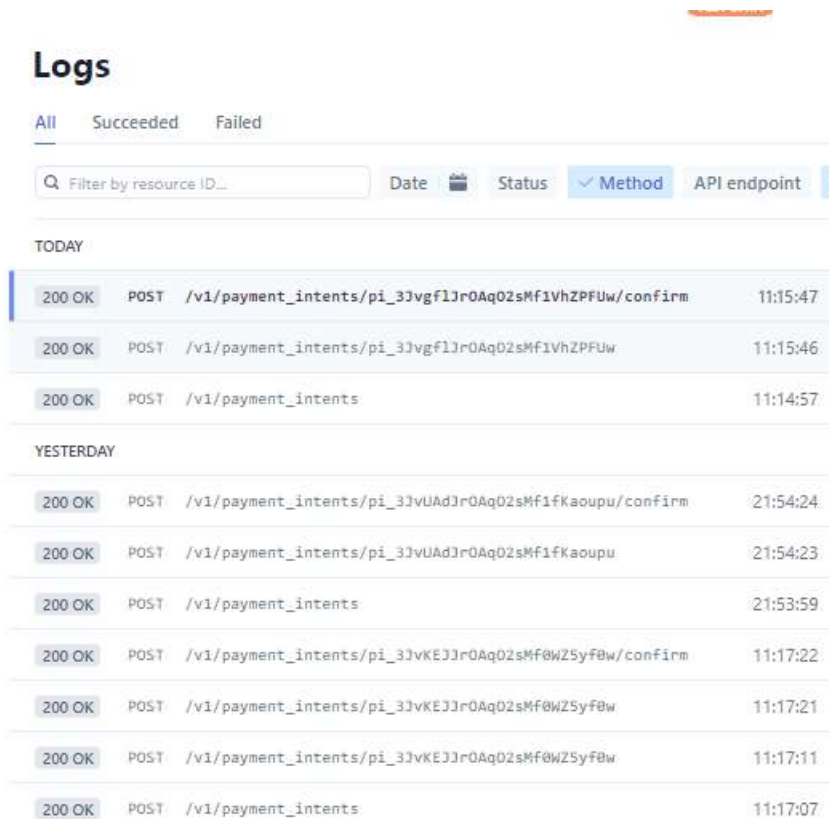


Automated Testing

Stripe Logs – there are no errors in the logs, all payments go through correctly.



The screenshot shows the Stripe Logs interface. At the top, there's a 'Logs' header with tabs for 'All', 'Succeeded', and 'Failed'. Below this is a search bar 'Filter by resource ID...' and a table of log entries. The table has columns for 'Date', 'Status', 'Method', and 'API endpoint'. The entries are grouped by 'TODAY' and 'YESTERDAY'. All entries show a '200 OK' status and a 'POST' method. The API endpoints are for payment intent confirmations and creation.

Date	Status	Method	API endpoint
TODAY			
	200 OK	POST	/v1/payment_intents/pi_3Jvgf1Jr0Aq02sMf1VhZPFUw/confirm
	200 OK	POST	/v1/payment_intents/pi_3Jvgf1Jr0Aq02sMf1VhZPFUw
	200 OK	POST	/v1/payment_intents
YESTERDAY			
	200 OK	POST	/v1/payment_intents/pi_3JvUAd3r0Aq02sMf1fKaoupu/confirm
	200 OK	POST	/v1/payment_intents/pi_3JvUAd3r0Aq02sMf1fKaoupu
	200 OK	POST	/v1/payment_intents
	200 OK	POST	/v1/payment_intents/pi_3JvKE3Jr0Aq02sMf0WZ5yf0w/confirm
	200 OK	POST	/v1/payment_intents/pi_3JvKE3Jr0Aq02sMf0WZ5yf0w
	200 OK	POST	/v1/payment_intents/pi_3JvKE3Jr0Aq02sMf0WZ5yf0w
	200 OK	POST	/v1/payment_intents

HTML – all pass except basket (see below)

1. Homepage

Document checking completed. No errors or warnings to show.

2. Product page

Document checking completed. No errors or warnings to show.

3. Product detail page

Document checking completed. No errors or warnings to show.

4. Add product page

Document checking completed. No errors or warnings to show.

5. Basket

The basket view doesn't pass because of mobile screen, there are 2 displays. One will always be hidden depending on device size, however the modify/delete buttons do the exact same job, so the IDs both have to be there – as one of the below forms will always be

hidden due to “d-block-none” and “d-block” attributes in parent div.

- Error** Duplicate ID `form`.
From line 205, column 37 to line 205, column 124
`<form class="form update-form" id="form" method="POST" action="/basket/modify/NS00010/">`
- Warning** The first occurrence of ID `form` was here.
From line 149, column 37 to line 149, column 124
`<form class="form update-form" id="form" method="POST" action="/basket/modify/NS00010/">`
- Error** Duplicate ID `remove_NS00010`.
From line 234, column 37 to line 234, column 114
``
- Warning** The first occurrence of ID `remove_NS00010` was here.
From line 180, column 37 to line 180, column 114
``

6. Checkout page

Document checking completed. No errors or warnings to show.

7. Checkout success page

Document checking completed. No errors or warnings to show.

8. Profile page

Document checking completed. No errors or warnings to show.

9. Change password page

Document checking completed. No errors or warnings to show.

10. Reset password page

Document checking completed. No errors or warnings to show.

11. Set password page (for social accounts)

Document checking completed. No errors or warnings to show.

12. Change email address page

Document checking completed. No errors or warnings to show.

Javascript – both pass with no major issues.

1. Stripe.js

<pre>74 name: \$.trim(form.full_name.value), 75 phone: \$.trim(form.phone_number.value), 76 email: \$.trim(form.email.value), 77 address: { 78 line1: \$.trim(form.street_address1.value), 79 line2: \$.trim(form.street_address2.value), 80 city: \$.trim(form.town_or_city.value), 81 state: \$.trim(form.county.value), 82 country: \$.trim(form.country.value), 83 } 84 }, 85 shipping: { 86 name: \$.trim(form.full_name.value), 87 phone: \$.trim(form.phone_number.value), 88 address: { 89 line1: \$.trim(form.street_address1.value), 90 line2: \$.trim(form.street_address2.value), 91 city: \$.trim(form.town_or_city.value), 92 state: \$.trim(form.county.value), 93 postal_code: \$.trim(form.postcode.value), 94 country: \$.trim(form.country.value), 95 } 96 } 97 }).then(function(result) { 98 if (result.error) { 99 var errorDiv = document.getElementById('card-errors'); 100 var html = ` 101 102 <i class="fas fa-times"></i> 103 104 \${result.error.message}`; 105 \$(errorDiv).html(html); 106 \$('#payment-form').fadeToggle(100); 107 \$('#loading-overlay').fadeToggle(100); 108 card.update({ 'disabled': false }); 109 \$('#submit-button').attr('disabled', false); 110 } else { 111 console.log("Submitted"); 112 if (result.paymentIntent.status === 'succeeded') { 113 form.submit(); 114 } 115 } 116 }).fail(function () { 117 // just reload the page, the error will be in django messages 118 location.reload(); 119 });</pre>	<p>CONFIGURE</p> <p>Metrics</p> <p>There are 5 functions in this file. Function with the largest signature take 1 arguments, while the median is 1. Largest function has 11 statements in it, while the median is 5. The most complex function has a cyclomatic complexity value of 3 while the median is 1.</p> <p>Five warnings</p> <p>37 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 97 Expected `)` to match `{` from line 70 and instead saw `)`. 100 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 116 Expected `)` and instead saw `.`. 116 Missing semicolon.</p> <p>Two undefined variables</p> <p>8 \$ 9 \$ 43 \$ 55 \$ 56 \$ 57 \$ 59 \$</p>
---	---

2. Script.js

<pre>7 \$.decrement-qty_\${productsku}`).prop('disabled', minDisabled); 8 \$.increment-qty_\${productsku}`).prop('disabled', plusDisabled); 9 } 10 11 // Ensure proper enabling/disabling of all inputs on page load 12 var allQtyInputs = \$('qty_input'); 13 for (var i = 0; i < allQtyInputs.length; i++){ 14 var productsku = \$(allQtyInputs[i]).data('product_sku'); 15 handleEnableDisable(productsku); 16 } 17 18 // Check enable/disable every time the input is changed 19 \$\$('.qty_input').change(function() { 20 var productsku = \$(this).data('product_sku'); 21 handleEnableDisable(productsku); 22 }); 23 24 // Increment quantity 25 \$\$('.increment-qty').click(function(e) { 26 e.preventDefault(); 27 var productsku = \$(this).data('product_sku'); 28 var closestInput = \$(this).closest('.input-group').find('qty_input')[0]; 29 var allQuantityInputs = \$('input-group-\${productsku} input[name=quantity]'); 30 var currentValue = parseInt(closestInput.val()); 31 \$(allQuantityInputs).val(currentValue + 1); 32 handleEnableDisable(productsku); 33 }); 34 35 // Decrement quantity 36 \$\$('.decrement-qty').click(function(e) { 37 e.preventDefault(); 38 var productsku = \$(this).data('product_sku'); 39 var closestInput = \$(this).closest('.input-group').find('qty_input')[0]; 40 var allQuantityInputs = \$('input-group-\${productsku} input[name=quantity]'); 41 var currentValue = parseInt(closestInput.val()); 42 \$(allQuantityInputs).val(currentValue - 1); 43 handleEnableDisable(productsku); 44 }); 45 46 // Update quantity on click 47 \$\$('.update-link').click(function(e) { 48 var form = \$(this).parent().prev().children('update-form'); 49 form.submit(); 50 }); 51 52 // Show toast messages 53 \$\$('.toast').toast('show');</pre>	<p>CONFIGURE</p> <p>Metrics</p> <p>There are 5 functions in this file. Function with the largest signature take 1 arguments, while the median is 1. Largest function has 7 statements in it, while the median is 5. The most complex function has a cyclomatic complexity value of 1 while the median is 1.</p> <p>Five warnings</p> <p>4 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 8 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 9 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 30 'template literal syntax' is only available in ES6 (use 'esversion: 6'). 41 'template literal syntax' is only available in ES6 (use 'esversion: 6').</p> <p>One undefined variable</p> <p>4 \$ 8 \$ 9 \$ 13 \$ 15 \$ 20 \$</p>
--	--

CSS – passes with no issues

1. Base.css

W3C CSS Validator results for TextArea (CSS level 3 + SVG)

Congratulations! No Error Found.

This document validates as **CSS level 3 + SVG** !

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:



Python – checked using www.pep8online.com

All below apps were tested through the PEP 8 validator and are now showing no errors:

All right

1. Manage.py
2. Custom_storages.py
3. Gamestore app
 - a. __init__.py
 - b. Adapter.py
 - c. Asgi.py
 - d. Settings.py
 - e. Urls.py
 - f. Wsgi.py
4. Home app
 - a. __init__.py
 - b. Admin.py
 - c. Apps.py
 - d. Models.py
 - e. Tests.py
 - f. Urls.py
 - g. Views.py
5. Products app
 - a. __init__.py
 - b. Admin.py
 - c. Apps.py
 - d. Forms.py
 - e. Models.py
 - f. Tests.py
 - g. Urls.py
 - h. Views.py
 - i. Widgets.py
6. Profiles app
 - a. __init__.py
 - b. Admin.py
 - c. Apps.py
 - d. Forms.py
 - e. Models.py
 - f. Tests.py
 - g. Urls.py
 - h. Views.py
7. Basket app
 - a. __init__.py
 - b. Admin.py

- c. Apps.py
- d. Context.py
- e. Models.py
- f. Tests.py
- g. Urls.py
- h. Views.py

8. Checkout app

- a. __init__.py
- b. Admin.py
- c. Apps.py
- d. Forms.py
- e. Models.py
- f. Signals.py
- g. Tests.py
- h. Urls.py
- i. Views.py
- j. Webhook_handler.py
- k. Webhooks.py