Test report

# Technologies Used

• Language: C# (.NET 7)  
• Framework: Playwright with xUnit  
• Browsers Tested: Chromium, Firefox, WebKit  
• IDE: Visual Studio Code with Playwright Test Explorer

# Scope

This report documents the integration-level tests automated for Tokero functionalities, with a focus on:

•Language switching

•Policy page validation

•Form interactions (including validation)

•Cross-browser behavior

•Performance constraints (load time)

# Flows Covered:

1. Footer policy links navigation and content validation.  
2. Language switcher validation across RO, EN, DE.  
3. SocialFi join form submission with success message verification.  
4. Contact form field character validation (positive and negative cases).

# Test Cases Summary

All test cases are defined using xUnit and executed against the live environment.  
Functional, cross-browser, multi-language and performance aspects were addressed.

# Functional Tests:

• **FooterPolicyLinks\_ShouldOpenAndContainExpectedText** – navigates and checks content for each footer policy link.  
• **LanguageSwitcher\_ShouldChangeLanguage** – asserts expected key UI text after switching languages.  
• **JoinSocialFi\_ShouldShowCongratsMessage** – simulates completing the SocialFi form and confirms success.  
• **ShouldShowError\_WhenInputIsLessThan20Characters** – checks error validation logic on contact form.  
• **ShouldHideError\_WhenInputIsExactly20Characters** – validates positive form behavior at character limit.

# Cross-Browser:

• **ShouldWorkInAllBrowsers** – runs core validation across Chromium, Firefox, WebKit.

# Performance:

• **HomePage\_ShouldLoadWithin5Seconds** – uses Stopwatch to validate that the homepage loads within 5 seconds.

# Trade-offs and Design Decisions •Used BasePage for lightweight abstraction instead of a full Page Object Model to prioritize speed and clarity.

•Skipped authentication testing due to reCAPTCHA limitations, as instructed.

• Implemented screenshot-on-failure for footer policy tests, including logic to auto-create the Screenshots/ directory if missing.

• Focused form validations on client-side behavior (e.g., input length, visibility) — backend validations were excluded.

• Centralized browser and page context setup in BrowserHelper.cs to avoid duplication.

• Negative UI tests were limited to character count checks rather than full input validation scenarios.

• Kept test scaffolding minimal to ensure simplicity and maintainability over architectural complexity.

• Moved utility logic (browser initialization, cookie handling, test data generation) into helper files like BrowserHelper.cs, LanguageHelper.cs, and TestDataGenerator.cs for better separation of concerns.

• LanguageHelper.cs encapsulates multilingual switching logic and validation, avoiding repetitive locator assertions across test cases.

# Known issues

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Status** | **Priority** |
| 1 | WebSocket disconnect breaks login | Open | High |
| 2 | Email verification message shows {0} | Open | Medium |
| 3 | Password rule accepts exactly 8 chars but says >8 | Open | Medium |
| 4 | No email received after signup | Open | High |
| 5 | Dropdown fails intermittently due to async handler | Open | Medium |
| 6 | Underline on "SocialFi Waiting List" appears thick and broken – inconsistent link styling | Open | Depends on req. |

# Output and Delivery

• Tests were executed via `dotnet test` and Playwright Test Explorer.  
• Screenshots saved for failed validations (e.g., policy link content).  
• Source code delivered as a .zip archive (alternatively GitHub-ready).  
• Report delivered in .docx and convertible to .pdf as needed.

# Bug Tracking

**1. Login fails due to WebSocket/Blazor server disconnection (intermittent)**

* **Steps to Reproduce:**
  1. Navigate to the login page: https://tokero.dev/en/login/
  2. Enter a valid email and password.
  3. Click on the "Login" button.
* **Expected Result:**  
  User should be successfully authenticated and redirected to the appropriate dashboard.
* **Actual Result:**  
  No action occurs on UI. Console displays disconnection errors:

- Error: Circuit not initialized

- Error: Connection disconnected

- Error: Cannot send data if the connection is not in the 'Connected' State

A screenshot of a computer

AI-generated content may be incorrect.

* **Environment:**  
  Chrome

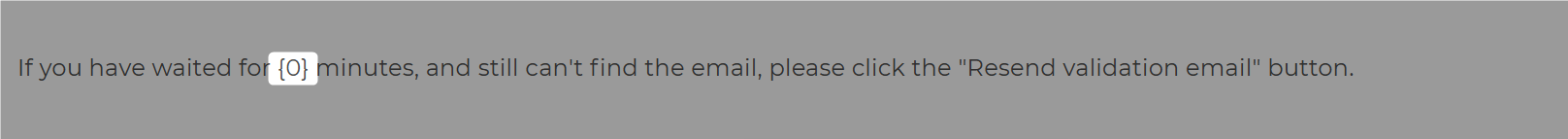
Dev environment

No network issues present

* **Impact:**  
  High — Prevents users from logging in and blocks further interaction.

**2. Email verification screen shows {0} placeholder instead of actual wait time**

* **Steps to Reproduce:**
  1. Create a new user account.
  2. Proceed to the email verification page.
  3. Observe the instructional message.
* **Expected Result:**  
  Message should display the actual number of minutes (e.g., “If you have waited for 5 minutes…”).
* **Actual Result:**  
  Message contains an unresolved placeholder:  
  If you have waited for {0} minutes...



* **Environment:**  
  Chrome

Dev environment

No network issues present

* **Impact:**  
  Medium — Affects UI clarity and user trust.

**3. Password rule inconsistency regarding minimum length**

* **Steps to Reproduce:**
  1. Go to the registration form: https://tokero.dev/en/register/
  2. Enter a password with exactly 8 characters (e.g., Test123!).
  3. Submit the form.
* **Expected Result:**

If passwords with 8 characters are accepted, the message should state:

“at least 8 characters”.

* **Actual Result:**

UI displays: Password must be more than 8 characters(>8)

However, 8-character passwords are accepted.

A white box with black text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Environment:**  
Chrome

Dev environment

No network issues present

* **Impact:**  
  Medium — May confuse users and suggests rule mismatch between frontend and backend.

**4. User does not receive verification email post-registration**

* **Steps to Reproduce:**
  1. Register a new user with a valid email.
  2. Observe the post-submission state.
* **Expected Result:**  
  A confirmation email is sent, and a success message is shown.
* **Actual Result:**  
  No email is received (not in inbox or spam). No feedback or retry option is displayed.
* **Environment:**  
  Chrome

Dev environment

No network issues present

* **Impact:**  
  High — Blocks progression for new users and results in account dead-ends.

**5. Dropdown component fails due to unhandled async message error (intermittent)**

* **Steps to Reproduce:**
  1. Go to the Contact page: https://tokero.dev/en/contact/
  2. Click on the first dropdown element.
* **Expected Result:**  
  Dropdown expands to show items, and no console errors appear.
* **Actual Result:**  
  Dropdown is unresponsive. Console error:

A screenshot of a computer

AI-generated content may be incorrect.

Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed before a response was received

* **Environment:**

Chrome

Dev environment

No network issues present

* **Impact:**  
  Medium — Interferes with form submission and damages user experience on critical flows.

**6. Underline on "SocialFi Waiting List" appears thick and broken – inconsistent link styling**

* **Steps to Reproduce:**
  1. Navigate to the login page: https://tokero.dev/en/login/
  2. Observe the **orange banner** with the call-to-action text:  
     *"Join the official SocialFi Waiting List today to earn Founder status!"*



* **Expected Result:**  
  The link **“SocialFi Waiting List”** should be styled cleanly, with either:

a normal underline (text-decoration: underline), or a button-like highlight.

* **Actual Result:**  
  The underline is unusually thick and appears split into two disconnected segments, as if it's rendered on multiple inline boxes.

Visually, it appears broken or malformed, even if the link works.

* **Environment:**

Chrome

Dev environment

No network issues present

* **Impact:**  
  Poor visual appearance reduces professionalism and trust.

# Informal Findings

While exploring the Tokero platform out of curiosity during the test automation process, I came across a few visual inconsistencies that might be considered UI bugs. Some of them may not be critical or might not technically qualify as “bugs”, but they stood out enough to raise attention and potentially impact the user experience.

These findings were not part of the automated flows but were identified incidentally during manual validation or while debugging test elements.

Although the original task did not require identifying or reporting UI bugs, I decided to include them since I encountered them naturally during the testing process.

It’s worth discussing these with someone familiar with the design and front-end implementation to clarify whether:

* they are known and accepted design decisions,
* they require further refinement,
* or they indicate rendering or layout issues under certain conditions (e.g., browser, resolution, CSS conflict).

# Author

Name: Cristina Moga  
Date: 09.05.2025  
Contact: [mogacristina74@yahoo.com](mailto:mogacristina74@yahoo.com)