Pavel Anhur

QA Automation Engineer

TypeScript · Playwright · Test Framework Design

pavelanhur@gmail.com · linkedin.com/in/pavel-anhur · github.com/PavelAnhur · Minsk, Belarus

PROFESSIONAL SUMMARY

QA Automation Engineer with 4 years of experience designing, building, and maintaining test frameworks for complex web applications. At AlgoSec, I own the automated test strategy for the security and firewall management domain — a product where a missed regression has real security consequences. My main stack is TypeScript and Playwright, complemented by Java and Python. I focus on test suites that are fast, stable, and readable enough that developers run them locally before pushing. My portfolio includes a public Playwright + TypeScript framework with 250+ tests, and a five-week LLM test harness covering judge calibration, RAG metrics — including a real retriever — and streaming. Both have published Allure reports. I also built an MCP-based AI layer with four agents — requirements analysis, failure triage, adversarial review, and test generation with a heal loop.

CORE SKILLS

Languages	TypeScript, Java, Kotlin, Python, SQL (MySQL, PostgreSQL)
Test Automation	Playwright, Selenium, REST Assured, JUnit, Pytest
AI in QA Workflows	AI-assisted test generation, LLM log analysis, MCP multi-agent workflows, prompt engineering, local models (llama.cpp, Ollama)
LLM Testing	Property-based assertions, pass-rate thresholds, adversarial inputs, LLM-as-a-judge, RAG metrics (faithfulness, relevance, context precision, recall), vector retrieval (Qdrant, embeddings), streaming tests
Framework Design	Page Object Model, custom fixtures, step decorators, data-driven tests, test tagging, visual regression
CI/CD	GitHub Actions, Jenkins, GitLab CI
Reporting	Allure, Playwright HTML, JUnit XML
API Testing	REST, OpenAPI/Swagger, Postman, contract testing basics
Infrastructure	Docker, Linux (incl. SSH on production), AWS (basic)
Version Control	Git, GitHub, Bitbucket, Jira
Build Tools	npm, Maven, Gradle

EXPERIENCE

Automation Engineer · AlgoSec · Prevasio

Jan 2023 – Aug 2024

Prevasio is AlgoSec's cloud security product — a CSPM solution covering container security, vulnerability assessment, and automated cloud network security. I joined as the first and only automation engineer, alongside one architect, three developers, and a small product team.

- **Combined manual and automated testing** — ran exploratory manual sessions alongside the automated suite to catch issues automation didn't cover.
- **Built the automation practice from zero** for a greenfield CSPM product — chose the stack, defined the test strategy, and wrote the first test before the UI had stable locators.
- **Automated 300+ tests** across UI and backend for cloud security scanning, vulnerability assessment, and container posture workflows.
- **Defined the test strategy for a product with no existing test process** — identified the highest-risk workflows, wrote the coverage plan, and set the standard for what "tested" means on the team.
- **Covered multi-cloud workflows end-to-end** across AWS, Azure, and GCP — credentials, scanning, findings, remediation — catching cross-provider regressions that single-cloud testing would have missed.
- **Built visual regression coverage** (screenshot comparison) into the suite.

Automation Engineer · AlgoSec · Devices

Jan 2022 – Dec 2022

AlgoSec's core security policy management product for enterprise firewalls. I was on the Devices team, responsible for the firewall policy change workflow, until moving to the Prevasio product in January 2023.

- **Owned the automated test suite** for the firewall policy change workflow — 400+ UI and API tests covering policy creation, validation, deployment, and rollback.
- **Authored test designs and reviews** for the security-policy domain, collaborating with developers to catch spec-level ambiguities before code was written.
- **Reduced suite runtime and flakiness** through targeted waits, and elimination of cross-test contamination.

Automation Test Engineer · EPAM Systems

May 2021 – Dec 2021

- Automated test cases for web services, mobile, and web applications using Java and Selenium.
- Worked alongside developers to keep automation aligned with application changes; debugged failures and reported on suite health.
- Produced detailed test result reports for stakeholders across the client and delivery teams.

PROJECTS

Playwright + TypeScript Test Framework

`github.com/PavelAnhur/playwright-typescript-framework`

A public reference framework for end-to-end testing. **250+ tests** across UI and API, organized by spec, with custom fixtures, typed page objects, a custom `@Step` decorator for Allure reporting, and CI that publishes a live Allure report to GitHub Pages on every push.
An AI layer built on top: an MCP server and four agents — a requirements analyst, a failure analyst, an adversarial judge that reviews the failure analyst's output, and a spec generator that produces tests from requirements and refines them in a heal loop until they pass.

`Live Allure report:` `pavelanhur.github.io/playwright-typescript-framework`

LLM Test Harness

`github.com/PavelAnhur/llm-test-harness`

A test harness for non-deterministic LLM outputs, built over five weeks and documented week-by-week. **Property-based assertions** replace exact-match checks; **pass-rate thresholds** gate multi-run suites; **adversarial input tests** cover prompt injection, role confusion, and payload splitting. Includes a rubric-based **LLM-as-a-judge** calibrated against 25 hand-labeled examples, and a **real RAG retriever** — Qdrant vector database, `nomic-embed-text` embeddings via Ollama, and a deterministic chunker — with **faithfulness**, **relevance**, **context precision**, and **context recall** metrics. **Streaming tests** cover token delivery, time-to-first-token, and mid-stream disconnect. Every finding and calibration round is written up, including the ones where the judge disagreed with the human label.

`Live Allure report:` `pavelanhur.github.io/llm-test-harness`

EDUCATION

Belarusian National Technical University — Minsk, Belarus 2002 – 2008

LANGUAGES

Russian (native) · **English** (professional B2+) · **Belarusian**