PROJECT REPORT | AUTOMATION & TESTING FRAMEWORK |

Automation Portal

Overview

A full-stack automation portal enabling developers and QA teams to run UI sanity tests and regression test suites via a unified interface. Integrated with real-time log viewing, test execution status, and metrics dashboards, it significantly improved test cycle efficiency and traceability.

Services

Automation Framework Design/Test Execution Orchestration/Frontend Integration/ CI-CD Setup/Real-time Monitoring/Log Management/Role-Based Access

---

Technolgies used

React.js/Node.js (Express)/MongoDB/MySQL/Selenium/JMeter/Prometheus/ Grafana/Azure DevOps/Kubernetes

OBJECTIVE

To automate test execution and reporting within the development pipeline, empowering both dev and QA teams to achieve fast, consistent, and traceable quality checks. The portal aimed to simplify test orchestration while maintaining transparency and performance metrics.

How do we integrate UI and backend testing into a common execution framework?

What’s the best way to provide real-time feedback and logs for test runs?

How can we minimize manual QA dependencies without compromising test depth?

Process

The portal allowed users to trigger test runs, observe live logs, and view test metrics. Selenium scripts handled UI tests, while JMeter covered regression. Test executions were tracked using Prometheus and visualized via Grafana. The backend handled orchestration and result storage. Azure DevOps handled deployment to a Kubernetes cluster for isolated testing.

Result

The portal brought measurable improvements in test coverage, response time, and release confidence. It automated the bulk of regression and sanity testing workflows, provided quick turnaround for validations, and empowered QA and dev teams with immediate insights. The result was faster, more reliable releases and a significant reduction in testing overhead.

60+

tests/day automated on average

40%

reduction in operation inefficiancy

40%

improving resolution time