```
document.addEventListener('DOMContentLoaded', function() {
    // Handle drag & drop functionality
    const uploadArea = document.querySelector('.document-upload-area');
    const fileInput = document.getElementById('document');
    const uploadForm = document.getElementById('uploadForm');
    const uploadStatus = document.getElementById('uploadStatus');
    const uploadProgress = document.getElementById('uploadProgress');
    const uploadBtn = document.getElementById('uploadBtn');

    if (uploadArea && fileInput) {
        // Prevent default drag behaviors
        ['dragenter', 'dragover', 'dragleave', 'drop'].forEach(eventName => {
            uploadArea.addEventListener(eventName, preventDefaults, false);
        });

        function preventDefaults(e) {
            e.preventDefault();
            e.stopPropagation();
        }

        // Highlight drop area when item is dragged over it
        ['dragenter', 'dragover'].forEach(eventName => {
            uploadArea.addEventListener(eventName, highlight, false);
        });

        ['dragleave', 'drop'].forEach(eventName => {
            uploadArea.addEventListener(eventName, unhighlight, false);
        });

        function highlight() {
            uploadArea.classList.add('bg-light');
        }

        function unhighlight() {
            uploadArea.classList.remove('bg-light');
        }

        // Handle dropped files
        uploadArea.addEventListener('drop', handleDrop, false);

        function handleDrop(e) {
            const dt = e.dataTransfer;
            const files = dt.files;

            if (files.length > 0) {
                fileInput.files = files;

                // Show file name in a meaningful way
                const fileName = files[0].name;
                const fileSize = formatBytes(files[0].size);

                // You could add an element to display the selected file info
                const fileInfoEl = document.createElement('div');
                fileInfoEl.className = 'selected-file-info mt-3 p-2 border rounded';
                fileInfoEl.innerHTML = `
                    <div class="d-flex align-items-center">
                        <i class="fas fa-file-alt text-primary me-2"></i>
                        <div>
                            <strong>${fileName}</strong>
                            <div class="text-muted small">${fileSize}</div>
                        </div>
                        <button type="button" class="btn-close ms-auto" aria-label="Remov
e file"></button>
                    </div>
                `;

                // Replace any existing file info
                const existingFileInfo = uploadArea.querySelector('.selected-file-info');
                if (existingFileInfo) {
                    existingFileInfo.remove();
```

```
                }

                uploadArea.appendChild(fileInfoEl);

                // Add event listener to the close button
                const closeBtn = fileInfoEl.querySelector('.btn-close');
                if (closeBtn) {
                    closeBtn.addEventListener('click', function() {
                        fileInput.value = '';
                        fileInfoEl.remove();
                    });
                }
            }
        }
    }

    // Handle form submission and show progress
    if (uploadForm) {
        uploadForm.addEventListener('submit', function(e) {
            if (fileInput.files.length === 0) {
                e.preventDefault();
                showAlert('Please select a file to upload', 'danger');
                return;
            }

            // Show the upload status
            uploadStatus.classList.remove('d-none');
            uploadBtn.disabled = true;

            // Simulate progress (in a real app, this would use XHR or Fetch API)
            simulateProgress();
        });
    }

    function simulateProgress() {
        let progress = 0;
        const interval = setInterval(() => {
            progress += Math.random() * 10;
            if (progress > 100) progress = 100;

            uploadProgress.style.width = `${progress}%`;

            if (progress === 100) {
                clearInterval(interval);
            }
        }, 300);
    }

    function showAlert(message, type) {
        const alertHtml = `
            <div class="alert alert-${type} alert-dismissible fade show" role="alert">
                ${message}
                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-labe
l="Close"></button>
            </div>
        `;

        // Insert alert before the form
        uploadForm.insertAdjacentHTML('beforebegin', alertHtml);
    }

    function formatBytes(bytes, decimals = 2) {
        if (bytes === 0) return '0 Bytes';

        const k = 1024;
        const dm = decimals < 0 ? 0 : decimals;
        const sizes = ['Bytes', 'KB', 'MB', 'GB', 'TB'];

        const i = Math.floor(Math.log(bytes) / Math.log(k));
```

```
        return parseFloat((bytes / Math.pow(k, i)).toFixed(dm)) + ' ' + sizes[i];
    }
});
```