PracticeQAAutomation

Start here.
Finish job-ready.

Your guide to a free playground for test automation: where to begin, what to do next, and everything the site can teach you.

Selenium **Playwright** **Cypress** **API · SQL · Performance**

34	52	559	502
realistic practice apps	element challenges	ready-made test cases	interview questions

Free · no sign-up needed · no paywall

www.practiceqaautomation.com

WELCOME

What this site is, and how to use it

PracticeQAAutomation is a set of real-feeling web apps and tricky pages built to be automated. You write the tests; the site gives you something worth testing.

34	52	559	502	130
practice apps	challenges	test cases	interview questions	companies covered

Your first 15 minutes

1 Pick one tool

Selenium, Playwright or Cypress. Open its page and follow the set-up box. Page 4 helps you choose.

2 Run the first test

Copy the ready-made first test from the tool's page and run it against this site. Nothing to install locally but the tool.

3 Open the guided path

Go to **/start** and work down the 10 steps. Tick each one when your own test passes.

What makes it different

- ✓ **Real apps, not toy pages.** Banking, e-commerce, travel, insurance, HR and more, each with full flows and business rules.
- ✓ **Unhappy paths included.** Locked accounts, declined cards, limits, timeouts and role-based access.
- ✓ **Stable test data on every page.** Test logins, coupons and OTPs are shown where you need them.
- ✓ **Working code to compare with.** Every app has a complete sample suite for all three tools, in several languages.
- ✓ **Every element has a `data-testid`**, and the sample suites also show id, name, role, label, text, CSS and XPath.
- ✓ **Everything saves in your browser.** No account is needed. Sign-in is optional, only for the daily challenge streak.

How to read this guide

IF YOU ARE...	START WITH	THEN
New to automation	Pages 3 and 4: the guided path and choosing a tool	Page 5, the starter apps
Already writing tests	Page 8: deep dives, then the bug hunts on page 5	Page 7, testing beyond the UI
Preparing for interviews	Page 9: interview prep and the daily challenge	Page 10, the four-week plan

STEP BY STEP

The guided path: 10 steps in order

This is the order we would learn it in. Each step names what to automate and how you know you are done. Your progress is saved in your browser.

1. **Set up your tool and run a first test** Beginner
   Install Selenium, Playwright or Cypress and run the ready-made first test against this site – no local app to start.
2. **Nova Shop – every login outcome** Beginner
   Cover the standard, locked and slow users, wrong and empty credentials, and the route guard on /shop/cart.
3. **Nova Shop – buy something end to end** Intermediate
   Search, add to cart, apply SAVE10, check the totals maths, pay by card and find the order in My orders.
4. **TrustBank – transfer with OTP** Intermediate
   Pass the two-step login, add a payee in the modal, transfer by IMPS, enter the OTP and reconcile the balance.
5. **Two roles in one test** Advanced
   Place an order as demo_user and ship it as admin_user in Nova Shop, then send a ₹30,000 transfer for approval in TrustBank and approve it as the branch manager.
6. **ShowTime – seats, timers and OTP** Advanced
   Pick seats on the live seat map, mock the clock to expire the 5-minute hold, then book again and pay with an OTP.
7. **Tricky elements** Intermediate
   Shadow DOM, iframes, alerts, new windows, drag and drop, uploads and dynamic IDs – one small page each.
8. **Find the seeded bugs** Advanced
   Sign in to Nova Shop as buggy_user and to TrustBank with the buggy build's customer ID. Each has five defects on purpose – write the tests that catch them.
9. **Check the UI against the API** Advanced
   Ask the shop API for a quote and the bank API to validate a transfer, then assert the UI shows exactly the same numbers and messages.
10. **Explain what you built** Beginner
    Turn your suite into interview answers: framework design, waits, locators, flaky tests and CI.

Open it: /start · Finishing all 10 steps earns the “Path finisher” badge on the daily challenge page.

CHOOSE YOUR TOOL

Selenium, Playwright or Cypress?

All three get the same treatment: a set-up guide, a sample suite for every app, recipes for every tricky element, and deep dives. Pick one and stay with it until the guided path is done.

Se Selenium

The standard in most enterprise teams and the one interviewers ask about most. Choose it if your job or target job uses Java, C# or Python.

Languages on this site
Java · Python · C# · JavaScript · TypeScript

Deep dives
8 topics

/selenium

Pw Playwright

Fast, auto-waiting and modern. Choose it if you are starting fresh or want the shortest path to stable tests.

Languages on this site
TypeScript · JavaScript · Python · Java · C#

Deep dives
9 topics

/playwright

Cy Cypress

Runs inside the browser with a friendly test runner. Choose it if your team is JavaScript-first and tests a web front end.

Languages on this site
JavaScript · TypeScript

Deep dives
9 topics

/cypress

What each tool's section gives you

Overview and set-up

Install steps and a first test you can run in minutes, written against this site.

Domain apps

For each of the 34 apps: scenarios to automate from beginner to advanced, plus a complete sample suite.

Elements

Short recipes for inputs, tables, alerts, frames, windows, drag and drop, uploads, shadow DOM and more.

Deep dives

The ideas that separate a script from a suite: waits, page objects or fixtures, sign in once, network control, UI versus API.

Two switches worth knowing

- ✓ **Language.** Sample suites, element recipes and deep dives all have a language switcher. Your choice is remembered for that tool across the site.
- ✓ **Locator style.** Each sample suite can be shown with **all locator types** (id, name, role, label, text, CSS, XPath) or with `data-testid` **only**. Start with test ids, then study the mixed version.
- ✓ **Not sure which locator to write?** Try them live in the Locator Playground: /elements/locators

PRACTICE APPS

34 apps from real industries

Start with these six, in this order. Each adds a technique the previous one did not need. Use only the test data shown on each page; everything you enter is fake and stays in your browser.

1. Nova Shop — E-commerce

Online store

URL: `/shop` User: `demo_user`

2. TrustBank — Banking

Net banking

URL: `/apps/bank` Customer ID: `100200300`

3. ShowTime — Entertainment

Movie ticket booking

URL: `/tickets` Login: `not required`

4. SkyWays — Travel & Airline

Flight booking

URL: `/apps/flights` Login: `not required`

5. StayEase — Hospitality

Hotel booking

URL: `/apps/hotel` Login: `not required`

6. FoodDash — Food delivery

Food ordering & delivery

URL: `/apps/food` Login: `not required`

...and 28 more all at /apps

SkyTel · CoreERP · AdPulse · SecureLife · MediCare+ · ShipFast · LearnHub · HomeNest · PeopleHub · SalesPulse · Chirp · TradeX · GovServe · RideGo · EventHive · FantasyXI · AutoDrive · StreamBox · PowerGrid · MediKart · DeskFlow · TaskBoard · HireHub · FarmMart · GiveHope · LexDesk · FactoryOS · KeyVault DRM

Bug hunts

Nova Shop and TrustBank each have a buggy build with five planted defects. Write tests that catch them. /shop/bug-hunt · /apps/bank/bug-hunt

Project KT for every app

The hand-over document you get on a real project: roles, rules and glossary. Read it first. /projects

ELEMENTS AND TEST CASES

52 challenges for the hard parts

Real apps hide tricky elements inside long flows. These pages isolate them, one idea per page, so you can learn each technique on its own.

15 Elements	**3** Forms & Login	**3** Alerts, Frames & Windows	**8** Widgets
7 Interactions	**3** Dynamic Behavior	**5** Media & Scrolling	**8** More

You will find shadow DOM (open, nested and closed), single and nested iframes, alerts and prompts, new tabs and windows, drag and drop, file upload and download, dynamic ids, hidden elements, sortable and paginated tables, date pickers, sliders, tooltips and OTP inputs. Browse them all: **/challenges**

A sensible order for the challenges

First
Text Box, Check Box, Radio Button, Buttons, Links, then the Login Form and Practice Form.

Next
Web Tables, Selects and Autocomplete, Date Picker, Alerts, Frames and Browser Windows.

Then
Dynamic Properties, Shadow DOM, drag and drop, uploads, hidden elements and the XPath functions page.

Test case library 559 cases

Do not start from a blank page

Every app has written test cases: positive, negative, edge and end-to-end. Pick a case, automate it, and compare with the sample suite. **/test-cases**

"Automate this case" links

Each case links to the app it runs on and to the guide for Selenium, Playwright and Cypress, so you never have to search for where to begin.

Tip: write the test first using only `data-testid` . Once it passes, rewrite two or three locators by role, label or XPath. That is how you learn which locators survive a redesign.

BEYOND THE UI

The other kinds of testing QA roles ask for

A modern tester is expected to do more than click through screens. Each of these has hands-on practice and working examples.

API testing — Runs on your computer

Run a practice REST API on your own machine and test it with Postman, Rest Assured or code. A buggy mode hides defects.
/api-practice

SQL & database — 37 live exercises

Write real SQL in the browser, from SELECT and joins to window functions.
/sql-playground

Network mocking — Errors & timeouts

Fake empty lists, server errors and slow responses for a page that loads its data.
/dynamic/network-mocking

Accessibility — 15 planted problems

A page with accessibility problems planted on purpose. Find them with axe or by hand.
/misc/accessibility

BDD & Cucumber — Java, TS & Python

Feature files, step definitions and runners, ready to download and run.
/bdd

Framework & CI — TestNG · Jenkins · GitHub

A complete framework with page objects, config, reporting and CI pipelines to copy.
/framework-ci

Visual testing — Catch a 2px shift

Two versions of a page that differ slightly, to practise screenshot comparison.
/misc/visual-testing

Performance — k6 & JMeter, safely

Load-test scripts to run against your own machine, not the live site.
/performance-testing

LLM and AI testing

A help-centre chatbot with a buggy mode: hallucinations, wrong citations, prompt injection and data leaks to catch. No real AI model, so results repeat. /ai/rag-bot

UI and API together

Nova Shop and TrustBank have small APIs that use the same rules as their screens. Ask the API for the right answer, then assert the UI shows it. /api-practice

Please do not load-test the live site. It is free and shared. The performance page shows how to point k6 or JMeter at the practice API running on your own computer.

DEEP DIVES

From a script that works to a suite you can trust

Short, focused lessons with complete examples. Every example is run against this site, in every language shown, so what you copy is known to pass.

Se Selenium 8 topics

1. Locators that survive a redesign
2. Explicit waits, never sleep
3. Page objects: locators in one place
4. Sign in once, reuse it everywhere
5. Control the network
6. Check the UI against the API
7. Two roles in one test
8. Evidence when a test fails

/selenium/deep-dives

Pw Playwright 9 topics

1. Locators that survive a redesign
2. Auto-waiting and web-first assertions
3. Fixtures: set-up you can name
4. Sign in once, reuse it everywhere
5. Control the network
6. Check the UI against the API
7. Two roles in one test
8. Control time
9. A config you can debug with

/playwright/deep-dives

Cy Cypress 9 topics

1. Locators that survive a redesign
2. Retry-ability, never cy.wait(ms)
3. Custom commands: set-up you can name
4. Sign in once with cy.session
5. Control the network
6. Check the UI against the API
7. Two roles in one test
8. Control time
9. A config you can debug with

/cypress/deep-dives

Read them in this order

1 • Foundations
Locators, then waits. Most flaky tests come from one of these two.

2 • Structure
Page objects, fixtures or custom commands, then signing in once.

3 • Control
Stub the network, control time, and check the UI against the API.

4 • Real life
Two roles in one test, then evidence and config for when a test fails in CI.

When to read them

- ✓ **After step 3 of the guided path.** Once you have one end-to-end test, the deep dives show how to make the next ten cheaper to write.
- ✓ **When a test is flaky.** Go straight to the waits topic for your tool before adding a sleep.
- ✓ **Before an interview.** Each topic is a question you are likely to be asked, with code you can explain.

KEEP GOING

A daily habit, and interview preparation

Skill comes from showing up. The daily challenge takes two minutes; the interview section turns what you built into answers.

🔥 Daily challenge

One question a day

The same question for everyone, on the homepage and at **/daily**. One try. The server checks your answer and explains it.

One hands-on task

A different element challenge each day to automate. Tick it when your own test passes.

Streaks and 10 badges

Correct answers on consecutive days build a streak. Badges mark 3, 7 and 30 days, and more.

As a guest your streak is kept in your browser. Sign in with Google (optional) to keep a verified streak that follows you to any device. No email is collected, and you can delete your data at any time.

🎯 Interview preparation

Question bank 502 questions

Theory, coding and behavioural questions with answers, by topic and experience level: Selenium, Playwright, Cypress, API, SQL, Java, Python, JavaScript, C#, BDD, frameworks and AI testing.
/interview

Company pages 130 companies

The interview rounds and the kinds of questions reported for each company, with sources linked.

Timed quiz 135 MCQs

Multiple-choice questions with explanations. Choose a topic, a level or a company style.
/interview/quiz

Share your interview

Had an interview recently? Add the questions you were asked to help the next person.
/interview/share

The best interview answer is your own suite. Step 10 of the guided path is “Explain what you built”. Pick one test you wrote here and be ready to explain every choice in it: the locator, the wait, the data, and what it would catch.

YOUR PLAN

Four weeks from first test to job-ready

An hour a day is enough. Adjust the pace, but keep the order.

Week 1 • First tests

- ✓ Set up one tool and run the first test
- ✓ Guided path steps 1 and 2: every Nova Shop login outcome
- ✓ Five element challenges from the “First” group
- ✓ Answer the daily question each day

Week 2 • Real flows

- ✓ Guided path steps 3 and 4: Nova Shop checkout, TrustBank transfer with OTP
- ✓ Deep dives on locators and waits
- ✓ Alerts, frames and windows challenges
- ✓ Automate ten cases from the test case library

Week 3 • Harder problems

- ✓ Guided path steps 5 to 7: two roles, timers, tricky elements
- ✓ Deep dives on page objects or fixtures, and signing in once
- ✓ Network mocking and one API test
- ✓ Start the SQL Playground

Week 4 • Prove it

- ✓ Guided path steps 8 to 10: bug hunts, UI versus API, explain your suite
- ✓ Put your suite in a framework with CI
- ✓ Two timed quizzes and your target company's page
- ✓ Share your streak on LinkedIn

Quick links

Start	/start · /daily	**Tools**	/selenium · /playwright · /cypress
Practice	/apps · /challenges	**Reference**	/test-cases · /projects
Beyond UI	/api-practice · /sql-playground	**Career**	/interview · /framework-ci
What's new	/releases	**Help**	/about for bug reports and requests

Stay in touch

Follow the PracticeQAAutomation page on LinkedIn for new apps, challenges and release notes: linkedin.com/company/practice-qa-automation

The site is free and stays free

If it helped you, you can support it from the link in the site footer. Sharing it with a friend who is learning helps just as much.

Open the site, pick a tool, and write your first test today.