testVibe

// PLAIN ENGLISH IN · REAL PLAYWRIGHT OUT

Tests that write themselves.

Load it. Run it. Watch it live.

Describe what to check in plain English. TestVibe's AI writes the Playwright code and runs it in isolated cloud sandboxes — then load-tests the same flows while live telemetry streams from your running servers.

AN INTRODUCTION

A product of Ice Tea Group · testvibe.com

THE PROBLEM

UI testing is the bottleneck

Slow to write

Every scenario is hand-coded — selectors, waits, test data. Real coverage takes weeks of engineering time.

Brittle to maintain

Every UI change breaks selectors. Suites go flaky, red builds get ignored, and trust in the tests evaporates.

Split across tools

Functional tests, load tests, and server monitoring live in three products that never see the same picture.

SOUND FAMILIAR?

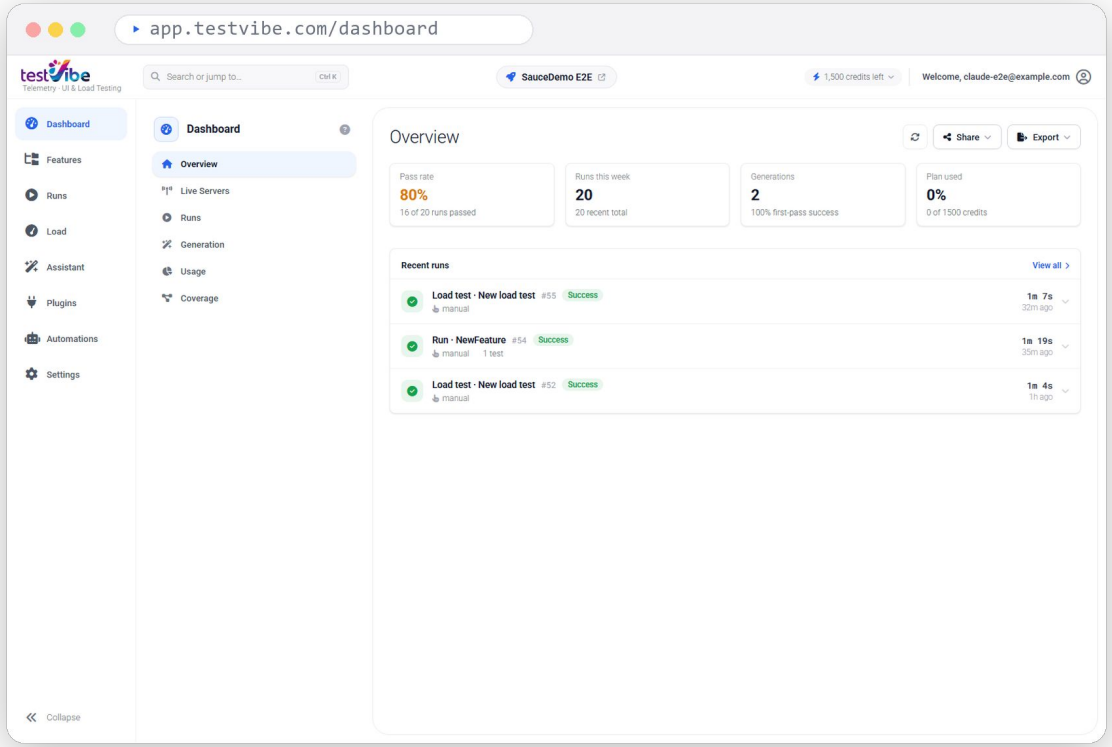
Coverage lags. Confidence drops. Bugs reach users first.

Most teams test a fraction of what they ship — not because they don't care, but because every test is expensive to create and keep alive.

WHAT IT IS

AI-native testing, end to end

TestVibe turns plain-English descriptions into real Playwright tests, runs them in isolated cloud sandboxes, then load-tests the same flows while streaming live telemetry from your servers — every run, trace, and chart in one place.



Describe → real tests

Plain English, in-browser recordings, or AI exploration become Playwright.

Load test the same flows

Replay journeys as concurrent users; chart latency, throughput & errors.

Watch it live

Stream CPU, memory, sessions and GC heap from your running servers.

THE BUILDING BLOCKS

Gherkin in, Playwright out

Gherkin

A plain-language format for describing behavior in Given / When / Then steps — readable by your whole team.

login.feature

```
Scenario: Standard user can sign in
  Given I open the home page
  When I sign in as "standard_user"
  Then I see the products page
```

Open standard · .feature files · the “what”

Playwright

Microsoft's open-source browser automation — it drives a real browser to click, type, and assert exactly like a user.

sign-in.spec.js

```
await page.goto('/');
await login(page, 'standard_user');
await expect(page.getByRole('heading',
  { name: 'Products' })).toBeVisible();
```

Chromium · Firefox · WebKit · the “how”

How they relate TestVibe’s AI turns each readable Gherkin step into the Playwright code that runs it against your app. Both are open standards — your tests stay readable, portable, and never locked in.

HOW IT WORKS

From a sentence to a green run

01

Describe it

Type plain English, record a click-through, or point the AI at your URL to explore. It all lands as clean Gherkin .feature files.

02

AI writes the code

In an isolated sandbox the agent generates Playwright, probes your real app, and self-corrects until the script passes — frontier models, no API keys to bring.

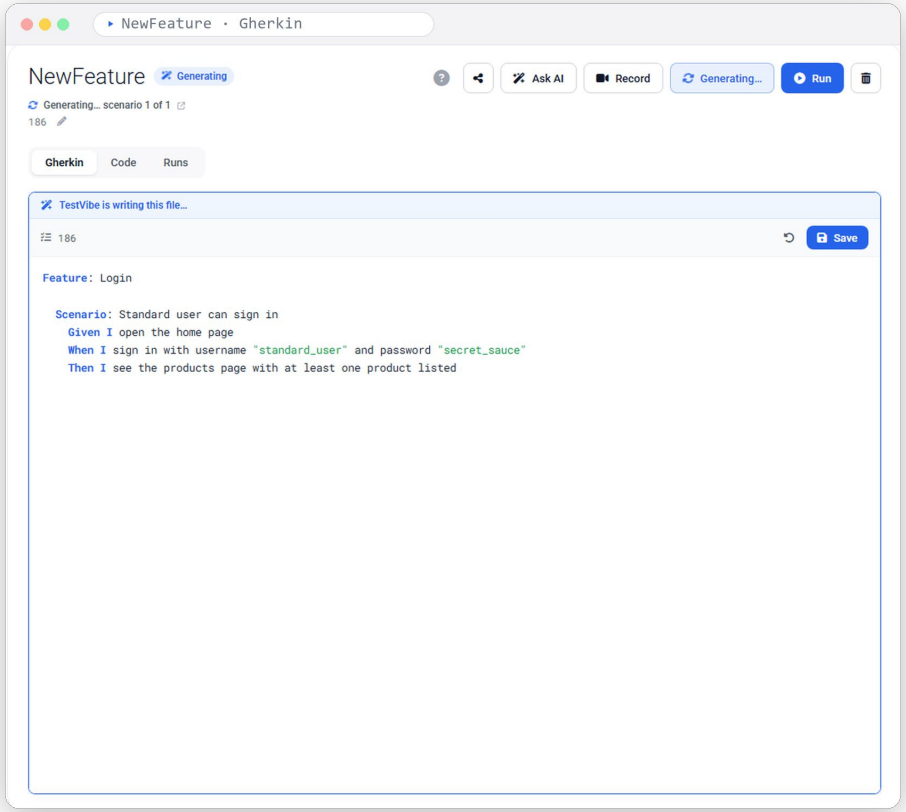
03

Run, load, observe

Deterministic Playwright, no AI tokens per run. Watch steps tick green, open the trace, replay the video — then load-test and watch telemetry.

THE GENERATION LOOP

Plain English in. Real Playwright out.



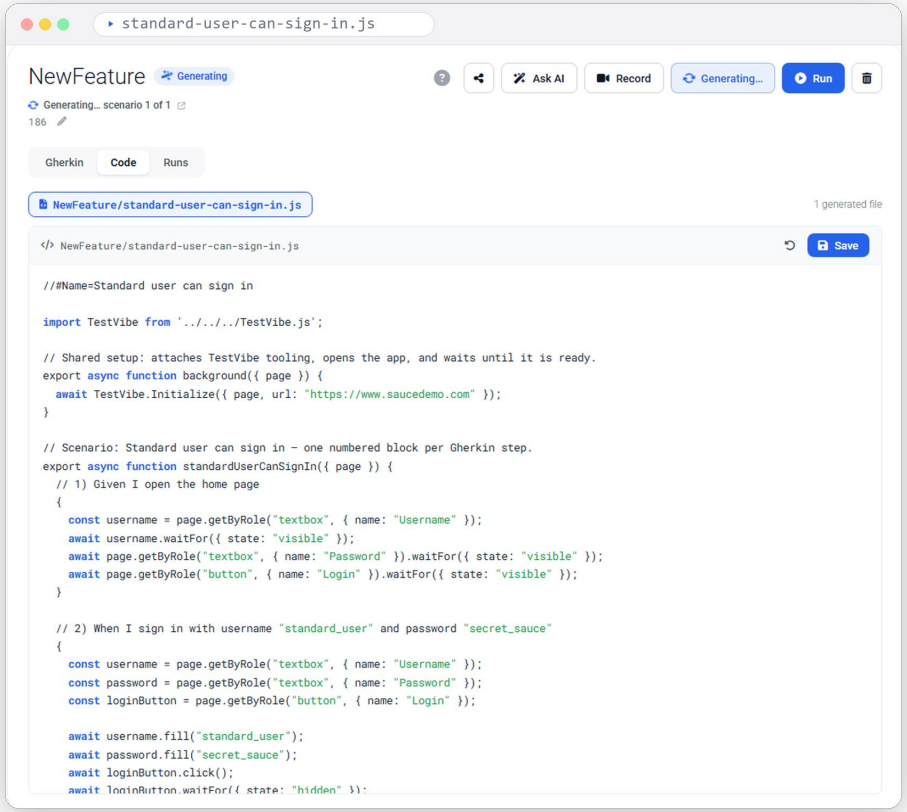
Gherkin · plain English

AI generates

probes your app

self-corrects

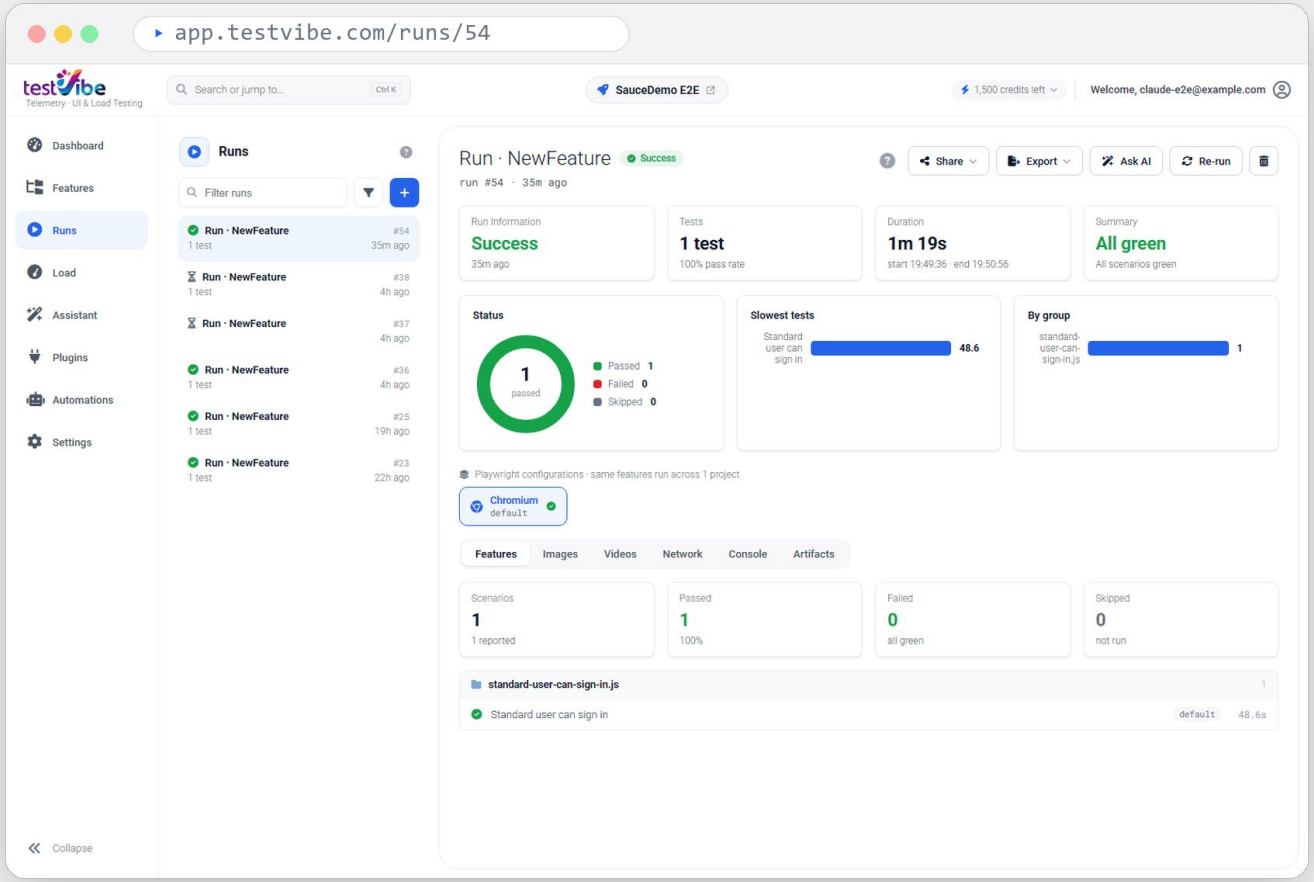
runs until green



Generated Playwright

RESULTS

Every run, in plain sight



1m 19s

RUN DURATION

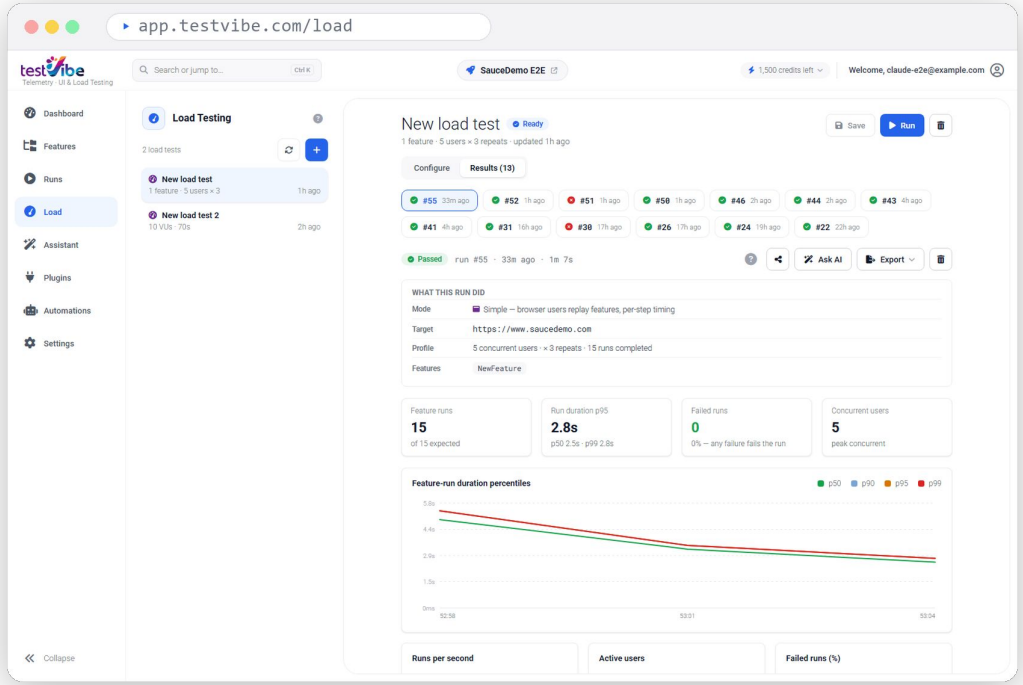
All green

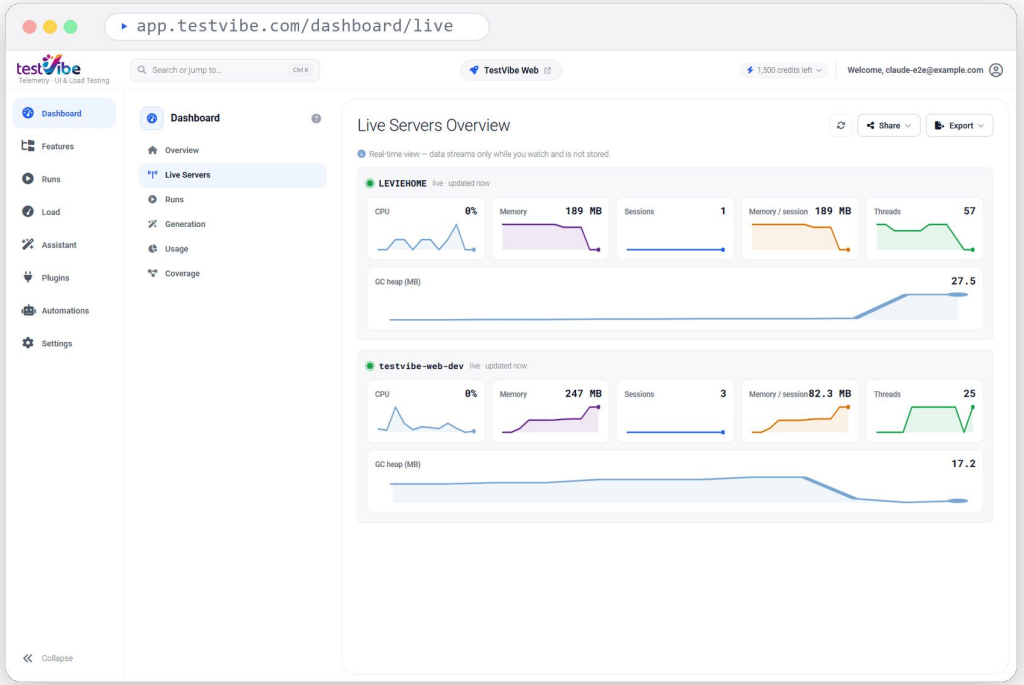
1 / 1 PASSED

- ✓ Live status as each step ticks green
- ✓ Trace, video, screenshots, console & network kept per run
- ✓ Re-run, share, or export — no digging through CI logs

BEYOND FUNCTIONAL

Push the same flows — and watch them hold up





Load testing Replay journeys as concurrent browser users or drive k6 traffic — latency percentiles, throughput, and error rate, charted live.

Live telemetry Drop a lightweight agent into any .NET or Node app; CPU, memory, sessions, threads and GC heap stream per server while you test.

THE WHOLE WORKFLOW

Everything around the tests

AI exploration

Point it at a URL; it browses, screenshots, and proposes what to test.

Record in browser

Click a flow once; it becomes a scenario, ready to generate.

Traces & replays

Every run keeps its Playwright trace, video, and screenshots.

Coverage

See which app areas have tests — and where the gaps are.

Automations

Run a suite or load test on a schedule, or when an event fires.

CLI & MCP

Drive it all from the terminal, your AI tools, and CI.

Plugins

Assert with an LLM, post to Slack, file a Jira, compare snapshots.

Team workspaces

Orgs, roles, invites; share projects, runs, and dashboards.

Test behind the wall

A secure tunnel lets cloud sandboxes reach localhost or VPN apps.

DEPLOYMENT

Same platform — your choice of where it runs

RECOMMENDED

FULLY MANAGED

Cloud-hosted

Sign in and start testing in seconds — we run the sandboxes, store every trace, and meter the AI. Nothing to install.

- ✓ Isolated cloud sandboxes, warm and ready
- ✓ First-party AI built in — no API keys
- ✓ Usage-based credits, scale anytime
- ✓ Automatic updates, backups & patches

SELF-HOSTED

On-premise

Run the whole platform inside your own network or private cloud — for data-residency, compliance, and air-gapped teams.

- ✓ Ships as containers — Docker or Kubernetes
- ✓ Your code & data never leave your network
- ✓ Bring your own AI endpoint or keys
- ✓ Annual license · runs fully air-gapped

testvibe

Tests that write themselves.

See your first run go green.

Get early access →

docs.testvibe.com

BUILT FOR TEAMS WHO SHIP

SaaS & web app teams · QA & SDET squads · Platform & DevOps · E-commerce · Fintech & regulated