Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Кафедра анализа данных и машинного обучения

Факультет информационных технологий и анализа больших данных

**ОтЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ**

**по дисциплине** «Технологии работы с открытыми данными»

**на тему** «Основные форматы данных. REST API»

**Вариант 25**

Направление подготовки: 09.03.03 – Прикладная информатика

Профиль: «ИТ-сервисы и технологии обработки данных в экономике и финансах»

**Выполнила студентка**

4 курса, группы ПИ21-6,

очной формы обучения

Филиппова Е. Р.

**Москва 2024**

**Цель работы:** изучить нормативные документы, стандарты, синтаксис данных JSON. Выполнить примеры заданий. Написать код сценарий с форматом данных JSON.

**Задание:** КХЛ. Информация об играх: команда 1, основной состав (достаточно указать 3 игроков), команда 2, основой состав (достаточно указать 3 игроков), счет, информация о голах.

**Решение**

Программное обеспечение

1. HTML: Использован для создания структуры веб-страницы.
2. CSS: Использован для стилизации элементов веб-страницы, включая фоновую картинку, цвета и шрифты.
3. JavaScript: Использован для динамического создания и вставки данных на веб-страницу. Скрипт обрабатывает данные о матче и создает соответствующие HTML-элементы.
4. Bootstrap: Использован для упрощения стилизации и создания отзывчивого дизайна. Bootstrap предоставляет готовые стили для таблиц, контейнеров и других элементов.
5. Visual Studio Code (VS Code): Использован для написания и редактирования HTML, CSS и JavaScript кода. VS Code предоставляет удобные инструменты для разработки, такие как подсветка синтаксиса, автодополнение и интеграция с системами контроля версий.
6. Google Chrome: Использован для тестирования и просмотра веб-страницы. Chrome предоставляет инструменты разработчика (DevTools), которые позволяют проверять и отлаживать HTML, CSS и JavaScript код.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 1 – структура каталога

Код

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>КХЛ</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

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

</head>

<body>

    <div class="container">

        <div class="main\_\_wrapper">

            <h1 class="text-center">Континентальная хоккейная лига</h1>

            <div id="score-container" class="mb-4"></div>

            <div id="teams-container" class="mb-4"></div>

            <div id="table-container"></div>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

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

</body>

</html>

**db.json**

{

    "игра": {

        "команда\_1": {

            "название": "СКА",

            "игроки": [

                "Иван Иванов",

                "Сергей Петров",

                "Алексей Смирнов"

            ]

        },

        "команда\_2": {

            "название": "ЦСКА",

            "игроки": [

                "Дмитрий Кузнецов",

                "Михаил Орлов",

                "Андрей Павлов"

            ]

        },

        "счет": {

            "команда\_1": 3,

            "команда\_2": 2

        },

        "голы": [

            {

                "минута": 12,

                "команда": "СКА",

                "автор": "Иван Иванов"

            },

            {

                "минута": 34,

                "команда": "ЦСКА",

                "автор": "Дмитрий Кузнецов"

            },

            {

                "минута": 45,

                "команда": "СКА",

                "автор": "Сергей Петров"

            },

            {

                "минута": 50,

                "команда": "ЦСКА",

                "автор": "Михаил Орлов"

            },

            {

                "минута": 58,

                "команда": "СКА",

                "автор": "Алексей Смирнов"

            }

        ]

    }

}

**Main.js**

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

  const scoreContainer = document.getElementById('score-container');

  const teamsContainer = document.getElementById('teams-container');

  const tableContainer = document.getElementById('table-container');

  // Загрузка данных из файла db.json

  fetch('db.json')

      .then(response => response.json())

      .then(data => {

          // Создаем и добавляем информацию о счете

          const scoreDiv = document.createElement('div');

          scoreDiv.classList.add('text-center', 'mb-4');

          scoreDiv.innerHTML = `<h2>Счет: ${data.игра.счет.команда\_1} - ${data.игра.счет.команда\_2}</h2>`;

          scoreContainer.appendChild(scoreDiv);

          // Создаем и добавляем информацию о членах команд

          const teamsDiv = document.createElement('div');

          teamsDiv.classList.add('row', 'mb-4');

          const team1Div = document.createElement('div');

          team1Div.classList.add('col-md-6');

          team1Div.innerHTML = `<h3>${data.игра.команда\_1.название}</h3><ul>${data.игра.команда\_1.игроки.map(player => `<li>${player}</li>`).join('')}</ul>`;

          const team2Div = document.createElement('div');

          team2Div.classList.add('col-md-6');

          team2Div.innerHTML = `<h3>${data.игра.команда\_2.название}</h3><ul>${data.игра.команда\_2.игроки.map(player => `<li>${player}</li>`).join('')}</ul>`;

          teamsDiv.appendChild(team1Div);

          teamsDiv.appendChild(team2Div);

          teamsContainer.appendChild(teamsDiv);

          // Создаем и добавляем таблицу с голами

          const table = document.createElement('table');

          table.classList.add('table', 'table-bordered');

          const thead = document.createElement('thead');

          const tbody = document.createElement('tbody');

          const headerRow = document.createElement('tr');

          const headers = ['Минута', 'Команда', 'Автор гола'];

          headers.forEach(headerText => {

              const th = document.createElement('th');

              th.textContent = headerText;

              headerRow.appendChild(th);

          });

          thead.appendChild(headerRow);

          data.игра.голы.forEach(goal => {

              const row = document.createElement('tr');

              const minuteCell = document.createElement('td');

              minuteCell.textContent = goal.минута;

              const teamCell = document.createElement('td');

              teamCell.textContent = goal.команда;

              const authorCell = document.createElement('td');

              authorCell.textContent = goal.автор;

              row.appendChild(minuteCell);

              row.appendChild(teamCell);

              row.appendChild(authorCell);

              tbody.appendChild(row);

          });

          table.appendChild(thead);

          table.appendChild(tbody);

          tableContainer.appendChild(table);

      })

      .catch(error => console.error('Ошибка при загрузке данных:', error));

});

**Styles.css**

body {

    background-image: url('https://hctorpedo.ru/upload/iblock/47e/1in4kjthetftwt5ulh78e13hse4m1u7z/cc0c5d0f26a40c45e787762e4702d232.jpg');

    background-size: cover;

    background-position: center;

    background-repeat: no-repeat;

    color: white;

    font-family: Arial, sans-serif;

}

.container {

    background-color: rgba(0, 0, 0, 0.7);

    padding: 20px;

    border-radius: 10px;

    margin-top: 5rem;

    margin-bottom: 5rem;

}

h1, h2, h3 {

    color: #ffd700;

}

.table {

    background-color: rgba(255, 255, 255, 0.8);

    color: black;

}

.table th, .table td {

    border: 1px solid #ddd;

    padding: 8px;

}

.table th {

    background-color: #ffd700;

    color: black;

}

.table tbody tr:nth-child(even) {

    background-color: #f9f9f9;

}

.table tbody tr:nth-child(odd) {

    background-color: #ffffff;

}

Результат

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 - Результат работы сценария

**Выводы:** В рамках данного задания была создана веб-страница, отображающая информацию о матче КХЛ. Использовались HTML, CSS и JavaScript для создания и стилизации страницы, а также Bootstrap для упрощения дизайна. Редактор кода Visual Studio Code и веб-браузер Google Chrome были использованы для разработки и тестирования.

Кроме того, был изучен синтаксис формата JSON и применен на практике для хранения и отображения данных о результатах игры хоккейных команд в виде таблицы. JSON позволяет структурировать данные в удобном и легко читаемом формате, что значительно упрощает их обработку и отображение на веб-странице.

**Список литературы**

1. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
2. <https://developer.mozilla.org/ru/docs/Learn/JavaScript/Objects/JSON>