

QA

ADVANCED

4 DAYS

Selenium Test Automation with Java

Build a strong foundation in test automation using Selenium by understanding its framework and setup. Learn how to write and debug test cases, apply advanced Selenium techniques, and generate clear test reports to support reliable and scalable automation testing.

Why Selenium with Java?

Improved Testing Efficiency

Selenium allows for automated testing of web applications, which significantly reduces the time and effort required for manual testing.

Cost-effective

Automated testing with Selenium can be a cost-effective alternative to manual testing, particularly in the long run.

Better Quality Assurance

Your team will witness a reduction in the likelihood of bugs and errors, an improvement in the quality of software, and enhanced customer satisfaction.

What you'll be able to do

- ✓ Gain a strong understanding of the Selenium framework
- ✓ Learn how to set up Selenium Environment
- ✓ Learn writing Test Cases
- ✓ Decoding common Debugging techniques used in Selenium
- ✓ Get equipped with the skills and knowledge to Reporting Test Results
- ✓ Learn Advanced Techniques used in Selenium

Who this is for

- Software developers
- Quality assurance engineers
- Test automation engineers
- IT professionals
- Project managers
- Data analysts
- Digital marketing professionals

Curriculum

01 Foundations

- Apply foundations using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Environment Setup

- Apply environment setup using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Locators and Interactions

- Apply locators and interactions using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Waits and Dynamic Elements

- Apply waits and dynamic elements using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Alerts, Frames, and Windows

- Apply alerts, frames, and windows using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 POM and Design Patterns

- Apply pom and design patterns using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 Data-Driven Tests

- Apply data-driven tests using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Frameworks and Reporting

- Apply frameworks and reporting using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 Cross-Browser and Parallel

- Apply cross-browser and parallel using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 CI/CD Integration

- Apply ci/cd integration using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Debugging and Best Practices

- Apply debugging and best practices using Selenium (Java) in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks