```
C: > Users > Компик > Desktop > вебка 5 > ♦ index.html > ...
       <!DOCTYPE html>
       <html lang="ru">
           <meta charset="UTF-8">
           <meta name="viewport" content="width=device-width, initial-scale=1.0">
           <title>Управление постами</title>
           <link rel="stylesheet" href="style.css">
      </head>
<body>
           <h1>Управление постами</h1>
           <div class="form-container">
               <h2>Создать новый пост</h2>
               <form id="createPostForm">
                   <input type="text" id="title" placeholder="Заголовок" required>
                   <input type="text" id="body" placeholder="Текст поста" required>
                   <button type="submit">Добавить пост</button>
               </form>
           </div>
           <div id="postsContainer"></div>
           <script src="script.js"></script>
      </html>
```

```
body {
    font-family: Arial, sans-serif;
    margin: 20px;
}

.form-container {
    margin-bottom: 20px;
}

input, button {
    padding: 10px;
    margin: 5px;
}

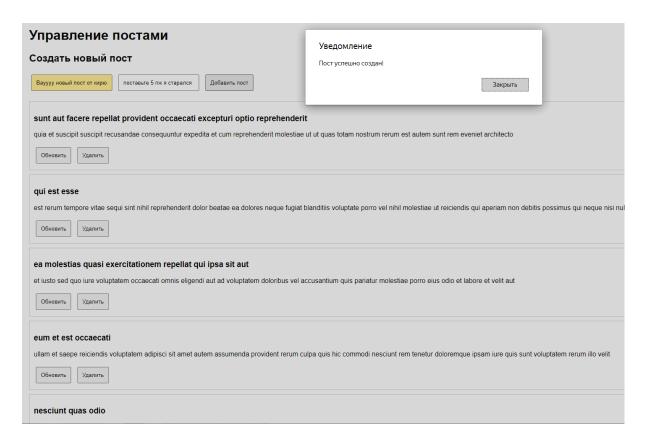
.post {
    border: 1px solid  #ccc;
    padding: 10px;
    margin-bottom: 10px;
}
```

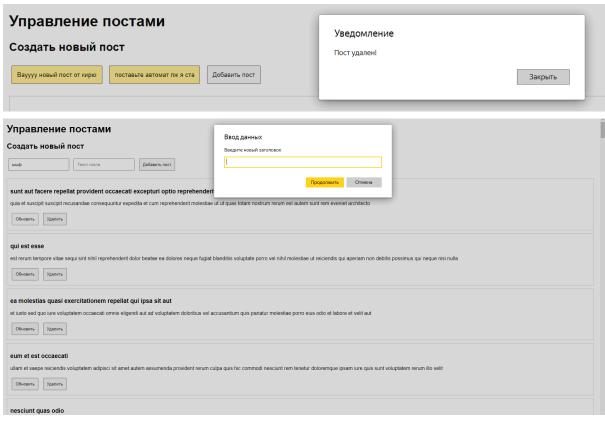
```
document.addEventListener('DOMContentLoaded', () => {
    fetchPosts();
// Функция для получения всех постов
function fetchPosts() {
    fetch('https://jsonplaceholder.typicode.com/posts')
        .then(response => {
            if (!response.ok) {
               throw new Error('Сетевая ошибка при получении постов');
            return response.json();
        .then(data => displayPosts(data))
        .catch(error => console.error('Ошибка при получении постов:', error));
function displayPosts(posts) {
    const postsContainer = document.getElementById('postsContainer');
    postsContainer.innerHTML = ''; // Очищаем контейнер перед повторной загрузкой
    posts.forEach(post => {
        const postElement = document.createElement('div');
        postElement.className = 'post';
        postElement.innerHTML =
            <h3>${post.title}</h3>
            ${post.body}
            <button onclick="updatePost(${post.id})">Обновить</button>
            <button onclick="deletePost(${post.id})">Удалить</button>
        postsContainer.appendChild(postElement);
    });
```

```
document.getElementById('createPostForm').addEventListener('submit', function(event) {
   event.preventDefault(); // Предотвращаем перезагрузку страницы
   const title = document.getElementById('title').value;
const body = document.getElementById('body').value;
    console.log('Отправка поста:', { title, body });
    fetch('https://jsonplaceholder.typicode.com/posts', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        body: JSON.stringify({
            title: title,
            body: body,
            userId: 1 // Используем фиктивный userId
    .then(response => {
        if (!response.ok) {
            throw new Error('Ошибка при создании поста');
        return response.json();
    .then(data => {
        console.log('Пост успешно создан:', data);
        alert('Пост успешно создан!');
        fetchPosts(); // Обновляем список постов
    .catch(error => console.error('Ошибка при создании поста:', error));
```

```
function updatePost(id) {
   const newTitle = prompt('Введите новый заголовок');
   const newBody = prompt('Введите новый текст поста');
    if (newTitle && newBody) {
        fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
           method: 'PUT',
           headers: {
                'Content-Type': 'application/json'
           body: JSON.stringify({
               title: newTitle,
               body: newBody,
               userId: 1
        .then(response => {
            if (!response.ok) {
               throw new Error('Ошибка при обновлении поста');
           return response.json();
        .then(data => {
            console.log('Пост обновлен:', data);
            alert('Пост обновлен!');
           fetchPosts(); // Обновляем список постов
       .catch(error => console.error('Ошибка при обновлении поста:', error));
       alert('Поле не может быть пустым!');
```

```
// Удаление поста
function deletePost(id) {
    fetch(`https://jsonplaceholder.typicode.com/posts/${id}`, {
        method: 'DELETE'
    })
    .then(response => {
        if (!response.ok) {
            throw new Error('Ошибка при удалении поста');
        }
        alert('Пост удален!');
        fetchPosts(); // Обновляем список постов
    })
    .catch(error => console.error('Ошибка при удалении поста:', error));
}
```





Уведомление Поле не может быть пустым! Больше не показывать сообщения от этого сайта Закрыть Ввод данных Введите новый текст поста выфвфыв Больше не показывать сообщения от этого сайта Продолжить Отмена Уведомление Пост обновлен! Закрыть

ehenderit