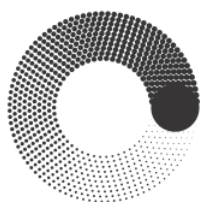


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»

Лабораторная работа № 5

Дисциплина: API-технологии

Тема: Построение статистики

Выполнил: студент группы 221-375

Яковлев Р. А.

(Фамилия И.О.)

Дата, подпись _____

(Дата)

(Подпись)

Проверил: _____

(Фамилия И.О., степень, звание)

(Оценка)

Дата, подпись _____

(Дата)

(Подпись)

Замечания:

Москва

2025

График криптовалюты

Структура HTML

HTML-код начинается с объявления типа документа и определения языка страницы

В `<head>` секции:

- Устанавливается кодировка UTF-8, что позволяет корректно отображать русские символы.
- Настраивается адаптивность с помощью мета-тега `viewport`, который важен для правильного отображения на мобильных устройствах.
- Подключается библиотека Bootstrap для стилизации элементов интерфейса.
- Загружается библиотека Chart.js для визуализации графиков.

```
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Крипто-график</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/
bootstrap.min.css">
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
```

Основной контент

В самом теле документа добавлены элементы для выбора криптовалюты и периода времени:

```
<body>
<div class="container mt-5">
<h1>Изменения курса криптовалюты</h1>
<div class="form-group">
<label for="currency">Выберите криптовалюту:</label>
<select id="currency" class="form-control">
<option value="bitcoin">Bitcoin</option>
<option value="ethereum">Ethereum</option>
```

```
<option value="ripple">Ripple</option>
</select>
</div>
<div class="form-group">
<label for="days">Выберите период (дней):</label>
<input type="number" id="days" class="form-control" min="1" max="30" value="7">
</div>
<button id="fetchData" class="btn btn-primary">Показать график</button>
<canvas id="myChart" class="mt-4"></canvas>
</div>
```

Кнопка "Показать график" инициирует получение данных и отображение графика.

JavaScript функциональность

```
JavaScript код начинается с добавления обработчика события на кнопку:
document.getElementById('fetchData').addEventListener('click', async function() {
  const currency = document.getElementById('currency').value;
  const days = parseInt(document.getElementById('days').value);
  if (days < 1 || days > 30) {
    alert("Пожалуйста, выберите период от 1 до 30 дней.");
    return;
  }
}
```

Ключевые моменты в коде:

1. Проверка периода: Убедитесь, что пользователь выбрал допустимый диапазон (от 1 до 30 дней).

2. Получение данных: Используя Fetch API, выполняется запрос к Coin-Gecko API, чтобы получить данные о ценах.

```
const response = await fetch(`https://api.coingecko.com/api/v3/coins/${currency}/market_chart?vs_currency=usd&days=${days}`);
```

3. Обработка данных: После получения данных, они переводятся в формат JSON и извлекаются нужные значения.

4. Создание графика: С помощью Chart.js создается линейный график, где ось Y отображает цену, а ось X — дату.

```
const ctx = document.getElementById('myChart').getContext('2d');
if (window.myChart && window.myChart instanceof Chart) {
  window.myChart.destroy(); // Удаляем предыдущий график
}
window.myChart = new Chart(ctx, {
  type: 'line',
  data: { ... },
  options: { ... }
});
```

5. Обработка ошибок: В случае ошибки при загрузке данных пользователю отображается предупреждение.

```
catch (error) {
  console.error('Ошибка:', error);
  alert('Произошла ошибка при получении данных о ценах.\n' + error.message);
}
```

lr5.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Крипто-график</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/
bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

```
</head>
<body>
  <div class="container mt-5">
    <h1>Изменения курса криптовалюты</h1>
    <div class="form-group">
      <label for="currency">Выберите криптовалюту:</label>
      <select id="currency" class="form-control">
        <option value="bitcoin">Bitcoin</option>
        <option value="ethereum">Ethereum</option>
        <option value="ripple">Ripple</option>
      </select>
    </div>
    <div class="form-group">
      <label for="days">Выберите период (дней):</label>
      <input type="number" id="days" class="form-control" min="1" max="30"
value="7">
    </div>
    <button id="fetchData" class="btn btn-primary">Показать график</button>
    <canvas id="myChart" class="mt-4"></canvas>
  </div>

  <script src="lr5.js"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js/dist/chart.umd.js"></script>
</body>
</html>
```

lr5.js

```
document.getElementById('fetchData').addEventListener('click', async function() {
    const currency = document.getElementById('currency').value;
    const days = parseInt(document.getElementById('days').value);
```

```
    if (days < 1 || days > 30) {
        alert("Пожалуйста, выберите период от 1 до 30 дней.");
        return;
    }
```

```
    try {
        const response = await fetch(`https://api.coingecko.com/api/v3/coins/${currency}/
market_chart?vs_currency=usd&days=${days}`);
```

```
        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
```

```
        const data = await response.json();
        console.log("Полученные данные:", data);
```

```
if (!data.prices || data.prices.length === 0) {  
    alert("Не удалось получить данные о ценах.");  
    return;  
}
```

```
const labels = data.prices.map(price => {  
    const date = new Date(price[0]);  
    return date.toLocaleDateString();  
});
```

```
const prices = data.prices.map(price => price[1]);
```

```
const ctx = document.getElementById('myChart').getContext('2d');
```

```
if (window.myChart && window.myChart instanceof Chart) {  
    console.log("Удаление предыдущего графика...");  
    window.myChart.destroy();    } else {  
    console.log("myChart не найден или не является экземпляром Chart.");  
}
```

```
window.myChart = new Chart(ctx, {  
    type: 'line',
```

```
data: {
  labels: labels,
  datasets: [{
    label: `Цена ${currency.charAt(0).toUpperCase() + currency.slice(1)}
(USD)`,
    data: prices,
    borderColor: 'rgba(75, 192, 192, 1)',
    backgroundColor: 'rgba(75, 192, 192, 0.2)',
    fill: true,
  }]
},
options: {
  responsive: true,
  scales: {
    y: {
      title: {
        display: true,
        text: 'Цена (USD)',
      },
      beginAtZero: false
    },
    x: {
      title: {
```



```
display: true,  
text: 'Дата',  
}  
}  
}  
}  
});
```

```
    } catch (error) {  
        console.error('Ошибка:', error);  
        alert('Произошла ошибка при получении данных о ценах.\n' + error.message);  
    }  
});
```

Картиночки

Изменения курса криптовалюты

Выберите криптовалюту:

Bitcoin

Выберите период (дней):

7

Показать график

Изменения курса криптовалюты

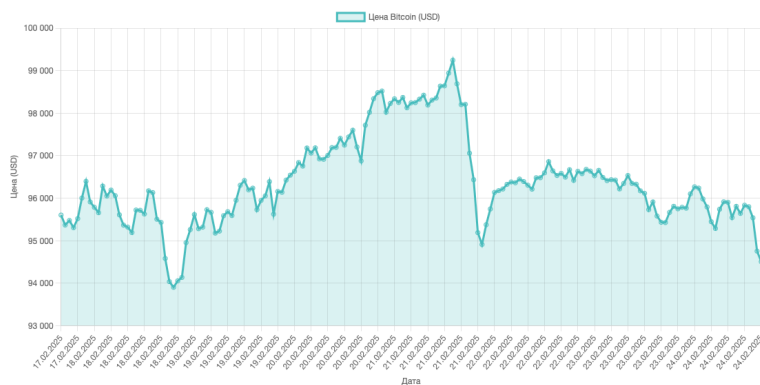
Выберите криптовалюту:

Bitcoin

Выберите период (дней):

7

Показать график



Ссылка на репозиторий с кодом:

<https://github.com/Ry0u14iY0Ru/APIshechki.git>