

Rajasekhar Aala

Automation Engineer | Playwright, TypeScript, CI/CD, API and Web Automation

Hyderabad, Telangana | +91 7842928271 | arajasekhar43@email.com

LinkedIn: linkedin.com/in/rajasekhar-aala | GitHub: github.com/rajasekhar-aala

Total Experience: 3 years | Notice Period: [20 days, negotiable]

PROFESSIONAL SUMMARY

Senior Automation Engineer with 3+ years in test automation, currently owning a **Playwright** and **TypeScript** framework that covers 750+ web and API tests across 3 product modules. Rebuilt the regression pipeline with built in workers and CI sharding, containerised browsers and **Jenkins**, taking the full regression from 6 hours to 40 minutes and lifting release confidence enough to move the team from fortnightly to weekly releases. Mentors a team of 5 engineers on **Playwright** standards, **TypeScript** coding practices and code review, and reports coverage, flakiness and defect leakage numbers to delivery leadership every sprint.

TECHNICAL SKILLS

Playwright · TypeScript · Automation Framework · Page Object Model · API Testing · CI/CD · Docker · Agile

Automation Frameworks Playwright (@playwright/test), Page Object Model, custom fixtures and helpers, Data Driven Testing

Languages TypeScript, JavaScript, Node.js, HTML and CSS selectors, XPath, JSON, YAML

Playwright and Libraries @playwright/test, Playwright APIRequestContext, expect assertions, fixtures and test.extend

CI/CD and DevOps GitHub Actions (workflows, matrix and sharding), Jenkins

Testing Types Regression, functional, REST API and contract testing, cross browser (Chromium, Firefox, WebKit)

Tools and Reporting the Playwright HTML report and trace viewer, Allure Report, Jira

Methodologies Agile, Scrum, Test Automation strategy

WORK EXPERIENCE

Senior Automation Engineer **Optum**

Feb 2026 to Present

Project: Multi tenant SaaS platform, 3 modules, 40+ engineers, weekly release cycle

- Designed and owned the **Playwright** automation framework from scratch in **TypeScript**, with the **Page Object Model**, shared fixtures, environment config in playwright.config.ts and reusable helper utilities, now covering 750+ tests across web and API layers.
- Cut full regression runtime from 6 hours to 40 minutes by running built in workers and CI sharding with 8 workers against containerised browsers, which removed the overnight wait and let the team release weekly instead of fortnightly.
- Built a **jenkins** pipeline that runs smoke on every merge, regression nightly and a full cross browser run weekly, then publishes the **Playwright** HTML report and **trace viewer** and an **Allure** report, giving stakeholders one link instead of 8 scattered reports.
- Reduced flaky failures from around 18% of runs to under 4% by turning on traces and video on retry, replacing brittle waits with web first assertions, and fixing the top 20 root causes over 2 sprints.
- Migrated 220 legacy **Selenium** tests to **Playwright** for the newest module, cutting average per test execution time by 35% and removing all explicit sleep based waits.
- Mentored [5] engineers through weekly code reviews and a written **Playwright** and **TypeScript** coding standard, which brought new joiner ramp up time from 6 weeks to 2 weeks and lifted first time pull request approval from 45% to 80%.

Automation Test Engineer **Mindtree**

[Aug 2024] to [Dec 2025]

Project: Core banking web portal for 12 branches, Java backend, regulated release process

- Automated 400+ regression tests in **Playwright** with @playwright/test, taking regression cycles from 5 days of manual effort to a 2 hour automated run.
- Introduced tag based suite slicing (smoke, regression, critical, module names) so release managers could trigger a 12 minute critical path run before every production push.
- Added end to end database checks to 90+ tests to verify records from UI submission to database row, which caught 15 data integrity defects that UI only tests were missing.
- Built a reusable test data generator in **TypeScript** that creates unique customer and account records on demand, removing the shared test data conflicts that were causing around 25 false failures per week.

Test Engineer **PQR Solutions**

[April 2023] to [Jan 2024]

Project: Insurance policy administration system

- Executed and maintained 500+ functional and regression test cases across 14 sprints with an average defect detection rate of 88% before UAT.
- Automated the first 120 smoke and regression tests in **Playwright**, which became the base suite the project still runs today.
- Validated 70+ REST endpoints in **Postman** and later moved 50 of them into **Playwright** API tests, cutting API regression time from 4 hours to 25 minutes.

PROJECTS

Enterprise **Playwright** Automation Framework, SaaS Platform (**TypeScript**)

Three product modules each had their own scripts, no shared standards, and regression took a full night. Built a single **Playwright** and **TypeScript** framework with a layered structure (tests, page objects, fixtures, helpers, config), **Playwright** workers parallel execution against containerised browsers, environment driven configuration, traces on retry, and one merged **Allure** report published from **GitHub Actions**. Result: 750+ tests, 40 minute regression, under 4% flakiness.

CI/CD Quality Gate for [Product Name] (**TypeScript**)

Broken builds were reaching QA and burning a full day of triage each week. Built a **Jenkins** workflow that runs a tagged **Playwright** smoke pack of 65 tests on every pull request inside a container, blocks merge on failure, and posts the pass rate as a PR comment. Build breaks reaching the QA environment dropped by around 70%.

EDUCATION

B.Tech Mechanical Engineering, Gudlavalleru Engineering College, 2023, Jan

Internal quality champion award for reducing regression cycle time, [Year]. Speaker at internal QA guild on **Playwright** fixture design in **TypeScript**. Author of the team **Playwright** coding standard, now used by 3 projects.

CERTIFICATIONS

- **Playwright** End to End Testing with **TypeScript**, [SDET-QA], [2023]

AWARDS AND ACCOLADES

- Star Performer of the Quarter for the framework migration, [2023]
- Client Appreciation for a Zero defect release, [2025]