JS Validation

Breadscrumb.js

```
1   // JavaScript code for generating breadcrumbs
2   document.addEventListener('DOMContentLoaded', function () {
3       generateBreadcrumbs();
4   });
5
6   function generateBreadcrumbs() {
7       var breadcrumbContainer = document.getElementById('breadcrumb-container');
8       var pathArray = window.location.pathname.split('/').filter(function (path) {
9           return path !== '';
10      });
11
12      var breadcrumbHtml = '<a href="/">Home</a>';
13      var path = '/';
14      for (var i = 0; i < pathArray.length; i++) {
15          path += pathArray[i] + '/';
16          breadcrumbHtml += ' > <a href="' + path + '">' + pathArray[i].charAt(0).toU
17      }
18
19      breadcrumbContainer.innerHTML = breadcrumbHtml;
20  }
```

CONFIGURE

Metrics

There are 3 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.
Largest function has 8 statements in it, while the median is 1.
The most complex function has a cyclomatic complexity value of 2 while the median is 1.

JSHint
version 2.13.6

About
Documentation
Install
Contribute
Blog

Star rating

```
1   $(document).ready(function () {
2       let selectedRating = 0;
3
4       $('.star-rating i').click(function () {
5           const rating = $(this).data('rating');
6           selectedRating = rating;
7           updateStars(selectedRating);
8       });
9
10      function updateStars(rating) {
11          $('.star-rating i').removeClass('checked');
12          $('.star-rating i').each(function () {
13              if ($(this).data('rating') <= rating) {
14                  $(this).addClass('checked');
15              }
16          });
17          $('#rating').val(rating);
18      }
19  });
```

CONFIGURE

Metrics

There are 4 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.
Largest function has 3 statements in it, while the median is 3.
The most complex function has a cyclomatic complexity value of 2 while the median is 1.

Two warnings

2   'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
5   'const' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).

JSHint
version 2.13.6

About
Documentation
Install
Contribute
Blog

Product list

```
1   document.addEventListener('DOMContentLoaded', function () {
2       // Find all elements with the 'delete-product' class
3       var deleteButtons = document.querySelectorAll('.delete-product');
4
5       // Add a click event listener to each delete button
6       deleteButtons.forEach(function (button) {
7           button.addEventListener('click', function (event) {
8               event.preventDefault();
9
10              // Get the product ID from the data-product-id attribute
11              var productId = button.getAttribute('data-product-id');
12
13              // Display a confirmation dialog
14              var isConfirmed = confirm('Are you sure you want to delete this pro
15
16              // If the user confirms, redirect to the delete URL
17              if (isConfirmed) {
18                  // Construct the delete URL using the url template tag
19                  var deleteUrl = '{% url "products:delete_product" product_id=0 
20                      '0', productId);
21                  window.location.href = deleteUrl;
22              }
23          });
24      });
25  });
```

CONFIGURE

Metrics

There are 3 functions in this file.
Function with the largest signature take 1 arguments, while the median is 1.
Largest function has 6 statements in it, while the median is 2.
The most complex function has a cyclomatic complexity value of 2 while the median is 1.

JSHint
version 2.13.6

About
Documentation
Install
Contribute
Blog

Product detail

```
1   <script>
2         function changeMainImage(event) {
3             var thumbnail = event.target;
4             if (thumbnail.classList.contains('thumbnail')) {
5                 var mainImage = document.getElementById('mainImage');
6                 var newImageSource = thumbnail.getAttribute('data-main-image');
7                 mainImage.src = newImageSource;
8             }
9         }
10  </script>
11  <script type="text/javascript">
12        // Disable +/- buttons outside 1-99 range
13        function handleEnableDisable(itemId) {
14            var currentValue = parseInt($(`#id_qty_${itemId}`).val());
15            var minusDisabled = currentValue < 2;
16            var plusDisabled = currentValue > 98;
17            $(`#decrement-qty_${itemId}`).prop('disabled', minusDisabled);
18            $(`#increment-qty_${itemId}`).prop('disabled', plusDisabled);
19        }
20
21        // Ensure proper enabling/disabling of all inputs on page load
22        var allQtyInputs = $('.qty_input');
23        for (var i = 0; i < allQtyInputs.length; i++) {
24            var itemId = $(allQtyInputs[i]).data('item_id');
25            handleEnableDisable(itemId);
26        }
27
28        // Check enable/disable every time the input is changed
29        $('.qty_input').change(function () {
30            var itemId = $(this).data('item_id');
31            handleEnableDisable(itemId);
32        });
33
34        // Increment quantity
35        $('.increment-qty').click(function (e) {
36            e.preventDefault();
37            var closestInput = $(this).closest('.input-group').find('.qty_input')[
38            var currentValue = parseInt($(closestInput).val());
39            $(closestInput).val(currentValue + 1);
40            var itemId = $(this).data('item_id');
41            handleEnableDisable(itemId);
42        });
43
44        // Decrement quantity
45        $('.decrement-qty').click(function (e) {
46            e.preventDefault();
47            var closestInput = $(this).closest('.input-group').find('.qty_input')[
48            var currentValue = parseInt($(closestInput).val());
```

Products

```
1   // Get the button
2   var mybutton = document.getElementById("back-to-top-btn");
3
4   // When the user scrolls down 20px from the top of the document, show the button
5   window.onscroll = function () {
6       scrollFunction();
7   };
8
9   function scrollFunction() {
10      if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20)
11          mybutton.style.display = "block";
12      } else {
13          mybutton.style.display = "none";
14      }
15  }
16
17  // When the user clicks on the button, scroll to the top of the document
18  function topFunction() {
19      document.body.scrollTop = 0; // For Safari
20      document.documentElement.scrollTop = 0; // For Chrome, Firefox, IE, and Oper
21  }
22  </script>
23  <script>
24      // Function to get the CSRF token from cookies
25      function getCookie(name) {
26          var cookieValue = null;
27          if (document.cookie && document.cookie !== '') {
28              var cookies = document.cookie.split(';');
29              for (var i = 0; i < cookies.length; i++) {
30                  var cookie = cookies[i].trim();
31                  // Check if this cookie string begins with the name we want
32                  if (cookie.substring(0, name.length + 1) === (name + '=')) {
33                      cookieValue = decodeURIComponent(cookie.substring(name.length +
34                      break;
35                  }
36              }
37          }
38          return cookieValue;
39      }
40  </script>
41
42  <script>
43      // Add click event to all elements with the "addToWishlistBtn" class
44      $('.addToWishlistBtn').click(function () {
45          var productId = $(this).attr('data-product-id');
46          console.log('Add to Wishlist clicked for product ID:', productId);
47
48          // Perform the AJAX request to add the product to the wishlist
49          $.ajax({
50              type: "GET",
51              url: `/wishlist/pluswishlist/?prod_id=${productId}`,
52              success: function (data) {
53                  console.log('AJAX Success:', data);
54                  // Handle success, e.g., show a notification or update UI
55              },
56              error: function (error) {
57                  console.log('AJAX Error:', error);
58                  // Handle error, e.g., show an error message
59              }
```