Faysal Aziz

Robot Framework & AI for QA

Build readable, reusable tests for web, API and mobile applications with Cursor.

FA-0828 · 2 days

Software Development · Agentic & Generative AI

Develop readable Robot Framework tests for web, API and mobile applications, using Cursor with human review, meaningful assertions and execution evidence.

Introduction

Good automation expresses test intent clearly, checks meaningful outcomes and gives the team useful evidence when something fails. This practical workshop helps QA professionals turn their testing knowledge into Robot Framework tests they can understand, execute, troubleshoot and maintain. AI assistance is integrated into the work, with participants learning to judge the quality of generated code as carefully as code they write themselves.

The programme supports manual QA professionals moving into automation and experienced automation engineers adopting Python and Robot Framework. Guided activities build confidence with syntax and test execution, while more demanding tasks develop refactoring, reuse and failure-diagnosis skills. Participants apply their existing testing judgement to automated checks and practise writing code that their colleagues can read and maintain.

The training emphasises practical engineering judgement, connecting readable test code with reliability, maintainability and team adoption.

Through guided individual practice and collaborative activities, participants develop, execute and improve a focused set of tests against prepared web applications, APIs and mobile apps. Cursor is used throughout to explain code, assist development and investigate failures, with human review and execution evidence guiding acceptance. An instructor-led demonstration connects mobile test automation to parallel execution on BrowserStack through GitHub Actions.

Learning outcomes

By the end of the course, participants will be able to:

- Translate suitable manual test cases into readable Robot Framework tests with meaningful assertions.
- Apply Robot Framework syntax, variables, keywords, setup and teardown correctly.
- Organise reusable keywords, resource files, configuration and test data for easier maintenance.
- Develop and execute web, API and mobile tests using SeleniumLibrary, RequestsLibrary and AppiumLibrary.
- Interpret reports, logs and failure evidence to troubleshoot and improve test reliability.
- Use Cursor to generate, explain, refactor and debug tests while reviewing and verifying its output.
- Explain how a prepared GitHub Actions workflow runs mobile tests in parallel on BrowserStack and retains test results.

Prerequisites

- Working QA experience: independently write test cases, execute tests and report defects.
- A laptop with permission to install and run compatible Python, Robot Framework, Git, Cursor and testing-library versions for the prepared training environment.
- Cursor access with AI features enabled under the organisation’s approved settings.
- Access to the training GitHub repository with GitHub Actions enabled.
- BrowserStack App Automate access with device capacity for group practice and parallel execution.
- Access to the agreed training applications, API endpoints, mobile test build and test accounts.
- Reliable internet access to the approved training tools and services.
- Authorised SSH access to remote training servers where required by the prepared exercises.

Training outline

1. 1. Robot Framework Foundations and Test Execution

1. Project environment and dependency verification
2. Test structure, syntax, spacing, variables and library imports
3. Keywords, arguments, return values and assertions
4. Command-line execution, tags and test selection
5. Reports, logs and common syntax failures

2. 2. Writing Reusable and Maintainable Tests

1. Translating manual test intent into readable automation
2. User keywords, resource files and shared configuration
3. Test setup, teardown and independent test data
4. Parameterised tests and data-driven templates
5. Refactoring duplication and maintaining tests after application changes

3. 3. AI-Assisted Automation with Cursor

1. Repository context, prompting and project rules
2. Generating tests from test cases and acceptance criteria
3. Explaining unfamiliar Robot Framework and Python code
4. Refactoring keywords and diagnosing failures with execution evidence
5. Human review, assertion quality, output verification and sensitive-data handling

4. 4. Web Application Testing with SeleniumLibrary

1. Browser sessions, element inspection and stable locators
2. User interactions and meaningful assertions
3. Explicit waits, synchronisation and flaky-test prevention
4. Hands-on web test development, execution and failure diagnosis

5. 5. API Testing with RequestsLibrary

1. HTTP requests, headers, authentication and JSON payloads
2. Status codes, response content and negative-path assertions
3. Hands-on API test development, execution and test-data cleanup

6. 6. Mobile Application Testing with AppiumLibrary

1. Appium sessions, capabilities and BrowserStack device configuration
2. Mobile element inspection, locators, interactions and waits
3. Hands-on mobile test development and execution
4. Failure diagnosis using session logs, screenshots and recordings

7. 7. Parallel Mobile Testing and CI Demonstration

1. Prepared GitHub Actions workflow, dependencies and secrets
2. BrowserStack parallel sessions, device selection and capacity limits
3. Independent test data and isolated concurrent execution
4. Pipeline status, Robot Framework artifacts and BrowserStack results

Disclaimer

This course outline serves as a guideline for the proposed training. The trainer reserves the right to amend, reorder, substitute or otherwise adjust the content, activities and delivery approach at his discretion, without prior notice, taking into account participant readiness, available systems, technical constraints and the agreed learning objectives.

© 2026 Faysal Aziz · Kuala Lumpur