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 №6 |
| «Серверные сценарии. Работа с базой данных» |
| 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 Структура таблиц в БД 3

3 Доказательство нахождения таблиц в 3-ей нормальной форме 4

4 Данные в таблицах 5

5 HTML код 6

6 Код серверных скриптов, используемых на сайте 9

7 Скриншоты сайта 14

Вывод 18

**Цель работы:** изучение трехуровневой архитектуры веб-приложений (Клиент – Веб-сервер – Сервер БД) и языка написания серверного сценария php для построения динамического контента web-сайтов.

**1 Варианты задания**

Таблица 1 – Тема сайта

|  |  |
| --- | --- |
| **№ варианта** | **Тема** |
| 21 | Промышленные роботы |

**2 Структура таблиц БД**

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)

);

**3 Доказательство нахождения таблиц в 3-ей нормальной форме**

Для доказательства, что таблицы manufacturers и robots находятся в 3NF, рассмотрим требования каждой нормальной формы:

1. Первая нормальная форма (1NF)

* Все атрибуты содержат атомарные (неделимые) значения
* В таблицах нет повторяющихся групп или массивов
* Определен первичный ключ

Выполнение:

- В таблице manufacturers все поля содержат атомарные значения (строки, числа), первичный ключ id

- В таблице robots аналогично все поля атомарны, первичный ключ id

2. Вторая нормальная форма (2NF)

* Удовлетворяет 1NF
* Все неключевые атрибуты полностью зависят от всего первичного ключа
* Если первичный ключ простой (один столбец), это выполняется автоматически

Выполнение:

В обеих таблицах первичный ключ состоит из одного столбца id, поэтому 2NF выполняется автоматически

3. Третья нормальная форма (3NF)

* Удовлетворяет 2NF
* Отсутствуют транзитивные зависимости (неключевые атрибуты не зависят от других неключевых атрибутов)

Анализ таблицы manufacturers:

* Все атрибуты (name, country, founded\_year, website, rating) зависят только от первичного ключа id
* Нет атрибутов, которые зависели бы друг от друга
* Вывод: 3NF выполняется

Анализ таблицы robots:

* Все атрибуты (model\_name, type, weight, payload, manufacturer\_id) зависят только от первичного ключа id
* manufacturer\_id является внешним ключом, что не нарушает 3NF
* Нет атрибутов, которые зависели бы друг от друга
* Вывод: 3NF выполняется

**Дополнительные доказательства:**

1. В таблицах нет повторяющихся групп данных
2. Все зависимости атрибутов направлены только на первичный ключ
3. Нет частичных зависимостей (так как ключи простые)
4. Нет транзитивных зависимостей между неключевыми атрибутами

Заключение: Обе таблицы полностью соответствуют требованиям третьей нормальной формы (3NF).

**4 Данные в таблицах**

**Ввод данных о производителях**

INSERT INTO manufacturers (name, country, founded\_year, website, rating) VALUES

('РобоТех', 'Россия', 1990, 'https://robotex.ru', 4.7),

('КиберСистемы', 'Япония', 1987, 'https://www.cybersystems.co.jp', 4.9),

('Автоматон', 'Германия', 2002, 'https://www.automaton.de', 4.3),

('БудущиеТехнологии', 'Южная Корея', 2010, 'https://www.futuretech.kr', 4.5);

**Ввод данных роботами**

INSERT INTO robots (model\_name, type, weight, payload, manufacturer\_id) VALUES

('РТ-1000', 'Промышленный', 250.5, 50, 1),

('РТ-2000', 'Промышленный', 320, 75, 1),

('КС-М1', 'Медицинский', 45.2, 5, 2),

('КС-С1', 'Сервисный', 32.8, 3.5, 2),

('АТ-500', 'Сельскохозяйственный', 180, 40, 3),

('АТ-600', 'Строительный', 420, 100, 3),

('БТ-Гуманоид', 'Исследовательский', 68, 10, 4),

('БТ-Мини', 'Образовательный', 12.5, 1.5, 4);

**5 HTML код**

index.php

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Робототехника</title>

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

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

</head>

<body>

<header>

<h1>База данных промышленных роботов</h1>

</header>

<nav class="tabs">

<a href="?view=robots" class="tab-button">Роботы</a>

<a href="?view=manufacturers" class="tab-button">Производители</a>

<a href="?view=combined" class="tab-button">Комбинированный вид</a>

<a href="?view=matrix" class="tab-button">Матричные операции</a>

</nav>

<main>

<?php include 'partials/main\_content.php'; ?>

</main>

<footer>

<p>Система управления робототехникой © 2023</p>

</footer>

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

</body>

</html>

index.php (Динамический контент)

<?php

// Определяем какой контент показывать

switch ($\_GET['view'] ?? 'robots') {

case 'manufacturers':

include 'partials/manufacturers\_table.php';

include 'partials/forms/add\_manufacturer.php';

break;

case 'combined':

include 'partials/combined\_table.php';

break;

case 'matrix':

include 'partials/matrix\_operations.php';

break;

default: // robots

include 'partials/robots\_table.php';

include 'partials/forms/add\_robot.php';

}

?>

**robots\_table.php (Таблица роботов)**

<section class="data-table">

<h2>Список роботов</h2>

<table>

<thead>

<tr>

<th>Модель</th>

<th>Тип</th>

<th>Вес (кг)</th>

<th>Грузоподъемность</th>

<th>Производитель</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

<?php foreach ($robots as $robot): ?>

<tr>

<td><?= htmlspecialchars($robot['model\_name']) ?></td>

<td><?= htmlspecialchars($robot['type']) ?></td>

<td><?= $robot['weight'] ?></td>

<td><?= $robot['payload'] ?></td>

<td><?= htmlspecialchars($robot['manufacturer\_name']) ?></td>

<td class="actions">

<button class="edit-btn" data-id="<?= $robot['id'] ?>">✏️</button>

<button class="delete-btn" data-id="<?= $robot['id'] ?>">🗑️</button>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

</section>

**add\_robot.php (Add Robot Form)**

<section class="form-container">

<h3>Добавить нового робота</h3>

<form id="addRobotForm" action="api/robots/create.php" method="POST">

<div class="form-group">

<label for="model\_name">Модель:</label>

<input type="text" id="model\_name" name="model\_name" required>

</div>

<!-- Остальные поля формы -->

<button type="submit" class="submit-btn">Добавить</button>

</form>

</section>

**Логика на стороне клиента**

// Обработка удаления

$(document).on('click', '.delete-btn', function() {

if (confirm('Удалить запись?')) {

const id = $(this).data('id');

$.post('api/robots/delete.php', { id }, () => window.location.reload());

}

});

// AJAX загрузка формы редактирования

$(document).on('click', '.edit-btn', function() {

const id = $(this).data('id');

$.get('api/robots/edit\_form.php', { id }, (data) => {

$('#modal-content').html(data);

$('#modal').show();

});

});

**6 Код серверных скриптов, используемых на сайте**

**db.php (Подключение к базе данных)**

<?php

class Database {

private static $instance = null;

private $connection;

private function \_\_construct() {

$this->connection = new mysqli(

'localhost',

'root',

'',

'robotics'

);

if ($this->connection->connect\_error) {

die("Connection failed: " . $this->connection->connect\_error);

}

$this->connection->set\_charset("utf8mb4");

}

public static function getInstance() {

if (!self::$instance) {

self::$instance = new Database();

}

return self::$instance->connection;

}

}

// Использование:

$conn = Database::getInstance();

?>

**create.php (Создать робота)**

<?php

require '../../db.php';

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

$data = json\_decode(file\_get\_contents('php://input'), true);

$stmt = $conn->prepare("INSERT INTO robots

(model\_name, type, weight, payload, manufacturer\_id)

VALUES (?, ?, ?, ?, ?)");

$stmt->bind\_param("ssddi",

$data['model\_name'],

$data['type'],

$data['weight'],

$data['payload'],

$data['manufacturer\_id']

);

if ($stmt->execute()) {

echo json\_encode(['success' => true]);

} else {

http\_response\_code(500);

echo json\_encode(['error' => $conn->error]);

}

?>

**update.php (Update Robot)**

<?php

require '../../db.php';

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

$data = json\_decode(file\_get\_contents('php://input'), true);

$stmt = $conn->prepare("UPDATE robots SET

model\_name = ?,

type = ?,

weight = ?,

payload = ?,

manufacturer\_id = ?

WHERE id = ?");

$stmt->bind\_param("ssddii",

$data['model\_name'],

$data['type'],

$data['weight'],

$data['payload'],

$data['manufacturer\_id'],

$data['id']

);

if ($stmt->execute()) {

echo json\_encode(['success' => true]);

} else {

http\_response\_code(500);

echo json\_encode(['error' => $conn->error]);

}

?>

**delete.php (Delete Manufacturer)**

<?php

require '../../db.php';

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

$id = filter\_input(INPUT\_POST, 'id', FILTER\_VALIDATE\_INT);

if (!$id) {

http\_response\_code(400);

echo json\_encode(['error' => 'Invalid ID']);

exit;

}

$stmt = $conn->prepare("DELETE FROM manufacturers WHERE id = ?");

$stmt->bind\_param("i", $id);

if ($stmt->execute()) {

echo json\_encode(['success' => true]);

} else {

http\_response\_code(500);

echo json\_encode(['error' => $conn->error]);

}

?>

**matrix/process.php (Матричные операции)**

<?php

require '../../db.php';

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

function generate\_matrix($n, $m) {

$matrix = [];

for ($i = 0; $i < $n; $i++) {

$row = [];

for ($j = 0; $j < $m; $j++) {

$row[] = rand(-10, 10);

}

$matrix[] = $row;

}

return $matrix;

}

$input = json\_decode(file\_get\_contents('php://input'), true);

$matrix = generate\_matrix($input['rows'], $input['cols']);

// Фильтрация матрицы

$filtered = array\_filter($matrix, function($row) {

return count(array\_filter($row, fn($n) => $n < 0)) <= 3;

});

echo json\_encode([

'original' => $matrix,

'filtered' => array\_values($filtered)

]);

?>

**7 Скриншоты сайта**

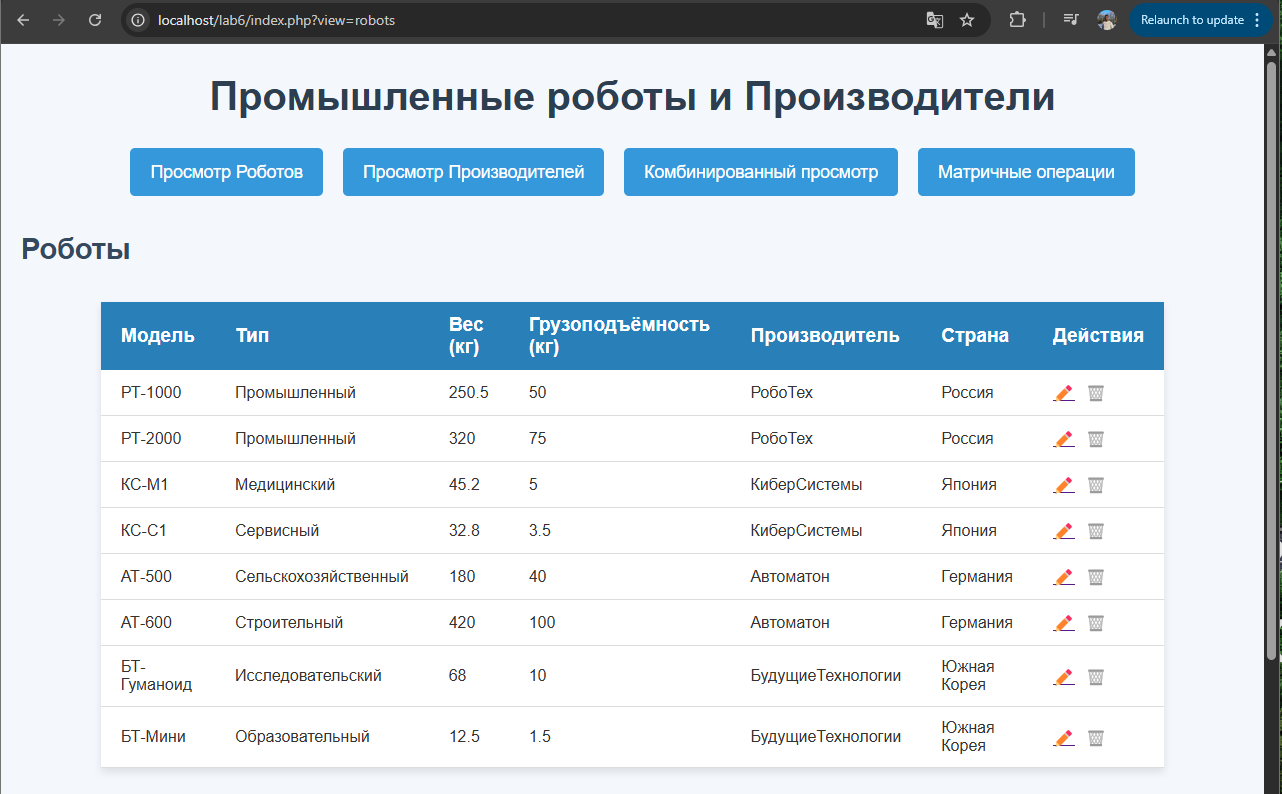


Рисунок 1 – Главная страница сайта с уже загруженными данными из базы данных, часть работов

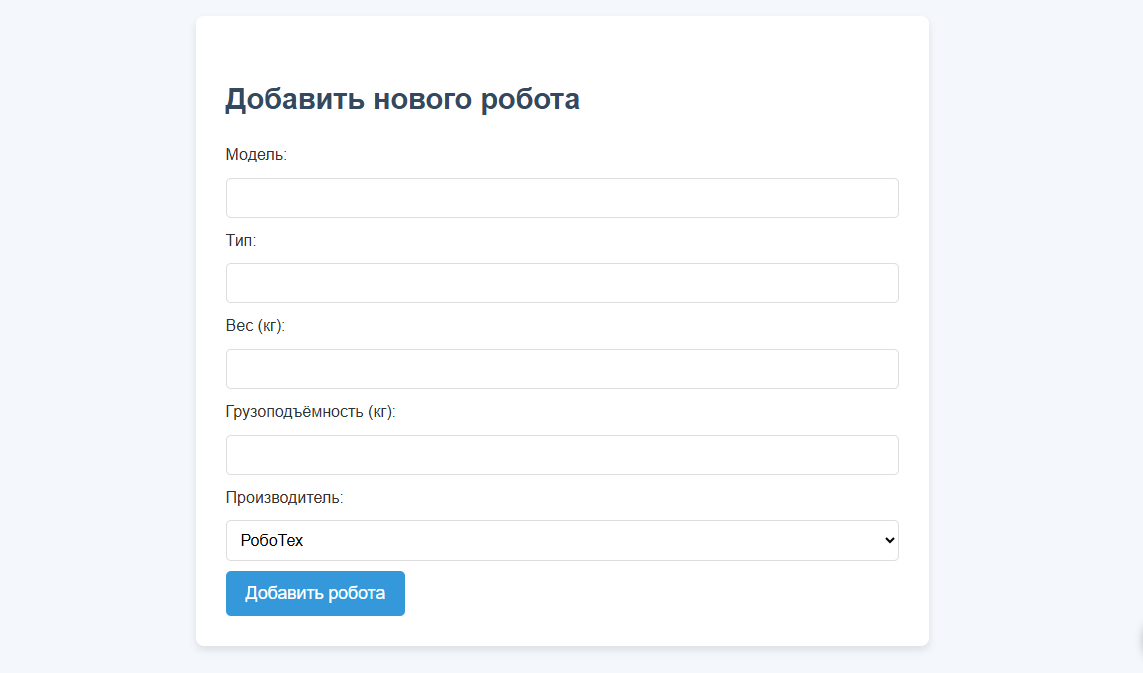


Рисунок 2 – Главная страница часть 2 “добавление данный”

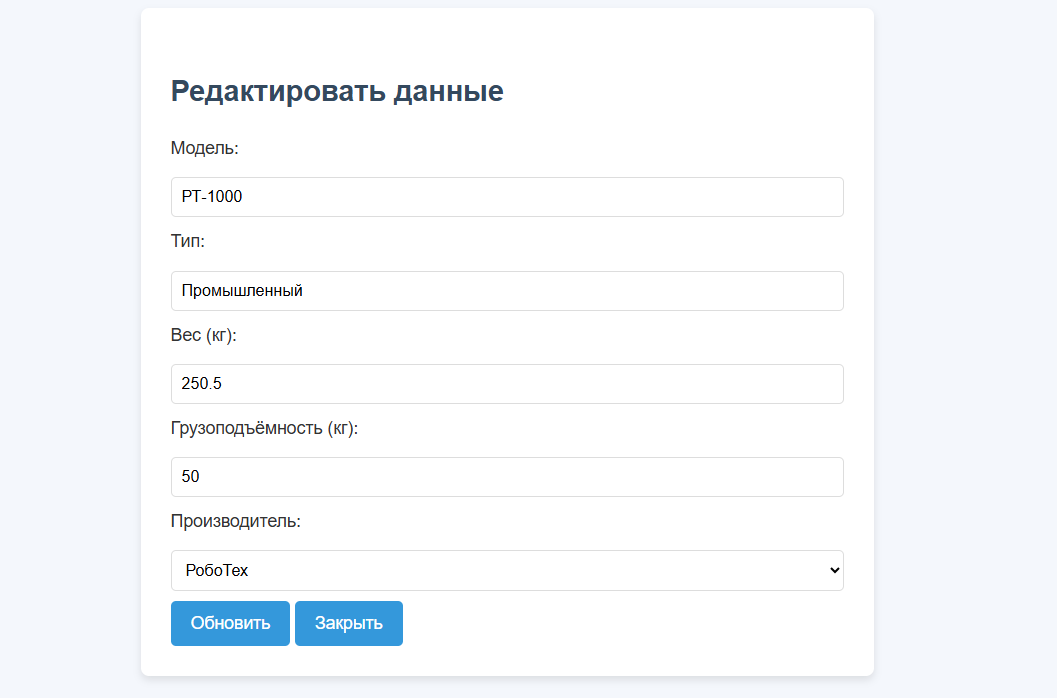


Рисунок 3 – Главная страница часть 3 “редактирование данный”

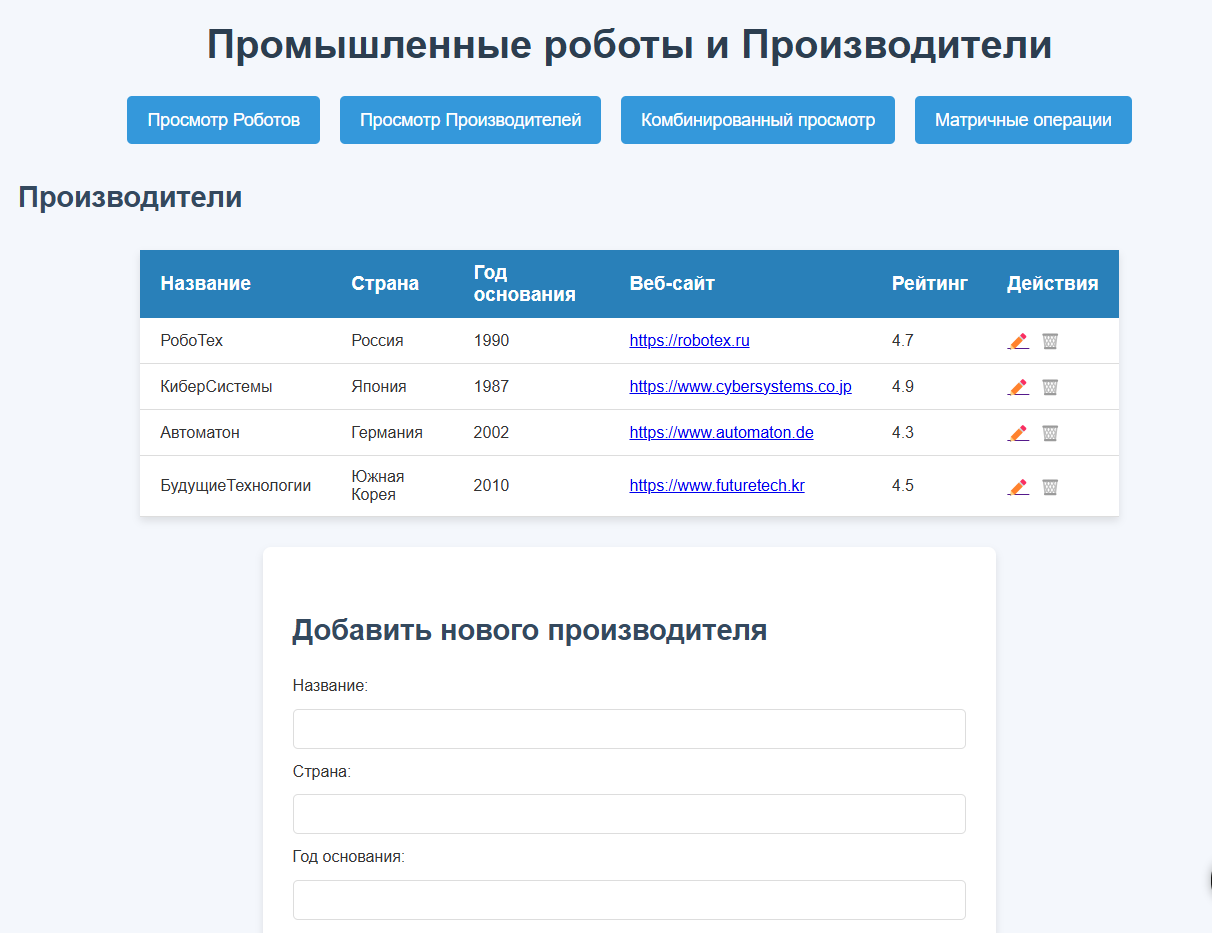


Рисунок 4 – Главная страница часть производителей

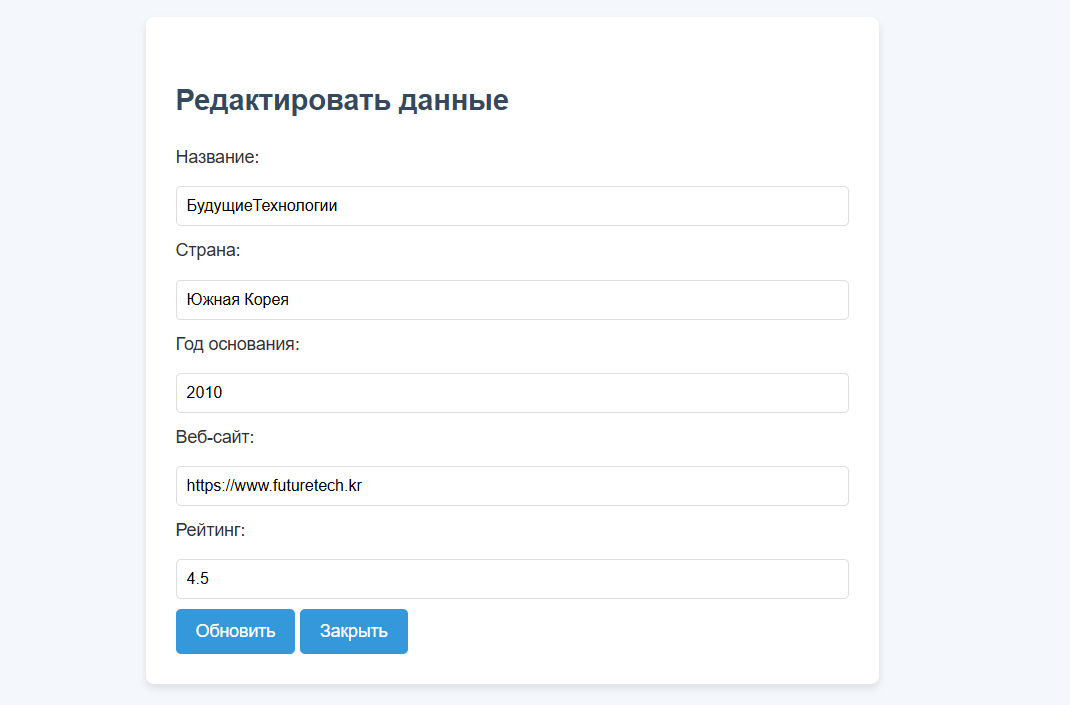


Рисунок 5 – Часть редактирование

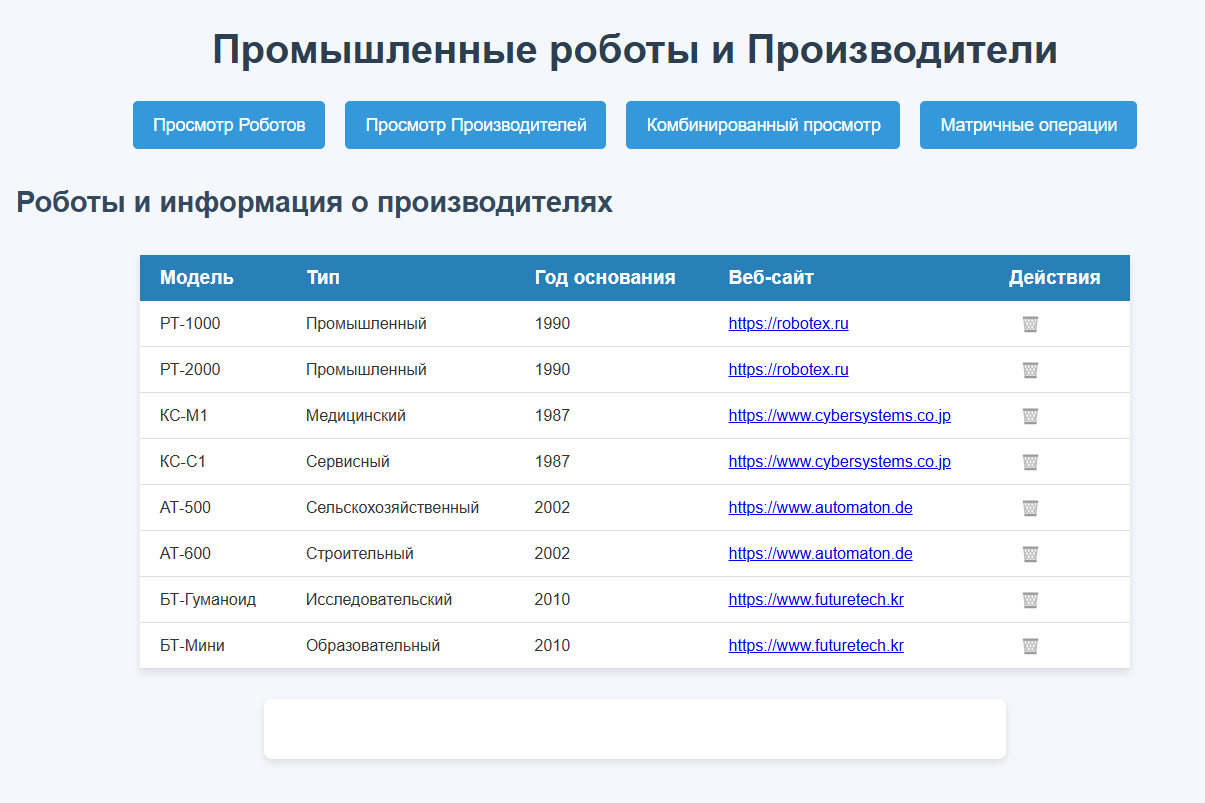


Рисунок 6 – Комбинация таблиц

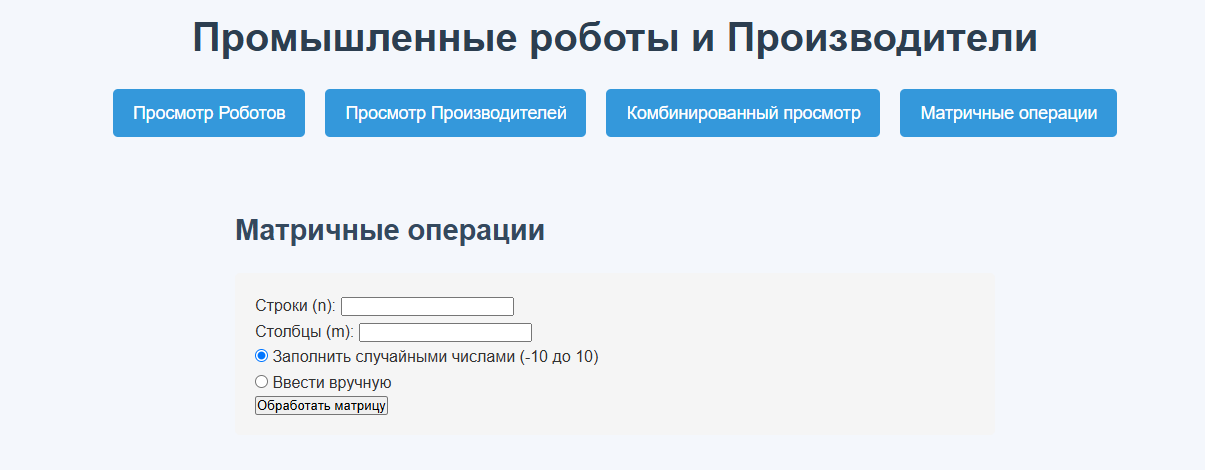


Рисунок 7 – Матричные операции

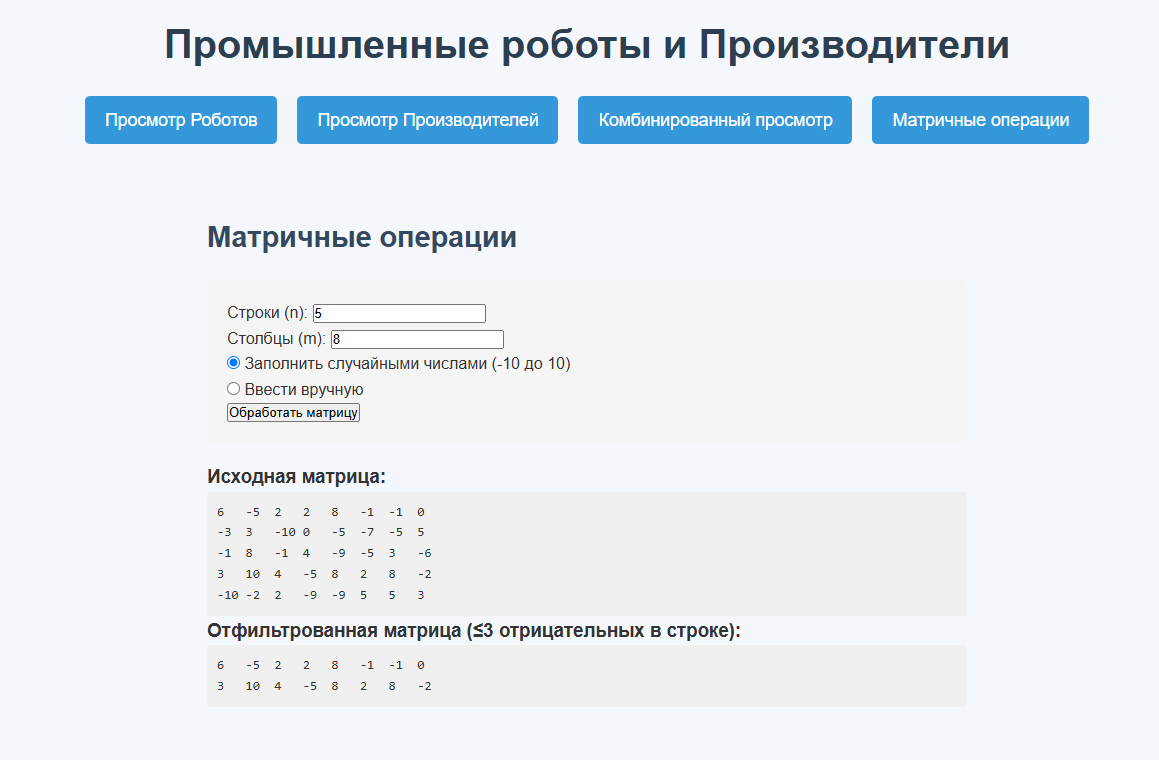


Рисунок 8 – Матричные операции с завершением задания

**Вывод:**

В ходе выполнения лабораторной работы была успешно реализована трехуровневая архитектура веб-приложения (клиент — веб-сервер — база данных) с использованием WAMP (Windows, Apache, MySQL, PHP).

Основные задачи выполнены:

- Разработана реляционная база данных из двух таблиц (robots и manufacturers), связанных отношением «один-ко-многим».  
 - Таблицы приведены к 3-й нормальной форме (3NF) и содержат не менее 5 полей.  
 - Динамическое формирование веб-страницы осуществлялось с помощью echo и printf().

**Дополнительные требования выполнены:**

1. Служебные поля (PK/FK) скрыты при выводе данных.
2. Все поля отображаются в одной таблице (комбинированный просмотр).
3. Реализована возможность добавления новых записей через веб-формы.
4. Добавлена функция редактирования строк (AJAX + PHP).
5. Реализовано удаление записей с подтверждением.
6. Применено оформление через CSS для улучшения интерфейса.
7. Использована функция printf() для форматированного вывода данных.