Resource File: script.js

1. Executive Summary

A supporting file used by the application for configuration or data storage.

2. Code Logic & Functionality

Contains static data or configuration parameters read by the application at runtime.

3. Key Concepts & Definitions

- **Static Asset**: A file that is not generated dynamically (e.g., images, text files).
- **Configuration**: Settings that determine the behavior of the software.

4. Location Details

Path: `static\js\script.js` **Type**: .JS File

5. Source Code Preview (Snippet)

Running typical software analysis on this file:

```
/**
 * School Activity Booking System - JavaScript
 * Client-side interactivity and validation
 */

// Utility Functions
function showAlert(message, type = 'info') {
    const alertDiv = document.createElement('div');
    alertDiv.className = `alert alert-${type} alert-dismissible fade sh
    alertDiv.role = 'alert';
    alertDiv.innerHTML = `
        ${message}
        <button type="button" class="btn-close" data-bs-dismiss="alert"
    `;

    const container = document.querySelector('main') || document.queryS
    if (container) {
        container.insertBefore(alertDiv, container.firstChild);

        // Auto dismiss after 5 seconds
        setTimeout(() => {
            alertDiv.remove();
        }, 5000);
    }
}
```

```
function getCsrfToken() {
    const meta = document.querySelector('meta[name="csrf-token"]');
    return meta ? meta.getAttribute('content') : '';
}

function validateEmail(email) {
    const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(email);
}

function formatPrice(price) {
    return new Intl.NumberFormat('en-US', {
        style: 'currency',
        currency: 'USD'
    }).format(price);
}

function formatDate(dateString) {
    const options = { year: 'numeric', month: 'long', day: 'numeric' };
    return new Date(dateString).toLocaleDateString('en-US', options);
}

// Form Validation
document.addEventListener('DOMContentLoaded', function() {
    // Validate registration form
    const registerForm = document.getElementById('registerForm');
    if (registerForm) {
        registerForm.addEventListener('submit', function(e) {
            const password = document.getElementById('password').value;
            const confirmPassword = document.getElementById('confirm_pa

            if (password.length < 8) {
                e.preventDefault();
                showAlert('Password must be at least 8 characters', 'wa
                return false;
            }

            if (password !== confirmPassword) {
                e.preventDefault();
                showAlert('Passwords do not match', 'warning');
                return false;
            }
        });
    }

    // Initialize date pickers with minimum date as today
    const dateInputs = document.querySelectorAll('input[type="date"]');
    const today = new Date().toISOString().split('T')[0];
    dateInputs.forEach(input => {
        input.setAttribute('min', today);
    });
```

```
    // Add Bootstrap validation feedback
    const forms = document.querySelectorAll('.needs-validation');
    Array.from(forms).forEach(form => {
        form.addEventListener('submit', event => {
            if (!form.checkValidity()) {
                event.preventDefault();
                event.stopPropagation();
            }
            form.classList.add('was-validated');
        }, false);
    });
});

// Debounce function for search/filter operations
function debounce(func, delay) {
    let timeoutId;
    return function(...args) {
        clearTimeout(timeoutId);
        timeoutId = setTimeout(() => func.apply(this, args), delay);
    };
}

// Loading spinner
function showSpinner(element, show = true) {
    if (show) {
        element.innerHTML = '<span class="spinner-border spinner-border
        element.disabled = true;
    } else {
        element.disabled = false;
    }
}

// API Helper Functions
async function fetchJSON(url, options = {}) {
    try {
        const headers = {
            'Content-Type': 'application/json',
            'X-CSRFToken': getCsrfToken(),
            ...options.headers
        };

        const response = await fetch(url, {
            ...options,
            headers
        });

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

        return await response.json();
    } catch (error) {
        console.error('Fetch error:', error);
```

```
        showAlert('An error occurred. Please try again.', 'danger');
        throw error;
    }
}

// Booking Management
async function bookActivityAsync(activityId) {
    const childId = document.getElementById('bookingChild').value;
    const bookingDate = document.getElementById('bookingDate').value;

    if (!childId) {
        showAlert('Please select a child', 'warning');
        return;
    }

    if (!bookingDate) {
        showAlert('Please select a date', 'warning');
        return;
    }

... [Code Truncated for Documentation Readability - See Source File for
```