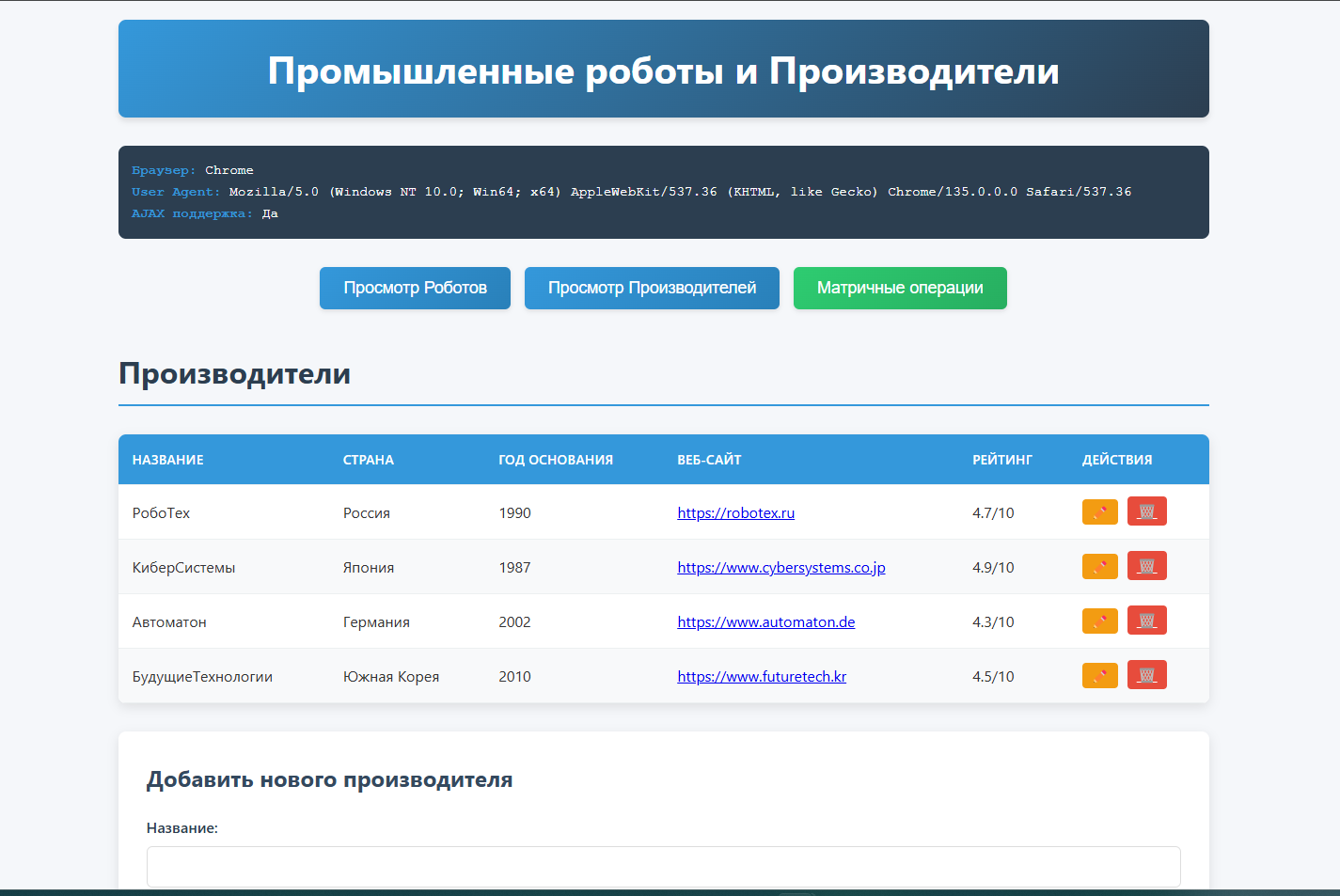


Рисунок 4 – Главная страница часть производителей

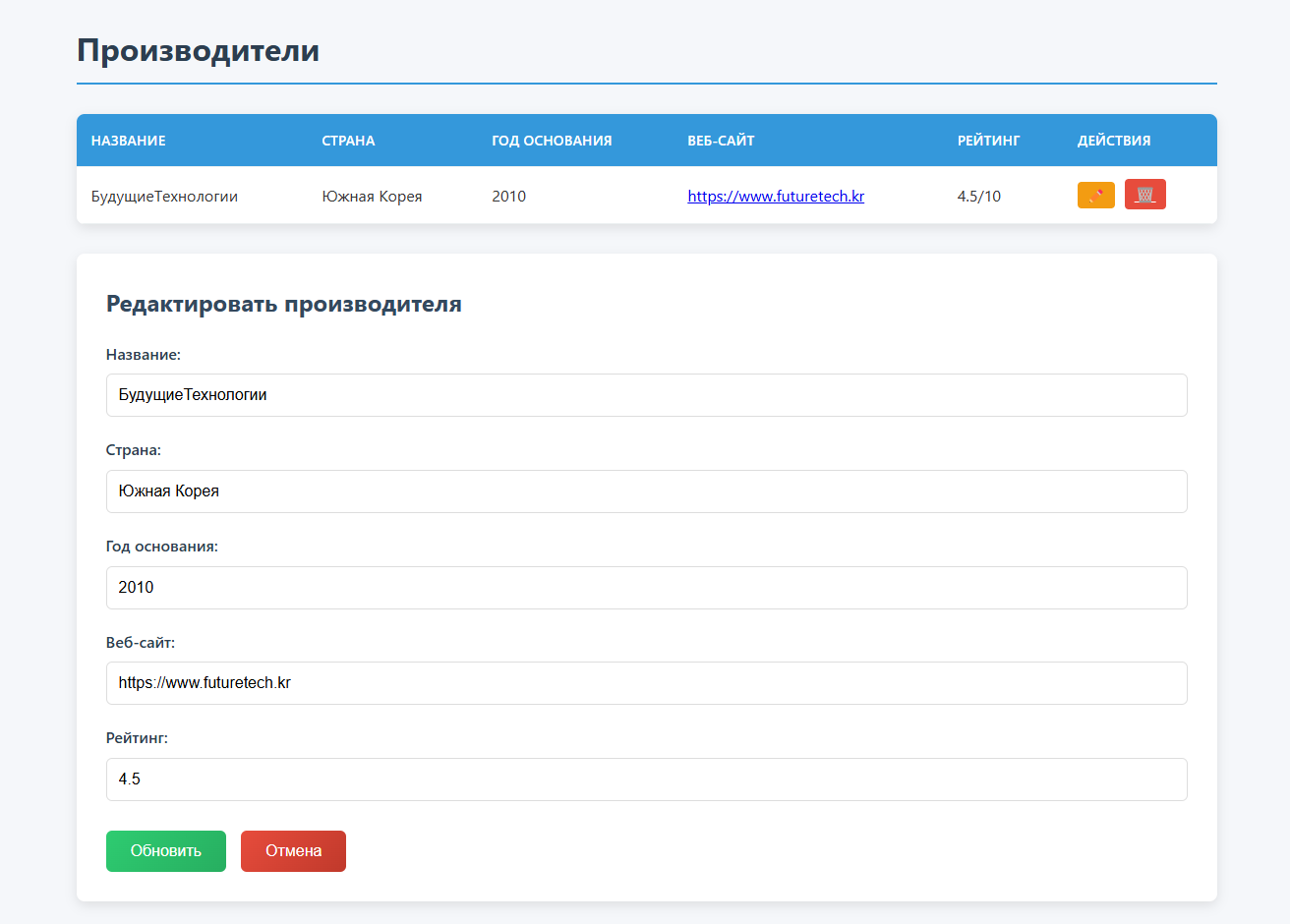


Рисунок 5 – Часть редактирование

**4 Код JavaScript**

/\*\*

\* Загружает данные с сервера через AJAX

\* @param {string} table - Название таблицы ('robots' или 'manufacturers')

\* @param {number|null} editId - ID записи для редактирования (опционально)

\*/

function loadData(table, editId = null) {

// Скрываем матричный раздел и показываем загрузку

$('#matrix-section').hide();

$('#content-section').show().html('<div class="loading">Загрузка данных...</div>');

// AJAX-запрос с использованием jQuery

$.ajax({

url: 'api.php',

method: 'GET',

data: { table: table, id: editId || '' },

dataType: 'json',

success: function(response) {

if (response.success) {

renderTable(response.data, response.columns, table, editId);

} else {

showError(response.error);

}

},

error: function(xhr, status, error) {

showError('Ошибка загрузки: ' + error);

}

});

}

/\*\*

\* Определяет браузер пользователя и выводит информацию

\*/

function detectBrowser() {

const userAgent = navigator.userAgent;

let browser = 'Неизвестно';

if (userAgent.includes('Firefox')) browser = 'Firefox';

else if (userAgent.includes('Chrome')) browser = 'Chrome';

else if (userAgent.includes('Safari')) browser = 'Safari';

else if (userAgent.includes('Edge')) browser = 'Edge';

// Вывод информации в блок #browser-info

$('#browser-info').html(`

<strong>Браузер:</strong> ${browser}<br>

<strong>User Agent:</strong> ${userAgent}<br>

<strong>Поддержка AJAX:</strong> ${'XMLHttpRequest' in window ? 'Да' : 'Нет'}

`);

}

// Инициализация при загрузке страницы

$(document).ready(function() {

detectBrowser();

loadData('robots'); // Загружаем роботов по умолчанию

});

**5 PHP-код**

Файл api.php (обработка AJAX-запросов)

<?php

header('Content-Type: application/json');

include 'db.php';

$response = ['success' => false];

try {

// Get input data from POST

$requestData = $\_SERVER['REQUEST\_METHOD'] === 'POST' ? $\_POST : $\_GET;

$table = $requestData['table'] ?? 'robots';

$id = isset($requestData['id']) ? (int)$requestData['id'] : null;

$action = $requestData['action'] ?? 'get'; // Default to 'get' if no action specified

// Validate table

if (!in\_array($table, ['robots', 'manufacturers'])) {

throw new Exception("Invalid table specified");

}

switch ($action) {

case 'get':

if ($table === 'robots') {

$query = "SELECT robots.\*, manufacturers.name AS manufacturer\_name, manufacturers.country

FROM robots

JOIN manufacturers ON robots.manufacturer\_id = manufacturers.id";

if ($id) {

$query .= " WHERE robots.id = $id LIMIT 1";

}

} elseif ($table === 'manufacturers') {

$query = "SELECT \* FROM manufacturers";

if ($id) {

$query .= " WHERE id = $id LIMIT 1";

}

}

$result = $conn->query($query);

if (!$result) {

throw new Exception($conn->error);

}

$data = [];

while ($row = $result->fetch\_assoc()) {

$data[] = $row;

}

$response = [

'success' => true,

'data' => $data,

'columns' => $table === 'robots' ?

['Модель', 'Тип', 'Вес (кг)', 'Грузоподъёмность (кг)', 'Производитель', 'Страна'] :

['Название', 'Страна', 'Год основания', 'Веб-сайт', 'Рейтинг']

];

break;

case 'delete':

if (!$id) {

throw new Exception("ID is required for deletion");

}

$query = "DELETE FROM $table WHERE id = $id";

if ($conn->query($query)) {

$response['success'] = true;

} else {

throw new Exception($conn->error);

}

break;

default:

// For other actions (insert/update), let the existing files handle them

throw new Exception("Invalid action specified");

}

} catch (Exception $e) {

$response = [

'success' => false,

'error' => $e->getMessage()

];

}

echo json\_encode($response);

?>

**6  HTML-код**

Фрагмент index.php (основная страница)

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Промышленные роботы</title>

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

<!-- Подключение jQuery -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

</head>

<body>

<div class="container">

<h1>Управление роботами и производителями</h1>

<!-- Блок информации о браузере -->

<div id="browser-info" class="browser-info"></div>

<!-- Кнопки переключения между таблицами -->

<div class="button-group">

<button onclick="loadData('robots')">Роботы</button>

<button onclick="loadData('manufacturers')">Производители</button>

</div>

<!-- Основной контент (загружается через AJAX) -->

<div id="content-section"></div>

</div>

<!-- Подключение JavaScript -->

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

</body>

</html>

**Вывод:**

В ходе выполнения лабораторной работы №7 было разработано AJAX-приложение для управления базой данных промышленных роботов и их производителей.

**Основные результаты:**

1. Реализовано асинхронное взаимодействие

- Данные загружаются без перезагрузки страницы.

- Использованы AJAX-запросы (jQuery + чистый JavaScript).

- Сервер возвращает ответы в формате JSON.

1. Интеграция с базой данных

- Работа с таблицами robots и manufacturers.

- Поддержка CRUD-операций (добавление, чтение, обновление, удаление).

1. Кросс-браузерная совместимость

- Реализована функция detectBrowser(), определяющая браузер пользователя.

- Проверена работа в Chrome, Firefox, Edge.

1. Использование jQuery

- Упрощение AJAX-запросов ($.ajax, $.post).

- Динамическое обновление интерфейса (формы, таблицы).