Metrics

There are 52 functions in this file.

Function with the largest signature take 3 arguments, while the median is 1.

Largest function has 42 statements in it, while the median is 1.

The most complex function has a cyclomatic complexity value of 6 while the median is 1.5.

```
1    /*jshint esversion: 6 */
2    /*global bootstrap, Stripe */
3
4    // Utility function to get a cookie value by name (used for CSRF tokens)
5    function getCookie(name) {
6        let cookieValue = null;
7        if (document.cookie && document.cookie !== '') {
8            const cookies = document.cookie.split(';');
9            for (let i = 0; i < cookies.length; i++) {
10               const cookie = cookies[i].trim();
11               if (cookie.substring(0, name.length + 1) === (name + '=')) {
12                   cookieValue = decodeURIComponent(cookie.substring(name.length + 1)
13                   break;
14               }
15           }
16       }
17       return cookieValue;
18   }
19
20   // Quantity controls for cart items
21   function EnableDisableQuantityChange(itemId) {
22       let input = document.getElementById(`id_qty_${itemId}`);
23       if (!input) return;
24       let currentValue = parseInt(input.value);
25       let minValue = parseInt(input.getAttribute('min')) || 1;
26       let maxValue = parseInt(input.getAttribute('max'));
27       let minusDisabled = currentValue <= minValue;
28       let plusDisabled = currentValue >= maxValue;
29       let decBtn = document.getElementById(`decrement-qty_${itemId}`);
30       let incBtn = document.getElementById(`increment-qty_${itemId}`);
31       if (decBtn) decBtn.disabled = minusDisabled;
32       if (incBtn) incBtn.disabled = plusDisabled;
33   }
34
35   // Tooltip for quantity input validation
36   function showTooltip(input, message) {
37       let oldTip = input.parentElement.querySelector('.qty-tooltip');
38       if (oldTip) oldTip.remove();
39       let tooltip = document.createElement('div');
40       tooltip.className = 'qty-tooltip';
41       tooltip.textContent = message;
42       tooltip.style.top = (input.offsetTop - 30) + 'px';
43       tooltip.style.left = input.offsetLeft + 'px';
44       input.parentElement.style.position = 'relative';
45       input.parentElement.appendChild(tooltip);
46       setTimeout(function () {
47           tooltip.remove();
48       }, 2000);
49   }
50
51   // Helper function to poll for order and redirect to success page
52   function pollForOrderAndRedirectByPid(pid, attempts = 0, maxAttempts = 30) {
53       fetch(`/checkout/get-order-number-by-pid/${pid}/`)
54           .then(response => response.json())
55           .then(data => {
56               if (data.order_number) {
57                   window.location.href = `/checkout/checkout_success/${data.order_nu
58               } else if (attempts < maxAttempts) {
59                   setTimeout(function () {
60                       pollForOrderAndRedirectByPid(pid, attempts + 1, maxAttempts);
61                   }, 2000); // 2 seconds
62               } else {
63                   alert('Order processing is taking longer than expected. Please che
64                   window.location.href = '/checkout/';
65               }
66           })
67           .catch(() => {
68               if (attempts < maxAttempts) {
69                   setTimeout(function () {
70                       pollForOrderAndRedirectByPid(pid, attempts + 1, maxAttempts);
71                   }, 2000);
72               } else {
73                   alert('Order processing is taking longer than expected. Please che
74                   window.location.href = '/checkout/';
75               }
76           });
77   }
78
79   // Main DOMContentLoaded event for all interactive features
80   document.addEventListener('DOMContentLoaded', function () {
81       // Filter dropdowns for releases page
82       const allFiltersElem = document.getElementById('all-filters-data');
83       if (allFiltersElem) {
84           const allFilters = JSON.parse(allFiltersElem.textContent);
85           const genreSelect = document.getElementById('genre-select');
86           const subgenreSelect = document.getElementById('subgenre-select');
87           const directorSelect = document.getElementById('director-select');
88           const decadeSelect = document.getElementById('decade-select');
89           const resetBtn = document.getElementById('reset-filters-btn');
90
91           // Dynamically update dropdown options based on current selections
92           const updateDropdowns = function() {
```