Vlad Georgescu
Cluj-Napoca, Romania

Tokero QA Automated Testing – Project Documentation

Repository: https://github.com/GeorgescuMVlad/Tokero-QA-Automated-Testing
Environment Under Test: https://tokero.dev/en/
Tech Stack: Playwright with .NET (C#) - NUnit Test Project (.NET Core) – Target framework: .NET 8.0, Visual Studio Community 2022

1. Purpose:

The aim of this project was to design and implement automated integration tests for the Tokero platform's staging environment. The tests were developed to simulate real user interactions and ensure that key website flows function correctly across languages and browsers.

2. Testing Strategy

What was tested:

1. Policy Pages Crawl & Validation

- From the footer section of the homepage, the script identifies all policy-related links.
- It visits each policy page (like Policies list, Terms and Conditions, etc.) and checks:
  - That the page loads without error (HTTP 200).
  - That it contains meaningful content (e.g., a title or specific keyword).
- This simulates how a user might explore legal info and verifies site integrity.

2. Multi-Language Switching

- The script interacts with the language selector.
- It tests for:
  - Proper URL changes (e.g., /ro, /en).
  - Visible UI elements reflecting the correct language.
- This ensures Tokero's internationalization behaves as expected.

3. Multi-Browser Compatibility

- Using Playwright's capabilities, the same tests run on:
  - Chromium (Chrome-based)
  - Firefox
  - WebKit (Safari-based)
- Validates consistency in rendering and behavior.

3. Design Decisions & Trade-offs

- Focus on Deliverability

- Due to limited time, priority was given to delivering functional tests over building a full-blown test framework.
- The architecture is pragmatic - test scripts are straightforward and concise, with minimal abstraction.

- Skipped Authentication

- Authentication features were intentionally avoided due to reCAPTCHA, as mentioned in the requirements.

- Performance Testing

- Not implemented in this version to keep the scope manageable.
- However, Playwright's tracing and time measurements could be added in future iterations.

4. Outputs & Reports

- **Reports:** Generated reports with all tests run and passed in .trx and .md formats.
- **Screenshot:** Screenshot with all tests run and passed from Visual Studio Test Explorer.

5. Future improvements

- Add performance benchmarks (load time, time-to-interactive) for key pages.
- Add accessibility (a11y) checks using Playwright plugins.
- Mock login or use test-specific tokens to unlock authenticated flows.
- Integrate tests into CI pipelines (e.g., GitHub Actions).

6. Conclusion

This automated testing suite provides a foundational framework to validate critical functionalities of the Tokero website. By focusing on key user flows and ensuring compatibility across languages and browsers, it aims to enhance the website's reliability and user experience.