QA Automation Engineer: Complete 6-Month Roadmap

Goal: Get interview-ready and land your first QA Automation role

Study Time: 15-20 hours/week (3-4 hours/day on weekdays, more on weekends)

Month 1-2: Foundation & Core Skills

Week 1-2: Programming Fundamentals

Choose ONE language (recommended: Java or Python)

Java Path:

- Variables, data types, operators
- Control structures (if/else, loops)
- Methods and functions
- OOP concepts (classes, objects, inheritance, polymorphism)
- Collections (ArrayList, HashMap)
- Exception handling

Python Path:

- Basic syntax and data types
- Control flow and loops
- Functions and modules
- OOP in Python
- Lists, dictionaries, tuples
- File handling

Resources:

- Java: "Java Programming Masterclass" (Udemy) or Codecademy
- Python: "Python for Everybody" (Coursera) or Automate the Boring Stuff
- Practice: HackerRank, LeetCode (Easy problems only)

Daily Practice: 1 hour coding exercises

Week 3-4: Testing Fundamentals + SQL

Testing Concepts:

- What is software testing and why it matters
- SDLC and STLC
- Types of testing (functional, regression, smoke, sanity)
- Test case design techniques
- Bug life cycle and bug reporting
- Agile/Scrum basics

SQL Basics:

- SELECT, WHERE, JOIN operations
- INSERT, UPDATE, DELETE
- GROUP BY, ORDER BY
- Aggregate functions
- Subqueries basics

Resources:

- ISTQB Foundation syllabus (free PDF)
- "Software Testing Tutorial" (Guru99)
- SQL: Mode Analytics SQL Tutorial or SQLBolt
- Practice: LeetCode SQL problems (Easy)

Hands-on Project:

- Manually test a public website (Amazon, Netflix)
- Write 20-30 test cases in Excel/Google Sheets
- Find and document 5+ real bugs with screenshots

Week 5-6: APIs & Postman

Learn:

- What are APIs and REST

- HTTP methods (GET, POST, PUT, DELETE)
- Status codes (200, 404, 500, etc.)
- JSON structure and manipulation
- Authentication (Basic, Bearer tokens)

Postman:

- Create collections
- Write API tests
- Environment variables
- Basic assertions
- Running collections

Resources:

- "Postman Beginner's Course" (YouTube - FreeCodeCamp)
- Practice APIs: JSONPlaceholder, ReqRes

Project:

- Test 10+ APIs from a public API
- Create a Postman collection with 30+ test cases
- Generate test reports

Week 7-8: Git & Version Control

Learn:

- Git basics (init, add, commit, push, pull)
- Branching and merging
- GitHub/GitLab basics
- Pull requests
- .gitignore files

Resources:

- "Git and GitHub for Beginners" (YouTube)
- Practice: Push your projects to GitHub

Action Items:

- Create GitHub profile
- Upload all practice projects
- Write good README files
- Start building your portfolio

Month 3-4: Automation Core

Week 9-10: Selenium WebDriver Basics

Learn:

- WebDriver architecture
- Locators (ID, Name, XPath, CSS Selector)
- WebElement interactions (click, sendKeys, getText)
- Waits (implicit, explicit, fluent)
- Handling alerts, frames, windows
- Taking screenshots

Resources:

- "Selenium WebDriver with Java" (Udemy - Rahul Shetty)
- Practice sites: The-Internet by Herokuapp, DemoQA

Daily Practice:

- Automate 2-3 scenarios per day
- Focus on different locator strategies

Week 11-12: TestNG/JUnit Framework

Learn:

- Annotations (@Test, @BeforeMethod, @AfterMethod)
- Assertions

- Test suite organization
- Parameterization
- Data-driven testing
- Parallel execution
- Generating reports

Resources:

- TestNG official documentation
- Practice: Convert previous Selenium scripts to TestNG

Project 1:

- Automate login, search, and checkout flow of an e-commerce site
- Use TestNG framework
- Implement data-driven testing with Excel/CSV
- Generate HTML reports

Week 13-14: Page Object Model (POM)

Learn:

- POM design pattern
- Page Factory
- Reusable components
- Base classes
- Utility classes

Resources:

- "Page Object Model Framework" tutorials
- Refactor existing projects

Project 2:

- Build a complete automation framework with POM
- Include 30+ test cases

- Add screenshots on failure
- Implement proper folder structure

Week 15-16: Maven/Build Tools + Advanced Concepts

Maven:

- pom.xml configuration
- Dependencies management
- Running tests from command line
- TestNG XML suite files

Advanced Selenium:

- JavaScript Executor
- Handling dynamic elements
- File uploads/downloads
- Headless browser execution
- Cross-browser testing basics

Project 3:

- Create Maven-based framework
- Add cross-browser capability
- Implement logging (Log4j)
- Push to GitHub with detailed README

Month 5-6: API Automation + Interview Prep

Week 17-18: REST Assured (API Automation)

Learn:

- REST Assured library basics
- Request/Response specifications
- JSON/XML parsing (JsonPath, XmlPath)

- Authentication in REST Assured
- Serialization and deserialization
- Response validation

Resources:

- "REST Assured API Testing" (YouTube tutorials)
- Practice: Automate public APIs

Project 4:

- Create API automation framework
- Test 20+ endpoints
- Implement response validation
- Generate Extent Reports

Week 19-20: CI/CD Basics

Learn:

- Jenkins basics
- Creating jobs in Jenkins
- Running Selenium tests in Jenkins
- GitHub Actions basics
- Understanding build pipelines

Resources:

- "Jenkins for Beginners" (YouTube)
- Jenkins official documentation

Action:

- Set up Jenkins locally
- Create a job to run your automation suite
- Trigger builds on GitHub commits

Week 21-22: Advanced Topics & Portfolio Polish

Learn ONE of these based on job market:

- Docker basics for test environments
- BDD with Cucumber (if market demands)
- Performance testing basics (JMeter intro)
- Mobile testing basics (Appium intro)

AI/ML Testing (Emerging Skill - Optional but Recommended):

- Understanding AI agent behavior testing
- Testing LLM-based applications
- Prompt injection testing
- Testing for AI bias and fairness
- Automated validation of AI outputs
- Tools: LangChain testing, OpenAI API testing

Why learn AI testing:

- Growing number of AI-powered applications
- Companies building AI agents need QA
- Differentiates you from other candidates
- Future-proof your career

Resources for AI Testing:

- "Testing AI Systems" articles on Medium
- OpenAI API documentation
- Practice: Test ChatGPT API responses for consistency

Week 23-24: Interview Preparation

Resume:

- Highlight 4-5 projects with metrics
- Use action verbs and quantify impact

- Include tech stack clearly
- Add GitHub profile link

Interview Prep:

- Manual testing concepts (50+ questions)
- Selenium WebDriver (50+ questions)
- Framework design questions
- Java/Python coding (Easy to Medium problems)
- SQL queries
- Scenario-based questions

Resources:

- "Top 100 Selenium Interview Questions"
- LeetCode (focus on Easy-Medium)
- Mock interviews with peers
- Record yourself answering questions

Action Items:

- Apply to 10+ jobs per week
- Customize resume for each application
- Practice explaining your projects
- Prepare STAR format answers for behavioral questions

Key Portfolio Projects Summary

By end of 6 months, you should have:

1. **E-commerce Automation Suite** (Selenium + TestNG + POM)
    - 30+ UI test cases
    - Data-driven framework
    - Screenshot on failure
2. **API Automation Framework** (REST Assured)
    - 20+ API tests

    - JSON validation
    - Extent Reports
3. **Hybrid Framework** (UI + API)
    - Maven-based
    - Jenkins integration
    - Cross-browser support
    - Detailed documentation
4. **AI/ML Testing Project** (Optional but impressive)
    - Test an AI API (OpenAI, Hugging Face)
    - Validate response consistency
    - Test for edge cases and prompt injection

All projects should be on GitHub with:

- Clean, commented code
- Professional README with setup instructions
- Screenshots/GIFs of execution
- Test execution reports

Daily Schedule Template

Weekdays (3-4 hours):

- 1 hour: Theory/Tutorial videos
- 1.5 hours: Hands-on coding/practice
- 0.5 hour: Interview questions/revision
- 0.5-1 hour: Build portfolio projects

Weekends (6-8 hours):

- Focus on building projects
- Review and refactor code
- Write documentation
- Apply to jobs (from Month 5)

What NOT to Do

❌ Don't learn every tool superficially (Cypress, Playwright, etc.)
❌ Don't skip coding fundamentals to jump to Selenium
❌ Don't build frameworks without understanding basics
❌ Don't ignore manual testing concepts
❌ Don't wait until Month 6 to start applying
❌ Don't copy-paste code without understanding
❌ Don't neglect your GitHub profile

When to Start Applying

- **Month 4:** Start applying to junior/trainee positions
- **Month 5:** Actively apply to SDET/Automation QA roles
- **Month 6:** Interview intensively

Don't wait to feel "ready" — you'll learn more on the job!

Tools & Technologies You'll Master

Core:

- Java/Python
- Selenium WebDriver
- TestNG/JUnit
- Maven
- Git/GitHub

Essential:

- SQL
- Postman
- REST Assured
- Jenkins/CI-CD basics
- Page Object Model

Bonus (Market differentiators):

- Docker basics
- AI/ML testing concepts
- Cloud basics (AWS/Azure)
- Cucumber BDD (if time permits)

Success Metrics

By Month 6, you should be able to:
✅ Write clean, maintainable automation code
✅ Design a framework from scratch
✅ Explain your projects confidently
✅ Solve LeetCode Easy problems in 15 minutes
✅ Answer 80% of common interview questions
✅ Debug failing tests independently
✅ Understand CI/CD pipeline basics

Final Tips

1. **Consistency > Intensity:** 3 hours daily beats 20 hours on Sunday
2. **Build in public:** Share your learning journey on LinkedIn
3. **Network:** Connect with QA professionals, join communities
4. **Practice explaining:** Can you explain your project to a 10-year-old?
5. **Stay updated:** Follow QA blogs, YouTube channels, newsletters
6. **Be patient:** Job search can take 2-3 months; keep learning

---

Remember: This roadmap is aggressive but achievable. Adjust pace based on your background. The goal isn't perfection—it's getting job-ready and landing interviews.

Questions? Stuck somewhere? Reach out—I'm here to help!

Good luck! 🚀