IltizamTech
التزام تك

// USER FIELD GUIDE · VERSION 1.0

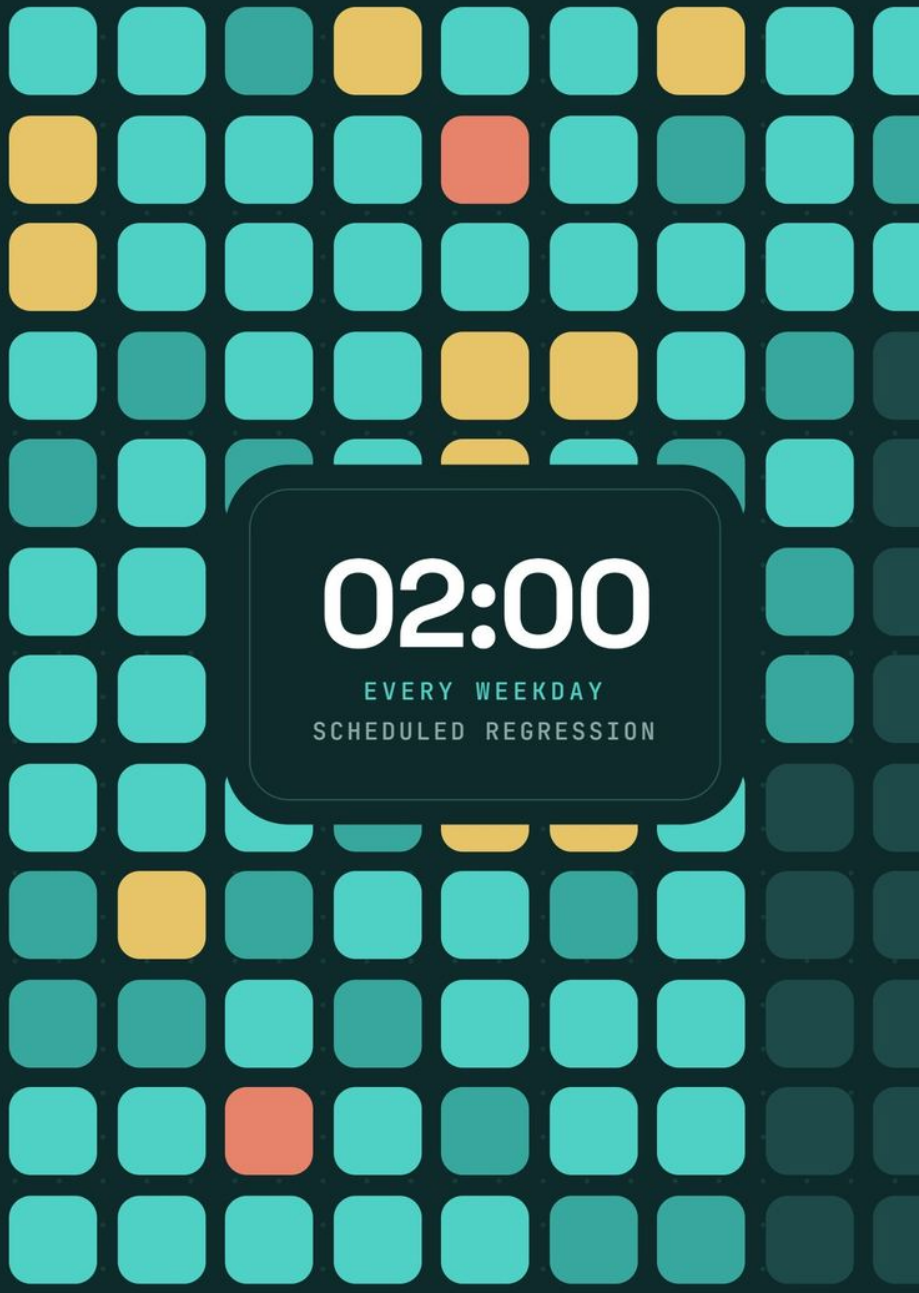
Atimtha

أتمتها

"AUTOMATE IT"

From release-week regression by hand to automated tests your own testers build, run overnight inside your own network. Everything you need to build, run and report automated tests for web applications and APIs.

web · api | arabic · english | cloud · on-prem · offline



NOVEMBER 2026 ARABIC · ENGLISH ILTIZAMTECH.COM

Find yourself.
Skip the rest.

This isn't a manual to read cover to cover. Pick your role, jump to your playbook, and come back to the big-picture pages when you want the whole story.

QA tester
Automate what you already test — no code.
JUMP TO PAGE 12 →

Automation engineer
Keep the suite healthy as screens change.
JUMP TO PAGE 13 →

QA lead
Decide what changes, and when it runs.
JUMP TO PAGE 14 →

Delivery manager
Know where quality stands without asking.
JUMP TO PAGE 15 →

Client stakeholder
Get evidence of testing, in your own colours.
JUMP TO PAGE 16 →

Organisation admin
Set it up once. Govern it always.
JUMP TO PAGE 17 →

SHORT ON TIME? The five-minute tour
03 Why Atimtha · 05 How it works · 08 Healed vs clean · 19 Golden rules

HOW TO READ THIS GUIDE

AUTO Atimtha does this for you

RULE The system enforces it

TIP Work smarter

SAMPLE Illustrative screens and data

Release week, replaced.

BEFORE · by hand · brittle scripts · waiting

Who's running regression this time?

Fifty tests red. The button just moved.

"We'll automate it when an engineer is free."

Is that failure real, or is it the environment?

Can test passwords go to that cloud tool?

Did anyone check the Arabic screens?

AFTER · one governed path

01 Describe → 02 Build → 03 Approve → 04 Run → 05 Report

EVERY TEST ENDS ONE OF THREE WAYS

- ✓ Passed
- Passed — healed
- ✕ Failed — sorted by cause

No code

Testers build tests from dropdowns and a data grid.

A second pair of eyes

Every change is approved before it applies.

Inside your network

Agents call out to the portal. Nothing connects in.

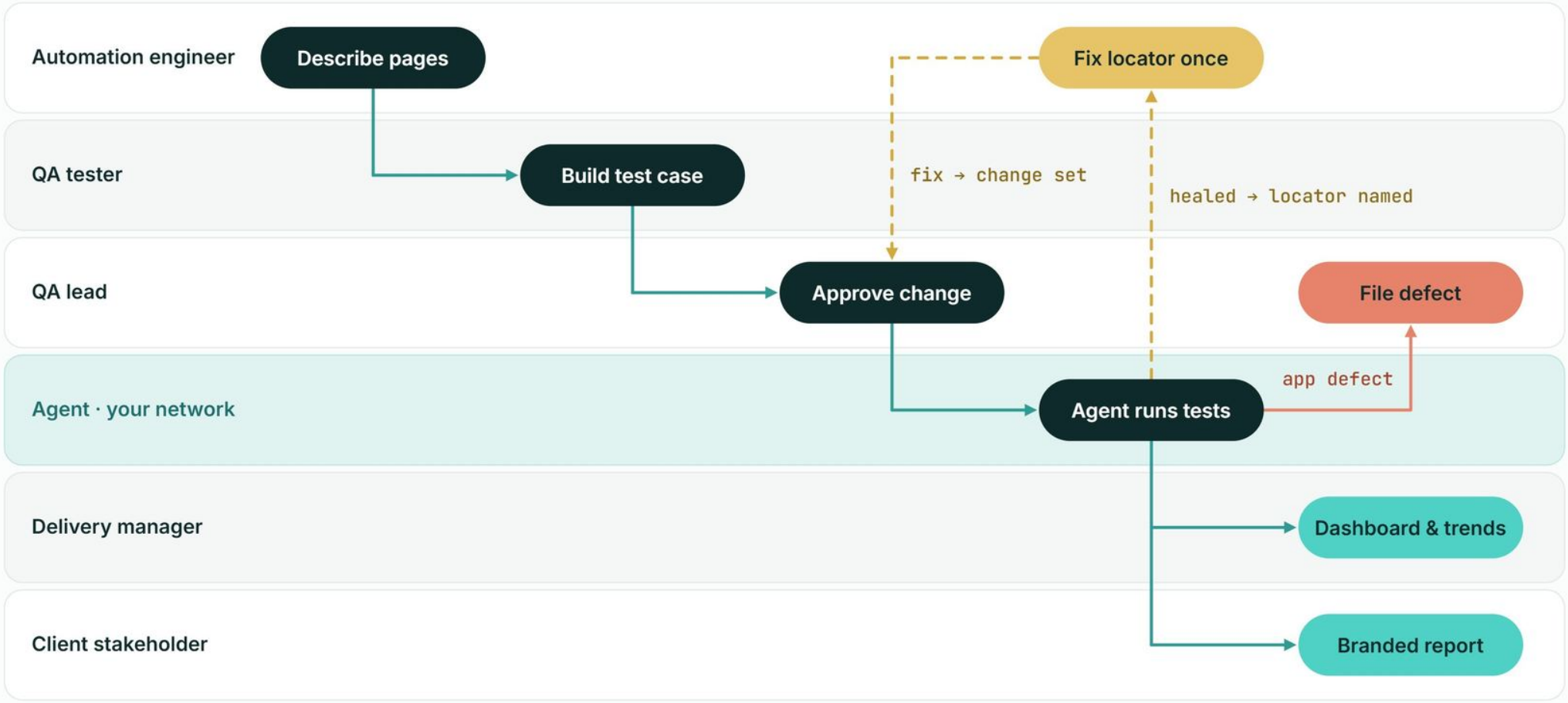
Arabic and English

Portal, tests and reports work right-to-left.

Same release. None of the scramble.

Every role. One path.

Every test change takes the same route to the shared suite, and every run reports back the same way.

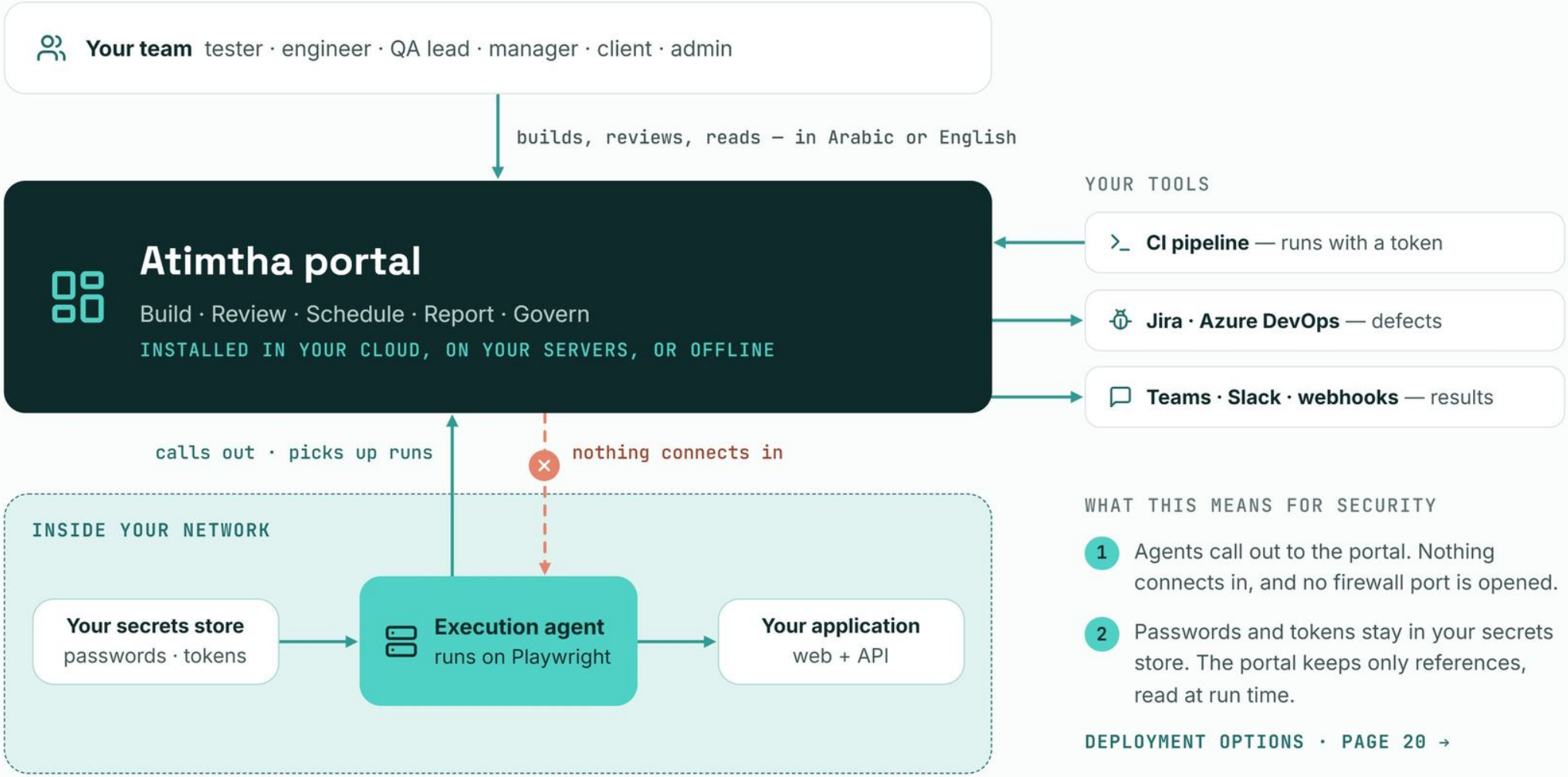


Main path — Failure → filed as a defect ---- Healed → fix once, review again

ADMIN SETS THE STAGE · SEE PAGE 17

The portal plans. Your network runs.

Your team works in the portal. An agent inside your network takes each run, tests the application where it lives, and sends the results back. Failures then reach your team's own tools.



WHAT THIS MEANS FOR SECURITY

1. Agents call out to the portal. Nothing connects in, and no firewall port is opened.
2. Passwords and tokens stay in your secrets store. The portal keeps only references, read at run time.

DEPLOYMENT OPTIONS · PAGE 20 →

Nothing changes
without a second pair of eyes.

Every edit to a test, a page or a workflow travels the same road. It reaches the shared suite only after someone else approves it.

C1

Edited

A test, page, workflow or test data is changed — in the portal or offline.

→

C2

Change set

The edits are bundled and sent for review.

→

C3

In review

A second person checks the change.

→

C4

Approved

The reviewer signs it off.

→

C5

Applied

Live in the shared suite. A page fix reaches every test that uses it.

Side road: Not approved

A change that isn't approved **never applies**. The suite your team runs stays exactly as it was.

`REVIEW → NOT APPROVED → UNCHANGED`

Same road: Offline work

Engineers can edit offline. Their work comes back as a change set and takes **the same review**.

`OFFLINE → CHANGE SET → IN REVIEW`

On the record

Every change and approval is logged **with who and when**, so the suite's history is never a guess.

`WHO · WHAT · WHEN`

NO UNREVIEWED EDIT REACHES THE SHARED TEST SUITE.

One run. Every result accounted for.

From the moment a run starts to the moment a defect reaches your developers — every stop it makes.

R1
Started
Now, on a schedule, or from CI.

→

R2
Picked up
An agent in your network takes it.

→

R3
Running
Tests run on the Playwright engine.

→

R4
Reported
Results go back to the portal.

→

R5
Sorted
Clean, healed or failed — and why.

→

R6
Delivered
Smart Report, Jira, Teams, Slack.

EVERY TEST ENDS ONE OF THREE WAYS

Passed

CLEAN PASS

Every element was found the main way. Nothing to do — it counts as a clean pass.

Passed — healed

A WARNING, NOT A WIN

The main locator missed and a backup found the element. Reported apart from clean passes, with the original failure and the locator to fix.

HOW HEALING WORKS · PAGE 08 →

Failed

SORTED BY CAUSE

Each failure arrives with screenshots, a video and a trace, sorted into one of three causes:

- **Application defect** — file it to Jira or Azure DevOps
- **Environment problem**
- **Infrastructure problem**

A healed pass is not a clean pass.

Screens change. When a button moves, Atimtha can still find it through a backup locator, so a cosmetic change doesn't turn your suite red. What it never does is hide that it happened.

HOW A HEAL HAPPENS



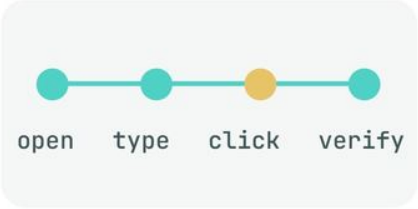
01 The main locator misses

The button moved in the new release. Its main locator no longer finds it.



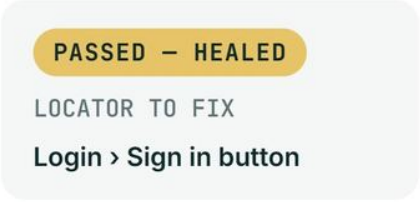
02 A backup locator finds it

The object repository keeps a main locator and backups for every element.



03 The test carries on

A cosmetic change doesn't stop the run or turn it red.



04 It's reported as healed

Apart from clean passes, with the original failure and the locator to fix.

TWO WAYS TO HEAL

	QUIET HEALING	ATIMTHA
The result	Counted as a pass	Reported apart from clean passes
Original failure	Gone	Kept, next to the result
What to fix	Left for you to find	The exact locator is named
The fix	Patched test by test	Fixed once, then approved

WHAT TO DO WITH A HEALED TEST

1. Open the healing report from the run.
2. Fix the named locator once, in the object repository.
3. Send the change for review. Once approved, every test using that element picks it up.

Atimtha keeps testing, even while your team sleeps.

AUTO

Scheduled runs

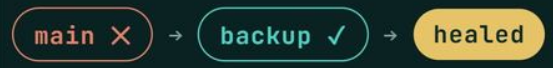
Runs start on time, on any environment — for example every weekday at 02:00.



AUTO

Self-healing

When a main locator misses, a backup finds the element — and says so.



AUTO

Failure sorting

Every failure is sorted: application, environment or infrastructure.



AUTO

Evidence capture

Each failure arrives with screenshots, a video and a trace.



RULE

Secrets at run time

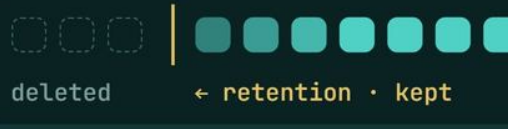
Passwords are read from your secrets store only when a run needs them.



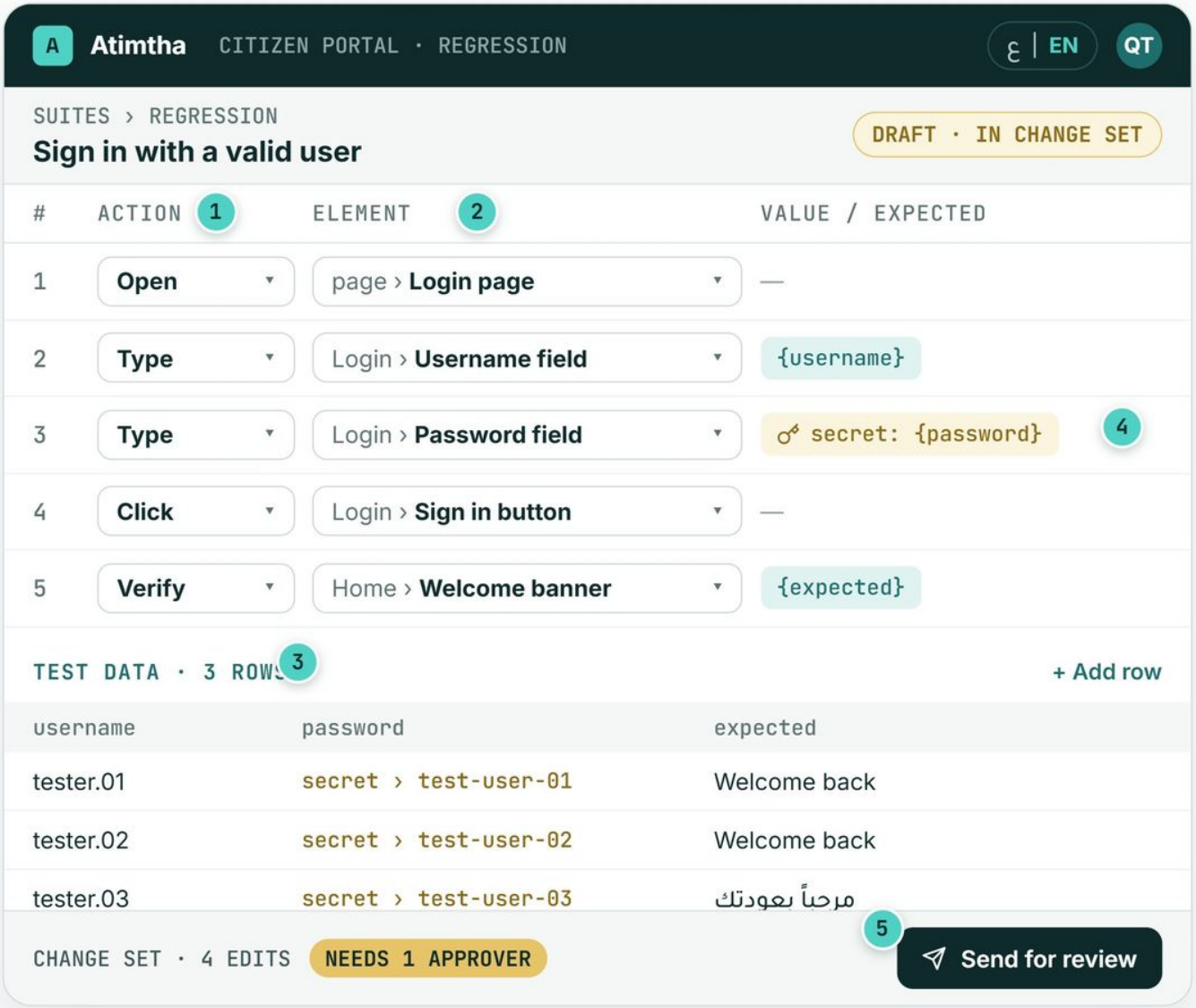
AUTO

Retention clean-up

Evidence is deleted when your organisation's retention period ends.



Build a test the way you'd write one.

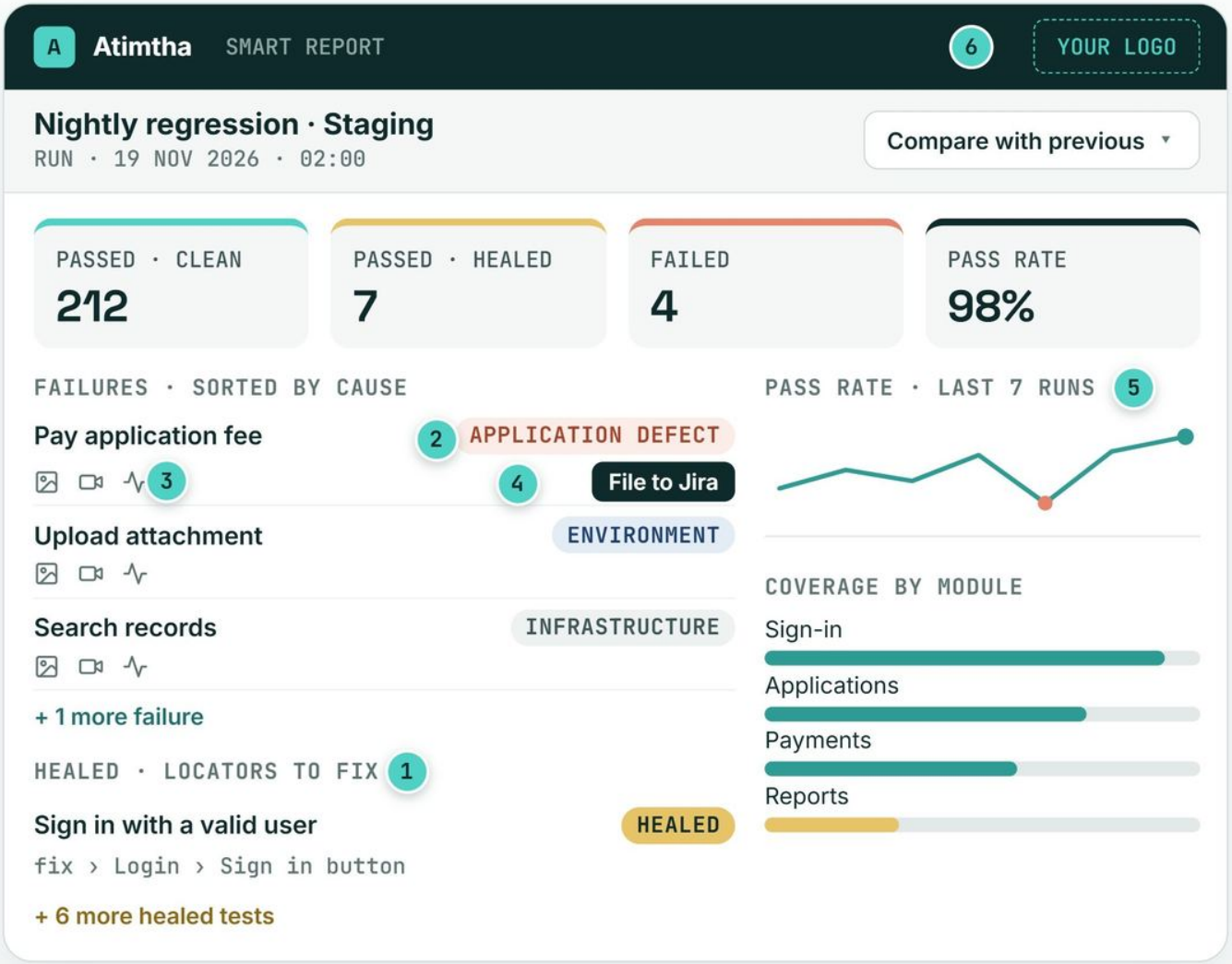


1. **Actions from a list**
   Open, type, click, verify. Pick the action — there's no code to write.
2. **Elements by name**
   Choose the element by its name. Its locators live in the object repository.
3. **Data in a grid**
   Each row runs the same case again, with its own data and expected result.
4. **Passwords stay out**
   A reference to your secrets store, read only at run time.
5. **Sent for review**
   Your edits leave as a change set. A second person approves them.

ILLUSTRATIVE SCREEN WITH SAMPLE DATA.
YOUR LAYOUT WILL DIFFER.

Answers, not just results.

What broke, what got fixed, and what is still untested — on one page.



1. **Healed, counted apart**
   Never mixed into clean passes. Each one names the locator to fix.
2. **Sorted by cause**
   Application defect, environment or infrastructure problem.
3. **Evidence, one click away**
   Screenshots, a video and a trace for every failure.
4. **Straight to your tracker**
   File a failure to Jira or Azure DevOps as a defect.
5. **Trends and comparison**
   Follow the pass rate, compare runs, see coverage by module.
6. **In your colours**
   Your logo, colours and report title, set per project.

ILLUSTRATIVE SCREEN WITH SAMPLE DATA.

PLAYBOOK 01 / 06

QA tester

Automate what you already test.

WHERE YOU WORK

Test Builder · Test data · Suites · Runs · Smart Report

YOU WORK WITH

- ← **Automation engineer** — gives you pages and elements to build with
- → **QA lead** — approves your change sets

PLAYBOOK 01 OF 06

YOUR JOURNEY

STEP 01 — Build

Pick an action — open, type, click, verify — and the element it acts on.

STEP 02 — Feed

Add data rows and expected results. One case runs once per row.

STEP 03 — Submit

Your edits go out as a change set for a second person to approve.

STEP 04 — Run

Run a suite on any environment, or let the schedule run it overnight.

STEP 05 — Read

Open the Smart Report. Every failure has screenshots, a video and a trace.

WHAT YOU GAIN

> **“I automate without writing code — and the same case runs in Arabic and English.”**

TIP

Need another scenario? Add a data row instead of copying the whole test.

RULE

Your edit reaches the shared suite only after a second person approves it.

Automation engineer

Keep the suite healthy as screens change.

WHERE YOU WORK

- Object repository
- Workflows
- Data generators
- Healing report
- CI

YOU WORK WITH

- → **QA tester** — builds tests on your pages and elements
- → **QA lead** — approves your change sets

YOUR JOURNEY

STEP 01 — Describe

Keep each page's elements, with a main locator and backup locators.

STEP 02 — Chain

Link test cases into workflows that follow a real user journey.

STEP 03 — Generate

Maintain the data generators your test data relies on.

STEP 04 — Heal

See which tests passed only through a backup locator, and which locator to fix.

STEP 05 — Connect

Run suites from your CI pipeline with a token.

WHAT YOU GAIN

> **"A screen change is fixed once, in one place — and I'm told exactly which locator to fix."**

TIP

Away from the portal? Edit offline. Your work comes back as a change set, ready for review.

RULE

Offline edits go through the same approval as edits made in the portal.

PLAYBOOK 03 / 06

QA lead

Decide what changes, and when it runs.

WHERE YOU WORK

- Change review
- Schedules
- Defects
- History

YOU WORK WITH

- ← **Testers and engineers** — send you their change sets
- → **Developers** — get your defects in Jira or Azure DevOps

PLAYBOOK 03 OF 06

YOUR JOURNEY

STEP 01

Review

Approve change sets before they reach the shared suite.

STEP 02

Schedule

Run on any environment, or schedule it — say, every weekday at 02:00.

STEP 03

Triage

Failures arrive sorted: application, environment or infrastructure.

STEP 04

File

Send failed tests to Jira or Azure DevOps as defects.

STEP 05

Trace

See who changed what, who approved it, and when.

WHAT YOU GAIN

> “Control over what changes, with a full history of who changed what and when.”

TIP

Treat healed passes as a to-do list for your engineers, not as wins.

RULE

Nothing applies to the shared suite until a second person approves it.

Delivery manager

Know where quality stands, without asking.

WHERE YOU WORK

Dashboard · Trends · Run comparison · Coverage

YOU WORK WITH

- ← **QA lead** — schedules and runs the suites
- → **Stakeholders** — get a clear picture of quality, per project

YOUR JOURNEY

STEP 01
Watch
Follow the dashboard for each of your projects.

STEP 02
Trend
Watch the pass rate move from run to run.

STEP 03
Compare
Put two runs side by side: what broke, and what got fixed.

STEP 04
Spot
See coverage by module — and what is still untested.

WHAT YOU GAIN

“A live view of quality per project, without asking the team for status.”

TIP

The morning after an overnight run, compare it with the one before. It shows exactly what changed.

RULE

You see only the workspaces you’ve been given access to.

PLAYBOOK 05 / 06

Client stakeholder

Evidence of testing, in your own colours.

WHERE YOU WORK

- Branded reports
- Jira · Azure DevOps
- Teams · Slack

YOU WORK WITH

- ← **The delivery team** — sends you reports and defect tickets
- → **Your developers** — pick defects up in your own tracker

PLAYBOOK 05 OF 06

YOUR JOURNEY

STEP 01

Receive

Reports carry your logo, colours and report title, per project.

STEP 02

Follow

Defect tickets arrive in the Jira or Azure DevOps you already use.

STEP 03

See

Screenshots, videos and traces show exactly what failed.

STEP 04

Alerts

Results can be posted to Teams, Slack or a webhook.

WHAT YOU GAIN

“Evidence of testing I can trust — in my own colours, in the tools I already use.”

TIP

Ask for results in your own Teams or Slack channel — no new tool to learn.

RULE

Evidence can only be downloaded by people allowed to see it.

PLAYBOOK 06 / 06

Organisation admin

Set it up once. Govern it always.

WHERE YOU WORK

Members & roles · Agents · Integrations · Retention

YOU WORK WITH

- **Every member** — gets only the workspaces they need
- **QA leads** — run the workspaces you set up

PLAYBOOK 06 OF 06

YOUR JOURNEY

STEP 01

People

Add members and give each a role — admin, QA lead, tester or viewer — per workspace.

STEP 02

Agents

Manage the execution agents that run tests inside your network.

STEP 03

Connect

Set up Jira, Azure DevOps, Slack, Teams and webhooks. Tokens stay in your secrets store.

STEP 04

Retain

Decide how long screenshots, videos and reports are kept.

WHAT YOU GAIN

“One place to govern access, tools and data for every project.”

TIP

Give each person only the workspaces they need. Roles are set per workspace.

RULE

Every access setting you change is logged with who and when.

One night, start to finish.

Follow one night of regression through Atimtha — including a cosmetic change that broke nothing, and a real defect that didn't hide.

EXAMPLE SCENARIO FOR ILLUSTRATION.

The point

A moved button didn't turn the suite red, a real defect didn't slip through, and every change was approved and on record.

RELEASE 4.2 · DEVELOPERS
The Sign-in button moves to a new place on the login page.

02:00 · AGENT — AUTO
The scheduled regression starts inside your network.

02:14 · RUN — PASSED — HEALED
Sign-in tests miss the button the main way. A backup locator finds it.

02:31 · RUN — FAILED · APP DEFECT
The payment test fails, and is sorted as an application defect.

08:30 · QA LEAD
The Smart Report lists the healed tests apart from the clean passes.

08:40 · QA LEAD — → JIRA
The payment failure is filed to Jira as a defect.

09:15 · DEVELOPER
Screenshots, a video and the trace show exactly what went wrong.

10:00 · AUTOMATION ENGINEER — CHANGE SET
The Sign-in locator is fixed once and sent for review.

10:20 · QA LEAD — ✓ APPLIED
Approved. Every test that uses that button picks up the fix.

Eight rules the system won't let you break.

01

A second pair of eyes

A change applies only after a second person approves it.

02

Healed is not clean

A healed pass is always reported apart from a clean pass.

03

Offline is no shortcut

Offline edits take the same approval as edits made in the portal.

04

No passwords inside

The portal keeps references. Your secrets store keeps the passwords.

05

Nothing connects in

Agents call out to the portal. No firewall port is opened.

06

Sealed organisations

The database itself keeps each organisation's data apart.

07

Everything on record

Every change, run, approval and access setting — with who and when.

08

Evidence expires

Deleted after your retention period. Downloadable only by people allowed to see it.

Installed where you need it. Even offline.

Atimtha runs as containers in your own cloud or on your own servers — including networks with no internet.

Client cloud

WHERE IT RUNS

Your own cloud account, as containers.

SUITS

Organisations with a cloud platform and their own operations team.

On-premises

WHERE IT RUNS

Your own servers, as containers.

SUITS

Government and regulated organisations that keep data in-house.

Offline

WHERE IT RUNS

An offline package with every image included. No internet needed.

SUITS

Closed and classified networks.

Local agents

WHERE IT RUNS

A tester's machine, or a server next to the application.

SUITS

Applications reachable only from inside a network.

WORKS WITH YOUR TOOLS

Jira · Azure DevOps · Microsoft Teams · Slack · Webhooks · CI pipelines · token · Your secrets store · Playwright engine · YAML / JSON import & export

IN VERSION 1.0

- ✓ Web application testing
- ✓ API testing

PLANNED FOR VERSION 1.1

- Mobile app testing
- Performance (load) testing

Release faster, with the testers you already have.

Faster releases
Regression that took days by hand can run overnight on a schedule, or on every build from CI.

More people can automate
Manual testers build automated tests without programming. Automation stops waiting for engineers.

Lower maintenance
One object repository, backup locators and healing reports keep tests working as screens change.

Results you can trust
Healed tests are reported apart from clean passes, and failures are sorted by cause.

Arabic-first
The portal, the tests and the reports work right-to-left in Arabic, as well as in English.

Fewer tools to buy
Web tests, API tests, scheduling, reporting and defect filing come in one portal.

No lock-in
Projects import and export in open YAML/JSON, and run on Playwright, a widely used open-source engine.

WE MEASURE IT WITH YOU

Your pilot, scored.

Every pilot tracks three numbers against where you start. Fill them in with your pilot team.

Time to first passing run

BASELINE	PILOT

Regression hours per release

BASELINE	PILOT

Defects found before release

BASELINE	PILOT

NO CLAIMED NUMBERS UNTIL WE'VE MEASURED THEM — TOGETHER.

The questions everyone asks.

The answers teams need most in their first weeks with Atimtha.

Do I need to know how to code?

A No. Testers build tests from dropdowns and a data grid. An engineer sets up each page's elements once; testers pick from them.

Can we test our APIs too?

A Yes. Send requests and check status, headers, JSON fields, schema and response time — in the same portal and reports.

A button moved. Will everything go red?

A Not for a cosmetic change. A backup locator keeps the test going, and it's reported as healed, with the locator to fix.

Which languages does it speak?

A Arabic and English. The portal, the tests and the reports all work right-to-left in Arabic.

Does our data leave our network?

A Tests run on agents inside your network. They only call out; nothing connects in. Passwords stay in your secrets store.

Can we take our tests with us?

A Projects import and export in open YAML/JSON, and run on Playwright, a widely used open-source engine.

Can it run with no internet?

A Yes. The offline package includes every image it needs, for closed and classified networks.

What about mobile apps and load testing?

A Both are planned for version 1.1. Version 1.0 covers web applications and APIs.

STILL HAVE A QUESTION?

Ask the IltizamTech team directly.

info@iltizamtech.com

The jargon, translated.

Sixteen words you'll meet in Atimtha, in plain language — with the common Arabic term.

ARABIC TERMS ARE COMMON TRANSLATIONS; YOUR PORTAL'S LABELS MAY DIFFER.

Locator
مُحدِّد العنصر
How a test finds an element — an address for a button.

Object repository
مستودع العناصر
The shared list of every page and its elements.

Self-healing
الإصلاح الذاتي
A backup locator finds a moved element; the test is flagged.

Test case
حالة اختبار
A sequence of actions and checks, like signing in.

Workflow
مسار عمل
Test cases chained into one real user journey.

Suite
مجموعة اختبارات
Test cases run together, such as smoke or regression.

Regression
اختبار الانحدار
Re-checking that what worked before still works.

Change set
حزمة تغييرات
A bundle of edits sent for a second person to approve.

Execution agent
وكيل التنفيذ
Runs the tests inside your network and reports back.

Trace
سجل التتبّع
A step-by-step recording of a run, to see what happened.

CI pipeline
خط التكامل المستمر
The automated build process. It can start runs with a token.

Webhook
ويب هوك
A message Atimtha sends another system with results.

Secrets store
خزنة الأسرار
The vault where your passwords and tokens are kept.

Retention period
فترة الاحتفاظ
How long evidence is kept before it is deleted.

Playwright
محرّك التشغيل
The widely used open-source engine Atimtha runs tests on.

YAML / JSON
صيغ نصية مفتوحة
Open text formats for importing and exporting projects.

Built for data that can't leave.

Atimtha runs where your data lives and records every action, so your security team can see exactly what happens.

Your environment
Install in your own cloud account, on your own servers, or offline in a closed network.

Nothing connects in
Execution agents call out to the portal. No inbound connection, and no firewall port is opened.

No passwords in the portal
Test passwords and integration tokens are references to your secrets store, resolved at run time.

Sealed organisations
The database itself separates organisations, keeping each one's projects and results apart.

Approved, and on record
Changes apply only after a second person approves. Every change, run, approval and access setting is logged.

Protected evidence
Encrypted at rest, deleted after your retention period, and downloadable only by people allowed to see it.

NEED A HAND?

info@iltizamtech.com

iltizamtech.com

IltizamTech
التزام تك

OUR MISSION · AUTOMATION & AI

Automation and AI that answer to people.

Iltizam means commitment. We build automation and AI for organisations that answer for every decision they make — governments, regulators, and the teams that serve them. Machines take on the work that repeats. People keep the approvals, the audit trail and the final word.

Automate the repetitive

Machines take the work that repeats. People keep the judgement.

Govern every change

Approvals, audit trails and clear roles — built in, not bolted on.

Honest by design

Show what really happened — like a heal reported as a heal.

Sovereign by default

Run where the data lives, even with no internet at all.

AI under governance

Switched on deliberately, watched closely, open to human override.

أتمتها Automate it — and keep it accountable.