

## Issues Found During Exploration

1. WordPress debug messages visible - Error notices for the woocommerce-smart-coupons plugin appear at the top of every page, creating an unprofessional appearance.
2. Placeholder product images - All products use generic placeholder images instead of actual product photos.
3. SQL injection vulnerability - Checkout form fields lack proper sanitization, potentially allowing SQL injection attacks.
4. No product categories - The shop lacks category navigation, making it difficult for users to browse effectively.
5. Inconsistent payment method behavior - PostFinance Card works, but SEPA Direct Debit fails with unhelpful error messages.
6. No input validation for special characters - Form fields accept potentially harmful special characters without proper validation.
7. WooCommerce deprecation notice visible - Technical deprecation message about outdated hooks is displayed at the bottom of the checkout page, exposing server file paths and implementation details.

## Bug Tickets

### Bug Ticket #1: WordPress Debug Messages Exposed in Production Environment

Priority: High

Severity: Major

Reported By: QA Testing Team

Date Reported: March , 2025

Affected Component: Core WordPress Configuration

Test Environment: Chrome 131.0.6575.205 / Windows 11

Reproducibility: 100% - Always

#### Description:

The WooCommerce showcase website is displaying WordPress debug/error messages at the top of every page. These technical error messages are visible to all users, revealing internal implementation details and creating an unprofessional appearance for customers.

#### Steps to Reproduce:

1. Navigate to any page on <https://woocommerce.showcase-wallee.com/>
2. Observe the top of the page where multiple error notices are displayed
3. The specific error shown is: "Notice: Function `_load_textdomain_just_in_time` was called incorrectly. Translation loading for the `woocommerce-smart-coupons` domain was triggered too early. This is usually an indicator for some code in the plugin or theme running too early. Translations should be loaded at the `init` action or later."
4. This error appears twice on every page of the website

#### Expected Result:

Debug messages should be hidden from end users in a production environment. Error logging should be configured to write to server logs rather than displaying on the frontend interface.

### Actual Result:

WordPress error notices are displayed prominently at the top of every page, exposing implementation details and creating a poor user experience.

### Business Impact:

This issue has several negative impacts:


1. Professional Appearance: Error messages make the site look broken or incomplete, reducing customer trust
2. Security Implications: Exposing internal details could provide information useful to potential attackers
3. User Experience: Technical messages are confusing to typical users and take up valuable screen space
4. Brand Perception: Public-facing error messages suggest poor quality control and can damage brand reputation

### Screenshots:

**Notice:** Function `_load_textdomain_just_in_time` was called **incorrectly**. Translation loading for the `woocommerce-smart-coupons` domain was triggered too early. This is usually an indicator for some code in the plugin or theme running too early. Translations should be loaded at the `init` action or later. Please see [Debugging in WordPress](#) for more information. (This message was added in version 6.7.0.) in `/var/www/dist/wp-includes/functions.php` on line **6114**


**Notice:** Function `_load_textdomain_just_in_time` was called **incorrectly**. Translation loading for the `woocommerce-smart-coupons` domain was triggered too early. This is usually an indicator for some code in the plugin or theme running too early. Translations should be loaded at the `init` action or later. Please see [Debugging in WordPress](#) for more information. (This message was added in version 6.7.0.) in `/var/www/dist/wp-includes/functions.php` on line **6114**

Demo Shop

Cart Checkout My account Shop 

## Shop

Showing all 8 results

Default sorting 



### Bug Ticket #2: SQL Injection Vulnerability in Checkout Form

Priority: Critical

Severity: Critical

Reported By: QA Testing Team

Date Reported: March, 2025

Affected Component: Checkout Form Processing

Test Environment: Chrome 131.0.6575.205 / Windows 11

Reproducibility: 100% - Always

### Description:

The checkout form has insufficient input sanitization, potentially allowing SQL injection attacks through multiple form fields. Testing with common SQL injection patterns revealed that the system does not properly validate or sanitize user input before processing it.

**Steps to Reproduce:**

1. Add any product to cart
2. Proceed to checkout
3. Enter SQL injection test patterns in form fields:
  - First Name: ' OR 1=1 --
  - Last Name: Smith
  - Email: test@example.com
  - Address: 123 Main St
4. Complete the checkout process
5. The system accepts the input without validation or sanitization

**Expected Result:**

The system should detect and reject potential SQL injection attempts, either by sanitizing the input or displaying a validation error.

**Actual Result:**

The system accepts SQL injection test patterns without any validation, potentially allowing attackers to execute malicious SQL queries against the database.

**Business Impact:**

1. Data Security: SQL injection vulnerabilities could allow unauthorized access to customer data, including personal and payment information
2. System Integrity: Attackers could potentially modify or delete database records
3. Regulatory Compliance: This vulnerability could result in data breaches that violate GDPR, PCI-DSS, and other regulations
4. Business Reputation: Security breaches resulting from this vulnerability could seriously damage brand reputation and customer trust

**Bug Ticket #3: Inconsistent Payment Method Functionality**

Priority: High

Severity: Major

Reported By: QA Testing Team

Date Reported: March, 2025

Affected Component: Payment Gateway Integration

Test Environment: Chrome 131.0.6575.205 / Windows 11

Reproducibility: 100% - Always

**Description:**

The WooCommerce Wallee payment gateway integration shows inconsistent behavior across different payment methods. While PostFinance Card works correctly with a proper simulation interface, SEPA Direct Debit payment method fails with unhelpful error messages that don't guide users on how to proceed.

**Steps to Reproduce:**

1. Add any product to cart
2. Proceed to checkout
3. Fill in all required billing fields with valid data

4. Select Direct Debit (SEPA) as payment method
5. Enter an IBAN (tried multiple formats including "CH93 0076 2011 6238 5295 7")
6. Click "Pay" button
7. Observe error message "The provided IBAN is not accepted"

#### Expected Result:

All payment methods should work consistently in the test environment. If specific test data is required (like test IBANs), clear guidance should be provided either through tooltips, help text, or proper error messages that explain exactly what format is expected.

#### Actual Result:

When using SEPA Direct Debit, the system displays the error message "The provided IBAN is not accepted" without indicating what would constitute a valid IBAN for testing purposes. This contrasts with PostFinance Card payment, which provides a clear simulation interface with sample data.

#### Business Impact:

This issue prevents customers from using SEPA Direct Debit as a payment option, which could lead to lost sales if customers prefer this payment method. For merchants who want to offer this payment option, this inconsistency creates a poor testing experience and could lead to support tickets and confusion when implementing in production.

#### Screenshots:

**Notice:** Function `_load_textdomain_just_in_time` was called **incorrectly**. Translation loading for the `woocommerce-smart-coupons` domain was triggered too early. This is usually an indicator for some code in the plugin or theme running too early. Translations should be loaded at the `init` action or later. Please see [Debugging in WordPress](#) for more information. (This message was added in version 6.7.0.) in `/var/www/dist/wp-includes/functions.php` on line **6114**

**Notice:** Function `_load_textdomain_just_in_time` was called **incorrectly**. Translation loading for the `woocommerce-smart-coupons` domain was triggered too early. This is usually an indicator for some code in the plugin or theme running too early. Translations should be loaded at the `init` action or later. Please see [Debugging in WordPress](#) for more information. (This message was added in version 6.7.0.) in `/var/www/dist/wp-includes/functions.php` on line **6114**

The screenshot shows the top of a WooCommerce checkout page. At the top, there is a red error message box that says "The provided IBAN is not accepted." Below this, the page header includes "Demo Shop" on the left and "Cart Checkout My account Shop" on the right. The main heading is "Checkout". The checkout process is divided into two columns. The left column is titled "1. Contact information" and includes a link "Already have an account? [Log in.](#)". Below this, it says "We'll use this email to send you details and updates about your order." There is a text input field for "Email address" containing "gvvgcgcg@erfg.com". At the bottom of this section is a checkbox labeled "Create an account?". The right column is titled "Order summary" and shows a list of items: "2 Glasses" for "€12.00", with a total of "€24.00". Below the order summary is a link "Add a coupon".

#### Bug Ticket 4: WooCommerce Deprecation Notice Visible on Checkout Page

Priority: High

Severity: Major

Reported By: QA Testing Team

Date Reported: March, 2025

Affected Component: WooCommerce Checkout

Test Environment: Chrome 131.0.6575.205 / Windows 11

Reproducibility: 100% - Always

### Description:

The website is displaying a WooCommerce deprecation notice at the bottom of the checkout page. This technical error message is visible to all users, revealing internal implementation details and creating an unprofessional appearance for customers.

### Steps to Reproduce:

1. Navigate to <https://woocommerce.showcase-wallee.com/>
2. Add any product to cart
3. Proceed to checkout
4. Scroll to the bottom of the checkout page
5. Observe the deprecation notice displayed at the bottom: "Deprecated: Hook `__experimental_woocommerce_blocks_checkout_update_order_meta` is deprecated since version 6.3.0! Use `woocommerce_store_api_checkout_update_order_meta` instead. This action was deprecated in WooCommerce Blocks version 6.3.0. Please use `woocommerce_store_api_checkout_update_order_meta` instead. in `/var/www/dist/wp-includes/functions.php` on line 6114"

### Expected Result:

Deprecation notices should not be displayed to end users in a production environment. These technical messages should be hidden and only logged in development environments.

### Actual Result:

A detailed WooCommerce deprecation notice is displayed at the bottom of the checkout page, exposing implementation details including file paths and function names.

### Business Impact:

1. Professional Appearance: Error messages make the site look broken or unprofessional, reducing customer trust
2. Security Implications: The message reveals server file paths (`/var/www/dist/wp-includes/functions.php`) that could be useful to potential attackers
3. User Experience: Technical messages are confusing to typical users and create a poor impression
4. Customer Confidence: Visible error messages may cause customers to question the reliability of the payment system

### Screenshots:

